ASTES

Advances in Science, Technology & Engineering Systems Journal

VOLUME 10-ISSUE 6|NOV-DEC 2025

www.astesj.com
ISSN: 2415-6698

EDITORIAL BOARD
Editor-in-Chief

Editorial Board Members

Regional Editors

Editorial

As digital infrastructures, intelligent algorithms, and cyber–physical systems continue to expand across sectors, new challenges emerge around security, efficiency, reliability, learning, and system observability. The eight research papers featured in this editorial collectively reflect how contemporary scholarship is responding to these challenges through innovative architectures, adaptive models, data-driven assessment frameworks, and practical tools. Spanning topics from Zero Trust security and private 5G systems to neural network design, learning analytics, reliability engineering, healthcare AI, network visualization, and drone detection, these works highlight the growing convergence of theory, technology, and real-world applicability.

The first paper addresses a foundational concern in modern cybersecurity: securing resource-constrained Internet of Things devices under a Zero Trust Architecture. By treating every device as untrusted and enforcing continuous verification, Zero Trust presents particular challenges for IoT environments. The proposed Time-based Identity Management and Flow Rule Control Engine (TIMeFoRCE) introduces a lightweight, time-based authentication mechanism that aligns with Zero Trust principles while remaining feasible for constrained devices. By providing concrete metrics and a viable implementation pathway, this work advances the practical adoption of Zero Trust in IoT ecosystems where prior studies largely identified problems without offering implementable solutions [1].

The second contribution explores private 5G as an alternative to traditional wired cable TV services, particularly in multi-dwelling and rural contexts. Recognizing spectrum limitations in the sub-6 GHz band, the paper proposes a hybrid Multiple-Input Multiple-Output (MIMO) approach that simultaneously supports reliable broadcasting and efficient data communication. By combining diversity MIMO and multi-stream MIMO within a single framework, the study addresses frequency efficiency challenges and strengthens the case for private 5G as a flexible and scalable broadcasting infrastructure [2].

Automating the design of deep neural networks is the focus of the third paper, which introduces structurally adaptive DNNs, termed StradNet models. Rather than relying on manual trial-and-error or post hoc pruning, this approach integrates structural adaptation directly into the training process. By progressively pruning weak connections and refining network structure during learning, StradNet produces efficient, partially connected architectures that perform well in dynamic and high-dimensional environments. Experimental validation demonstrates superior scalability and performance compared to conventional pruning strategies, underscoring the potential of adaptive architectures in real-world machine learning applications [3].

The fourth study shifts attention to education, proposing a method to assess learners' conceptual understanding of data science through open-ended responses. By combining natural language processing techniques, such as Word2vec embeddings, with machine learning models including random forests, Naive Bayes, and logistic regression, the framework identifies both understood and misunderstood concepts. The integration of teacher–learner interaction data and electrodermal activity further enriches the analysis. Despite a limited sample size, the results show strong performance, highlighting the promise of linguistic and behavioral analysis in diagnosing learning difficulties and providing targeted instructional support [4].

Reliability engineering is addressed in the fifth paper, which investigates the persistent gap between laboratory reliability predictions and real-world field performance. By introducing a closed-loop reliability correlation framework, the study significantly improves alignment between lab-tested and field-observed failure modes. The integration of traditional DFMEA with system-level tools such as Function Block Diagrams, Interface Matrices, and usage-context analysis enables a more holistic, user-centered understanding of product behavior. This approach enhances predictive accuracy and supports proactive mitigation strategies that better reflect operational realities [5].

The sixth paper reviews privacy-preserving artificial intelligence in the context of the Internet of Medical Things, focusing on Federated Learning enhanced by Differential Privacy and Blockchain technologies. Through a comprehensive comparison of FL-DP and FL-BC frameworks, the study highlights trade-offs in privacy guarantees, trust, scalability, and energy efficiency. The analysis reveals that while differential privacy offers strong mathematical protection, blockchain-based approaches ensure transparency and traceability. Emerging hybrid architectures are identified as a promising direction for secure, trustworthy, and regulation-compliant healthcare AI systems [6].

Observability and analysis of complex networked systems are the subject of the seventh contribution, which extends the PerfVis tool into a comprehensive timestamp data analyzer. By integrating statistical outputs, customizable traffic patterns, and external data sources, the enhanced tool supports deeper insights into system and protocol behavior. Case studies involving real 5G networks and established measurement protocols demonstrate the value of flexible visualization and analysis in uncovering temporal patterns, performance anomalies, and protocol dynamics [7].

The final paper responds to growing concerns around drone proliferation by presenting a comprehensive dataset of drone acoustic signatures and an interactive web-based exploration tool. Covering 32 drone categories, the dataset includes raw audio, spectrograms, and MFCC representations, supporting research in acoustic-based drone detection and classification. The accompanying web application enhances accessibility and educational value, enabling users to explore and analyze drone sounds interactively. This contribution fills a critical gap in publicly available resources for acoustic drone detection research [8].

Collectively, these eight papers illustrate how modern research is tackling complexity across technological, educational, and operational domains. From securing IoT devices and optimizing wireless spectrum usage to automating neural network design, assessing learning comprehension, improving product reliability, safeguarding medical data, visualizing network behavior, and enabling drone detection, each study emphasizes practical relevance grounded in rigorous methodology. Together, they underscore a broader shift toward adaptive, data-driven, and user-aware systems that are better aligned with real-world constraints and evolving societal needs, offering valuable foundations for future research and deployment.

References:

[1] V. Morris, K. Kornegay, J. Falaye, S. Richardson, M. Tienteu, L.J.D. Tchuenkou, “TIMeFoRCE: An Identity and Access Management Framework for IoT Devices in A Zero Trust Architecture,” Advances in Science, Technology and Engineering Systems Journal, 10(6), 1–22, 2025, doi:10.25046/aj100601.

[2] H. Ito, H. Ohno, H. Kitano, S. Matsumoto, “Private 5G MIMO for Cable TV IP Broadcasting,” Advances in Science, Technology and Engineering Systems Journal, 10(6), 23–28, 2025, doi:10.25046/aj100602.

[3] D. Degbor, H. Xu, P. Singh, S. Gibbs, D. Yan, “StradNet: Automated Structural Adaptation for Efficient Deep Neural Network Design,” Advances in Science, Technology and Engineering Systems Journal, 10(6), 29–41, 2025, doi:10.25046/aj100603.

[4] K. Yasuda, H. Shimakawa, F. Harada, “Identifying Comprehension Faults Through Word Embedding and Multimodal Analysis,” Advances in Science, Technology and Engineering Systems Journal, 10(6), 42–54, 2025, doi:10.25046/aj100604.

[5] R. Lawrance, N.K.R. Gorla, “System-Level Test Case Design for Field Reliability Alignment in Complex Products,” Advances in Science, Technology and Engineering Systems Journal, 10(6), 55–64, 2025, doi:10.25046/aj100605.

[6] S.A. Farooqi, A.A.R. Rahman, A. Saad, “Federated Learning with Differential Privacy and Blockchain for Security and Privacy in IoMT A Theoretical Comparison and Review,” Advances in Science, Technology and Engineering Systems Journal, 10(6), 65–76, 2025, doi:10.25046/aj100606.

[7] M. B¨ohmer, T. Herfet, “PerfVis+: From Timestamps to Insight through Integration of Visual and Statistical Analysis,” Advances in Science, Technology and Engineering Systems Journal, 10(6), 77–87, 2025, doi:10.25046/aj100607.

[8] M.Y. Wang, M. Linn, A.P. Berg, Q. Zhang, “A Multi-class Acoustic Dataset and Interactive Tool for Analyzing Drone Signatures in Real-World Environments,” Advances in Science, Technology and Engineering Systems Journal, 10(6), 88–96, 2025, doi:10.25046/aj100608.

Editor-in-chief

Prof. Hamid Mattiello

ADVANCES IN SCIENCE, TECHNOLOGY AND ENGINEERING SYSTEMS JOURNAL

Volume 10 Issue 6 **November-December 2025**

CONTENTS

Advances in Science, Technology and Engineering Systems Journal
Vol. 10, No. 6, 1-22 (2025)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTES Journal
ISSN: 2415-6698

TIMeFoRCE: An Identity and Access Management Framework for IoT Devices in A Zero Trust Architecture

Vinton Morris[*,1], Kevin Kornegay[2], Joy Falaye[1], Sean Richardson[1], Marcial Tienteu[1], Loic Jephson Djomo Tchuenkou[1]

[1]*Morgan State University, Cybersecurity Assurance and Policy Center, Baltimore, 21251, United States*

[2]*Morgan State University, Electrical and Computer Engineering, Baltimore, 21251, United States*

ARTICLE INFO

Article history:
Received: 01 October, 2025
Revised: 19 October, 2025
Accepted: 21 October, 2025
Online: 10 November, 2025

Keywords:
Zero Trust
Zero Trust Architecture
Identity and Access Management
Access Control
Fingerprinting
Authentication
Policy Enforcement Point
Policy Decision Point

ABSTRACT

Zero Trust Architecture offers a transformative approach to network security by emphasizing ”never trust, always verify.” IoT devices, while increasingly integral to modern ecosystems, pose unique challenges for identity management and access control due to their constrained processing power, memory, and energy capabilities. In a Zero Trust framework, every IoT device is treated as a potential security risk, necessitating continuous, adaptive authentication and strict access control policies. Despite their limited resources, IoT devices must undergo identity verification and operate within the principle of least privilege, granting access only to specific services and data based on each device’s role, context, and risk profile. This paper examines the challenges of implementing Zero Trust in IoT environments, focusing on scalable identity management and access control mechanisms that account for the inherent resource constraints of IoT devices while ensuring robust security against emerging threats. Previous works highlighted the inherent issues with IoT devices in a Zero Trust environment; however, they offer no viable solution. This paper proposes Time-based Identity Management and Flow Rule Control Engine (TIMeFoRCE), an identity management and access control solution that satisfies the tenet of Zero Trust for resource-constrained IoT devices through time-based authentication. This work builds upon prior research and provides metrics to show the solution’s efficacy.

1. Introduction

This paper is an extension of work originally presented in the 2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC) [1]. IoT devices, while integral to modern ecosystems, are inherently constrained by their limited computational resources, including processing power, memory, and energy capacity [2, 3, 4]. These limitations make traditional security models, which often assume more robust device capabilities, unsuitable for IoT environments [5, 6, 7]. As a result, securing these devices against an ever-increasing array of cyber threats has become a pressing concern. One promising solution to these challenges is the implementation of Zero Trust Architecture (ZTA) [5, 8, 9, 10, 11]. ZTA operates on the principle of ”never trust, always verify,” where no entity, whether internal or external to the network, is automatically trusted [5, 12, 13]. In the context of IoT, this approach is crucial because it treats every device as a potential security risk, regardless of its location on the network [8]. One of the fundamental tenets of Zero Trust is continuous adaptive authentication and strict access control for all devices, including those with limited resources [8]. Despite the inherent limitations of IoT devices, ZTA requires them to undergo identity verification and adhere to the principle of least privilege, granting access only to specific services and data based on the device’s role, context, and associated risk profile [8, 14].

This paper addresses the inherent limitations of IoT devices by proposing a novel approach to identity and access management, specifically designed for resource-constrained IoT devices within a Zero Trust framework. By leveraging behavior-based fingerprinting and time-based authentication, we aim to offer a lightweight yet effective solution that balances security with the device limitations inherent to the IoT landscape by removing the security responsibility from devices. Vendors such as Cisco, Microsoft, Palo Alto, Armis, and Zscaler [15, 16, 17, 18, 19] provide access control solutions for IoT; however, these proprietary black-box solutions with limited visibility simply exist for financial gains and require significant

[*]Corresponding Author: Vinton Morris, Email: vinton.morris@morgan.edu

https://dx.doi.org/10.25046/aj100601

financial investment and infrastructure change. TIMeFoRCE is one that can be deployed without incurring significant overhead. We performed a comprehensive analysis of the proposed architecture's efficacy through detailed metrics and performance evaluations. In doing so, we aim to advance the state of the art in securing IoT environments, ensuring that they remain resilient against emerging threats while maintaining the integrity of the underlying infrastructure.

Our contribution presents Time-based Identity Management and Flow Rule Control Engine (TIMeFoRCE), an Identity and Access Management (IAM) framework tailored to support IoT devices within a ZTA. The main contributions of this paper are:

- Proposed *TIMeFoRCE*, a novel Identity and Access Management (IAM) system that authenticates IoT devices by extracting discriminative features from network packet headers to generate unique device fingerprints, which are matched against a reference database to validate devices attempting communication.

- Introduced a continuous identity verification and dynamic access control mechanism that integrates time-based authentication, software-defined networking (SDN), and behavioral fingerprinting, thereby mitigating reliance on pre-shared credentials and static configurations.

- Presented a comprehensive evaluation using a testbed of 26 IoT devices under conditions representative of real-world network environments. The system employs machine learning to analyze traffic patterns and classify newly introduced devices in real time.

- Demonstrated that the IAM system enforces authentication and access control policies through continuous traffic monitoring and adaptive interaction with the SDN switch, enabling context-aware security decisions based on device behavior.

The remainder of this paper is structured as follows: Section 2 discusses related work. Section 3 details our methodology and experimental setup. Section 4 presents our results. Sections 5 and 6 conclude the paper and discuss future directions.

2. Related Work

This research spans several areas, including device identification, Zero Trust, microsegmentation, and software-defined networking to develop TIMeFoRCE and are organized accordingly.

2.1. Device Identification

Extensive research has been conducted on device identification and classification [3, 20, 21, 22, 23, 24, 25, 26, 27], and show the various ways to identify devices. However, minimal research has been conducted in the Zero Trust space on the use of such identities for identity management.

One of the more promising works in the area of IoT device identification is by [28]. In [28], the author evaluated their IoT device identification method using two publicly available datasets: the Aalto University IoT dataset and the USNW dataset. With their mixed approach, they achieved accuracies of 94% on the UNSW dataset and 83% on the Aalto dataset. Unlike [28], who grouped similar devices (TP-LinkPlugHS100 and TP-LinkPlugHS110) to form their aggregate model and improve accuracy, our work grouped identical devices (make and model). For example, in our study, two Sengled light bulbs were treated as distinct devices and not grouped because of a color difference (soft white vs. daylight). The work of [28] focused on device classification, while our work focuses on developing device identities and using them to satisfy the identity and access management tenet of a ZTA.

2.2. Zero Trust

While many studies [29, 30, 31, 32, 33] discuss the principles and components of Zero Trust, few provide practical implementations or concrete methods for utilizing such identities for authentication. The authors in [31], investigated global Zero Trust research and its application in IoT environments. In [32], the authors examined advancements in biometrics, blockchain, and AI to enhance IoT security through improved authentication, and explored the integration of Zero Trust principles into IoT frameworks to mitigate cyber threats and strengthen resilience; offering insights from recent research and real-world implementations on the evolution of secure IoT authentication. In [33], the authors sought to reinterpret Zero Trust from a network security perspective by exploring how elements of trust still exist within Zero Trust frameworks; defining the concept and key characteristics of trust in Zero Trust systems, establishing foundational principles, and discussed future research trends and applications across various scenarios. These authors focused primarily on providing a comprehensive review of current literature and Zero Trust trends.

In [34], the authors present results from a cybersecurity test bed, which incorporates elements of a Zero Trust data communication network through the use of autonomic OODA loops. They demonstrate results from experiments in which identity management was integrated with automated threat response and packet-based authentication, alongside the dynamic management of eight unique network trust levels. In their testing environment, they deployed a dynamic orchestration of firewall access control lists (ACLs) and incorporated authentication gateways, each configured with its own dynamic trust levels. The authentication mechanisms employ First Packet Authentication (FPA) alongside Transport Access Control (TAC), with explicit trust established by generating a network identity token at the start of each session. They reported that the system successfully detected and mitigated DDoS attacks, blocking over 100 unauthorized access attempts within 60 seconds. However, despite their effectiveness, IoT devices typically operate in an insecure manner and lack identities that can be tied to external authentication providers.

In [5], the authors conducted a comprehensive review of the literature on Zero Trust, identifying current knowledge and research gaps. They presented relevant literature on Zero Trust. They aim to systematically find new research streams and organize the body of knowledge on Zero Trust. They determined that the majority of the literature they reviewed highlighted the benefits of Zero Trust but did not discuss its potential shortcomings. They noted that although experts contend that implementing Zero Trust principles produces a highly scalable infrastructure, no design for such massive networks has been put forth, proven, or tested to date. They also highlighted that they found no research into sectors such as healthcare or energy, where security is paramount. While they conducted an in-depth review, no relevant solution was proposed to address the inherent lack of identity and access management (IAM) in IoT devices.

The authors in [9], presented the core principles of Zero Trust using a descriptive approach and also reviewed a variety of approaches to effectively implement this paradigm. In addition to providing a thorough analysis of cutting-edge authentication and access control methods across various contexts, they explain the role of these technologies in ZTA. They also go into great detail about standard encryption methods, security automation, and microsegmentation that can be used to implement a ZTA. The paper also examines several challenges that may impede the practical implementation of Zero Trust, including issues with contemporary authentication methods, access control systems, trust and risk assessment frameworks, microsegmentation strategies, and software-defined perimeter solutions. They also identified potential future research directions for the successful realization of zero trust in critical infrastructures. Our work focuses on developing a method for authenticating devices using discrete identities.

In [6], the authors sought to bridge the current knowledge gap and investigate the complexities of the Zero Trust framework. Like [5], [9], [31], and [33] they reviewed the ZTA literature and offered a basic analysis of its application and efficacy in light of earlier research. They examined the advantages and disadvantages of the Zero Trust security architecture, provided a general overview of the model, and addressed the knowledge gap regarding the effectiveness of implementing a Zero Trust philosophy. They supported the widely held belief that Zero Trust has numerous meanings. Like the previous authors, they also focus on the component that equates to a robust Zero Trust policy, but no solution to address the inherent problems.

In [35], the authors presents ZTA-IoT, a novel Zero Trust Architecture tailored for IoT systems, designed to extend and adapt the NIST ZTA framework for cloud-enabled IoT environments. They further proposes the ZTA-IoT Access Control Framework (ZTA-IoT-ACF) to manage diverse interactions among IoT layers and components, to reduce implicit trust and reinforcement of authentication and authorization mechanisms. Additionally, they presents the Object-Level Zero Trust Score-Based Authorization Framework (ZTA-IoT-OL-SAF), which governs access to devices and data objects through dynamic, context-aware authorization based on real-time calculated trust scores. The framework supports continuous authorization decisions by evaluating multiple attributes related to actors, targets, and actions. The study culminates in the UCONIoT model, a formally defined usage control policy governing user-to-object and device-to-object interactions validated through a proof-of-concept implementation on AWS IoT using Lambda functions and DynamoDB. The results highlight the potential of fine-grained, score-based access control in enhancing IoT security. Several concepts were introduced; however, they were later presented as future work. This include; Real-time dynamically calculated score and threshold values, virtual-object level, the cloud level, the application level interactions, and usability. Additionally, the study focused heavily on access control and not authentication. While the work appears to provide effective access control, it contrasts with TIMeFoRCE, which is capable of operating in real time.

The National Institute of Standards and Technology (NIST) special publications 800-207 [8] and 800-207A [36] outline the core principles of any ZTA implementation, and are the primary documents referenced by researchers and practitioners. A thorough set of Zero Trust principles and a referenced ZTA for bringing those ideas to life are outlined in NIST Special Publication 800-207. A

Table 1: Feature Comparison of prior works across device identification, microsegmentation, and SDN controller functionality externalization with TIMeFoRCE.

Work	Scalability	IoT Device Support	Behavioral Identity / SDN Integration	Performance / Strengths
IoTDevID [28]	✓	✓	✗	High classification accuracy on public IoT datasets; behaviour-based device identification; limited to offline analysis and non-dynamic enforcement.
Zero Trust Microsegmentation [29]	✓	✗	✗	Improves Zero Trust posture via fine-grained segmentation and workload isolation; achieves 60–90% reduction in lateral movement risk; scalability policy tradeoff present.
Externalization of Packet Processing in SDN [30]	✓	✗	✓	Architectural scalability through distributed control-plane microservices; minimal performance overhead; no behavioral identification or access control capability.
TIMeFoRCE (This Work)	✓	✓	✓	Provides **real-time authentication and access control** using behavioral fingerprints integrated within SDN; supports continuous Zero Trust validation; scalable to large heterogeneous IoT ecosystems.

key paradigm shift in obtaining a ZTA involves transitioning from traditional security models that enforce segmentation and isolation based on network constructs to models that enforce policy decisions primarily based on identity [8].

While the previous literature identifies some of the benefits of Zero Trust, most do not offer a meaningful solution, especially regarding authentication. Most of the Zero Trust solutions we find exist in the commercial space and are focused on financial gains. Organizations such as [16], [37], [38], and [39] offer IoT platforms; however, these are designed for IoT manufacturers as cloud platforms for user interaction with devices or as aggregation points for analytics.

2.3. *Microsegmentation*

In [29], the authors conducted an impact assessment to develop an analytical framework for characterizing and quantifying the efficacy of microsegmentation in enhancing network security. Their methodology employed a dual graph-feature-based framework that integrated network connectivity and attack graphs to assess network exposure and robustness. To evaluate robustness, they used MulVAL, incorporating inputs such as Nessus XML data and network firewall rules to generate the network attack graph. The study examined a range of metrics, including the number of misconfigurations, counts of shortest paths, average and minimum shortest path lengths, minimum shortest path counts, and average and maximum out-degrees, as well as average betweenness. Their findings indicate that the average and maximum out-degrees of compromised network privileges substantially illustrate the impact of microsegmentation in constraining lateral movement and exploration by attackers, thereby reducing the potential number of attacks by 93% and 69%, respectively. Furthermore, microsegmentation reduced average betweenness by over 98% and altered the network topology, resulting in a more linear distribution of betweenness. In [29], the authors focused on developing an evaluation framework for microsegmentation, a tenet of Zero Trust; however, our work focuses on identity management, which authenticates devices communicating within the network.

2.4. *Software-Defined Networking*

For our research framework, we examined the work of [20] and [28] in the context of device identification, the work of [29] in microsegmentation, and the contributions of [30] in the SDN domain. In [30], the authors present an architecture that decouples controller functionality and externalizes packet processing, thereby decentralizing microservices at the control plane level. Their goal was to disaggregate core subsystems of SDN controllers into cooperative microservices, allowing flexibility in programming language selection and converting a monolithic control plane into a microservice-based architecture. They also integrated support for external reactive applications beyond traditional SDN APIs. To achieve this, they propose using Kafka as an event distribution system to transmit incoming packets from network devices to external management apps. ICMP packets were transmitted to evaluate the system, and the corresponding response times were recorded for analysis. Although this work aligns closely with ours, it focuses on measuring response time rather than addressing an actual use case. Additionally, the preliminary work reported in [1] presents ongoing efforts to identify and manage access to IoT devices in a ZTA. Table 1 provides a comparison of TIMeFoRCE in terms of scalability, device type, identity methods, and performance. Table 1 compares four approaches, IoTDevID, Zero Trust Microsegmentation, Externalized SDN Processing, and TIMeFoRCE across scalability, IoT device support, behavioral identity integration, and performance. TIMeFoRCE distinguishes itself by providing real-time behavioral fingerprint-based authentication and access control within SDN, demonstrating both adaptability and scalability across diverse environments.

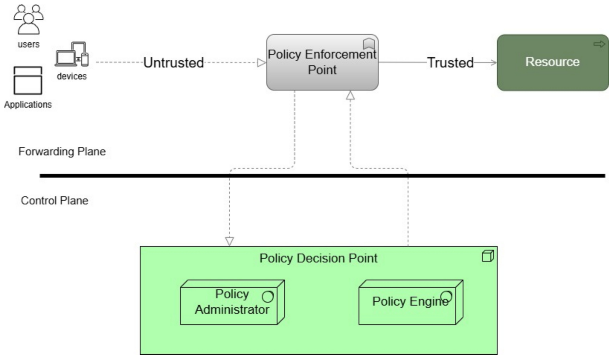


Figure 1: Zero Trust Architecture covering the policy enforcement point and policy decision point with policy administrator and policy engine.

3. Methodology

To present our Identity and Access Management (IAM) solution (i.e., TIMeFoRCE), we divided the implementation into several Phases. We first investigated and created the testbed. Second, data collection was performed on the testbed. Next, we performed device classification during the analysis phase. Following the analysis phases, we implemented the proposed solution. Finally, metrics were obtained on the performance of the proposed solution. The first three items were performed in [1] as part of a work-in-progress paper.

3.1. *The Simplified Zero Trust Architecture*

A ZTA is often depicted as two distinct components: the control and data plane. The control plane serves as the Policy Decision Point (PDP), where security policies are defined and decisions are made, while the data plane functions as the Policy Enforcement Point (PEP), where these policies are enforced during data transmission and access requests [8]. The PDP can be further broken down into the Policy Administrator (PA) and the Policy Engine (PE). Figure 1 portrays a simplified ZTA showing the delineation of control and forwarding/data plane. TIMeFoRCE is integrated into the PDP, where it evaluates and determines the outcomes of access requests.

3.2. *IoT Testbed*

For the study, the Linksys WRT 3200ACM, a dual-band MU-MIMO Gigabit Wi-Fi router [40] was utilized as the infrastructure, due to its architectural advantages, compatibility with OpenWRT, support for the Open vSwitch (OVS) package, and wireless isolation feature as outlined in [1]. For this study, primarily home and small-office IoT devices were the focus, as they are readily available as commercial-off-the-shelf (COTS) devices. A total of 26 IoT devices, representing various types, such as switches, plugs, light bulbs, sensors, media, Gateway hubs, and home assistants, typically deployed in real-world environments, comprise the testbed based on [1], are presented in Figure 2 and Table 2.

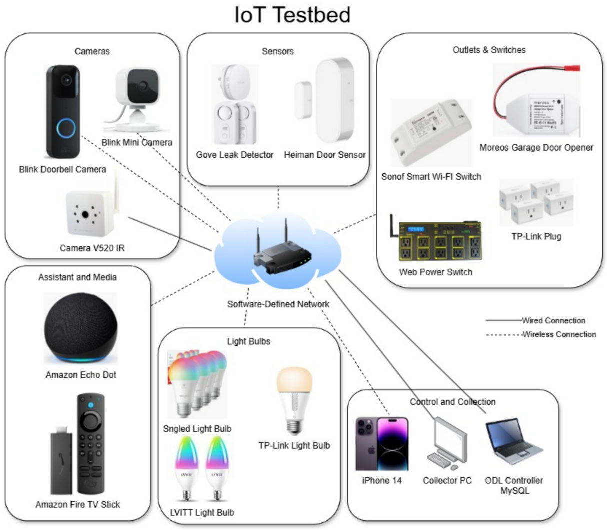


Figure 2: IoT testbed showing the various device types used in the experiment

Table 2: IoT devices in the testbed listing device type, quantity of each, and communication protocol

Device Name	Device Type	Quantity	Protocol
Amazon Alexa	Home Assistant	1	WiFi
Amazon Fire TV	Media	1	WiFi
Blink Doorbell Camera	Camera	1	WiFi
Blink Mini Camera	Camera	1	WiFi
Camera VR520 IR	Camera	1	Ethernet
Govee Gateway	Gateway	1	WiFi
Govee Leak Detector	Sensor	2	RF
Heiman Door Sensor	Sensor	3	WiFi
LVWIT Light Bulb	Light Bulb	2	WiFi
Sengled Light Bulb	Light Bulb	5	WiFi
Sonof WiFi Smart Switch	Smart Switch	1	WiFi
TP-Link Light Bulb	Light Bulb	1	WiFi
TP-Ling Plug	Smart Plug	4	WiFi
Web Power Switch	Smart Switch	1	WiFi

Figure 3 presents the internal architecture of the Linksys WRT 3200ACM after installing OVS and configuring the various ports. The Linksys WRT 3200ACM, when operating with the OpenWRT firmware version 22.03.2, utilized the Distributed Switching Architecture (DSA), which provides a Linux network interface for these user ports (lan1 - lan4 and wan), known as 'slave' interfaces [41], allowing the devices to function as a traditional layer two switch and forward all packets that it receives.

3.3. Data Collection and Analysis

3.3.1. Data Collection

Data collection began during the commissioning process of introducing the IoT devices to the testbed [1]. Data were collected for approximately 20 days over 3 months, with devices added at various stages of the capture period. During this stage, the required data volume was determined, and the most suitable features for subsequent analysis [1] were identified. Specifically, key packet header attributes essential for device identification were identified, along with additional features beyond the packet header, derived through feature engineering that could further improve this process.

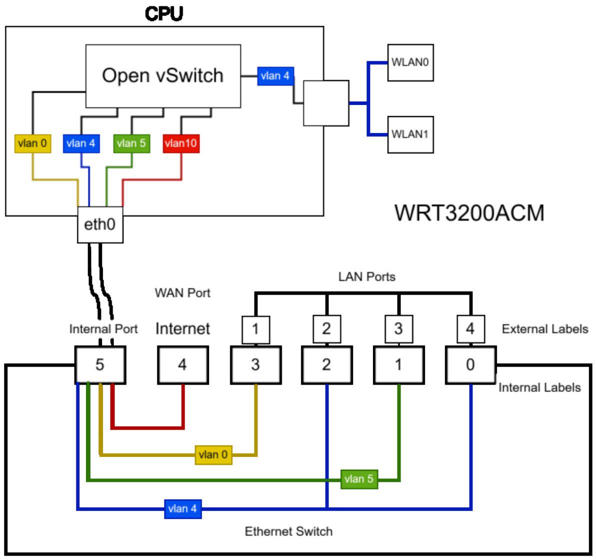


Figure 3: Linksys WRT 3200ACM Router showing the internal switch architecture after installing OVS

3.3.2. Data Analysis

Our intention was not to develop our own identification and classification algorithm but to leverage existing techniques based on previously completed scholarly work as a foundation for the research. We reviewed several such works that focused on packet-based classification [2, 28, 42, 20], as well as flow-based classification [22, 43]. The scope was narrowed to research on packet-based classification, specifically the work by [28]. However, gaps in the research prevented the use of their method as a viable solution in its current state, particularly when dealing with data the model has not encountered during its training phase. The authors in [1] provide a framework for the proposed method.

To address the limitations identified in [28], the feature set was expanded to include the z-scores of both packet size and payload for each packet. Additional binary features were introduced to indicate the presence of specific protocols, including Internet Protocol

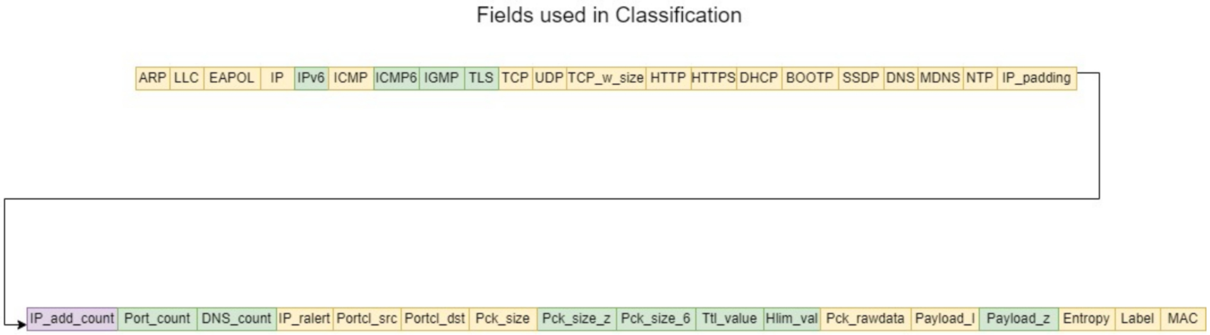


Figure 4: Feature set utilized in the classification process, showing original features and new features added.

version 6 (IPv6), Internet Group Management Protocol (IGMP), and Transport Layer Security (TLS). Furthermore, the Time-to-Live (TTL), Header Limit (HLIM), and packet size for IPv6 packets were incorporated. These IPv6-related features were added to enhance support for IPv6 traffic [1]. Figure 4 illustrates the complete set of features used in the classification process, with features from [28] shown in yellow, newly added features in green, and modified features in purple.

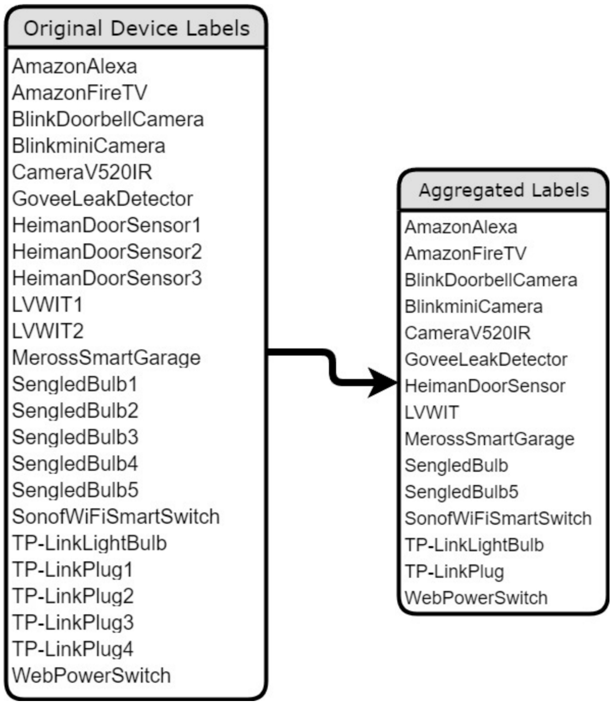


Figure 5: Original labels of devices and labels after grouping.

While adding the additional features allowed the model to perform well, utilizing the training and test data, and improving accuracy to over 99% in comparison to the 94% obtained in [28], it did not perform well with a validation set. Validation sets often contain data the trained model has not previously encountered. The validation set was a new dataset collected over eight days outside the training and test data. This is important because the identity and access management solution, TIMeFoRCE, will review this data to identify new IoT devices attempting to communicate on the network. The ”destination IP Address count” feature utilized in [28] resets to zero when parsing a new dataset. It is conceivable that, as the dataset size increases, the number of destination IP addresses will increase as well. Each time a new destination IP Address is detected, the count value is incremented by one (1). The issue is that it increments the value of the current row/packet. In a large dataset, this value can become substantial. As a result, the number of IP addresses associated with a device at the start of the dataset may differ significantly from the count observed later. This feature is essential in the classification process as a discriminatory value for each device. To overcome the limitation, we created a dictionary that maintains the counts obtained during the initial capture period. That dictionary is then used during the validation process. This dictionary updates the IP_add_count after parsing all the dataset packets. This is accomplished by using the dictionary values to update each row’s column values for specific MAC identifiers. Additionally, we added feature values for unique destination port counts and query name (QNAME) counts for each observed MAC address, further strengthening the model with more contextual information and better capturing patterns in the data. The QNAME is an important feature used during the discovery, classification, and addition of new devices. The following represents the collection of destination ports (dst_port), destination IPs (dst_ip), and query names (qname) per MAC Address, used to update their respective unique count columns.

Let:

- M be the set of unique MAC addresses.
- P_m be the set of unique destination ports visited by a MAC address m.
- I_m be the set of unique destination IPs visited by a MAC address m.
- Q_m be the set of unique query names resolved by a MAC address m.

For each observed packet t, extract:

$$(m, p, i, q) = (\text{MAC}, \text{dst\_port}, \text{dst\_ip}, \text{qname}) \tag{1}$$

$$P_m = P_m \cup \{p\}, \quad I_m = I_m \cup \{i\}, \quad Q_m = Q_m \cup \{q\} \tag{2}$$

For each MAC address m, we compute the number of unique values:

$$C_m^{IP} = |I_m|, \quad C_m^{Port} = |P_m|, \quad C_m^{Qname} = |Q_m| \tag{3}$$

where:

- C_m^{IP} represents the number of unique destination IPs visited by m.
- C_m^{Port} represents the number of unique destination ports contacted by m.
- C_m^{Qname} represents the number of unique query names resolved by m.

We achieved accuracy, precision, and recall above 99% using Random Forest (RF), as illustrated in Table 3 and based on the aggregated labels of identical devices as depicted in Figure 5. These are the labels obtained by combining identical devices into a single label. Furthermore, Figure 6 shows a comparison of dictionary utilization in TIMeFoRCE on the validation dataset. Utilizing a dictionary for the unique destination IPs and ports, and unique query names, provides for much greater accuracy. Figure 6a shows misclassifications for devices that share a similar manufacturer. For example, in Figure 6a, there is a high number of misclassifications. Several ”AmazonFireTV” devices are misclassified as ”AmazonAlexa” devices. This indicates that they share some behavioral traits. Figure 6b, on the other hand, shows that using the dictionary enables higher precision, recall, F1-score, and accuracy. Figure 6c and 6d show the classification report corresponding to Figure 6a and 6b, respectively.

Table 3: Classification Report using the aggregation method for identical model IoT Devices

Device	Precision	Recall	F1-Score	Support
AmazonAlexa	1.00	1.00	1.00	178483
AmazonFireTV	1.00	1.00	1.00	145786
BlinkDoorbellCamera	1.00	1.00	1.00	62534
BlinkminiCamera	1.00	1.00	1.00	32783
CameraV520IR	1.00	1.00	1.00	6445
GoveeLeakDetector	1.00	1.00	1.00	15755
HeimanDoorSensor	1.00	1.00	1.00	1227
LVWIT	1.00	1.00	1.00	9757
MerossSmartGarage	1.00	1.00	1.00	2938
SengledBulb	1.00	1.00	1.00	64959
SengledBulb5	1.00	1.00	1.00	2760
SonofWiFiSmart Switch	1.00	1.00	1.00	1231
TP-LinkLightBulb	1.00	1.00	1.00	3145
TP-LinkPlug	1.00	1.00	1.00	13944
WebPowerSwitch	1.00	1.00	1.00	1906
Accuracy			**1.00**	**543653**
Macro Avg	1.00	1.00	1.00	543653
Weighted Avg	1.00	1.00	1.00	543653

The classification process aims to generate unique device fingerprints, enabling each testbed device to be reliably identified by its intrinsic characteristics. Fingerprints are derived as follows. D is the set of all IoT devices. F is the set of all fingerprints. $d \in D$ is an individual IoT device represented by $f \in F$, a fingerprint. $\mathrm{MAC}(d) \in \{0,1\}^{48}$ is the 48-bit MAC address of device d while $\vec{h} = [h_1, h_2, \ldots, h_{16}] \in \mathbb{R}^{16}$ is a vector of 16 packet header features. $\phi : \{0,1\}^{48} \times \mathbb{R}^{16} \to F$ is the fingerprint generation function that maps $\psi : F \to D$ each fingerprint to its originating device.

The fingerprint is generated as follows:

$$f = \phi(\mathrm{MAC}(d), \vec{h}) \tag{4}$$

Where:

1. Each fingerprint is uniquely associated with a single device:
$$\forall f_1, f_2 \in F, \quad f_1 = f_2 \Rightarrow \psi(f_1) = \psi(f_2) \tag{5}$$
2. A single device may have multiple fingerprints:
$$\exists f_1, f_2 \in F, \; f_1 \neq f_2 \text{ such that } \psi(f_1) = \psi(f_2) = d \tag{6}$$
3. The fingerprint-to-device mapping is injective:
$$\forall f \in F, \quad \psi(f) = d \in D \tag{7}$$
4. The device-to-fingerprint mapping is one-to-many:
$$\forall d \in D, \quad |\psi^{-1}(d)| \geq 1 \tag{8}$$

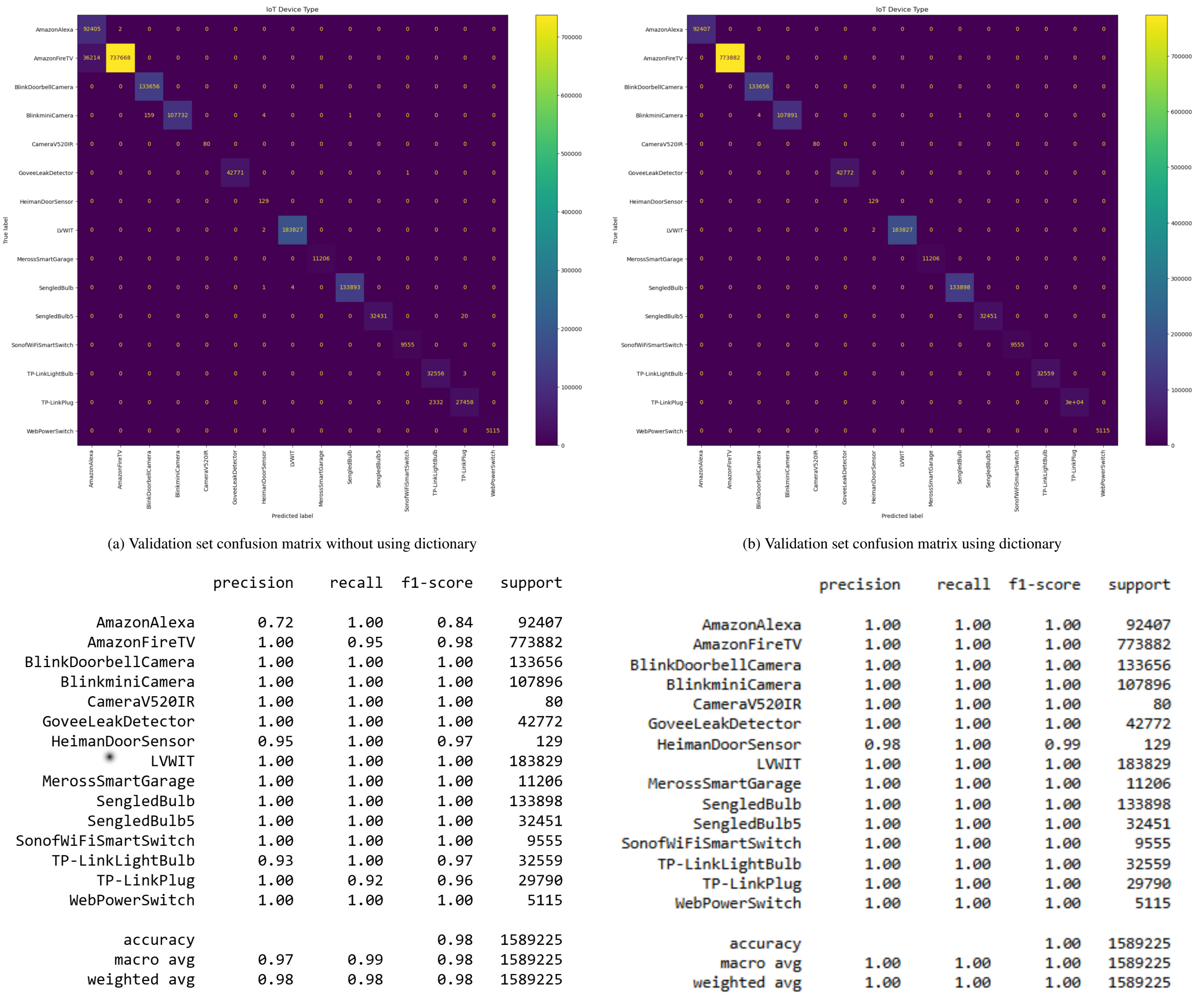
Resulting in:

$$\psi(\phi(\mathrm{MAC}(d), \vec{h})) = d \tag{9}$$

Table 4: Features used in the classification

Value Types	Feature names
Direct Values	TCP_w_size, Pck_size, Pck_size_6, payload_l
Binary Values	ARP, LLC, EAPOL, HTTP, HTTPS, DHCP, BOOTP, SSDP, DNS, MDNS, NTP, IP, IPV6, ICMP, ICMP6, IGMP, TLS TCP, UDP, IP_padding, IP_ralert, Pck_rawdata
Grouping	Portcl_src, Portcl_dst, Ttl_value, Hlim_value
Mathematical Equation	Entropy, Packet_size_z, Payload_z
Count	IP_add_count, Port_count, DNS_count

The classification was carried out using 36 features, an increase in the set of characteristics from the initial work in [1]. Table 4 presents the feature names and value types of each feature. It includes features obtained directly from the packet header as well as those derived from various feature engineering methods. Two new features added are the z-score of the packet and the payload, respectively. To obtain the z-score, the mean is needed. Given a sufficiently large dataset, the mean will reach a point where it



	AmazonAlexa	AmazonFireTV	BlinkDoorbellCamera	BlinkminiCamera	CameraV520IR	GoveeLeakDetector	HeimanDoorSensor	LVWIT	MerossSmartGarage	SengledBulb	SengledBulb5	SonofWiFiSmartSwitch	TP-LinkLightBulb	TP-LinkPlug	WebPowerSwitch
AmazonAlexa	92405	2	0	0	0	0	0	0	0	0	0	0	0	0	0
AmazonFireTV	36214	737668	0	0	0	0	0	0	0	0	0	0	0	0	0
BlinkDoorbellCamera	0	0	133656	0	0	0	0	0	0	0	0	0	0	0	0
BlinkminiCamera	0	0	159	107732	0	0	4	0	0	1	0	0	0	0	0
CameraV520IR	0	0	0	0	80	0	0	0	0	0	0	0	0	0	0
GoveeLeakDetector	0	0	0	0	0	42771	0	0	0	0	0	1	0	0	0
HeimanDoorSensor	0	0	0	0	0	0	129	0	0	0	0	0	0	0	0
LVWIT	0	0	0	0	0	0	2	183827	0	0	0	0	0	0	0
MerossSmartGarage	0	0	0	0	0	0	0	0	11206	0	0	0	0	0	0
SengledBulb	0	0	0	0	0	0	1	4	0	133893	0	0	0	0	0
SengledBulb5	0	0	0	0	0	0	0	0	0	0	32431	0	0	20	0
SonofWiFiSmartSwitch	0	0	0	0	0	0	0	0	0	0	0	9555	0	0	0
TP-LinkLightBulb	0	0	0	0	0	0	0	0	0	0	0	0	32556	3	0
TP-LinkPlug	0	0	0	0	0	0	0	0	0	0	0	0	2332	27458	0
WebPowerSwitch	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5115

(a) Validation set confusion matrix without using dictionary

	AmazonAlexa	AmazonFireTV	BlinkDoorbellCamera	BlinkminiCamera	CameraV520IR	GoveeLeakDetector	HeimanDoorSensor	LVWIT	MerossSmartGarage	SengledBulb	SengledBulb5	SonofWiFiSmartSwitch	TP-LinkLightBulb	TP-LinkPlug	WebPowerSwitch
AmazonAlexa	92407	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AmazonFireTV	0	773882	0	0	0	0	0	0	0	0	0	0	0	0	0
BlinkDoorbellCamera	0	0	133656	0	0	0	0	0	0	0	0	0	0	0	0
BlinkminiCamera	0	0	4	107891	0	0	0	0	0	1	0	0	0	0	0
CameraV520IR	0	0	0	0	80	0	0	0	0	0	0	0	0	0	0
GoveeLeakDetector	0	0	0	0	0	42772	0	0	0	0	0	0	0	0	0
HeimanDoorSensor	0	0	0	0	0	0	129	0	0	0	0	0	0	0	0
LVWIT	0	0	0	0	0	0	2	183827	0	0	0	0	0	0	0
MerossSmartGarage	0	0	0	0	0	0	0	0	11206	0	0	0	0	0	0
SengledBulb	0	0	0	0	0	0	0	0	0	133898	0	0	0	0	0
SengledBulb5	0	0	0	0	0	0	0	0	0	0	32451	0	0	0	0
SonofWiFiSmartSwitch	0	0	0	0	0	0	0	0	0	0	0	9555	0	0	0
TP-LinkLightBulb	0	0	0	0	0	0	0	0	0	0	0	0	32559	0	0
TP-LinkPlug	0	0	0	0	0	0	0	0	0	0	0	0	0	3e+04	0
WebPowerSwitch	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5115

(b) Validation set confusion matrix using dictionary

```
                      precision    recall  f1-score   support

         AmazonAlexa       0.72      1.00      0.84     92407
        AmazonFireTV       1.00      0.95      0.98    773882
  BlinkDoorbellCamera       1.00      1.00      1.00    133656
     BlinkminiCamera       1.00      1.00      1.00    107896
        CameraV520IR       1.00      1.00      1.00        80
   GoveeLeakDetector       1.00      1.00      1.00     42772
    HeimanDoorSensor       0.95      1.00      0.97       129
               LVWIT       1.00      1.00      1.00    183829
   MerossSmartGarage       1.00      1.00      1.00     11206
         SengledBulb       1.00      1.00      1.00    133898
        SengledBulb5       1.00      1.00      1.00     32451
 SonofWiFiSmartSwitch       1.00      1.00      1.00      9555
    TP-LinkLightBulb       0.93      1.00      0.97     32559
          TP-LinkPlug       1.00      0.92      0.96     29790
      WebPowerSwitch       1.00      1.00      1.00      5115

            accuracy                           0.98   1589225
           macro avg       0.97      0.99      0.98   1589225
        weighted avg       0.98      0.98      0.98   1589225
```

(c) Validation set classification report without using dictionary

```
                      precision    recall  f1-score   support

         AmazonAlexa       1.00      1.00      1.00     92407
        AmazonFireTV       1.00      1.00      1.00    773882
  BlinkDoorbellCamera       1.00      1.00      1.00    133656
     BlinkminiCamera       1.00      1.00      1.00    107896
        CameraV520IR       1.00      1.00      1.00        80
   GoveeLeakDetector       1.00      1.00      1.00     42772
    HeimanDoorSensor       0.98      1.00      0.99       129
               LVWIT       1.00      1.00      1.00    183829
   MerossSmartGarage       1.00      1.00      1.00     11206
         SengledBulb       1.00      1.00      1.00    133898
        SengledBulb5       1.00      1.00      1.00     32451
 SonofWiFiSmartSwitch       1.00      1.00      1.00      9555
    TP-LinkLightBulb       1.00      1.00      1.00     32559
          TP-LinkPlug       1.00      1.00      1.00     29790
      WebPowerSwitch       1.00      1.00      1.00      5115

            accuracy                           1.00   1589225
           macro avg       1.00      1.00      1.00   1589225
        weighted avg       1.00      1.00      1.00   1589225
```

(d) Validation set classification report using dictionary

Figure 6: Comparison of TIMeFoRCE classification performance on unseen data with and without dictionary usage

changes little, if at all, with the addition of more packets or rows. As a result, the dataset's mean, which contains approximately 10 million packets, will be used as the classification mean in future classification.

Suppose there are n samples $x_1, x_2, \ldots, x_n$, each in the interval $[a, b]$. Define the sample mean by

$$\bar{x}_n = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{10}$$

When a new sample $x_{n+1} \in [a, b]$ is added, the updated mean is

$$\bar{x}_{n+1} = \frac{1}{n+1} \sum_{i=1}^{n+1} x_i = \bar{x}_n + \frac{x_{n+1} - \bar{x}_n}{n+1}. \tag{11}$$

Hence, the difference between the new mean and the old mean is

$$\bar{x}_{n+1} - \bar{x}_n = \frac{x_{n+1} - \bar{x}_n}{n+1}. \tag{12}$$

Since $x_{n+1}, \bar{x}_n \in [a, b]$, the numerator $|x_{n+1} - \bar{x}_n|$ is at most $b - a$, yielding

$$|\bar{x}_{n+1} - \bar{x}_n| = \left| \frac{x_{n+1} - \bar{x}_n}{n+1} \right| \leq \frac{b-a}{n+1}. \tag{13}$$

As $n \to \infty$, we have $\frac{b-a}{n+1} \to 0$, implying that once the number of samples becomes large, adding additional values in the range $[a, b]$ changes the mean by a negligibly small amount.

These are all standard practices utilized by many researchers in the IoT device identification and classification space [28, 42, 44, 45].

Creating a fingerprint for a device is akin to building a user identity in Active Directory (AD) or Azure, and using the Lightweight

Directory Access Protocol (LDAP) or Security Assertion Markup Language (SAML) as the authentication protocol. Fingerprints are proactively added to the database by taking all packets captured during the learning phase and removing certain packets, such as ARP. ARP packets are identical for all devices except for the MAC address. The dataset was reduced by removing duplicate rows and extracting features to derive the fingerprint.

3.4. Implementation

The solution was implemented using the OpenDaylight (ODL) SDN controller [46], version 0.20.1 (code name Calcium). A Flask Web Server (FWS) [47] (version 3.0.3) was employed as the external application responsible for interacting with the associated Python scripts [48]. MySQL 8 [49] served as the database (DB) repository for storing device fingerprints. An ODL Java bundle was developed using Maven [50, 51], enabling streamlined portability across compatible SDN controllers.

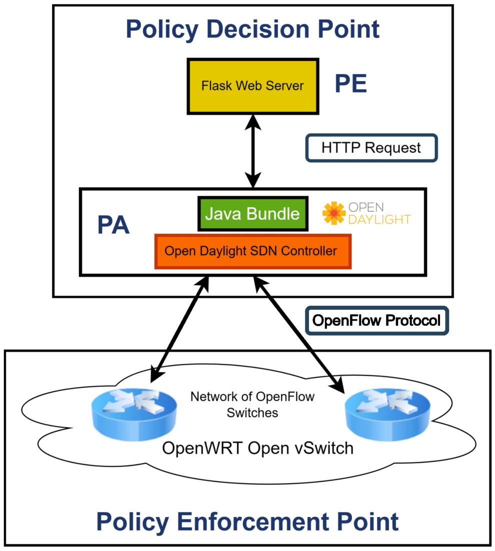


Figure 7: Proposed ZTA showing the various components that make up the architecture.

3.5. Experimental Setup

Multiple studies [27, 52, 53, 54, 55], have been completed utilizing Mininet as a virtual SDN switch; however, our study utilized the Linksys WRT 3200ACM because the testbed contains real-world devices that require a connection point [1]. A Samsung Notebook Model NP940X5J, Intel Core i7-4500U CPU @ 1.80 GHZ, 8 GB RAM with VirtualBox version 7.0.14 r161095 [56], and (2) Ubuntu 22.04.4 virtual machines functioned as the control plane of the architecture. We installed ODL Calcium 2024.01 [57] with three (3) GB RAM and one (1) CPU on the first virtual machine. MySQL 8 [49] was installed on the 2nd virtual machine with two (2) GB RAM and one (1) CPU. The MySQL 8 database was configured and populated with the fingerprints generated in section 3.3.2. This served as the fingerprints or the identity repository for the IoT devices in the testbed. Figure 7 presents the proposed ZTA showing the various components that are integrated into the PDP, while Figure 8 outlines the ZTA for the IAM solution (TIMeFoRCE), showing the process flows among the various components. The ODL controller, FWS, and DB makes up the PDP layer of the architecture. The OpenWRT OVS is the forwarding/data plane that forwards packets based on match criteria in the OpenFlow switch flow table. The control plane is the ODL SDN controller, which interacts with the OpenFlow switches via the OpenFlow protocol and the FWS. The ODL SDN controller operates via Java bundles, while the FWS uses a set of Python scripts. ODL functions as the policy administrator (PA), while FWS functions as the policy engine (PE), ultimately deciding whether to grant a given subject access to a resource. In this research, the decision is whether an IoT device is allowed to communicate.

4. Experimental Results

The previous section presented the methodology for our solution. We presented relevant literature on device identification, SDN, and Zero Trust. The team performed data analysis and completed the experimental setup. We investigated the various components that comprise the infrastructure to determine the optimal architecture. Additionally, we created the various Python scripts that the FWS (the PE) uses to perform the necessary operations. The fingerprints generated in section 3.3.2 were also added to the database. In this section, the focus is on results from the infrastructure's operational state. We examined the communication pattern of IoT devices. Additionally, several use cases are presented that demonstrate TiMeFoRCE's promise not only for authenticating network devices but also for providing access control.

4.1. Communication Patterns

An IoT device exhibits unique communication patterns. Some devices communicate quite frequently, even when not actively used, while others only communicate when a change-state event occurs. Table 5 provides the results for traffic collected over a five-hour period. 104783 packets were observed for the ”AmazonFireTV” with an average interval of 0.302449 milliseconds, a maximum interval of 38.528123 seconds, and a minimum interval of 0.0000001 seconds while it was not actively utilized. Compare this to the ”HeimanDoorSensor2,” which observed only 35 packets with an average interval of 0.350611 seconds, a maximum interval of 3.613992 seconds, and a minimum interval of 0.000626 seconds. While the ”HeimanDoorSensor2” may appear to have a lower maximum interval, it only measures traffic when it is activated; it did not see traffic again within the five-hour period. As a result, communication patterns cannot be garnered from devices like the HeimanDoorSensor2” as they operate on event-based triggers. Table 5 shows that devices can communicate from a few microseconds to several minutes.

4.2. Use Case

The following sections present three use cases for TIMeFoRCE: a legitimate device achieving successful authentication, an illegitimate device where authentication is rejected, and a new device being introduced into the network where classification is required.

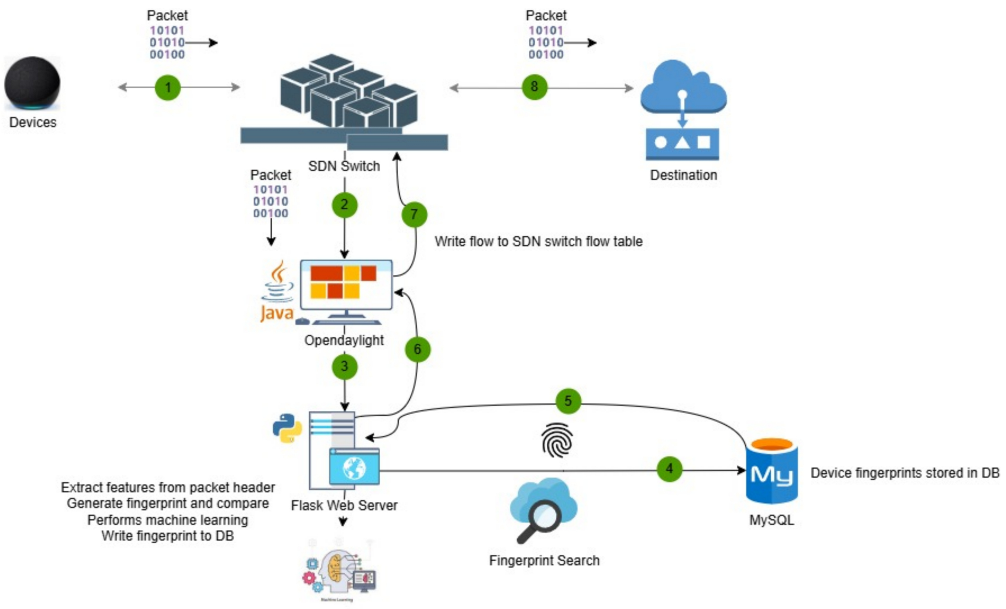


Figure 8: ZTA for Identity and Access Management (TIMeFoRCE) showing the various components that make up the architecture.

Table 5: Table showing device statistics for the number of packets observed, the average interval, the max interval, and the min interval between communication requests (intervals in seconds)

Device	Packets Sent	Avg Interval	Max Interval	Min Interval
BlinkminiCamera	8873	4.945252	65.825147	0.000005
SengledBulb2	4819	9.106290	60.037668	0.000051
SengledBulb4	8686	5.048959	49.121616	0.000054
SengledBulb3	5164	8.496532	42.142203	0.000061
AmazonAlexa	92001	0.476809	30.582975	0.000000
SengledBulb5	5289	8.272675	133.843222	0.000073
TP-LinkLightBulb	1984	22.071131	319.312205	0.000021
WebPowerSwitch	777	56.293847	502.737341	0.000066
SonofWiFiSmartSwitch	1363	32.197260	186.069789	0.000069
TP-LinkPlug1	1062	41.211247	245.402278	0.000196
TP-LinkPlug3	2612	16.782434	90.460782	0.000128
GoveeLeakDetector	3954	11.080574	102.742173	0.000147
TP-LinkPlug2	2651	16.529848	94.959352	0.000629
TP-LinkPlug4	2576	17.023061	88.578222	0.000050
MerossSmartGarage	1230	35.631615	191.624964	0.000284
BlinkDoorbellCamera	14404	3.024163	615.469474	0.000006
AmazonFireTV	104783	0.302449	38.528123	0.000000
HeimanDoorSensor2	35	0.350611	3.613992	0.000626
LVWIT1	2929	3.534108	10.117231	0.000145

4.2.1. Legitimate Device

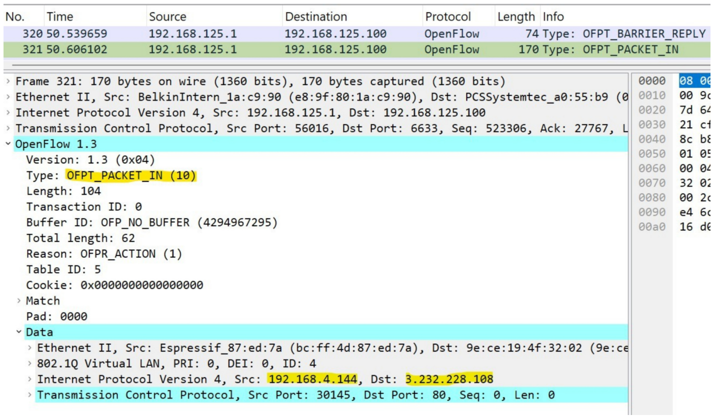
In this use case, with the ODL SDN controller, Java bundle, and FWS activated, which serves as the PDP and the central components of TIMeFoRCE, the device attempting to communicate sends a packet on the network. The OpenFlow switch or the PEP, which operates based on reactive flows, collects the packet and attempts to match it against its flow table. Due to the absence of a matching flow and a static flow rule directing certain packets to the PDP, the PEP sends the received packet to the PDP as an OFP_PACKET_IN message via the Southbound application programming interface (API). This is the original packet encapsulated in an OpenFlow packet, as illustrated in Figure 9.

The SDN controller using the Packet-Forwarder bundle (the PA) we created extracts the payload (the original packet ”data”) from the OpenFlow packet as a byte string, ensuring it is not a null packet. It then sends the extracted OpenFlow packet payload, ”the data,” to the PE via an HTTP Post request using the Northbound API. The PE receives the payload string and parses the string to obtain the packet header information. It then performs the sequence of activities as illustrated in Figure 10.

The primary activity is to generate a fingerprint from the parsed

packet header data as this is the initial task performed by TIMeFoRCE. We defined the fingerprint generated as follows:
Let $\vec{x}_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{i16}) \in \{0,1\}^{16}$ represent the 16 binary features extracted from the packet header of device i.
Let $\mathrm{MAC}_i \in \{0,1\}^{48}$ denote the 48-bit Ethernet source address of device i, obtained from the packet header.
We define the fingerprint f_i of device i as the binary concatenation of the MAC address and the feature vector:

$$f_i = \mathrm{MAC}_i \parallel \vec{x}_i \tag{14}$$



```
No.   Time       Source          Destination      Protocol  Length Info
  320 50.539659  192.168.125.1   192.168.125.100  OpenFlow      74 Type: OFPT_BARRIER_REPLY
  321 50.606102  192.168.125.1   192.168.125.100  OpenFlow     170 Type: OFPT_PACKET_IN

› Frame 321: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits)
› Ethernet II, Src: BelkinIntern_1a:c9:90 (e8:9f:80:1a:c9:90), Dst: PCSSystemtec_a0:55:b9 (0
› Internet Protocol Version 4, Src: 192.168.125.1, Dst: 192.168.125.100
› Transmission Control Protocol, Src Port: 56016, Dst Port: 6633, Seq: 523306, Ack: 27767, L
˅ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_PACKET_IN (10)
    Length: 104
    Transaction ID: 0
    Buffer ID: OFP_NO_BUFFER (4294967295)
    Total length: 62
    Reason: OFPR_ACTION (1)
    Table ID: 5
    Cookie: 0x0000000000000000
  › Match
    Pad: 0000
  ˅ Data
    › Ethernet II, Src: Espressif_87:ed:7a (bc:ff:4d:87:ed:7a), Dst: 9e:ce:19:4f:32:02 (9e:ce
    › 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 4
    › Internet Protocol Version 4, Src: 192.168.4.144, Dst: 3.232.228.108
    › Transmission Control Protocol, Src Port: 30145, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  08 0
0010  00 9
0020  7d 6
0030  21 c
0040  8c b
0050  01 0
0060  00 0
0070  32 0
0080  00 2
0090  e4 6
00a0  16 d
```

Figure 9: OpenFlow packet (OFPT_PACKET_IN) captured in Wireshark

Figure 11 presents a sample of fingerprints in the database representing the fingerprintID and deviceID for four distinct devices. A device is not limited to the number of fingerprints that it can have. The 16 features used to generate the fingerprints are listed in Table 6.

TIMeFoRCE's PE performs a series of activities. The series of activities includes parsing packets, generating fingerprints, creating a dataframe, querying the database, classifying packets, gathering values to send back to the PA, adding fingerprints to the database, generating flow IDs, and applying the table ID. These activities include:

- Parsing packets - This script parses the packet header into its various features.
- Generating fingerprints - This script generates a fingerprint by combining the MAC address and 16 binary features from the packet header.
- Create dataframe - This script creates a dataframe, which is used to classify new devices introduced into the network.
- Query the database - This script queries the database to obtain a matching fingerprint and matching deviceID.
- Classify packet - This script performs classification utilizing the previous model developed in the data analysis phase.
- Gather values to send back to PA - This script extracts information from the packet header to be sent back to the PA (i.e., the ODL controller). The values extracted from the packet header include table ID, VLAN ID, source MAC address, destination MAC address, source IP address, destination IP address, source port, and destination port.
- Generate flow ID - The PE (i.e., FWS) generates a unique flow ID that is attached to each flow. This is a randomly generated 6-digit hexadecimal string.
- Applying table ID - The PE returns the table ID value to the PA. In our example, the table ID is 5.

Table 6: Features used in fingerprint

Protocols Layers	**Protocols**
Layer 1 Protocol	MAC
Layer 2 Protocols	ARP, LLC
Layer 3 Protocols	IP, IPV6, ICMP, ICMP6
Layer 4 Protocols	TCP, UDP
Layer 7 protocols	HTTP, HTTPS, DHCP, BOOTP, SSDP, DNS, MDNS, NTP

These activities performed vary based on the observed condition of the device attempting to communicate (new vs. existing).

The MAC $M_s = \mathrm{MAC}_{\mathrm{src}}(P)$ and features $\mathbf{f} \in \{0,1\}^{16}$ are obtained from the header $h(P)$ of the incoming packet P and compared to a database of know fingerprints $\mathcal{F}$, each of the form $(M_i, \mathbf{f}_i)$. Therefore, the fingerprint of P is:

$$\mathrm{FP}(P) = (M_s, \mathbf{f}) \tag{15}$$

Define by the following match flag:

$$F = \begin{cases} 1, & \text{if } \mathrm{FP}(P) \in \mathcal{F} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

Let M_d be the destination MAC address. If $F = 1$, install bi-directional flows:

$$\mathrm{InstallFlow}(M_s, M_d)\mathrm{InstallFlow}(M_d, M_s) \tag{17}$$

Upon completing the required processing, the PE returns a set of parsed values to the PA. Figure 12 illustrates the output corresponding to a successfully extracted fingerprint. The returned values include the source and destination Ethernet (MAC) addresses, source and destination IP addresses, source and destination transport-layer ports, the protocol identifier, the flow ID, and the Table ID. The flow ID is a unique, randomly generated six-digit hexadecimal string. For reverse flows, an appended character "r" is used to denote the reversed direction of the same flow identifier.

The Packet-Forwarder bundle (the PA) uses the parsed values to update the PEP flow table with bidirectional flows, as illustrated in Figure 13.

$$\begin{aligned} \mathrm{InstallFlow}(M_s, M_d) &= \{\text{cookie} = c,\ \text{table\_id} = t,\ \ldots\} \\ \mathrm{InstallFlow}(M_d, M_s) &= \{\text{cookie} = c',\ \text{table\_id} = t',\ \ldots\} \end{aligned} \tag{18}$$

It adds a flow based on the original packet and a flow with the parsed values in reverse. SDN switches operate based on bi-directional flows. Therefore, the PA must add at least two flow rules to a PEP

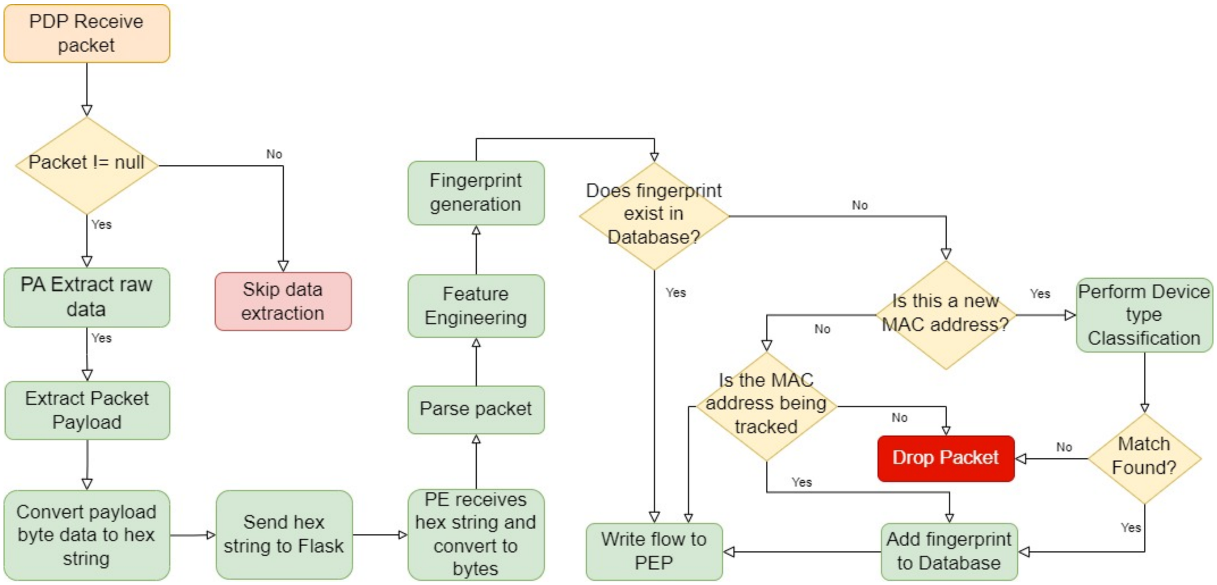


Figure 10: Flowchart depicting the sequence of activities performed by the PDP (PA and PE)

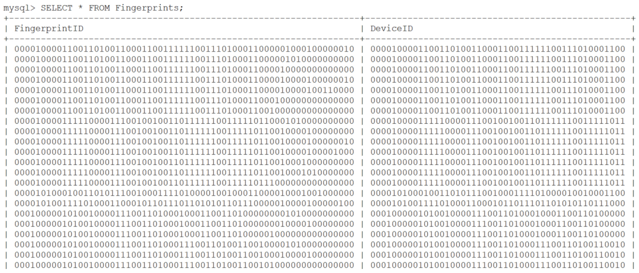

```
mysql> SELECT * FROM Fingerprints;
+------------------------------------------------------------------+--------------------------------------------------+
| FingerprintID                                                    | DeviceID                                         |
+------------------------------------------------------------------+--------------------------------------------------+
| 0000100001100110100110001100111111001110100011000001000100000010 | 000010000110011010011000110011111100111010001100 |
| 0000100001100110100110001100111111001110100011000001010000000000 | 000010000110011010011000110011111100111010001100 |
| 0000100001100110100110001100111111001110100011000010000000000000 | 000010000110011010011000110011111100111010001100 |
| 0000100001100110100110001100111111001110100011000010000100000010 | 000010000110011010011000110011111100111010001100 |
| 0000100001100110100110001100111111001110100011000010000100110000 | 000010000110011010011000110011111100111010001100 |
| 0000100001100110100110001100111111001110100011000100000000000000 | 000010000110011010011000110011111100111010001100 |
| 0000100001100110100110001100111111001110100011001000000000000000 | 000010000110011010011000110011111100111010001100 |
| 0000100001111100001110010010011011111100111110110001010000000000 | 000010000111110000111001001001101111110011111011 |
| 0000100001111100001110010010011011111100111110110010000100000000 | 000010000111110000111001001001101111110011111011 |
| 0000100001111100001110010010011011111100111110110010000100000010 | 000010000111110000111001001001101111110011111011 |
| 0000100001111100001110010010011011111100111110110010000100001000 | 000010000111110000111001001001101111110011111011 |
| 0000100001111100001110010010011011111100111110110010001000000000 | 000010000111110000111001001001101111110011111011 |
| 0000100001111100001110010010011011111100111110110010001010000000 | 000010000111110000111001001001101111110011111011 |
| 0000100001111100001110010010011011111100111110111000000000000000 | 000010000111110000111001001001101111110011111011 |
| 0000101000100110101110010001111010000100100011000010001001000000 | 000010100010011010111001000111101000010010001100 |
| 0000101001111010001100010110111011010101101110000010000100000100 | 000010100111101000110001011011101101010110111000 |
| 0001000000101001000011100110100010001100110100000000101000000000 | 000100000010100100001110011010001000110011010000 |
| 0001000000101001000011100110100010001100110100000001000010000000 | 000100000010100100001110011010001000110011010000 |
| 0001000000101001000011100110100010001100110100000100000000000000 | 000100000010100100001110011010001000110011010000 |
| 0001000000101001000011100110100011100110100110010000101000000000 | 000100000010100100001110011010001110011010011001 |
| 0001000000101001000011100110100011100110100110010001000010000000 | 000100000010100100001110011010001110011010011001 |
| 0001000000101001000011100110100011100110100110010100000000000000 | 000100000010100100001110011010001110011010011001 |
```

Figure 11: Sample of fingerprints in the database

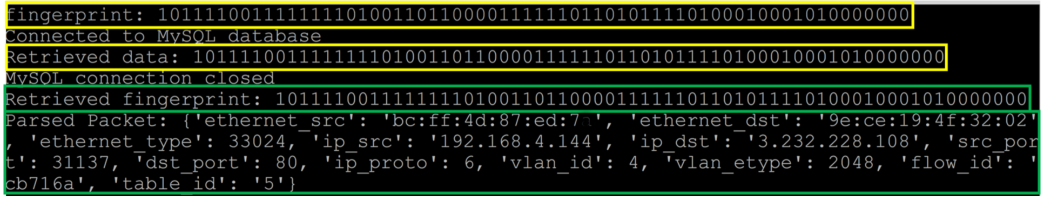

```
fingerprint: 1011110011111111101001101100001111110110101111010001000101000000
Connected to MySQL database
Retrieved data: 1011110011111111101001101100001111110110101111010001000101000000
MySQL connection closed
Retrieved fingerprint: 1011110011111111101001101100001111110110101111010001000101000000
Parsed Packet: {'ethernet_src': 'bc:ff:4d:87:ed:7 ', 'ethernet_dst': '9e:ce:19:4f:32:02'
, 'ethernet_type': 33024, 'ip_src': '192.168.4.144', 'ip_dst': '3.232.228.108', 'src_por
t': 31137, 'dst_port': 80, 'ip_proto': 6, 'vlan_id': 4, 'vlan_etype': 2048, 'flow_id': '
cb716a', 'table_id': '5'}
```

Figure 12: Show a successful fingerprint retrieval from the database

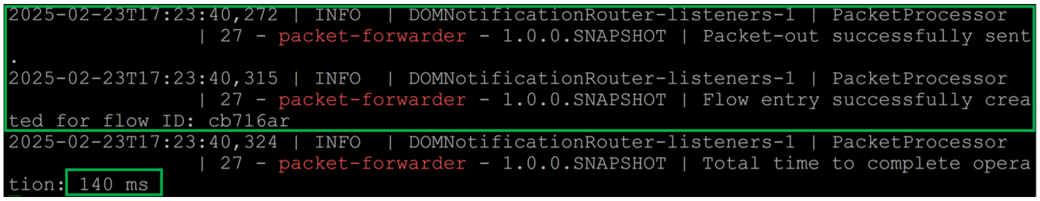

```
2025-02-23T17:23:40,272 | INFO  | DOMNotificationRouter-listeners-1 | PacketProcessor
                         | 27 - packet-forwarder - 1.0.0.SNAPSHOT | Packet-out successfully sent
.
2025-02-23T17:23:40,315 | INFO  | DOMNotificationRouter-listeners-1 | PacketProcessor
                         | 27 - packet-forwarder - 1.0.0.SNAPSHOT | Flow entry successfully crea
ted for flow ID: cb716ar
2025-02-23T17:23:40,324 | INFO  | DOMNotificationRouter-listeners-1 | PacketProcessor
                         | 27 - packet-forwarder - 1.0.0.SNAPSHOT | Total time to complete opera
tion: 140 ms
```

Figure 13: Shows a successful flow installation of original and reverse flow

flow table to receive return traffic. The PA adds a cookie ID to each flow and two timeout values to the flow rule: a hard timeout and a soft timeout. The PA uses the cookie ID to remove flows from the PEP after they expire, and the PA receives a FLOW_REMOVED message. The soft timeout removes the flow if no packet is received at the network device with matching criteria within a specified interval. In contrast, the hard timeout removes the flow rule after a specified amount of time that the network programmer sets, ensuring that devices will be re-authenticated periodically, fulfilling the time-based authentication requirement of a ZTA. In the experimental testing, the value of the hard timeout was twice the soft timeout. We performed several iterations to determine the optimal time for soft and hard timeouts. Traffic was captured for 20 minutes at each interval. We measured the flow table size at each interval. The bidirectional flows are written with the following parameters: cookie, table ID, idle_timeout, hard_timeout, send_flow_rem priority, Protocol, dl_vlan, dl_src, dl_dst, nw_src, nw_dst, tp_src, tp_dst, and the actions to take for the specified packet. The initial packets sent by a device provide authentication, while the flow(s) written to the PEP provide access control for the specific device. The initial packet is the only packet sent to the PDP during a communication request; subsequent packets bypass this process until the timeout expires.

The previous section outlines the steps for retrieving a successful fingerprint. Subsequent sections will present cases where the application did not obtain a successful fingerprint. This will result in one of two activities. First, if a fingerprint is not successfully identified, TIMeFoRCE will not allow communication for the specific traffic type. Second, TIMeFoRCE will further examine the device to determine if its MAC address is not in the DB.

4.2.2. Unsuccessful Fingerprint

The application performs the initial activities as outlined in 4.2.1 and Figure 10. In this scenario, a fingerprint was not found in the database. Figure 14 shows that it did not find a matching fingerprint. The ”Retrieved fingerprint,” which denotes the fingerprint returned by the query, is a 64-bit string of all zeros.

$$\mathrm{FP}(P) = \mathbf{0}_{64} \tag{19}$$

This indicates that it did not find a matching fingerprint. The next activity involves checking to see if the device has an existing presence in the database $\mathcal{D}$:

$$\mathrm{IsKnown}(M_s) = \begin{cases} 1, & \text{if } M_s \in \mathcal{D} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

The ”Retrieved data” indicates that the device is present in the database. As a result, the packet is determined not to originate from a legitimate device, or the fingerprint generated for the specific device does not match an allowed authentication fingerprint. If $\mathrm{FP}(P) = \mathbf{0}_{64}$ and $\mathrm{IsKnown}(M_s) = 1$, the device is not deemed legitimate:

$$\mathrm{Legitimate}(M_s) = 0 \tag{21}$$

The PE returns an error to the PA, which, in essence, is ”null” values for the parsed data; this, in turn, skips the flow entry update of the SDN switch flow table, and the packet is subsequently dropped. Figure 15 presents the output of the PA for an unsuccessful fingerprint or illegitimate device as indicated by the skipping of the flow and reverse flow entry update.

4.2.3. New Device

The previous sections outline cases in which a fingerprint was successfully retrieved and a case in which it was not, for a device. Section 4.2.2 presents a case where a successful fingerprint was not identified. Part of the activities outlined in Figure 10 and 14 was to determine if the device attempting to communicate is a ”new device”. This is accomplished by checking the database for the device’s MAC address. The device’s existence in the database indicates that it is not a new device; however, $\mathrm{IsKnown}(M_s) = 0$, which demonstrates that it is, in fact, a new device. Designating a new device does not determine whether it is legitimate. Further assessment is needed.

As outlined in Figure 10, one of the questions that is asked is whether or not this is a new MAC Address. If it is determined that $\mathrm{IsKnown}(M_s) = 0$, the application will attempt to perform device type classification. The PE first extracts the features from the packet header with an initial label of ”Unknown. Before device classification, it is essential to establish a ground truth reference to determine the best match accurately. Specific values result from counting and cannot be obtained from a single packet. As a result, the packet is checked against a list developed during model creation to identify specific conditions. Similar IoT devices should have similar behaviors. For example, all Amazon Alexa devices will connect to the same or a similar cloud provider. Similar devices use similar destination ports and DNS names, as presented in [22]. Using a structured approach, we can examine the similarities shared among devices and update the IP_add_count, the port_count, and the qname_count based on devices that share those traits. We used the following conditions to determine the values to assign to the device prior to classification. The match condition function C is defined as:

$$C = \begin{cases} 1 & \text{if Full MAC matches} \\ 2 & \text{if OUI} \wedge \mathrm{IP}_d \wedge P_d \wedge Q \text{ match} \\ 3 & \text{if OUI} \wedge \mathrm{IP}_d \wedge P_d \text{ match} \\ 4 & \text{if } \mathrm{IP}_d \wedge P_d \wedge Q \text{ match} \\ 5 & \text{if OUI} \wedge \mathrm{IP}_d \wedge Q \text{ match} \\ 6 & \text{if OUI} \wedge \mathrm{IP}_d \text{ match} \\ 7 & \text{if OUI} \wedge P_d \wedge Q \text{ match} \\ 8 & \text{if OUI} \wedge P_d \text{ match} \\ 9 & \text{if OUI} \wedge Q \text{ match} \\ 10 & \text{if } \mathrm{IP}_d \wedge P_d \text{ match} \\ 11 & \text{if } Q \text{ matches only} \\ 12 & \text{if OUI matches only} \\ 13 & \text{otherwise (default case)} \end{cases}$$

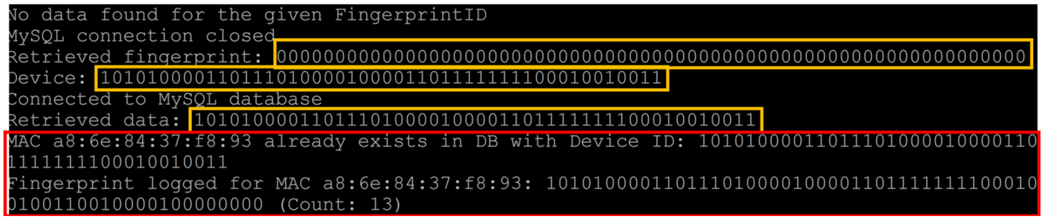
where:

- $\mathrm{MAC} \in \{0, 1\}^{48}$: Full MAC address
- $\mathrm{OUI} = \mathrm{MAC}_{[0:24]}$: First 24 bits of MAC address
- IP_d: Destination IP address
- P_d: Destination port
- Q: DNS QNAME
- $C \in \{1, 2, \ldots, 13\}$: Match condition code

The packet parsing script iterates through the conditions to find the best possible match. Matching a condition does not automatically guarantee that a device will match the type whose condition is matched during the classification process.

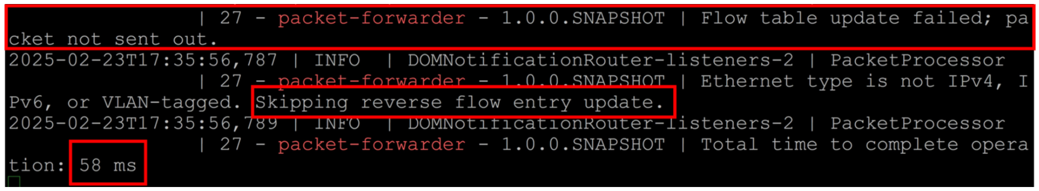
In this case, the MAC is not known, and no fingerprint is found, so device classification is performed:

$$\text{if } \mathrm{FP}(P) = \mathbf{0}_{64} \wedge \mathrm{IsKnown}(M_s) = 0 \Rightarrow \mathrm{Classify}(M_s) \tag{22}$$



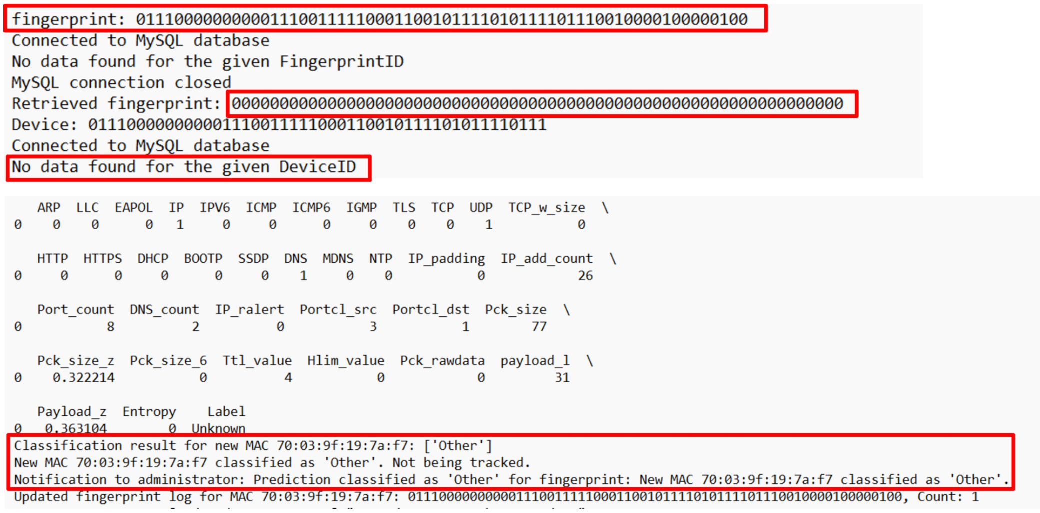
```
No data found for the given FingerprintID
MySQL connection closed
Retrieved fingerprint: 0000000000000000000000000000000000000000000000000000000000000000
Device: 1010100001101110100001000011011111111100010010011
Connected to MySQL database
Retrieved data: 1010100001101110100001000011011111111100010010011
MAC a8:6e:84:37:f8:93 already exists in DB with Device ID: 101010000110111010000100000110
1111111100010010011
Fingerprint logged for MAC a8:6e:84:37:f8:93: 1010100001101110100001000011011111111100010
0100110010000100000000 (Count: 13)
```

Figure 14: Depicts an unsuccessful fingerprint retrieval



```
                  | 27 - packet-forwarder - 1.0.0.SNAPSHOT | Flow table update failed; pa
cket not sent out.
2025-02-23T17:35:56,787 | INFO  | DOMNotificationRouter-listeners-2 | PacketProcessor
                  | 27 - packet-forwarder - 1.0.0.SNAPSHOT | Ethernet type is not IPv4, I
Pv6, or VLAN-tagged. Skipping reverse flow entry update.
2025-02-23T17:35:56,789 | INFO  | DOMNotificationRouter-listeners-2 | PacketProcessor
                  | 27 - packet-forwarder - 1.0.0.SNAPSHOT | Total time to complete opera
tion: 58 ms
```

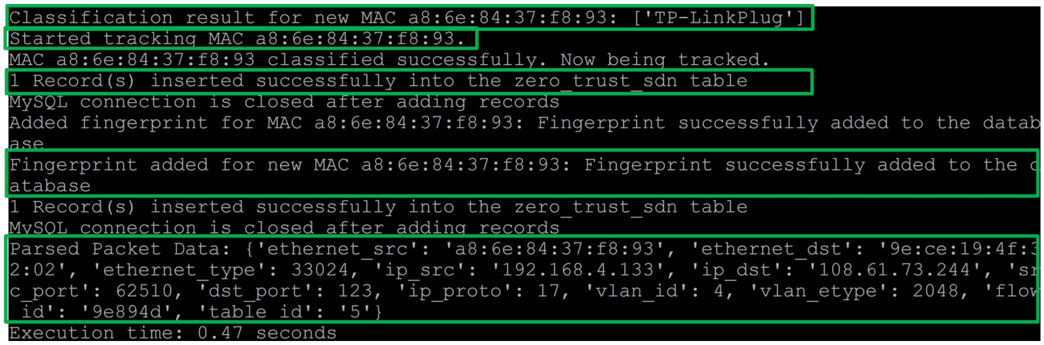
Figure 15: Depicts an unsuccessful fingerprint output message from ODL



```
fingerprint: 0111000000000011100111110001100101111010111101110010000100000100
Connected to MySQL database
No data found for the given FingerprintID
MySQL connection closed
Retrieved fingerprint: 0000000000000000000000000000000000000000000000000000000000000000
Device: 011100000000001110011111000110010111101011110111
Connected to MySQL database
No data found for the given DeviceID
```

```
   ARP  LLC  EAPOL  IP  IPV6  ICMP  ICMP6  IGMP  TLS  TCP  UDP  TCP_w_size  \
0    0    0      0   1     0     0      0     0    0    0    1           0

   HTTP  HTTPS  DHCP  BOOTP  SSDP  DNS  MDNS  NTP  IP_padding  IP_add_count  \
0     0      0     0      0     0    1     0    0           0            26

   Port_count  DNS_count  IP_ralert  Portcl_src  Portcl_dst  Pck_size  \
0           8          2          0           3           1        77

   Pck_size_z  Pck_size_6  Ttl_value  Hlim_value  Pck_rawdata  payload_l  \
0    0.322214           0          4           0            0         31

   Payload_z  Entropy    Label
0   0.363104        0  Unknown
Classification result for new MAC 70:03:9f:19:7a:f7: ['Other']
New MAC 70:03:9f:19:7a:f7 classified as 'Other'. Not being tracked.
Notification to administrator: Prediction classified as 'Other' for fingerprint: New MAC 70:03:9f:19:7a:f7 classified as 'Other'.
Updated fingerprint log for MAC 70:03:9f:19:7a:f7: 0111000000000011100111110001100101111010111101110010000100000100, Count: 1
```

Figure 16: Show a newly discovered device classified as ”Other”



```
Classification result for new MAC a8:6e:84:37:f8:93: ['TP-LinkPlug']
Started tracking MAC a8:6e:84:37:f8:93.
MAC a8:6e:84:37:f8:93 classified successfully. Now being tracked.
1 Record(s) inserted successfully into the zero_trust_sdn table
MySQL connection is closed after adding records
Added fingerprint for MAC a8:6e:84:37:f8:93: Fingerprint successfully added to the datab
ase
Fingerprint added for new MAC a8:6e:84:37:f8:93: Fingerprint successfully added to the d
atabase
1 Record(s) inserted successfully into the zero_trust_sdn table
MySQL connection is closed after adding records
Parsed Packet Data: {'ethernet_src': 'a8:6e:84:37:f8:93', 'ethernet_dst': '9e:ce:19:4f:3
2:02', 'ethernet_type': 33024, 'ip_src': '192.168.4.133', 'ip_dst': '108.61.73.244', 'sr
c_port': 62510, 'dst_port': 123, 'ip_proto': 17, 'vlan_id': 4, 'vlan_etype': 2048, 'flow
_id': '9e894d', 'table_id': '5'}
Execution time: 0.47 seconds
```

Figure 17: Show a newly discovered device being successfully classified

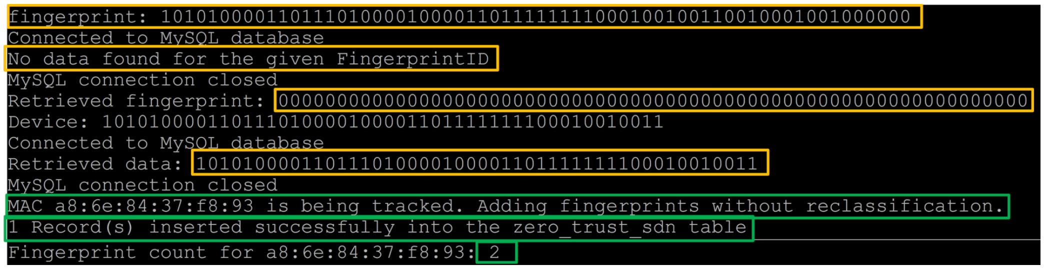
Classification is performed using a pre-trained model $\mathcal{M}$ and a condition list C:

$$\text{Classify}(M_s) = \begin{cases} \mathcal{M}(\mathbf{x}) \wedge C(\mathbf{x}), & \text{if predict\_proba}(\mathcal{M}, \mathbf{x}) \geq 0.75 \\ \text{reject}, & \text{otherwise} \end{cases} \tag{23}$$

where $\mathbf{x}$ is the feature vector extracted from P (e.g., header-based features):

$$\mathcal{D} \leftarrow \mathcal{D} \cup \{(M_s, \mathbf{f}_0)\} \tag{24}$$

In this particular example, we are adding a TP-Link Smart Plug model HS103. This device matches the device type (make and model) of a device already classified and operating within the ZTA. Device type classification will attempt to match the newly identified device type to an existing one in the database. A model was developed and stored as a pickle file during the data analysis phase.



```
fingerprint: 1010100001101110100001000011011111111000100100110010001001000000
Connected to MySQL database
No data found for the given FingerprintID
MySQL connection closed
Retrieved fingerprint: 0000000000000000000000000000000000000000000000000000000000000000
Device: 101010000110111010000100001101111111100010010011
Connected to MySQL database
Retrieved data: 101010000110111010000100001101111111100010010011
MySQL connection closed
MAC a8:6e:84:37:f8:93 is being tracked. Adding fingerprints without reclassification.
1 Record(s) inserted successfully into the zero_trust_sdn table
Fingerprint count for a8:6e:84:37:f8:93: 2
```

Figure 18: Show fingerprint added to the database for a newly discovered device where classification is bypassed

This model will be used during classification. Additionally, we used the Random Forest (RF) predict_proba function with a probability_threshold of 0.75 to identify newly discovered devices. If a newly discovered device does not fall within this threshold, it is assigned a classification of ”Other” during classification.

$$\text{Classify}(M_s) = \text{"Other"} \tag{25}$$

A classification label of ”Other” indicates that it was not able to successfully match the device against one that already exists via the model. In this scenario, a flow is not installed in the SDN switch’s flow table representing the PEP. Figure 16 presents the results of a failed classification. This results from the absence of the device type in the classification model. Devices labeled ”Other” are either illegitimate or exhibit unrecognized behavioral patterns. Thus:

$$\text{if Classify}(M_s) = \text{"Other"} \Rightarrow \text{RequiresManualReview}(M_s) = 1$$

The system then notifies the administrator:

$$\text{NotifyAdministrator}(M_s) \tag{26}$$

However, if the newly discovered device matches an existing device via the classification model, its fingerprint is added to the database for future use. Figure 17 presents the results of a successful classification. Additionally, the application will only allow one fingerprint to be added to the database. Subsequent unique fingerprints generated from the same MAC Address will be denied access due to the absence of matching fingerprints, but the MAC Address will still exist as outlined in the previous section. To overcome this limitation in the application, once a new device is successfully classified, the MAC address is tracked, and the first six fingerprints generated from the newly discovered device are automatically added to the database, eliminating the classification process as long as they are received during the first five minutes of the device being discovered and correctly classified.

$$\begin{aligned} \text{If } t_i \in [t_0, t_0 + \Delta t] \text{ and } \mathbf{f}_i \notin \mathcal{D}(M_s), \text{ then } \mathcal{D}(M_s) \\ \leftarrow \mathcal{D}(M_s) \cup \{\mathbf{f}_i\}, \quad \text{for } i = 1, \ldots, N \end{aligned} \tag{27}$$

We chose six fingerprints as most devices in our environment show six fingerprints or fewer in the database. Figure 18 presents the results of adding a second fingerprint following this process and bypassing device classification. Additionally, two log files are maintained that store fingerprints generated during unsuccessful fingerprint retrievals and device classifications.

4.3. Flow Table Size

Flow table overflow is always a pressing concern in any SDN architecture. TIMeFoRCE is discriminatory and restrictive, where each flow is required to match multiple criteria. For example, a flow that matches on source MAC address would only need two flows for all destinations. A flow that matches the source and destination MAC addresses will need two flows for each MAC address combination. Extrapolate that to the source and destination MAC addresses, the source and destination IP addresses, and the source and destination ports, and the exponential growth in the flow table size becomes apparent. An SDN switch flow table grows exponentially as more fields are added to the match criteria, leading to more flow table entries.
We derive a formula to illustrate this growth.

Let:

- N_s be the number of source MAC addresses,
- N_d be the number of destination MAC addresses,
- N_{IP_s} be the number of source IP addresses,
- N_{IP_d} be the number of destination IP addresses,
- N_{port_s} be the number of source ports,
- N_{port_d} be the number of destination ports.

If a flow matches only the source MAC address, then a single flow rule can handle all destinations:

$$F = N_s \tag{28}$$

When matching on both source and destination MAC addresses, the required flow table size increases to:

$$F = N_s \times N_d \tag{29}$$

When matching on both MAC and IP addresses, the number of flows further increases:

$$F = N_s \times N_d \times N_{IP_s} \times N_{IP_d} \tag{30}$$

If the flow table matches not only MAC and IP addresses but also source and destination ports, the total number of required flows is:

$$F = N_s \times N_d \times N_{IP_s} \times N_{IP_d} \times N_{port_s} \times N_{port_d} \tag{31}$$

The number of required flow entries grows exponentially with the number of fields included in the match criteria. This presents scalability challenges in SDN, particularly in large-scale deployments.

As a result, timeout values play a critical role in regulating flow table growth by helping to maintain a manageable table size. Open vSwitch (OVS) in its default state supports approximately 200,000 flows [58]. However, this can be adjusted and is highly dependent on the hardware used. More resources equal more capabilities. We performed flow table lookup for six idle and hard timeout combinations: 1 and 2 minutes, 2 and 4 minutes, 3 and 6 minutes, 4 and 8 minutes, 5 and 10 minutes, and 10 and 20 minutes, respectively. Figure 19 provides the metrics for the various timeout values. The higher the timeout threshold, the greater the accumulation of flows. If λ is the average flow arrival rate, and T be the timeout duration applied to each flow (in seconds). The expected number of active flow entries in the PEP flow table, denoted by N, is given by:

$$N = \lambda \cdot T \tag{32}$$

This indicates a linear relationship: As the timeout T increases, the size of the flow table N increases proportionally.

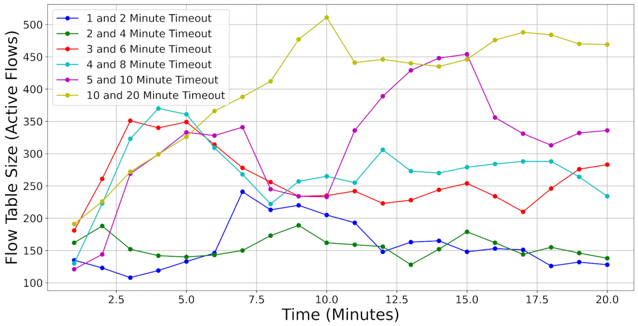


Figure 19: Size of the SDN switch flow table at 1-minute time intervals over a 20-minute period.

4.4. Packets Matched

In addition to the flow table size for idle and hard timeouts, we also examine the matched packets at each 60-second interval. Network traffic is highly unpredictable. As described in Table 5 of Section 4.1, devices can generate frequent communication requests even in standby mode or when not in use. To compensate for variability in packet transmission, we collected five sets of metrics for each idle/hard timeout combination and averaged the results. $M_k^{(j)}$ represent the number of packets matched in minute $k \in \{1, \ldots, 20\}$ for period $j \in \{1, \ldots, 5\}$. The average number of matched packets in minute k, denoted $\bar{M}_k$, is calculated as:

$$\bar{M}_k = \frac{1}{5} \sum_{j=1}^{5} M_k^{(j)} \tag{33}$$

Figure 20 presents the packets matched every 60 seconds. Irrespective of the timeout values set, packets sent across the network should not change much, as devices are abstracted away from the timeout values. Based on the observed results, the packets that matched appeared relatively uniform across the various idle/timeout combinations. The most noticeable distinction lies in the flow table size of the policy enforcement point (PEP). This is expected behavior, as flows are removed after the idle/hard timeout, limiting the size of the flow table.

$$\text{Remove}(\phi_i) = \begin{cases} \text{True} & \text{if } t - t_{\text{last}}^{(i)} \geq \tau_{\text{idle}} \\ \text{True} & \text{if } t - t_{\text{created}}^{(i)} \geq \tau_{\text{hard}} \\ \text{False} & \text{otherwise} \end{cases} \tag{34}$$

This led us to another question. Do the idle/hard timeout values significantly impact the number of authentication requests sent to the policy decision point (PDP)?

4.5. Result of Authentication Request

An authentication request is sent to the PDP when there are no matching policies; in this case, no flow rule exists at the PEP, creating a ”miss” condition. Assuming entries are removed based on the minimum of the idle and hard timeout, we can model:

$$P_{\text{miss}} \propto \frac{1}{\min(T_{\text{idle}}, T_{\text{hard}})} \tag{35}$$

The packet-in rate is approximately:

$$\text{Packet-In Rate} = \lambda \cdot P_{\text{miss}} \approx \frac{\lambda}{\min(T_{\text{idle}}, T_{\text{hard}})} \tag{36}$$

As mentioned in the previous section, there is a direct correlation between flow table size and authentication requests, especially for chatty devices. Lower timeout values lead to more frequent flow expirations, increasing the number of packets sent to the PDP as misses and requiring authentication via TIMeFoRCE. We utilized the putty logging mechanism to capture log files from the policy engine (PE), in this case, FWS, for the 20-minute period. We then filter the logs to obtain the number of authentication requests sent for the 20-minute period. Figure 21 provides the number of authentication requests sent for each idle/hard timeout combination. Based on the graph, the 1-2 minute idle/hard timeout has the highest number of authentication requests, while the 10-20 minute idle/hard timeout has the lowest. This is expected behavior, as the flows will persist for a longer period at the PEP before a device needs to re-authenticate.

4.6. Authentication Success Rate

Any effective authentication solution that provides identity and access management must perform accurately. This means it correctly identifies and authenticates legitimate subjects and rejects illegitimate ones. Authentication succeeds if either the fingerprint matches the database or the device's behavior is successfully classified (for new devices).
Let:

- M: MAC address of the device
- $\mathbf{x} \in \{0, 1\}^{16}$: binary feature vector extracted from packet headers
- $\mathcal{F}$: fingerprint database containing known $(M, \mathbf{f})$ pairs
- $F \in \{0, 1\}$: fingerprint match indicator

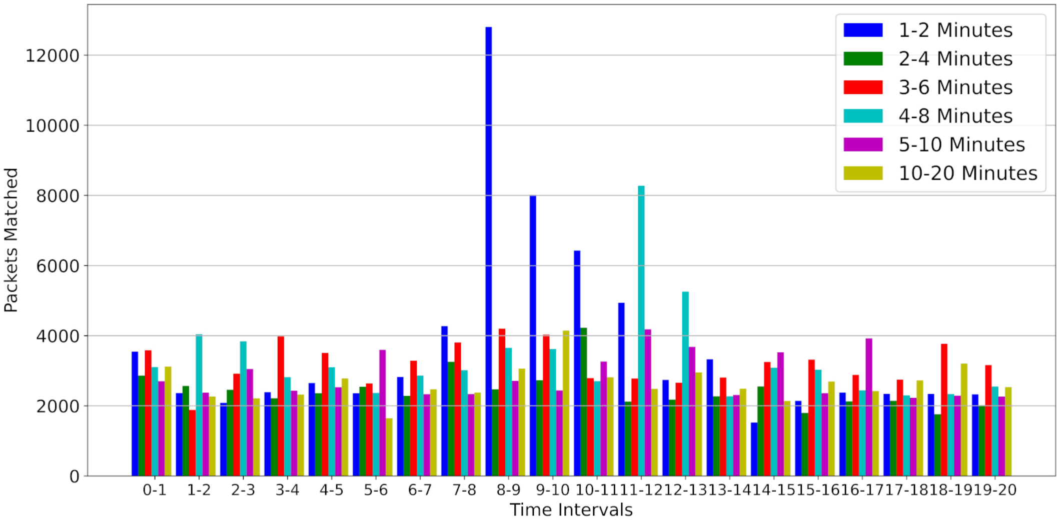


Figure 20: Packets matched every 60 seconds over 20 minutes

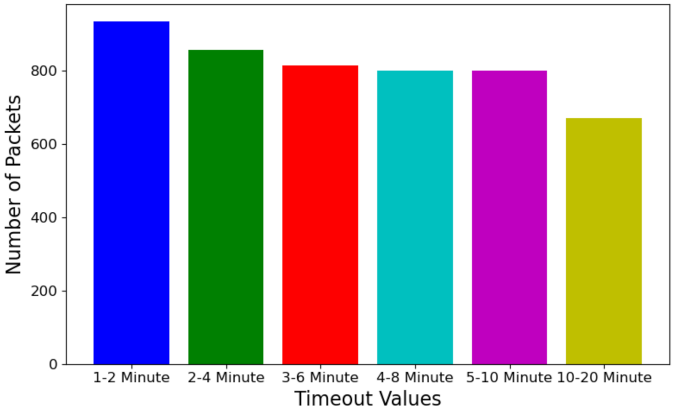


Figure 21: Authentication request sent to the PDP PE for each idle/hard timeout value.

- $\hat{p}_\gamma \in [0, 1]$: predicted classification confidence
- $\theta = 0.75$: minimum confidence threshold
- $\gamma \in \{0, 1\}$: classification acceptance outcome
- $A \in \{0, 1\}$: overall authentication decision

Define the fingerprint match:

$$F = \begin{cases} 1, & \text{if } (M, \mathbf{f}) \in \mathcal{F} \\ 0, & \text{otherwise} \end{cases} \tag{37}$$

Define classification acceptance based on confidence threshold:

$$\gamma = \begin{cases} 1, & \text{if } \hat{p}_\gamma \geq \theta \\ 0, & \text{otherwise} \end{cases} \tag{38}$$

Authentication is successful if either fingerprint or classification is accepted:

$$A = F \vee \gamma \tag{39}$$

Assuming independence:

$$P(A = 1) = 1 - (1 - p_F)(1 - p_\gamma) \tag{40}$$

To demonstrate the effectiveness of our solution for IoT device authentication in a ZTA, we present metrics for 1,000 access requests. We sent 1000 access requests to the PDP (TIMeFoRCE) and measured the percentage of successful authentication requests. Figure 22 shows that of the 1000 requests sent to the PDP, a successful authentication was returned for 961 of the requests, which is a 96.1% success rate. A successful authentication is one in which a fingerprint is retrieved from the database, and the PDP PA updates the PEP with bidirectional flows. Thirty-nine of the requests were unsuccessful, which is 3.9%. It is, however, important to note that the 39 unsuccessful requests are from only four devices, each sending the same repeated authentication request. If we treat those as only four requests, the resulting authentication accuracy is over 99%. Of the four (4) devices, two (2), representing 31 unsuccessful authentications, were not authorized to communicate based on the observed traffic match criteria and were successfully denied. A denial results in no flow being added to the PEP's flow table, enforcing Zero Trust. This occurred because of the removal of specific fingerprints from the database. Two (2) devices, representing eight requests, did not successfully authenticate, resulting in denied traffic. This is an incorrect assessment, as those represent legitimate device communications, which can be attributed to human error, as those fingerprints were not added to the database.

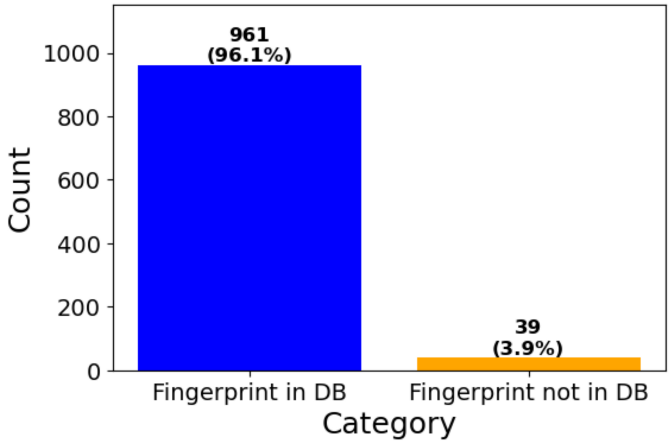


Figure 22: Authentication request sent to the PDP PE, resulting in successful authentication.

4.7. Introduced Latency For First Packets

In any network architecture, latency is key and can significantly impact performance. Latency was observed while using our Packet-Forwarder application to implement authentication. Figure 23 illustrates the additional latency introduced, including a trend line showing a decrease in the time it takes to perform the overall operation. This latency is incurred exclusively by the initial packet, as subsequent packets are not redirected to the PDP until the designated timeout has elapsed. T_n, is the delay between the PEP and PDP, T_p, PDP processing time, and T_f, is the delay to install the flow rule back to the PEP.

$$L_{\text{fwd}} = 2T_n + T_p + \max_{1\le i\le N} T_f^{(i)} \tag{41}$$

Since the PDP installs a reverse flow along with the forward flow, the reverse packet does not trigger a miss requiring authentication. Thus the reverse flow incurs no latency for authentication beyond L_{normal}, the normal network latency:

$$L_{\text{rev}} = L_{\text{normal}} \tag{42}$$

Thus the total setup latency for the bi-directional flow is:

$$L_{\text{total}} = L_{\text{fwd}} + L_{\text{rev}} = 2T_n + T_p + \max_{1\le i\le N} T_f^{(i)} + L_{\text{normal}} \tag{43}$$

It is important to note that this latency is in addition to the normal network operational latency. We elected to present the additional latency rather than the overall latency for two primary reasons: (1) each network environment is unique, and the latency observed in one setting may not accurately represent that in another, and (2) the additional latency specifically characterizes the delay introduced by the first transmitted packet, rather than applying uniformly to every packet sent by a device.

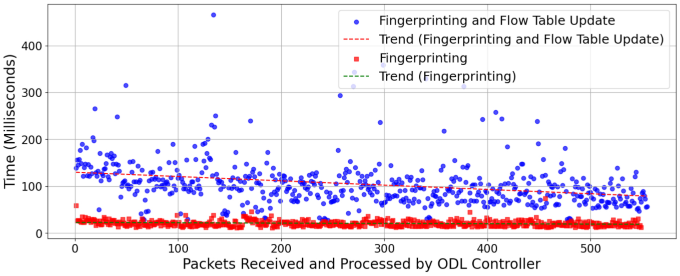


Figure 23: The additional latency incurred to perform Zero ZTA authentication measured for the first packet in each communication request, with results illustrated alongside a corresponding trend line.

4.8. Network Throughput

Network throughput is an important consideration when implementing any architecture. We present the throughput achieved with an SDN switch using static flow rules and our Packet-Forwarder solution. We obtained performance metrics using the iperf3 [59] tool and conducted the test over 60 seconds at 1-second intervals. We used a Raspberry Pi 4B with 4GB of RAM running Kali Linux 2024.4 as the iperf3 client, and a UP2 development board with 8GB of RAM running Ubuntu 22.04 as the iperf3 server. The results presented in Figure 24 show that using our Packet-Forwarder Zero Trust application, we obtained a throughput of approximately 895 Mbits/sec compared to 920 Mbits/sec using a default configuration SDN switch on a 1000 Mbits/sec interface.

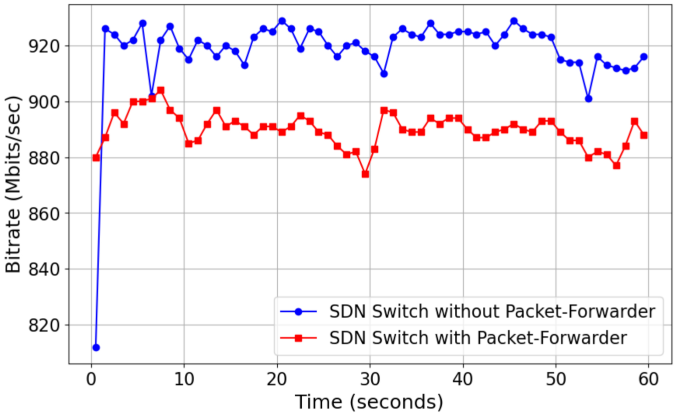


Figure 24: Network throughput showing the throughput for TiMEFoRCE and the SDN switch operating in default mode

4.9. Adversarial Model

In any identity and access management solution, several threats are ever-present. As it relates to our TIMeFoRCE, common threats include identity spoofing, evasion, replay, misclassifications or false positives, man-in-the-middle (MitM), and denial of identity. The identity management system TIMeFoRCE is designed to operate in a Zero Trust environment where devices are authenticated via behavioral fingerprinting and SDN-based flow enforcement. This section describes the adversarial assumptions, capabilities, objectives, and attack strategies targeting the system's core components. The adversary seeks to:

- Bypass identity verification by mimicking legitimate devices.
- Evade classification and enforcement by exploiting ambiguous or unclassifiable behavior.
- Trigger classification errors (false negatives) to obtain unauthorized access.
- Interfere with SDN flow rule installation or behavior.
- Overload the system with excessive identity attempts or fingerprint variations.

Given the following capabilities, the adversary may possess one or more of the following:

- Passive packet sniffing to observe MAC addresses, ports, and QNAMEs.
- Active injection of spoofed packets and manipulated fingerprints.
- Replay of legitimate traffic traces and behavioral fingerprints.
- Partial knowledge of the ML model's feature set and threshold logic.
- Ability to manipulate or observe SDN flow control messages (e.g., man-in-the-middle).
- Capability to launch resource exhaustion attacks using high volumes of malformed or ambiguous identity traffic.

The primary concern in IoT device security as it relates to TIMeFoRCE is identity spoofing, where an illegitimate device impersonates a legitimate one. We consider adversaries with varying levels of access and capabilities who seek to spoof the IoT fingerprinting system. Let the fingerprint for device i be defined as:

$$f_i = \text{MAC}_i \,\|\, \vec{x}_i, \quad \vec{x}_i \in \{0,1\}^{16} \tag{44}$$

where MAC_i is the device's 48-bit MAC address and $\vec{x}_i$ is the 16-bit binary feature vector extracted from packet headers.

To address this concern, we combined the device's MAC address with 16 binary features from the packet header. The 16 binary features are used to mitigate the possibility of a cloned MAC Address. A malicious device would need to clone the MAC address of an existing device and generate comparable feature variables. The probability that the attacker successfully spoofs a known MAC address (P_{mac}), and the probability that the attacker successfully replicates the correct 16-bit binary feature vector (P_{feat}), resulting in the probability that the attacker spoofs the entire fingerprint ($M^*, \mathbf{f}^*$) ($P_{\text{fingerprint}}$) is defined by:

$$P_{\text{fingerprint}} = P_{\text{mac}} \cdot P_{\text{feat}} \tag{45}$$

If each bit in the feature vector is guessed randomly:

$$P_{\text{feat}} = \frac{1}{2^{16}} = 1.5259 \times 10^{-5} \tag{46}$$

Then, if $P_{\text{mac}} = 0.9$ (A very high probability of being spoofed):

$$P_{\text{fingerprint}} = 0.9 \cdot \frac{1}{2^{16}} \approx 1.3733 \times 10^{-5} \tag{47}$$

resulting in extremely low probability. The addition of the timeout values ensures the device is re-authenticated for each subsequent communication request after the timeout expires. This will aid in determining if a device is potentially compromised, as its behavior will change. This will also guard against replay attacks.

Additionally, we make the following system assumptions:

- The adversary does not possess direct access to the fingerprint database.
- The classifier operates on 16 binary features extracted from packet headers.
- The classification threshold is fixed at $\theta = 0.75$.
- The SDN control plane is assumed to be protected, but may be targeted via MitM attacks.

We developed an identity management solution that can satisfy the tenets of Zero Trust, and while we present an adversarial model for one of the most common threats, spoofing of identities, other threats include:
The crafting feature vector $\mathbf{x}'$ such that:

$$\mathcal{M}(\mathbf{x}') \in \text{Other} \quad \text{and} \quad \text{predict\_proba}(\mathcal{M}, \mathbf{x}') < \tau \tag{48}$$

or applying perturbation δ to feature vector:

$$\mathbf{x}' = \mathbf{x} + \delta \quad \text{where} \quad ||\delta|| < \epsilon \quad \text{and} \quad \mathcal{M}(\mathbf{x}') \neq \mathcal{M}(\mathbf{x}) \tag{49}$$

These types of attacks are designed to weaken the model performance to create false negatives, where legitimate devices are classified as 'Other'. If the device being introduced into the system is not one that is approved, the classification of 'Other' is appropriate, as classification only occurs during the addition of new devices. The administrator is notified whenever a new device is given the 'Other' designation, prompting manual actions. Additionally, an adversary may seek to intercept or modify SDN control messages $m \in \mathcal{F}$ to create:

$$m' = f(m) \quad \text{where} \quad f \text{ alters flow behavior} \tag{50}$$

The controller operates on an out-of-band (OOB) channel that is separate from the data path. The attacker may also generate sequence $\{\mathbf{f}_1, \ldots, \mathbf{f}_k\}$ with:

$$\forall i, \mathbf{f}_i \notin \mathcal{D} \quad \text{and} \quad k \gg |\mathcal{D}| \tag{51}$$

We rate-limit new fingerprint additions by enforcing threshold-based rejection for unknown devices. If a device sends 10 identical packets that fail to generate a successful fingerprint, a flow is written to block that device from communicating for the specified communication type.

5. Conclusion

Security is a critical network element and the primary driver of a ZTA. Current ZTA solutions remain insufficient in delivering comprehensive Identity and Access Management for IoT devices. We demonstrated that authentication for IoT devices can be provided for a ZTA using device behavioral traits. TIMeFoRCE observed an overall success rate of 96.1% across all traffic and a 99% success rate for all devices that attempted authentication after removing duplicate packets resulting from retransmissions. Additionally, we achieved classification accuracy above 99% using the dictionary, even on validation data not previously encountered. The results are derived from a small testbed; however, we contend that, with sufficient resources and processing power, these findings can be extrapolated to accommodate significantly larger commercial network environments. Organizations can mitigate the blast radius of compromised or misbehaving devices by authenticating IoT devices and constraining their communication. The Packet-Forwarder application is modular, meaning the back-end classification algorithm used to identify devices can be updated as technology evolves. This research focuses solely on providing a solution to address identity management and access control. Further research is needed to address the other areas not discussed in this paper. Although the solution demonstrates strong performance, there remains a potential risk that traffic from a legitimate device may be denied if a corresponding fingerprint is not identified during the database query. To account for this, TIMeFoRCE maintains log files that store the fingerprints generated during unsuccessful retrievals and device classifications. A record of the device fingerprint and the number of times it was observed is maintained. These log files allow a network programmer to scrutinize devices add fingerprints manually to the database.

6. Future Work

TIMeFoRCE is implemented with a single SDN controller. The flows are written to the SDN switch flow table with hard-timeout values, causing the flow to expire and be removed. During implementation and testing, it was discovered that whenever the SDN switch (the PEP) loses access to the SDN controller (the PDP), the flows remain active on the SDN switch until the SDN controller is reconnected. Existing connections will remain active; however, new connections will not be possible. Exploration into a distributed SDN controller architecture or a fail-safe that installs a default flow in the SDN switch flow table until controller functionality returns is necessary. Machine learning models, large language models (LLMs), and generative AI, such as ChatGPT [60], are constantly evolving and have become integral to many security solutions. It is important to explore their capability to provide improved identification and classification. To examine the effectiveness of the proposed solution, IoT devices deployed in environments such as industrial, medical, and smart cities should be investigated. TIMeFoRCE used a probability threshold of 0.75, which is significantly higher than the standard random forest threshold of 0.50. While correct classification was achieved with a higher threshold, this could lead to legitimate devices being incorrectly classified as ”Other,” denying authentication and the successful addition of fingerprints to the database. Finally, the fingerprints are stored in plain text in the database; a more secure approach is to hash the values during generation.

Conflict of Interest The authors declare no conflict of interest.

Acknowledgment The authors would like to thank the Cybersecurity Assurance Policy (CAP) Center for providing the resources necessary to perform this research.

References

[1] V. Morris, K. Kornegay, “Flow Table Modification Using Behavioral-based Fingerprinting Technique to Facilitate Zero Trust Identity Management and Access Control,” in 2025 16th Annual IEEE Computing and Communication Workshop and Conference (CCWC), 325–332, IEEE, 2025, doi:DOI: 10.1109/CCWC62904.2025.10903861.

[2] B. Bezawada, I. Ray, I. Ray, “Behavioral fingerprinting of Internet-of-Things devices,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, **11**(1), 41–50, 2021, doi:DOI: 10.1002/widm.1337.

[3] H. Noguchi, M. Kataoka, Y. Yamato, “Device identification based on communication analysis for the internet of things,” IEEE Access, **7**, 52903–52912, 2019, doi:DOI: 10.1109/ACCESS.2019.2910848.

[4] R. Perdisci, T. Papastergiou, O. Alrawi, M. Antonakakis, “IoTFinder: Efficient Large-Scale Identification of IoT Devices via Passive DNS Traffic Analysis,” Available at `www.google.com`, accessed: [Please insert the access date].

[5] C. Buck, C. Olenberger, A. Schweizer, F. Völter, T. Eymann, “Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust,” Computers and Security, **110**, 102436, 2021, doi:DOI: 10.1016/j.cose.2021.102436.

[6] O. C. Edo, T. Tenebe, E. Etu, A. Ayuwu, J. Emakhu, S. Adebiyi, “Zero Trust Architecture: Trend and Impact on Information Security,” International Journal of Emerging Technology and Advanced Engineering, **12**(7), 140–147, 2022, doi:DOI: 10.46338/ijetae0722_15.

[7] Y. G. Wu, W. H. Yan, J. Z. Wang, “Real identity based access control technology under zero trust architecture,” Proceedings - 2021 International Conference on Wireless Communications and Smart Grid, ICWCSG 2021, 18–22, 2021, doi:DOI: 10.1109/ICWCSG53609.2021.00011.

[8] S. Rose, O. Borchert, S. Mitchell, S. Connelly, “Zero Trust Architecture,” NIST Special Publication 800-207, National Institute of Standards and Technology, 2020, doi:DOI: 10.6028/NIST.SP.800-207.

[9] N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig, R. Doss, “Zero Trust Architecture (ZTA): A Comprehensive Survey,” IEEE Access, **10**, 57143–57179, 2022, doi:DOI: 10.1109/ACCESS.2022.3174679.

[10] U. Mattsson, “Zero Trust Architecture,” Controlling Privacy and the Use of Data Assets, 127–134, 2022, doi:DOI: 10.1201/9781003189664-11.

[11] P. Assunção, “A Zero Trust Approach to Network Security,” in Proceedings of the Digital Privacy and Security Conference 2019, 2019.

[12] J. Kindervag, “No More Chewy Centers: Introducing the Zero Trust Model of Information Security,” Technical report, Forrester Research, 2010, accessed: January 24, 2024.

[13] M. Campbell, “Beyond Zero Trust: Trust Is a Vulnerability,” Computer, **53**(10), 110–113, 2020, doi:DOI: 10.1109/MC.2020.3011081.

[14] Defense Information Systems, Agency, National Security Agency : Zero Trust Engineering Team, “Department of Defense (DOD) Zero Trust Reference Architecture CLEARED For Open Publication Department of Defense OFFICE OF PREPUBLICATION AND SECURITY REVIEW,” 170, 2021.

[15] Cisco Systems, “Cisco Identity Services Engine (ISE),” `https://www.cisco.com/site/us/en/products/security/identity-services-engine/index.html`, 2024, accessed: 2025-06-13.

[16] Microsoft, "Introduction to Azure IoT," https://learn.microsoft.com/en-us/azure/iot/iot-introduction, accessed: January 21, 2025.

[17] Palo Alto Networks, "IoT Security for Zero Trust Enterprise," https://www.paloaltonetworks.com/network-security/iot-security, 2024, accessed: 2025-06-13.

[18] Armis Inc., "Armis Platform Overview," https://www.armis.com/platform/, 2024, accessed: 2025-06-13.

[19] Zscaler Inc., "Zscaler for IoT and OT Security," https://www.zscaler.com/solutions/iot-security, 2024, accessed: 2025-06-13.

[20] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. R. Sadeghi, S. Tarkoma, "IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT," Proceedings - International Conference on Distributed Computing Systems, 2177–2184, 2017, doi:DOI: 10.1109/ICDCS.2017.283.

[21] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, "IoTSense: Behavioral Fingerprinting of IoT Devices," arXiv:, 2018.

[22] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, V. Sivaraman, "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics," IEEE Transactions on Mobile Computing, **18**(8), 1745–1759, 2019, doi:DOI: 10.1109/TMC.2018.2866249.

[23] R. R. Chowdhury, S. Aneja, N. Aneja, E. Abas, "Network Traffic Analysis based IoT Device Identification," ACM International Conference Proceeding Series, 79–89, 2020, doi:DOI: 10.1145/3421537.3421545.

[24] S. Aneja, N. Aneja, M. S. Islam, "IoT Device Fingerprint Using Deep Learning," in Proceedings - 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS), 174–179, 2019, doi:DOI: 10.1109/IOTAIS.2018.8600824.

[25] P. Oser, F. Kargl, S. Lüders, "Identifying devices of the internet of things using machine learning on clock characteristics," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), **11342 LNCS**, 417–427, 2018, doi:DOI: 10.1007/978-3-030-05345-1_36.

[26] A. Hameed, "IoT Traffic Multi-Classification Using Network and Statistical Features in a Smart Environment; IoT Traffic Multi-Classification Using Network and Statistical Features in a Smart Environment," Technical report, 2020.

[27] S. V. Radhakrishnan, A. S. Uluagac, R. Beyah, "GTID: A Technique for Physical Device and Device Type Fingerprinting," IEEE Transactions on Dependable and Secure Computing, forthcoming or In Press; update fields once details are known.

[28] K. Kostas, M. Just, M. A. Lones, "IoTDevID: A Behavior-Based Device Identification Method for the IoT," 2021, doi:DOI: 10.1109/JIOT.2022.3191951.

[29] N. Basta, M. Ikram, M. Kaafar, A. Walker, "Towards a Zero-Trust Microsegmentation Network Security Strategy: An Evaluation Framework," 1–7, 2022, doi:DOI: 10.1109/noms54207.2022.9789888.

[30] D. Comer, A. Rastegarnia, "Externalization of Packet Processing in Software Defined Networking," IEEE Networking Letters, **1**(3), 124–127, 2019, doi:DOI: 10.1109/lnet.2019.2918155.

[31] C. Liu, R. Tan, Y. Wu, Y. Feng, F. Z. Z. Jin, Y. Liu, Q. Liu, "Dissecting zero trust: research landscape and its implementation in IoT," Cybersecurity, **7**, 20, 2024, doi:10.1186/s42400-024-00212-0.

[32] C. Bast, K. Yeh, "Emerging Authentication Technologies for Zero Trust on the Internet of Things," Symmetry, **16**(8), 993, 2024, doi:10.3390/sym16080993.

[33] H. Kang, G. Liu, Q. Wang, L. Meng, Lei, , J. Liu, "Theory and Application of Zero Trust Security: A Brief Survey," Entropy, **25**(12), 1–26, 2023, doi:10.3390/e25121595.

[34] D. Eidle, S. Y. Ni, C. Decusatis, A. Sager, "Autonomic Security for Zero Trust Networks," in 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), 288–293, 2017, doi:DOI: 10.1109/UEMCON.2017.8249053.

[35] S. Ameer, L. Praharaj, R. Sandhu, S. Bhatt, M. Gupta, "ZTA-IoT: A Novel Architecture for Zero-Trust in IoT Systems and an Ensuing Usage Control Model," ACM Transactions on Privacy and Security, **27**(3), 2024, doi:10.1145/3671147.

[36] R. Chandramouli, Z. Butcher, "NIST Special Publication NIST SP 800-207A ipd 2 A Zero Trust Architecture Model 3 for Access Control in Cloud-Native Applications in Multi-Location," .

[37] Amazon Web Services (AWS), "AWS IoT Core," https://aws.amazon.com/iot-core/, accessed: 21 January 2025.

[38] Siemens, "Industrial Internet of Things (IIoT)," https://www.sw.siemens.com/en-US/solutions/industrial-internet-of-things-iiot/#6Xl1D3sPnBHuo88QKMZ1EV, accessed: January 21, 2025.

[39] "IoT Cloud Platform Market," https://www.marketsandmarkets.com/Market-Reports/iot-cloud-platform-market-195182.html, accessed: 2021-03-20.

[40] U. Guide, "WRT 3200ACM Gigabit Wi-Fi Router," .

[41] A. Lunn, V. Didelot, F. Fainelli, "Distributed Switch Architecture, A.K.A DSA," Netdev 2.1, 2017.

[42] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, "Behavioral fingerprinting of IoT devices," in Proceedings of the ACM Conference on Computer and Communications Security, 41–50, Association for Computing Machinery, 2018, doi:DOI: 10.1145/3266444.3266452.

[43] M. R. Shahid, G. Blanc, Z. Zhang, H. Debar, "IoT Devices Recognition Through Network Traffic Analysis," in Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018, 5187–5192, Institute of Electrical and Electronics Engineers Inc., 2019, doi:DOI: 10.1109/BigData.2018.8622243.

[44] Y. Guo, Y. Wang, F. Khan, A. A. Al-Atawi, A. A. Abdulwahid, Y. Lee, B. Marapelli, "MAB with SDN Orchestration," 1–18, 2023.

[45] L. Fan, S. Zhang, Y. Wu, Z. Wang, C. Duan, J. Li, J. Yang, "An IoT Device Identification Method based on Semi-supervised Learning," in 2020 16th International Conference on Network and Service Management (CNSM), 1–7, IEEE, 2020, doi:DOI: 10.23919/CNSM50824.2020.9269044.

[46] "OpenDaylight project," https://en.wikipedia.org/wiki/OpenDaylig h_Project., accessed: Nov. 2, 2024.

[47] "Flask Documentation (v3.0.x)," https://flask.palletsprojects.com/en/3.0.x/, accessed: January 24, 2024.

[48] "Python," https://www.python.org/, accessed: January 24, 2024.

[49] "MySQL," https://www.mysql.com/, accessed: January 24, 2024.

[50] "Apache Maven," https://maven.apache.org/, accessed: January 24, 2024.

[51] "MVNRepository," https://mvnrepository.com/, accessed: January 24, 2024.

[52] P. K. Sharma, J. H. Park, Y. S. Jeong, J. H. Park, "SHSec: SDN-based Secure Smart Home Network Architecture for the Internet of Things," Mobile Networks and Applications, **24**(3), 913–924, 2019, doi:DOI: 10.1007/s11036-018-1147-3.

[53] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, M. Sain, "Lightweight and Secure Session-Key Establishment Scheme in Smart Home Environments," IEEE Sensors Journal, **16**(1), 254–264, 2016.

[54] P. Krishnan, K. Jain, K. Achuthan, R. Buyya, "Software-Defined Security-by-Contract for Blockchain-enabled MUD-aware Industrial IoT Edge Networks," IEEE Transactions on Industrial Informatics, 2021, doi:DOI: 10.1109/TII.2021.3084341.

[55] S. Bera, S. Misra, A. Jamalipour, "FlowStat: Adaptive Flow-Rule Placement for Per-Flow Statistics in SDN," IEEE Journal on Selected Areas in Communications, **37**(3), 530–539, 2019, doi:DOI: 10.1109/JSAC.2019.2894239.

[56] O. Corporation, “VirtualBox 7.0.14 for Windows,” https://www.virtualbox.org/wiki/Download_Old_Builds_7_0, accessed: March 16, 2024.

[57] OpenDaylight, “Karaf Integration 0.20.3,” https://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/integration/karaf/0.20.1, accessed: March 16, 2024.

[58] B. Pfaff, J. Pettit, T. Koponen, E. J. Jackson, A. Zhou, J. G. J. Rajahalme, A. Wang, J. Stringer, P. Shelar, K. Amidon, M. Casado, “The Design and Implementation of Open vSwitch,” Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2015, 117–130, 2015.

[59] T. iPerf Group, “iPerf,” https://iperf.fr/, 2025, accessed: February 6, 2025.

[60] OpenAI, “ChatGPT,” https://chat.openai.com/chat, 2023, accessed: April 17, 2024.

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 10, No. 6, 23-28 (2025)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTESJ
ISSN: 2415-6698

Private 5G MIMO for Cable TV IP Broadcasting

Hiroshi Ito*[1], Hideki Ohno[1], Hikaru Kitano[1], Shuichi Matsumoto[2]

[1] *NEC Networks & System Integration Corporation, Tokyo, 108-8515, Japan*

[2] *Japan Cable Laboratories, Tokyo, 103-0025, Japan*

ARTICLE INFO

Article history:
Received: 02 September, 2025
Revised: 30 October, 2025
Accepted: 01 November, 2025
Online: 14 November, 2025

Keywords:
Cable TV
IP Broadcasting
Private 5G
MIMO

ABSTRACT

Private 5G utilization by cable TV is expected to be an alternative to wired services, especially for multi-dwelling units and rural communal TV receiving areas. On the other hand, the 100 MHz of the sub-6 frequency band for private 5G is not sufficient for cable TV services consisting of multi-channel broadcasting and Internet, and some measures to improve frequency utilization efficiency are needed. In this paper, we propose a hybrid MIMO technology for private 5G that applies diversity MIMO for reliable broadcasting and multi-stream MIMO for efficient data communication at the same time.

1. Introduction

Cable television fulfils the function of a comprehensive information media that provides re-transmission of terrestrial broadcasting programs, multi-channel cable broadcasting, Internet and telephony together, and also plays an important role as a regional information and communications infrastructure as a medium for transmitting information essential for daily life, such as evacuation information during disasters. On the other hand, in order to bridge the digital divide, there is a strong need to reduce the cost of the last one mile of cable television transmission lines by FTTH(Fiber To The Home) in areas where infrastructure development lags behind that of urban areas, and expectations for FWA (Fixed Wireless Access) services using Private 5G (5th Generation Mobile Communication System), which can significantly reduce construction costs, are increasing as a solution to this problem.

In addition, when looking at the provision of advanced broadcasting services (4K and 8K TV) for cable television, there are many in-building facilities in older housing complexes that do not support these services due to a limited transmission bandwidth, and the aging of the buildings themselves makes it impossible to replace the coaxial cables in the in-building facilities, making the provision of these services an urgent issue. The use of FWA in private 5G systems is attracting attention as a promising solution.

If private 5G systems are not only used in the telecommunications field as FWAs, but are also expected to expand their use in the broadcasting field [1], for example, transmission capacity of 200 Mbps or more will be required for broadcasting 20 4K programs, and the shortage of sub-6 frequency bands (4.8 GHz to 4.9 GHz) for private 5G systems will be a serious problem.

On the other hand, Multicast and Broadcast Services (MBS) of 5G NR (New Radio) [2] is being considered in 3GPP, an international standards organization for mobile communication systems such as 5G, but there has been no consideration of technical measures to deal with the frequency crunch caused by the use of broadcasting. Additionally, there have been no product developments for private 5G radio equipment compatible with MBS.

Against this background, in order to meet the demand for the effective use of MBS frequencies, we have focused on MIMO(Multiple Input Multiple Output) technology [3][4], which has been studied mainly from the physical layer perspective including rarely for broadcast applications [5][6][7]. Namely we have applied a hybrid type of MIMO technology that effectively enables both media to coexist on MIMO from the application layer perspective, which consists of different requirements such as reliability for IP broadcasting and efficiency for data communication.

In the next section, we will explain the adaptive MBS proposed by the authors regarding the overall system configuration of IP broadcasting and unicast distribution, and the feature functions [8][9]. In Section 3, after explaining the multiplex configuration on MIMO for IP broadcasting and data communication streams

*Corresponding Author: Hiroshi Ito, NEC Networks & System Integration Corporation, Tokyo, 108-8515, Japan, ito.hiroshi@nesic.com

https://dx.doi.org/10.25046/aj100602

(including unicast distribution) based on the adaptive MBS, the concepts of time-division multiplexing and frequency-division multiplexing are described, and the comparison of the two is discussed qualitatively in terms of MIMO effect, transmission efficiency and transmission time delay from the perspective of the resource block scheduler. Section 4 introduces the private 5G radio propagation simulator developed by the authors for IP broadcasting to residential complexes, with the aim of quantitatively understanding the effect of IP broadcasting using the hybrid MIMO studied. The simulation results are partially reported.

The main contribution of this paper is the introduction of hybrid MIMO technology for private 5G to improve frequency utilization efficiency. It applies reliable diversity MIMO for broadcasting and high-efficiency multi-stream MIMO for data communication simultaneously, realizing IP broadcasting services through private 5G.

2. System Configuration for IP Broadcasting with Private 5G

2.1. Adaptive MBS

Fig.1 shows the fundamental structure of the adaptive MBS, which adds the following four functions to the MBS defined by 3GPP.

1. Limited viewing program distribution
2. Variable bit rate encoding linked with Modulation and Coding Scheme (MCS)
3. IP broadcast/unicast adaptive distribution
4. IP broadcast/unicast seamless synchronous playback

These functions prevent inefficient transmission of programs by stopping the IP broadcasting of non-viewing broadcast programs and single-viewing household programs, replacing them with unicast transmission. Furthermore, unicast transmission is supplemented for households that cannot receive IP broadcasts in low radio propagation environments. As a result transmission efficiency is improved by applying higher-order QAM modulation to households that receive IP broadcasts.

The IP broadcast stream is compressed into a file using MPEG-DASH (Dynamic Adaptive Streaming over HTTP) and then encapsulated using FLUTE (File Delivery over Unidirectional Transport) to add error correction code at the application layer. OFDM modulation is then performed together with the unicast stream of the DASH file. At this time, the transmitter decides whether each program needs to be broadcast or not and chooses the unicast alternative based on the program selection information from the receiver and the IP broadcast reception status information

The video coding rate applied to a program is determined in conjunction with the MCS determined from the reception signal status, thereby preventing transmission capacity overflow (exceeding the resource block) in a low reception power environment as much as possible. This variable rate coding is considered desirable in terms of ensuring broadcast quality if it is linked to the program resolution of 4K, HD and SD, for example.

At the receiver side, synchronous synthesis is performed by buffering both IP broadcast and unicast streams, taking into account the unicast reception start time, in order to seamlessly switch between IP broadcast and unicast in both directions.

2.2. MIMO System Configuration

A configuration for multiplexing both IP broadcasting and data communication streams including unicast over MIMO-OFDM under adaptive MBS is shown in Fig.2.

For the two QAM modulation symbol streams of IP broadcasting and data communication, the scheduler allocates resource blocks consisting of 14 OFDM symbols and 12 subcarriers as shown in Fig.3.

At this time, the MIMO layer is allocated to the IP broadcast stream with priority given to single-stream diversity MIMO for reliability to maintain broadcast quality, and to the data communication stream including unicast stream with priority given to multi-stream MIMO for transmission efficiency. The aforementioned resource block scheduling of both streams on the time-frequency axis results in MIMO multiplexing. As explained later, this multiplexing scheme can include Frequency Division Multiplexing (FDM) on the frequency axis and Time Division Multiplexing (TDM) on the time axis.

After MIMO multiplexing, precoding adapted to the MIMO propagation path is applied to both broadcast and communication streams independently, with common precoding for the receiving households applied to the resource blocks of the broadcast stream and individual household precoding for the resource blocks of the communication stream, which are then sent to the receiving side. Similarly, for MCS that varies according to the received power status, the common MCS for receiving households is applied to the broadcast stream, and the individual MCS for receiving households is applied to the communication stream.

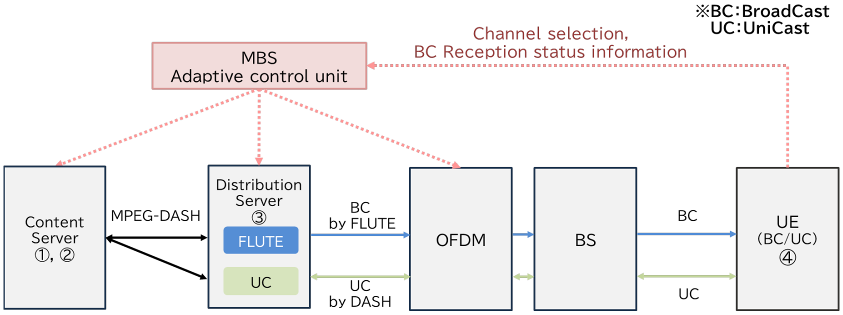


Figure 1. Private 5G system using hybrid MIMO for cable TV services

3. Hybrid MIMO

3.1. Diversity and Multi-stream

MIMO technology, utilizing multiple antennas encompasses two primary approaches Space-Time Coding (STC) for improving transmission quality and Space Division Multiplexing (SDM) for enhancing transmission speed. STC employs diversity MIMO techniques for single streams, while SDM uses multi-stream MIMO for the simultaneous processing of multiple streams. When the MIMO channel propagation path characteristics are known at the transmitter side, more advanced techniques can be employed: Maximal Ratio Combining (MRC) diversity and Eigenbeam-Space Division Multiplexing (E-SDM). These techniques demonstrate superior performance in terms of transmission quality and speed compared to STC and SDM, which operate without knowledge of the channel propagation path.

Rank adaptation, an adaptive technology, dynamically adjusts the number of streams and the modulation coding order based on reception conditions. 3GPP standards specify protocols for transitioning between diversity, low-order, and high-order multi-stream modes.

Figure 4 illustrates the Frequency Division Multiplexing (FDM) and Time Division Multiplexing (TDM) configurations when diversity MIMO is applied to IP broadcasting and multi-stream MIMO to data communication. In TDM, data communication and IP broadcasting alternate in time slots based on data volume, while in FDM, they are allocated to different subcarriers. The allocated resource blocks are transmitted using single-stream diversity MIMO for IP broadcasting and multi-stream MIMO for data communication, with the number of streams corresponding to the layer count. In a 4×4 MIMO configuration, this can result in up to approximately 6 dB of improvement in received power for IP broadcasting and a quadrupling of transmission throughput for data communication.

3.2. Considerations

Regarding the choice of the MIMO multiplexing scheme, several assumptions need to be taken into account, as listed below.

- Massive antennas on the transmit side for beamforming in mobile communications are not assumed, and single-user MIMO is assumed, where the MIMO transmission path is shared between terminals in a time-division manner.
- In 3GPP NR, subcarrier spacing of 30 kHz and 60 kHz is specified in the 100 MHz bandwidth of private 5G, and subcarrier spacing cannot be changed by frequency at the same time.
- The Reduction of inter-subcarrier interference caused by phase noise by increasing the subcarrier spacing.
- The OFDM symbol length affects the transmission delay time of the radio section.
- MIMO precoding and QAM modulation based on MCS can be applied to IP broadcasting resource blocks for all households commonly, and to data communication resource blocks for each receiving household individually.

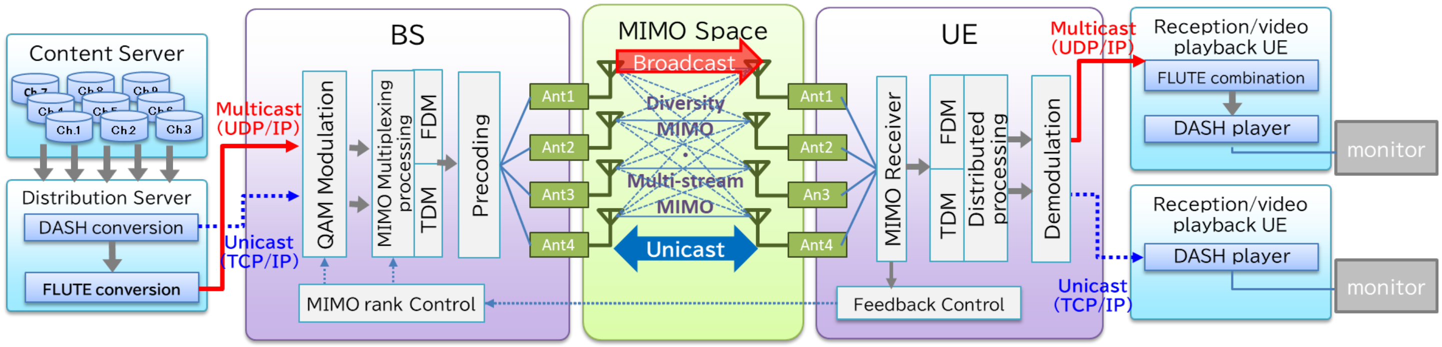
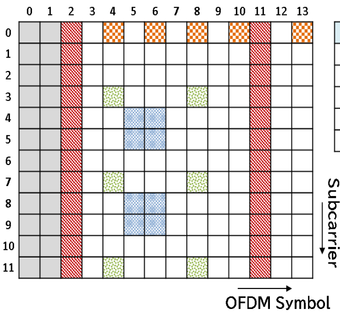


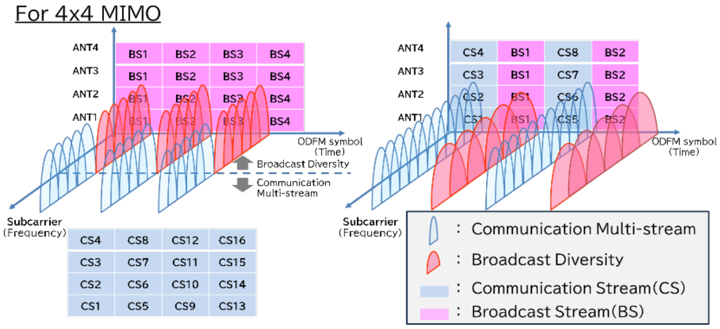
Figure 2. MIMO system for cable TV IP broadcasting



Reference Signal	Number of symbols	Ratio
PDCCH	24	14.28%
CSI-RS	8	4.76%
DM-RS	24	14.28%
TRS	6	3.57%
PT-RS	5	2.97%

□ PDCCH (Physical Downlink Control CHannel)
▨ CSI-RS (Channel State Information RS)
▨ DM-RS (DeModulation RS)
▨ TRS (Tracking RS)
▨ PT-RS (Phase Tracking RS)

Figure 3: Example of DL reference signal in 5G resource blocks



(1)FDM (2)TDM

Figure 4: Hybrid MIMO multiplexing

- The resource block allocation arrangement affects the transmission delay time.
- Each terminal is allocated priority in receiving power for data communication resource blocks (IP broadcasting resource blocks are shared by all terminals, so priority in receiving power is not applied).

Given the conditions, we consider a qualitative comparison between TDM and FDM.

(1) MIMO effect (frequency selectivity)

In wideband FWA, frequency fluctuations are more significant than time fluctuations, which are predominant in narrowband mobile communications. TDM enables individual optimization of receiving terminals using MIMO precoding that accounts for frequency selectivity across all frequency bands, providing a distinct advantage.

(2) Transmission delay

As illustrated in Figure 5-1, FDM distributes broadcasting resource blocks in the slot direction (time axis), while TDM concentrates them in the subcarrier direction. Consequently, TDM offers reduced packet transmission delay, rendering it more advantageous in this aspect.

(3) Scheduling accuracy

Figure 5-2 demonstrates that in FDM, resource blocks allocated for communication to each User Equipment (UE) are distributed in the slot direction. This necessitates received power prediction when scheduling UEs based on received power, potentially reducing scheduling accuracy. Conversely, TDM concentrates communication resource blocks in the subcarrier direction, allowing for more accurate distribution of resource blocks to receiving terminals based on received power.

(4) TDD configuration

As shown in Figure 5-3, FDM requires sharing of uplink and downlink resource block patterns for IP broadcasting and communication within a 10 ms unit radio frame, potentially decreasing efficiency. TDM, however, allows for optimization of the TDD pattern for both IP broadcasting and communication, enhancing radio frame efficiency. It should be noted that this advantage is predicated on the use of dynamic TDD.

(5) MIMO Precoding

Figure 5-4 illustrates that FDM necessitates the simultaneous application of two types of MIMO precoding (for broadcasting and communication) to each terminal for each slot. In contrast, TDM requires only one type of precoding (either broadcasting or communication) per slot time for each receiving terminal, potentially reducing the implementation load, particularly on the receiving terminal.

(6) Variable subcarrier spacing

If it is necessary to change the 30kHz and 60kHz subcarrier spacing specified in the 100MHz band of the sub-6 band for broadcasting and communication or synchronization control, in FDM, the subcarrier spacing cannot be changed by the frequency as shown in Figure 5-5 due to frequency interference, and TDM with variable subcarrier intervals is advantageous.

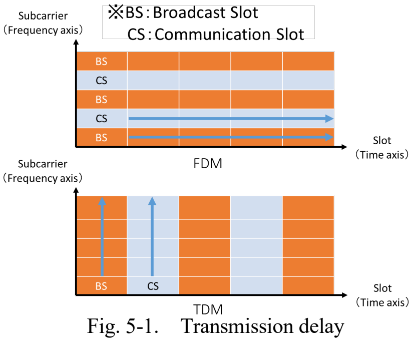


Fig. 5-1. Transmission delay

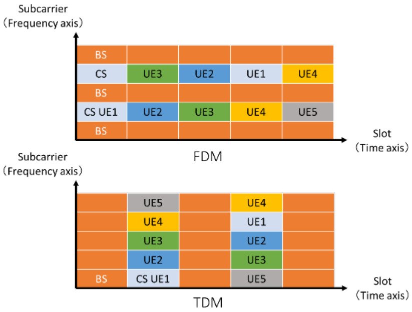


Fig. 5-2. Scheduling accuracy

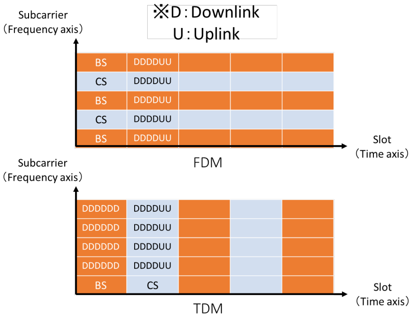


Fig. 5-3. TDD configuration

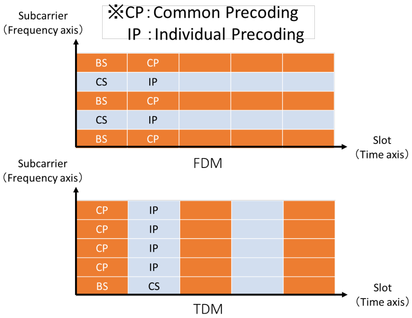


Fig. 5-4. MIMO Precoding

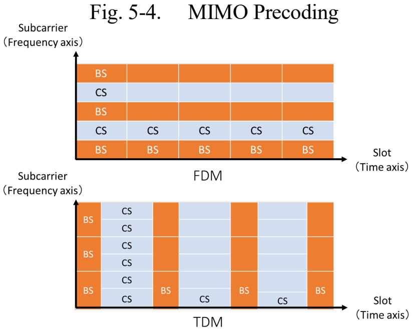


Fig. 5-5. Variable subcarrier spacing

Based on these qualitative considerations, it can be concluded that TDM is advantageous as a MIMO multiplexing method for both IP broadcasting and communication streams. More advanced multiplexing schemes that maximize the MIMO effect might be possible for further study.

4. A MIMO Radio Propagation Simulator for IP Broadcasting to Housing Complexes

4.1. Simulator Configuration

A simulator has been developed to model a cable TV IP broadcasting system utilizing MIMO for private 5G, with a focus on multi-dwelling reception. The simulator incorporates multi-dwelling house models, broadcasting models, and viewing models as input parameters to a conventional radio propagation simulator, while also considering MIMO parameters for enhancing transmission quality and speed. Fig.6 presents the overall configuration diagram.

The left portion of the diagram defines the one-to-one positional relationship between the transmitting base station and the housing complex as a parameter. The household structure within the complex is also defined as an input parameter, enabling the simulation of reception characteristics for each individual household.

This attenuation of radio waves as they penetrate from outdoors to indoors, known as O2I (Outdoor to Indoor) penetration loss, is modeled according to the methodology described in reference [10]. Simulations assume a distance of 100 m to 1 km between the base station and the housing complex, with the Fixed Wireless Access (FWA) line in a Line of Sight (LOS) environment. Considering the propagation loss at breakpoints in the LOS environment and the height of the base station antenna, we adopted the UMa (Urban Macro) model as the basic propagation loss calculation formula.

The upper part of the figure shows various configurable parameters from the transmitting antenna to the viewer model. These allow characteristic analysis and parameter optimization under each setting condition.

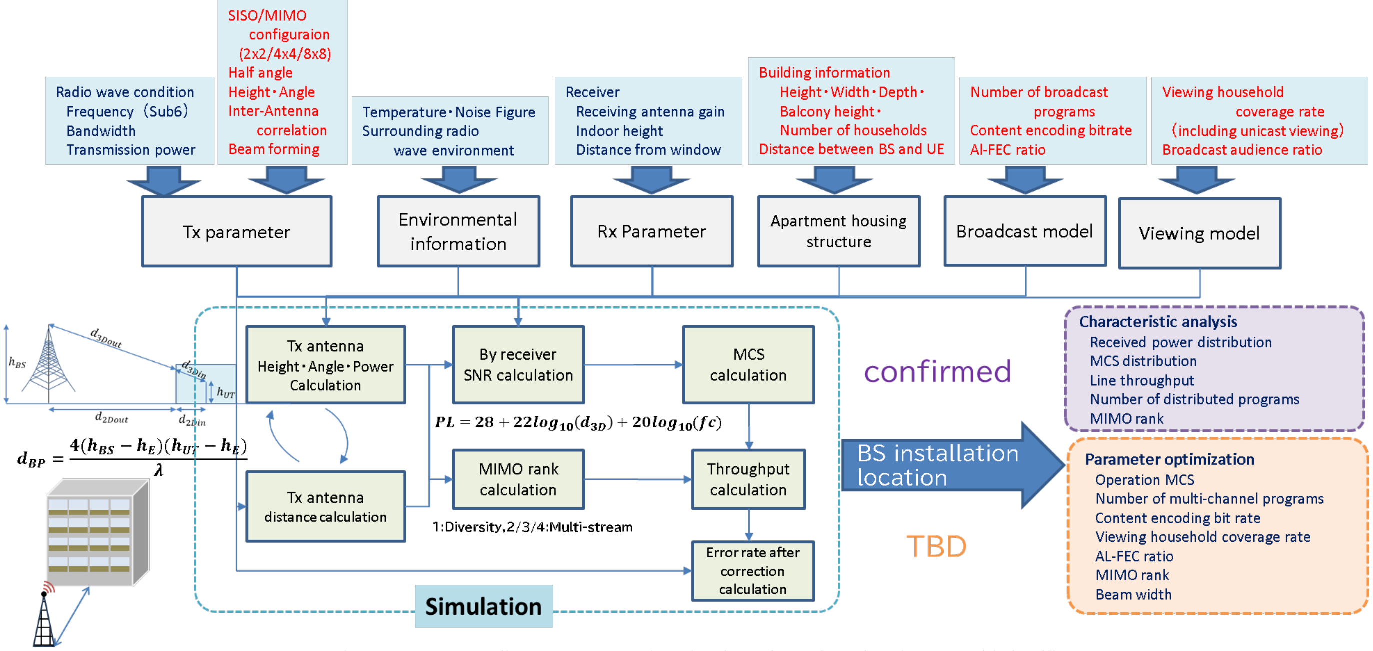
4.2. Input/Output Parameters and Analysis/Optimisation

The input parameters of the simulator encompass both SISO and MIMO configurations for the transmitting antennas. For MIMO, which is expected to provide a diversity effect on IP broadcast streams, the parameters can be adjusted to reflect the number of antennas, the correlation between MIMO propagation paths, and the implementation of beamforming. The structural parameters of the housing complex include building height, width, balcony height, number of households, and the distance between the base station and each household.

The broadcasting model incorporates parameters such as the number of IP broadcast programs, content coding bit rate, and the error correction coding rate (AL-FEC: Application Layer Forward Error Correction) applied in FLUTE encapsulation following MPEG-DASH. The viewing model allows for time-varying settings of household viewing rates and IP broadcast viewing rates. The simulator outputs include received power distribution, MCS distribution, transmission throughput, BLER distribution, MIMO rank distribution, and the number of IP broadcast-ready programs for each household in the complex.

The system can optimize parameters such as radio beamwidth relative to transmit power, MIMO rank, operational MCS, and AL-FEC ratio for a fixed base station distance.

In adaptive MBS, which assumes both IP broadcasting and unicast, the optimal configuration minimizes the sum of required resource blocks for IP broadcasting $B_b\left(k_b(t)\right)$ and broadcast complement unicast $B_u\left(k_u(t)\right)$. These are defined by the following equations.



Radio wave condition
Frequency（Sub6）
Bandwidth
Transmission power
SISO/MIMO configuraion (2x2/4x4/8x8)
Half angle
Height・Angle
Inter-Antenna correlation
Beam forming
Temperature・Noise Figure
Surrounding radio wave environment
Receiver
Receiving antenna gain
Indoor height
Distance from window
Building information
Height・Width・Depth・
Balcony height・
Number of households
Distance between BS and UE
Number of broadcast programs
Content encoding bitrate
Al-FEC ratio
Viewing household coverage rate
（including unicast viewing）
Broadcast audience ratio
Tx parameter
Environmental information
Rx Parameter
Apartment housing structure
Broadcast model
Viewing model
h_{BS}
d_{3Dout}
d_{3Din}
h_{UT}
d_{2Dout}
d_{2Din}
$d_{BP} = \frac{4(h_{BS}-h_E)(h_{UT}-h_E)}{\lambda}$
Tx antenna Height・Angle・Power Calculation
Tx antenna distance calculation
By receiver SNR calculation
$PL = 28 + 22log_{10}(d_{3D}) + 20log_{10}(fc)$
MIMO rank calculation
1:Diversity,2/3/4:Multi-stream
MCS calculation
Throughput calculation
Error rate after correction calculation
Simulation
confirmed
BS installation location
TBD
Characteristic analysis
Received power distribution
MCS distribution
Line throughput
Number of distributed programs
MIMO rank
Parameter optimization
Operation MCS
Number of multi-channel programs
Content encoding bit rate
Viewing household coverage rate
AL-FEC ratio
MIMO rank
Beam width

Figure 6: MIMO radio wave propagation simulator for IP broadcasting to multi-dwelling

$$B_b\left(k_b(t)\right) = L \cdot \frac{R\left(1 + A_f\right)}{D\left(k_b(t)\right)}$$

$$B_u\left(k_u(t)\right) = N \cdot \left\{1 - P\left(k_b(t)\right)\right\} \cdot \frac{R}{V_r D\left(k_u(t)\right)} \cdot \frac{1}{m(t)}$$

In the above equation

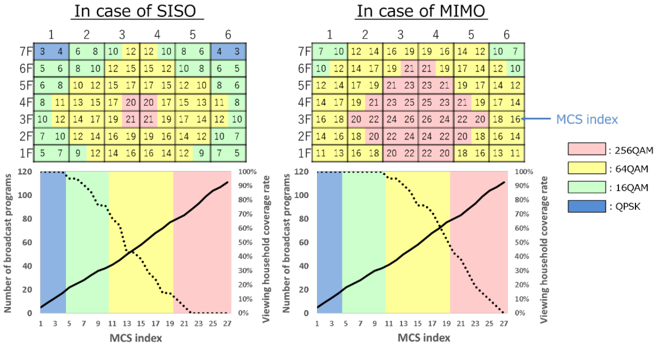
Content parameters: the number of broadcast programs is L, picture coding rate is R, and the FLUTE AL-FEC ratio is A_f.

Viewing parameters: the number of all households is N, the IP broadcast viewing household coverage $P(k)$, viewing rate is V_r. Transmission parameters: the common modulation index is $k_b(t)$ for all households for IP broadcasting, the individual household modulation index is $k_u(t)$ for unicast, the number of MIMO layers is $m(t)$, the bit capacity per resource block is $D\left(k(t)\right)$at modulation index k.

4.3. Some Results of the Simulation

Figure 7 presents the simulation results of the MCS index numbers derived from the received CN by two receiving antennas placed 3 meters apart on a balcony in each household in the (4×4) diversity MIMO and SISO configurations. The simulation assumes a medium-sized apartment complex with 7 floors and 42 households, and a base station (antenna output of 32 dBm) with a LOS environment located 90 m away. The figure compares the characteristics of IP broadcasting household coverage and the maximum number of multi-channel broadcast programs (10 Mbps/program) for each MCS index. The number of households capable of receiving 256QAM increases by over 30%, from 4 households with SISO to 18 households with MIMO.

Furthermore, under 100% viewing coverage conditions in the 100 MHz width of the sub-6GHz frequency band, the number of multi-channel broadcast programs doubles from 20 channels with SISO to 40 channels with MIMO.



In case of SISO

	1		2		3		4		5		6	
7F	3	4	6	8	10	12	12	10	8	6	4	3
6F	5	6	8	10	12	15	15	12	10	8	6	5
5F	6	8	10	12	15	17	17	15	12	10	8	6
4F	8	11	13	15	17	20	20	17	15	13	11	8
3F	10	12	14	17	19	21	21	19	17	14	12	10
2F	7	10	12	14	16	19	19	16	14	12	10	7
1F	5	7	9	12	14	16	16	14	12	9	7	5

In case of MIMO

	1		2		3		4		5		6	
7F	7	10	12	14	16	19	19	16	14	12	10	7
6F	10	12	14	17	19	21	21	19	17	14	12	10
5F	12	14	17	19	21	23	23	21	19	17	14	12
4F	14	17	19	21	23	25	25	23	21	19	17	14
3F	16	18	20	22	24	26	26	24	22	20	18	16
2F	14	16	18	20	22	24	24	22	20	18	16	14
1F	11	13	16	18	20	22	22	20	18	16	13	11

Figure 7: MCS distribution in multi-dwelling, number of broadcast programs and viewing household coverage rate for MCS index

5. Conclusion

The replacement of wired cable TV services by private 5G FWA is an urgent issue for the cable industry, and MIMO technology, which has conventionally been discussed at the physical layer, has been examined from the viewpoint of the higher layer of IP broadcasting.

Based on the recognition that MIMO technology is promising for large-capacity, real-time, high-quality transmission of IP broadcasting in limited wireless bandwidth, the study of system implementation as well as more detailed quantitative performance analysis remains an issue from the viewpoint of the coexistence of IP broadcasting and data communication.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

This research and development was carried out under the “Research and development of frequency effective use technology for wireless transmission of IP multicast broadcasting” commissioned by the Ministry of Internal Affairs and Communications in 2025.

References

[1] D. Gomez-Barquero, W. Li, M. Fuentes, J. Xiong, G. Araniti, C. Akamine, J. Wang, “IEEE Transactions on Broadcasting Special Issue on: 5G for Broadband Multimedia Systems and Broadcasting”, IEEE Transactions on Broadcasting, **65(2)**:351-355, 2019, DOI: 10.1109/TBC.2019.2914866

[2] 3GPP, “Architectural enhancements for 5G multicast-broadcast services”, 3rd Generation Partnership Project Technical Specification, **23.247(17.11.0)**:1-121, 2024.

[3] T. Ohgane, Y. Ogawa, “Easy to Understand MIMO System Technology”, Ohmsha, 1-208, 2009.

[4] A. Benjebbour, “Evolution of MIMO transmission technology in mobile communication systems”, The Journal of the Institute of Image Information and Television Engineers, **70(1)**:28-34, 2016, DOI: 10.3169/itej.70.28

[5] R. Zhang, C. K. Ho, “MIMO broadcasting for simultaneous wireless information and power transfer”, IEEE Transactions on Wireless Communications, **12(5)**:1989-2001, 2013, DOI: 10.1109/TWC.2013.031813.120224

[6] J. P. Lemayian, J. M. Hamamreh, “Hybrid MIMO: A New Transmission Method For Simultaneously Achieving Spatial Multiplexing and Diversity Gains in MIMO Systems”, RS Open Journal on Innovative Communication Technologies, **2(4)**: 2021, DOI: 10.46470/03d8ffbd.549d270b

[7] S. Asakura, S. Kawashima, T. Ijiguchi, K. Kambara, M. Okano, “Performance analysis of dual-polarized MIMO-SFN considering imbalance of received power”, ITE Transactions on Media Technology and Applications, **10(3)**:100-109, 2022, DOI: 10.3169/mta.10.100

[8] H. Shimizu, S. Matsumoto, “Proposal of adaptive MBMS for multi-channel video distribution by private 5G and its effectiveness”, Proceedings of the Institute of Image Information and Television Engineers Annual Convention, **2022(32C-2)**, 2022.

[9] ITU-T, “System architecture for cable television services to use IMT-2020 radio system”, International Telecommunication Union Telecommunication Standardization Sector Recommendation, **J.153(06/2024)**:1-13, 2024.

[10] 3GPP, “Study on channel model for frequencies from 0.5 to 100 GHz”, 3rd Generation Partnership Project Technical Report, **38.901(14.3.0)**:1-95, 2018.

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 10, No. 6, 29-41 (2025)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTESJ
ISSN: 2415-6698

StradNet: Automated Structural Adaptation for Efficient Deep Neural Network Design

David Degbor[1], Haiping Xu[*1], Pratiksha Singh[1], Shannon Gibbs[1], Donghui Yan[2]

[1] *Computer and Information Science Department, University of Massachusetts Dartmouth, Dartmouth, MA 02747, USA*

[2] *Mathematics Department, University of Massachusetts Dartmouth, Dartmouth, MA 02747, USA*

ARTICLE INFO

Article history:
Received: 14 October, 2025
Revised: 27 November, 2025
Accepted: 29 November, 2025
Online: 05 December, 2025

Keywords:
Machine learning
Deep neural network (DNN)
Structurally adaptive DNN
Automated network design
Dynamic environments

ABSTRACT

Deep neural networks (DNNs) have demonstrated remarkable success across a wide range of machine learning tasks. However, determining an effective network architecture, particularly the sizes of the hidden layers, remains a significant challenge and often relies on inefficient trial-and-error experimentation. In this paper, we propose an automated architecture design approach based on structurally adaptive DNNs, referred to as StradNet models. Our method begins with training a fully connected DNN (FC-DNN) initialized with standard hidden layer sizes. The structure is then progressively adapted by iteratively pruning weak connections, removing isolated neurons, and fine-tuning the network, producing a series of partially connected DNN (PC-DNN) models that converge toward an effective configuration. Unlike many pruning methods that reduce redundancy only after full training, StradNet integrates structural adaptation directly into the learning process, enabling the network to evolve as it learns. This adaptability makes StradNet well-suited for domains with shifting data distributions and complex, high-dimensional dependencies. Experiments on dynamic environments, such as marine datasets, demonstrate that StradNet produces efficient and scalable models that consistently outperform conventionally pruned FC-DNNs. Overall, this automated strategy enhances the efficiency of DNN design and provides a practical framework for real-world machine learning applications.

1. Introduction

Deep neural networks (DNNs) have achieved remarkable success across a wide range of machine learning applications, including speech recognition, image classification, and natural language processing [1]-[3]. Their ability to approximate highly complex nonlinear functions arises from the presence of multiple hidden layers and large numbers of parameters [4]. These hidden layers allow DNNs to extract hierarchical features from raw input data, capturing both low-level and high-level patterns. For instance, convolutional neural networks (CNNs) learn basic patterns such as edges and textures in their early layers, while deeper layers identify more abstract object features [5]. In contrast, recurrent neural networks (RNNs) and long short-term memory (LSTM) architectures are designed to capture temporal or sequential dependencies, making them effective for language modeling and time-series prediction [6]. In recent years, modern large-scale architectures, such as large language models (LLMs) [7], have pushed the boundaries of deep learning. With billions of parameters, these systems demonstrate extraordinary expressive power, excelling in tasks ranging from text summarization to code generation. These models also exhibit cross-task generalization, transferring knowledge from one domain to another and enabling zero-shot and few-shot learning scenarios that were previously unattainable with smaller networks. Moreover, innovations such as attention mechanisms and transformer architectures have further expanded the capacity of DNNs to model long-range dependencies and capture complex contextual information [8].

Despite these successes, the increasing size and complexity of DNNs present significant challenges. Large models require substantial computational resources and massive training datasets, making their deployment on resource-constrained devices often impractical [9]. Furthermore, a high number of parameters increases the risk of overfitting, where the model memorizes training noise rather than capturing generalizable patterns. Conversely, overly small models may suffer from underfitting, failing to capture important structures in the data. Thus, striking the right balance between model complexity, generalization capability, and computational efficiency is critical for building effective and scalable DNNs [10].

[*] Corresponding Author: Haiping Xu, University of Massachusetts Dartmouth, Dartmouth, MA 02747, Email: hxu@umassd.edu

https://dx.doi.org/10.25046/aj100603

A key determinant of this balance lies in the number of neurons and connections in the hidden layers, which directly influence both the representational capacity and computational cost of DNNs. Existing methods typically rely on manual hyperparameter tuning or grid/random search, which are computationally expensive, time-consuming, and heavily dependent on domain expertise [11]. To address these challenges, various technical approaches have been proposed to optimize network structures more systematically. Model compression methods, such as quantization and pruning, reduce resource demands while maintaining accuracy [12], [13]. However, traditional pruning methods typically operate within a fixed network structure, limiting their ability to fully exploit structural redundancy. Neural Architecture Search (NAS) frameworks attempt to automate network design by exploring a large architecture space using reinforcement learning, evolutionary algorithms, or gradient-based optimization [14]. While NAS can generate high-performance architectures, its computationally intensive nature may still result in overly parameterized networks that are difficult to deploy efficiently.

Given these limitations, there is a pressing need for adaptive and automated methods that can optimize both the number of neurons and their interconnections. Such approaches would not only improve computational efficiency but also reduce overfitting risks and enhance generalization capabilities. To address this need, this paper introduces structurally adaptive DNNs, referred to as StradNet models. Specifically, this work aims to develop StradNet to dynamically adapt network structures, improving computational efficiency while maintaining competitive accuracy, and automating pruning and tuning processes to handle complex, noisy datasets from dynamic environments. This results in both time savings and scalability improvements over conventional approaches. Our approach begins with training a fully connected DNN (FC-DNN) initialized with standard hidden layer sizes. The network is then progressively adapted through iterative pruning of weak connections, removal of isolated neurons, and fine-tuning of the remaining weights, resulting in lightweight yet high-performing network architectures. Unlike conventional pruning or NAS methods, StradNets dynamically adapt the network structure, enabling more efficient and generalizable partially connected DNN (PC-DNN) models for diverse real-world applications. The main contributions and novelties of this paper are summarized as follows:

- Proposed a systematic approach for identifying and pruning weak connections in DNNs, enhancing both computational efficiency and generalization capability.
- Introduced mechanisms to detect and remove isolated neurons along with their associated connections, simplifying the network structure without compromising accuracy.
- Demonstrated that iterative structural adaptation, combined with network fine-tuning, effectively balances model complexity and predictive performance.
- Demonstrated that efficient network architectures do not require a fixed hidden-layer ratio, and that StradNet adaptation allows models to adjust layer sizes for specific tasks.
- Evaluated the StradNet framework in dynamic environments, such as marine datasets, and demonstrated that it outperforms conventionally pruned FC-DNNs in terms of computational efficiency and scalability.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed automated framework for DNN design. Section 4 describes the procedures for automated structural adaptation of DNNs. Section 5 provides case studies that validate the feasibility and effectiveness of the StradNet architecture. Finally, Section 6 concludes the paper and outlines future research directions.

2. Related Work

In recent years, researchers in deep learning and neural networks have made substantial progress in developing strategies to improve classifier performance. These efforts include making existing approaches not only more accurate but also more resource-efficient, thereby advancing both model architectures and optimization techniques [15], [16]. In machine learning applications, models typically require both learned parameters and hyperparameters to be carefully adjusted. As network sizes grow, manual tuning becomes increasingly difficult, often requiring domain expertise or careful monitoring of experimental results, and if done poorly, it can degrade performance. To address these challenges, researchers have been exploring automated methods that enhance the performance of learning algorithms across a variety of tasks. For example, in [17], the researchers presented an optimization strategy that uses a population-based evolutionary search algorithm to identify promising hyperparameter settings in EEG classification. The optimization process proceeds in a single tuning cycle, where at each stage past evaluation results are compared and leveraged to generate improved parameter settings. Likewise, in [18] the authors described a software-testing-inspired approach to hyperparameter optimization. Presented as a constrained exhaustive method, also known as t-way combinatorial testing, the goal is to evaluate a subset of hyperparameter configurations on the model. In [19], the authors approached hyperparameter tuning from a different perspective by applying asynchronous reinforcement learning to seek optimal configurations. Their methodology employs a master agent to coordinate a group of worker agents that continuously evaluate hyperparameter settings. In [20], the authors introduced a NAS method aimed at mitigating the computational challenges of identifying effective DNN structures. They represented the initial network structure as a root node in a tree and recursively evaluated child nodes using an improved Monte Carlo Tree Search algorithm. In [21], the authors proposed a neural network compression method that removes redundant features from the convolutional layer within a multihead CNN architecture. Their approach computes multi-dimensional principal components on the convolutional layers using a statistically guaranteed hyperparameter optimization scheme. While the aforementioned approaches provide valuable insights into effective DNN design, some methods exhaustively evaluate possible configurations, whereas others attempt a one-shot search process. Unlike these approaches, StradNet employs an adaptive strategy that iterates through multiple cycles to identify an effective hyperparameter configuration, particularly the hidden layer sizes. The process can also terminate early if low accuracy is detected, saving significant time compared to other architecture search algorithms. Techniques such as NAS, combinatorial methods, and evolutionary algorithms are effective for navigating the complex, high-dimensional hyperparameter space; however, they are typically computationally expensive, and their application to partially connected architectures has not been fully explored. In contrast, a key aspect of our approach is the use of PC-DNNs as lightweight and efficient alternatives to conventional FC-DNN architectures. Accordingly, this work complements existing research that primarily focuses on FC-DNN-based designs.

Research on improving model efficiency through structural adaptation has become an active area of study, leveraging automated pruning techniques to facilitate the transition from FC-DNNs to PC-DNNs. Studies across various pruning strategies show that the automated conversion process can maintain or even improve the accuracy of FC-DNNs [22]. More broadly, automated machine learning (AutoML) integrates algorithms for dynamic model construction and deployment, as well as hyperparameter optimization [23]. Frameworks such as the Dynamic Processing Unit (DPU) presented in [24] show that network compression can take a hybrid approach, where weights are not permanently removed or masked but instead oscillate between active and inactive states depending on the DPU. The authors in [25] explored an Echo State Network (ESN) structure designed to provide simplicity for neural networks running on edge devices. Their approach prunes neurons based on a neuron-importance heuristic combined with iterative fine-tuning. In [26], the researchers presented an activation-profile-based neuron pruning framework that targets neurons during inference and groups them according to similar outputs. In their approach, neurons with outputs that are too small are pruned, while those with sufficiently large outputs are retained. In [27], the authors introduced direct connections from each layer to the output layer, providing multiple substructures to increase network capacity. They then apply a training scheme with L1 regularization to encourage the removal of redundant neurons and facilitate the adjustment of hidden layer sizes. The authors in [28] addressed the structural adaptation of neural networks by constructing uniform or non-uniform subnets. These subnets can be sampled across different epochs to leverage one-shot training and provide a runtime advantage during inference. In [29], the authors presented a DNN-based approach for building budget-constrained models for big data analysis. In their approach, they demonstrated how to eliminate less important features by identifying weak links and neurons, thereby bringing the model cost within a given budget. Building on existing methods for structural adaptation, StradNet introduces a distinct and effective strategy for optimizing network architecture. A key challenge in this domain is the continual growth of network size and the associated computation and storage overhead from weak connections. StradNet addresses this issue by identifying and pruning weights that contribute little to the model's output, thereby improving efficiency without compromising accuracy. Traditional structurally adaptive algorithms often depend on neuron importance scores to guide pruning, a process that can be both tedious and computationally intensive. In contrast, StradNet tracks the number of connections pruned for each neuron and removes a neuron only when all its connections have been eliminated, leading to more compact and efficient PC-DNN architectures. PC-DNNs and structural adaptation have gained increasing attention for their potential to balance model expressiveness and computational efficiency. Previous studies have explored how connectivity patterns affect network performance or specific aspects of partially connected designs. However, automated structural adaptation within PC-DNNs remains a relatively underexplored area. Recent advancements, such as adaptive learning rate algorithms [30], have further enhanced model convergence and generalization, yet structural adaptation in PC-DNNs continues to present promising opportunities for future research.

Previous research on optimizing hyperparameters for dynamically changing environments has explored how neural networks can be applied to tasks requiring adaptability and robustness under shifting conditions. Applications such as oceanography, weather forecasting, stock market analysis, and housing price prediction all involve underlying patterns that fluctuate over time, highlighting the need for flexible models. In such highly volatile domains, it is insufficient to train a model once and rely on it consistently; instead, a model must adapt to complex changes in data and restructure dynamically. In the field of Scientific Machine Learning (SciML), both machine learning and mechanistic modeling techniques have been independently and successfully applied in systems biology [31]. Machine learning excels at uncovering statistical relationships and generating quantitative predictions from data, whereas mechanistic modeling provides a robust framework for capturing domain knowledge and elucidating the causal mechanisms driving dynamic biological processes. In [32], the authors evaluated the impact of diverse marine environments on image classification performance using these methods. They reported that their methodology effectively demonstrates how well models can generalize across a wide range of conditions. The study in [33] examined a weather application, presenting a hybrid machine learning approach for rainfall prediction. They developed a useful machine learning tool capable of identifying potential threats in real time and issuing timely warnings. In [34], the authors proposed an Environmental Graph-Aware Neural Network (EGAN) for modeling and analyzing large-scale, multi-modal environmental datasets. The EGAN framework constructs a spatiotemporal graph representation that integrates physical proximity, ecological similarity, and temporal dynamics, and employs graph convolutional encoders to extract expressive spatial features. In [35], the authors investigated the housing market using DNN predictions. Forecasting the housing market is highly complex and high-dimensional, which can make it challenging for machine learning models to capture underlying patterns. Finally, [36] extended this analysis to the stock market. Despite the availability of years of data, financial markets are heavily influenced by external factors such as geopolitical events and investor sentiment, making accurate predictions difficult. These studies underscore the importance of designing models that can adapt continuously, rather than relying on static training procedures. In our approach, we selected datasets from dynamic environments, including oceanographic and marine settings, where conditions vary seasonally and data are often high-dimensional. Marine factors such as turbidity, depth, light variation, image resolution, camera calibration, and motion blur further increase the challenge of achieving robust generalization. In some applications, only a single training cycle has been used to capture chaotic patterns in a one-shot manner. While feasible, this approach often struggles in dynamic environments, requiring frequent retraining to maintain reliable predictions. In contrast, our multi-step pruning loop can be fine-tuned and restructured in real time to adapt to new data, providing a general solution for developing efficient PC-DNN models in dynamic environments.

3. An Automated Framework for DNN Design

In deep learning, a network's structure, particularly the configuration of its hidden layers, plays a crucial role in determining accuracy, efficiency, and generalization performance. Unlike generic hyperparameters such as learning rate, batch size, or number of epochs, which primarily control training behavior, structural parameters directly define the capacity, connectivity, and representational power of the network itself. Manually selecting an effective structure is especially challenging because the search space of possible layer sizes, neuron counts, and inter-layer connections is extremely large and combinatorially complex. To address this challenge, we propose an automated framework that adapts hidden layer sizes during training. Our method follows a two-phase approach. In Phase 1, the network undergoes aggressive pruning to remove less significant connections and neurons. If pruning exceeds the optimal point, Phase 2 readjusts by restoring connections and pruning more conservatively. This adaptive procedure ensures convergence toward an efficient structure without extensive manual intervention. The key idea is the progressive transformation of an initialized FC-DNN into a more efficient PC-DNN. By dynamically removing weak connections and isolated neurons, the framework reduces computation during forward and backward propagation, improving efficiency while maintaining accuracy. To emphasize the structural effects, all other hyperparameters are held constant. Figure 1 illustrates the process, beginning with dataset preprocessing, including cleaning and normalization, followed by iterative pruning and refinement until the network stabilizes at a desirable structure.

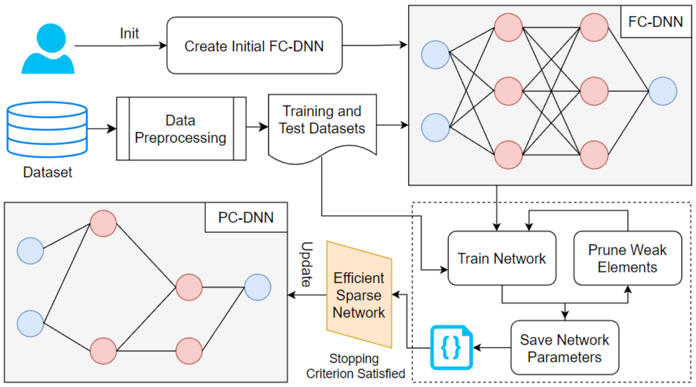


Figure 1: An automated framework for DNN design.

As shown in Figure 1, the dataset is first preprocessed by normalizing input features using the min-max scaling method as in (1). Min-max scaling ensures comparability across features, improves convergence speed, and preserves relative relationships among input values. This procedure rescales all features to the range [0, 1], transforming raw data into a normalized space that the network can effectively interpret while retaining the original distribution.

$$x_k{}' = \frac{x_k - x_{k\_min}}{x_{k\_max} - x_{k\_min}} \quad (1)$$

where x_{k_min} and x_{k_max} are the minimum and maximum values of input feature k, x_k is the original feature value, and $x_k{}'$ is the normalized feature value.

After preprocessing, the dataset is divided into training and test sets to enable evaluation on unseen data, typically using a 70:30 or 80:20 split to ensure sufficient data for both sets. The automated structural adaptation process requires user interaction only during network initialization. Hidden layer sizes are set in powers of two. For example, a network can be initialized with two hidden layers of 512 and 256 neurons, respectively, which serve as standard starting sizes. All other hyperparameters (e.g., number of layers, learning rate, loss function) remain fixed for comparability. Once initialized and prepared, the framework iteratively adjusts the network structure through a prune-train-test loop until a predefined stopping criterion is satisfied, resulting in an efficient sparse network as a PC-DNN. During each iteration, the network is trained and evaluated on the split dataset, and the maximum accuracy is tracked to identify the most promising state for subsequent refinements. Although the base network starts with random weights, pruning and retraining preserve previously learned knowledge by saving the network parameters instead of reinitializing them.

To enable iterative and adaptive structural refinement, we construct two-dimensional mask matrices as part of the saved network parameters to continuously track pruned connections. Each weight tensor is paired with a matrix of size $i \times j$, matching its dimensions to record pruning decisions precisely. At initialization, all entries are set to *false*, indicating that no connections have been removed; when a connection is pruned, the corresponding entry is switched to *true*. These mask matrices allow the framework to maintain an explicit and transparent record of which connections have been pruned, enabling reproducible and verifiable updates to the network structure during iterative training. The extent of pruning is governed by the pruning ratio, a tunable hyperparameter that controls pruning aggressiveness: higher ratios prune more aggressively, while lower ratios perform pruning more conservatively. This parameter provides flexibility to explore different strategies and balance computational efficiency against potential accuracy degradation. Pruning decisions specifically target weak elements, including individual connections and entire neurons, that contribute minimally to model performance while consuming computational resources. Retaining such redundant elements increases training time and memory usage while offering negligible performance gains. By systematically eliminating these weak components, the framework concentrates computational resources on the most informative parts of the network, thereby enhancing both training efficiency and learning effectiveness. Overall, our StradNet framework aims to reduce network size and training time by removing weak elements, resulting in compact and efficient PC-DNN models without sacrificing accuracy. Furthermore, by integrating mask-based tracking with controlled pruning, the framework ensures that network adaptation remains fully automated, reproducible, and robust across diverse datasets, initialization conditions, and dynamically evolving data distributions.

4. Automated Structural Adaptation of DNNs

4.1. Pruning Weak Connections in a DNN

The first step in network reduction is to prune weak connections by removing those with the smallest absolute weights, as determined by a predefined minimal pruning ratio, p_{min}. As defined in (2), the minimal pruning ratio p_{min} is the proportion of connections removed from the DNN during pruning relative to the total number of connections before pruning.

$$p_{min} = \frac{N_{\text{removed}}}{N_{\text{total}}} \times 100\% \qquad (2)$$

where N_{removed} is the number of pruned connections, and N_{total} is the total number of connections prior to pruning. For example, if p_{min} = 10%, the 10% weakest connections are removed. We refer to this as the *minimal* pruning ratio because, in the next step (described in Section 4.2), removing isolated neurons also requires eliminating their associated connections. Consequently, the actual pruning ratio in each round may exceed p_{min}.

This pruning step reduces the neural network's complexity while maintaining high performance in the resulting PC-DNN. The network to be pruned may be an FC-DNN in the first pruning round or a PC-DNN if pruning has already been applied. To identify weak connections, we sort all active weights in ascending order of magnitude. The pruning threshold is computed from the absolute weight values, since the influence of a connection depends on its magnitude rather than its sign: a large negative weight represents a strong inhibitory effect, while a large positive weight represents a strong excitatory effect. Magnitude-based pruning is effective because small-magnitude weights have minimal impact on downstream activations. In contrast, large-magnitude weights encode more informative features. Removing the weakest weights therefore eliminates low-impact parameters while preserving essential computational pathways of the network. As discussed in Section 3, mask matrices track which connections have been removed.

Algorithm 1 outlines the procedure for removing weak connections. As shown in the algorithm, the pruning threshold *t* is determined using the minimal pruning ratio p_{min} and the k^{th} smallest element (step 6 and 7), thereby identifying the weakest proportion p_{min} of weights. The algorithm then removes all weights whose magnitudes fall below this threshold. Given the array of weight values R_w and the number k_{w} of smallest elements to remove, the value t_{w} is set as the largest of these k_{w} elements and serves as the pruning cutoff for the entire network in the current iteration. When a weak connection in a weight tensor Θ is flagged for removal, the corresponding entry in the mask matrix is set to *true* (step 10), ensuring that the training process excludes this connection in subsequent updates. After all weak connections are marked, the algorithm returns the updated PC-DNN (step 11). This threshold-based procedure ensures that pruning decisions remain consistent across layers, regardless of their size or scale. Because the threshold is computed directly from the sorted weight magnitudes, the algorithm adaptively adjusts to the distribution of weights in each pruning round. This makes the pruning robust to training dynamics, including changes during fine-tuning or shifts in connection importance. Moreover, by eliminating only the weakest connections at each iteration, the algorithm avoids abrupt structural changes that could destabilize the learning process, allowing the model to gradually converge toward a compact and efficient PC-DNN.

Algorithm 1: Pruning Weak Connections

Input: DNN *N* with mask matrices, minimal pruning ratio p_{min}
Output: A PC-DNN with weak connections pruned

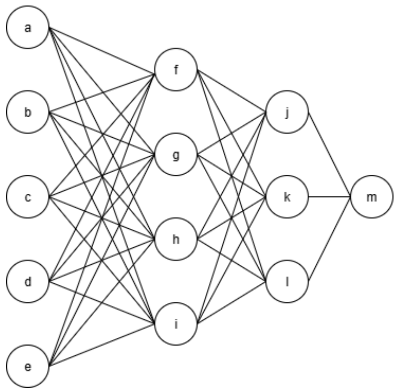
```
1.  Initialize an empty list $R_w$ for remaining (active) weights
2.  for each weight tensor Θ in N
3.      for each weight w in Θ with corresponding mask value = false
4.          Append w to $R_w$
5.  Sort $R_w$ in ascending order
6.  Compute $k_w = \lfloor p_{min} * len(R_w) \rfloor$   // the number of weights to prune
7.  Set pruning threshold $t_w = R_w[k_w]$  // the $k^{th}$ smallest weight
8.  for each weight tensor Θ in N
9.      for each connection c in weight tenor Θ
10.         Mark c as true (pruned) in the mask matrix if weight $w_c \le t_w$
11. return updated N
```

Figure 2 shows a sample DNN used to illustrate the pruning of the weakest weight connections. In this example, the network is initialized as an FC-DNN with two hidden layers, containing four and three hidden neurons, respectively. The weights for each connection are displayed in the three weight tensor tables on the left, providing a clear view of the initial parameter distribution. The pruning threshold is determined dynamically based on the current minimal pruning ratio and the weight values in the tables, ensuring that the weakest connections are consistently identified and removed during each pruning round.

	a	*b*	*c*	*d*	*e*
f	0.4	0.5	0.9	0.3	0.7
g	0.7	0.1	0.4	0.6	0.4
h	0.4	0.6	0.5	0.8	0.8
i	0.1	0.9	0.5	0.4	0.6

	f	*g*	*h*	*i*
j	0.2	0.4	0.6	0.5
k	0.2	0.7	0.8	0.5
l	0.1	0.3	0.9	0.8

	j	*k*	*l*
m	0.5	0.2	0.3



(a) Connection weights in the FC-DNN (b) Network structure of the FC-DNN

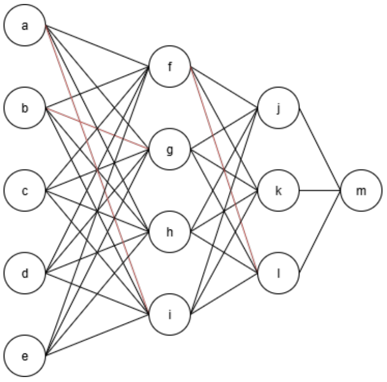
Figure 2: A sample FC-DNN initialized with two hidden layers

Figure 3 illustrates the DNN after pruning weak connections with a minimal pruning ratio of p_{min} = 10%. According to Algorithm 1, k_w is set to 3, and the pruning threshold t_w to 0.1, meaning any connection with a weight less than or equal to 0.1 is flagged for pruning and removed from the neural network. As shown in the figure, all connections flagged for pruning are highlighted in red in both the network diagram and its table representation. After compressing the model's connections by 10%, we can observe which neurons contribute to strong activations and thus influence the network's output. Since no isolated hidden neurons are present after this round of pruning, the same 10% minimal pruning ratio is applied in a second round without removing any neurons. This process can be repeated iteratively until a performance drop is observed.

	a	*b*	*c*	*d*	*e*
f	0.4	0.5	0.9	0.3	0.7
g	0.7	0.1	0.4	0.6	0.4
h	0.4	0.6	0.5	0.8	0.8
i	0.1	0.9	0.5	0.4	0.6

	f	*g*	*h*	*i*
j	0.2	0.4	0.6	0.5
k	0.2	0.7	0.8	0.5
l	0.1	0.3	0.9	0.8

	j	*k*	*l*
m	0.5	0.2	0.3



(a) Connection weights in the PC-DNN (b) Network structure of the PC-DNN

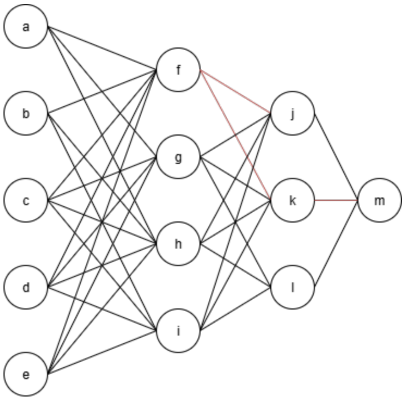
Figure 3: The PC-DNN after pruning weak connections with p_{min} = 10%

Figure 4 shows the updated PC-DNN after retraining. Previously pruned weights are marked as "n/a" in the weight tensor tables, indicating their removal from the network and ensuring that these inactive connections are permanently excluded from all future computations and parameter updates.

	a	*b*	*c*	*d*	*e*
f	1.4	0.6	1.3	0.5	1.7
g	2.7	n/a	1.0	2.0	0.9
h	1.1	1.5	0.7	0.6	0.5
i	n/a	1.9	1.2	2.4	2.2

	f	*g*	*h*	*i*
j	0.3	0.8	0.9	0.5
k	0.3	1.7	2.3	0.7
l	n/a	0.6	1.0	1.3

	j	*k*	*l*
m	2.5	0.3	0.8



(a) Connection weights in the PC-DNN (b) Network structure of the PC-DNN

Figure 4: The PC-DNN after pruning weak connections again with p_{min} = 10%

According to Algorithm 1, in the second round of weak connection pruning, k_w is set to 3, the pruning threshold t_w across all layers is set to 0.3. The computed threshold does not affect any connections in the first weight tensor. However, in the second and third weight tensors, connections (*f*, *j*), (*f*, *k*), and (*k*, *m*) are identified as meeting the pruning threshold and are therefore removed, further simplifying the PC-DNN structure.

4.2. Removing Isolated Neurons and Associated Connections

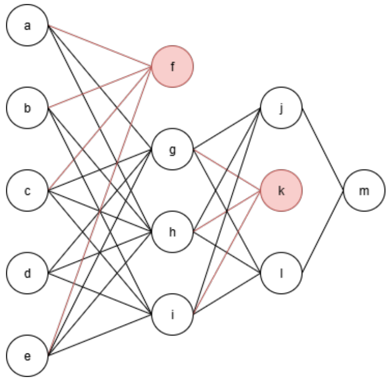
Following the example in Section 4.1, we are left with a PC-DNN containing isolated neurons whose input or output connections have been entirely pruned. We extend the concept of model compression by removing these neurons, which also eliminates their associated incoming or outgoing connections. Figure 5 provides a visual representation of this process, where neurons *f* and *k* are identified as isolated and subsequently removed along with their connections.

A key feature of the model design is that weak connections are removed before weak neurons. This ordering matters because a neuron becomes isolated only after all its input or output connections are pruned. As shown in Figure 3, some pruning rounds produce no isolated neurons, so the neuron-removal step is skipped. When a neuron does lose all incoming or outgoing connections, it is marked as isolated and removed. No pruning ratio is defined for neurons in our approach, because their removal depends solely on the absence of connections. Algorithm 2 summarizes the procedure for detecting and deleting isolated neurons and their associated connections.

	a	*b*	*c*	*d*	*e*
f	1.4	0.6	1.3	0.5	1.7
g	2.7	n/a	1.0	2.0	0.9
h	1.1	1.5	0.7	0.6	0.5
i	n/a	1.9	1.2	2.4	2.2

	f	*g*	*h*	*i*
j	n/a	0.8	0.9	0.5
k	n/a	1.7	2.3	0.7
l	n/a	0.6	1.0	1.3

	j	*k*	*l*
m	2.5	n/a	0.8



(a) Connection weights in the PC-DNN (b) Network structure of the PC-DNN

Figure 5: The PC-DNN with identified isolated neurons

Algorithm 2: Removing Isolated Neurons and Their Connections

Input: DNN *N* with mask matrices
Output: A PC-DNN with isolated neurons and their associated connections removed

```
1.  repeat
2.      Set removed = 0
3.      for each hidden layer $\Psi$ in N
4.          for each undeleted neuron $\theta$ in $\Psi$
5.              Let $W_{in}$ = incoming connections of $\theta$
6.              Let $W_{out}$ = outgoing connections of $\theta$
7.              if all connections in $W_{in}$ are pruned (mask = true)
8.                  Delete neuron $\theta$ and all connections in $W_{out}$
9.                  Increment removed by 1
10.             else if all connections in $W_{out}$ are pruned (mask = true)
11.                 Delete neuron $\theta$ and all connections in $W_{in}$
12.                 Increment removed by 1
13. until removed = 0
14. Update N and its mask matrices
15. return N
```

As shown in Algorithm 2, for an undeleted hidden neuron θ, let W_{in} and W_{out} denote its sets of incoming and outgoing connections, respectively. If all connections in W_{in} or W_{out} are pruned, θ is identified as isolated. In this case, θ is marked for deletion, and all its associated connections are removed from the network (step 8 and 11). All hidden neurons in each hidden layer must be examined for potential removal. To ensure that both existing isolated neurons and any new isolated neurons resulting from the removal of their connections are identified, we define a counter *removed* to record the number of hidden neurons eliminated in each deletion round. This process is repeated until no additional hidden neurons are identified in the current round. At that point, the PC-DNN is updated by removing all identified isolated hidden neurons and their associated connections, and the corresponding mask matrices are updated accordingly (step 14).

4.3. Automated Structural Adaptation Process

As discussed in Section 3, the automated structural adaptation process follows a two-phase pruning strategy. Phase 1 performs coarse pruning across the hidden-layer space, inspired by the lottery ticket hypothesis [37], which suggests that subnetworks within a large model can achieve performance comparable to the full network. Starting from an FC-DNN initialized with an input layer, a number of hidden layers, and an output layer, connections are pruned using a minimal pruning ratio p_{min} (e.g., 10%), and isolated neurons are removed according to Algorithms 1 and 2, respectively. Weights are preserved between iterations to speed convergence, and metrics such as accuracy, neuron count, total weights, and pruning thresholds are tracked. This coarse stage allows the model to rapidly explore a wide range of architectural variations and discard obviously redundant structures. By aggressively pruning early, the network is guided toward a more compact region of the structural space where promising sub-architectures reside. Pruning continues until accuracy drops by more than 1% below the best observed value, at which point the previous best model is restored as the baseline for Phase 2, which applies fine-grained adjustments to finalize the network structure. The duration of Phase 1 depends on the initial hidden layer sizes and the stopping criterion. Larger starting layers generally require more pruning cycles (e.g., 20-30 iterations), whereas well-chosen sizes converge faster (e.g., 5-10 iterations). The stopping criterion balances under-pruning, which leaves the network over-parameterized, and over-pruning, which degrades accuracy. This ensures that Phase 1 terminates at a structurally meaningful point, providing a stable foundation for the more targeted refinements performed in Phase 2.

Phase 2 further refines the network structure established in Phase 1 by applying a smaller pruning ratio, where p_{min} is halved (e.g., 5% in our implementation) to achieve finer-grained structural adjustments. Starting from the best-performing model obtained from Phase 1, Phase 2 executes a single prune-train-test cycle to fine-tune neuron counts and consolidate model stability. Only one iteration is required, as additional pruning at this scale would largely replicate Phase 1 outcomes without significant improvement. Phase 2 serves primarily to polish the architecture rather than reshape it, allowing the network to stabilize around the strongest connections and finalize the distribution of neurons across layers. Algorithm 3 summarizes the complete conversion process from an FC-DNN to a PC-DNN using StradNets. As shown in the algorithm, entering the *while*-loop (Step 10) initiates pruning, during which weak connections and isolated neurons are removed. The network is then retrained without reinitialization, and pruned connections are skipped during feedforward and backpropagation, strengthening the remaining ones. After retraining, accuracy is compared with the maximum observed so far. If accuracy improves, the maximum is updated. If accuracy decreases slightly but remains within tolerance, the process continues. The 1% tolerance margin ensures that natural fluctuations in training are distinguished from genuine performance loss, preventing the network from being pruned beyond its capacity. Once this condition is met in Phase 1 (step 20), the previous state is restored, and Phase 2 begins. A baseline accuracy a is recorded at the start of Phase 2 to measure its refinement. Since Phase 2 consists of only a single prune-train-test cycle, its stopping criterion is met immediately after that cycle. Likewise, if the condition is met in Phase 2 (step 17), the previous state is also restored. At this point, the optimal neural network configuration is finalized and returned to the user (step 19). This two-phase design ensures both global exploration (Phase 1) and local refinement (Phase 2), enabling the framework to produce compact architectures that preserve predictive accuracy while substantially reducing computational cost.

Algorithm 3: Automated Structural Adaptation

Input: FC-DNN *N*, Train dataset D_{Train}, test dataset D_{Test}
Output: A PC-DNN with optimized hidden layer sizes

```
1.  Scale $D_{Train}$and $D_{Test}$ using min-max normalization
2.  Initialize N with standard starting hidden layer sizes
3.  Initialize N with random weights
4.  for each weight tensor $\Theta$ in N
5.      Create an $i \times j$ mask matrix for $\Theta$, initialized to false
6.  Train and test N on $D_{Train}$ and $D_{Test}$, respectively
7.  Let $a$ be the current accuracy of N
8.  Initialize pruning ratio $p_{min}$ to a predefined value (e.g., 10%)
9.  Set phase = 1
10. while true
11.     Save the current network N to N_prev
12.     Prune weak connections in N according to ratio $p_{min}$
13.     Remove isolated neurons and their remaining connections in N
14.     Update N and the mask matrix for each weight tensor in N
15.     Train N with saved weights and evaluate accuracy $a'$ on $D_{Test}$
16.     if phase = 2 // in Phase 2
17.         if $a' < a - 0.01$ // $\geq$ 1% decrease from the best accuracy
18.             Reload previous network N_prev to N
19.         return N
20.     else if $a' < a - 0.01$  // in Phase 1
21.         Reload previous network N_prev to N
22.         Set phase = 2 and $p_{min} = p_{min}$*0.5
23.     else if $a' > a$, set $a = a'$  // set the best accuracy
```

In the automated structural adaptation process, the pruning and structural adaptation steps (step 12 and 13) in each iteration have a complexity of $O(N+E)$, where N and E denote the number of neurons and connections, respectively. The $O(E)$ component and the $O(N)$ component correspond to the time complexities of Algorithm 1 and Algorithm 2, respectively. Specifically, Algorithm 1 iterates through each connection in the network, while Algorithm 2 iterates through each node to evaluate and adjust neuron connectivity. Consequently, the overall complexity of the automated structural adaptation introduces only a small additional computational overhead due to structural adjustments, with the training step (Step 15) remaining the major time-consuming component of the process.

5. Case Studies

To evaluate the StradNet architecture, we conducted case studies on real-world machine learning applications. These studies demonstrate that StradNet's autotuning strategies enhance performance while reducing manual intervention. StradNets effectively adapt to complex patterns in noisy datasets, making them suitable for dynamic fields such as autonomous systems, healthcare, oceanography, and finance. In this paper, we focus on marine ecosystem datasets, which are inherently complex and require large neural networks for accurate prediction. Our primary goal is to evaluate the efficiency and scalability of StradNet relative to the conventional pruning approach, a well-recognized baseline. While comparisons with other adaptive pruning frameworks (e.g., dynamic sparse training [38], NAS-based methods [14], or other modern structured pruning techniques) are valuable, they are beyond the scope of this study due to differences in experimental settings, computational requirements, and model assumptions. In all subsequent experiments, each dataset was divided into 70% training data and 30% testing data. This split was chosen to ensure that each dataset contained sufficient samples to support a stable training process and accurate predictions.

5.1 Automated Two-Phase Pruning Process

In this case study, we apply the StradNet model to predict distant adversarial vessels using 15 input features derived from real-world maritime surveillance data [39]. The adversarial vessels dataset was originally generated to predict the fishing activity of a vessel using its position and behavioral features. In an effort to align the dataset with the objectives of marine research, the labels were repurposed to classify vessels as hostile or non-hostile. The structure of the dataset and the underlying datapoints remained unchanged; only the target labels were adapted to reflect relevant maritime scenarios. The dataset is large, comprising approximately 130,000 records, and was further enhanced using the Synthetic Minority Oversampling Technique (SMOTE) to address class imbalance and better preserve representative category distributions [40]. By applying SMOTE, we generated synthetic samples for the minority class while maintaining the overall feature distribution and diversity of the data. The input features include calendar variables capturing temporal and seasonal patterns; geographical indicators representing vessel detection locations; distance-based metrics estimating proximity to shores and ports; physical attributes such as speed, course, direction, and size; and environmental factors including weather and visibility conditions. Together, these features provide rich contextual information essential for accurately identifying potential adversarial vessels at a distance.

To construct the initial FC-DNN model, we first define its key hyperparameters, including the number of hidden neurons, activation functions, and training parameters. While our approach supports an arbitrary number of hidden layers, we restrict the network architecture to two hidden layers for simplicity and reproducibility. The remaining hyperparameters are set as follows: the Rectified Linear Unit (ReLU) activation function in all hidden layers, a learning rate of 0.01, 12 training epochs, and a batch size of 16. The number of hidden neurons is dataset-dependent; in this experiment, we use a standard configuration with 512 and 256 neurons in the first and second hidden layers, respectively. This configuration provides a balanced trade-off between model capacity and computational efficiency, serving as a strong baseline for evaluating the effects of iterative pruning. The setup aims to demonstrate that progressive pruning of the fully connected network can maintain strong generalization performance throughout the pruning process. Equation (3) defines the computation used to determine the initial total number of weights in the FC-DNN with n layers.

$$totalWeights = \sum_{i=1}^{n-1}(nNeu\_Layer_i * nNeu\_Layer_{i+1}) \quad (3)$$

where $nNeu\_Layer_i$ refers to the number of neurons in layer i, where $1 \leq i \leq n$.

We begin with a minimal pruning ratio p_{min} of 10%, removing 10% of the total weights at each pruning step to gradually simplify the architecture. If model performance drops beyond a predefined accuracy threshold, the model is reverted to a previously saved architecture, and pruning continues at a reduced rate to preserve stability. In this study, a 1% decrease from the best recorded accuracy in Phase 1 is used as the cutoff threshold, at which point the minimal pruning ratio p_{min} is reduced from 10% to 5% in Phase 2 to enable finer-grained structural refinement. Table 1 provides an illustrative example of the automated pruning process used to identify an effective StradNet architecture by progressively removing weak connections and isolated neurons while maintaining overall model generalization.

Table 1: An illustrative example of the automated pruning process.

Total Neurons	***HL1***	***HL2***	***Total Weights***	***Pruning Ratio (%)***	***Threshold***	***Accuracy (%)***
***Phase* 1 (*Coarse Pruning Step*)**						
768	512	256	139,264	N/A	N/A	96.30
765	512	253	125,337	10	0.007	97.10
748	512	236	112,803	10	0.015	97.31
742	512	212	101,523	10	0.022	97.27
704	512	192	91,371	10	0.029	97.65
686	512	174	82,234	10	0.036	97.74
613	454	159	74,010	10	0.045	97.66
569	440	129	66,609	10	0.070	97.85
555	433	122	59,948	10	0.127	97.65
543	422	121	53,953	10	0.206	***98.12***
526	409	117	48,558	10	0.302	97.96
512	401	111	43,702	10	0.413	98.04
495	396	99	39,332	10	0.526	97.97
484	390	94	35,399	10	0.651	98.12
475	384	91	31,859	10	0.785	97.97
463	382	81	28,673	10	0.924	97.93
448	374	74	25,805	10	1.081	97.38
423	363	60	23,225	10	1.217	***96.88***
***Phase* 2 (*Fine-Tuning Step*)**						
448	374	74	25,805	10	1.081	97.38
437	**372**	**65**	**24,514**	**5**	**1.133**	**97.36**

From Table 1, we observe that the pruning process completes its first phase when accuracy drops to 96.88%, 1.24% below the best recorded accuracy of 98.12%. The network is then reverted to its previous architecture and enters the fine-tuning step with a minimal pruning ratio of 5%. The final architecture obtained through this process contains 437 neurons: 372 in the first hidden layer and 65 in the second. This configuration achieves an accuracy of 97.36%, 1.06% higher than the initialized FC-DNN model (96.30%). These results indicate that the StradNet pruning process generalizes well to new data, achieving accuracy comparable to the much larger initial model, and demonstrate that many weights and neurons are unnecessary. In particular, these findings highlight how pruning effectively removes redundant structure while preserving the essential computational pathways needed for accurate prediction. While only the strongest connections contribute significantly to accurate predictions, excessive pruning can be detrimental; removing too many important connections reduces accuracy and impairs prediction quality. This underscores the importance of selecting appropriate pruning ratios to balance compactness and performance. Moreover, the smooth recovery of accuracy after reverting to the best-performing architecture reinforces the value of maintaining historical model states during the pruning process. This mechanism prevents the framework from drifting into overly aggressive pruning regimes and ensures that structural exploration remains both controlled and reversible. Figure 7 illustrates this effect with a constant minimal pruning ratio of 10%, showing a steady decline in accuracy as the network becomes too small to perform effectively.

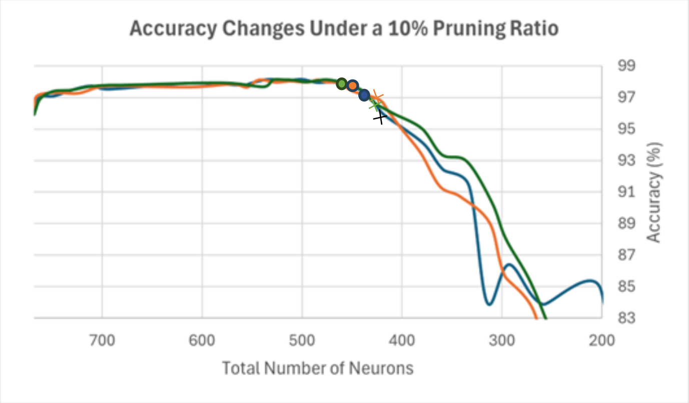


Figure 7: Accuracy vs. total number of neurons under a 10% pruning ratio

In Figure 7, we present three trials of the StradNet workflow, in which weak connections and neurons are progressively pruned from the network. Each trial begins with 768 neurons, consistently arranged as 512 in the first hidden layer and 256 in the second. Accuracy initially improves by approximately 2%, suggesting a reduction in overfitting as the network size decreases. The early pruning stages indicate that the network can still achieve accurate predictions on the test dataset. However, once the network size falls below roughly 450 neurons, accuracy declines sharply. Continuing to prune at a fixed 10% rate beyond this point results in a steady performance decline, which we attribute to the aggressiveness of the pruning strategy and the need for more careful dynamic adjustment. This trend confirms that the crosses in the figure mark the appropriate points for activating the stopping criterion. At these points, the network transitions to fallback architectures, represented by the dots, allowing pruning to continue under a more relaxed strategy.

The fine-tuning step begins by training and testing the reverted network using the stored weight parameters. In this phase, we apply a single prune-train-test cycle with a 5% minimal pruning ratio, after which network modifications are finalized. Figure 8 illustrates the reinstated neural network from Phase 1. At the start of Phase 2, further pruning remains possible at a smaller percentage; accordingly, this diagram reflects pruning 5% of the weakest connections. The resulting performance curve closely resembles that of the 10% pruning diagram, confirming that the observed accuracy decline is consistent across trials. The dots in Figure 8 correspond to the same total-neuron counts shown in Figure 7, while the crosses indicate the configurations returned to the user upon algorithm completion, representing the effective network architectures. As shown in the figure, all three trials demonstrate that accuracy continues to decline when pruning at 5% is repeated. Additional pruning at this rate is therefore unnecessary, as Phase 1 has already revealed the adverse effects of more aggressive 10% pruning. The convergence of results across multiple trials also indicates that the selected 5% pruning ratio provides a stable termination condition, ensuring that the final PC-DNN architecture reflects both structural efficiency and reliable predictive performance.

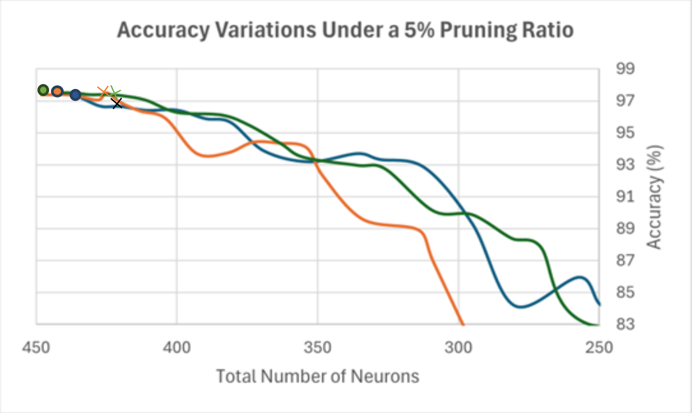


Figure 8: Accuracy vs. total number of neurons under a 5% pruning ratio

Figure 9 illustrates the relationship between the total number of weights in the neural network and its accuracy under a 10% minimal pruning ratio. Across three trials, we observe a positive trend in the early stages of pruning: as the number of weights decreases, the network achieves higher accuracy, suggesting that larger networks may be prone to overfitting. This trend continues until the networks become sufficiently small, around 20,000 total weights, at which point accuracy drops sharply from approximately 98% to 83%. These results are consistent with the accuracy decline observed in Figure 7 when excessive neurons and their associated weak connections are removed. Similarly, the crosses in Figure 9 mark the points at which the stopping criterion is activated, leading into the fine-tuning step. This pattern highlights the balance between removing redundant weights and keeping enough capacity for accurate predictions. The consistent results across trials also show that the stopping rule effectively prevents the network from being pruned too aggressively.

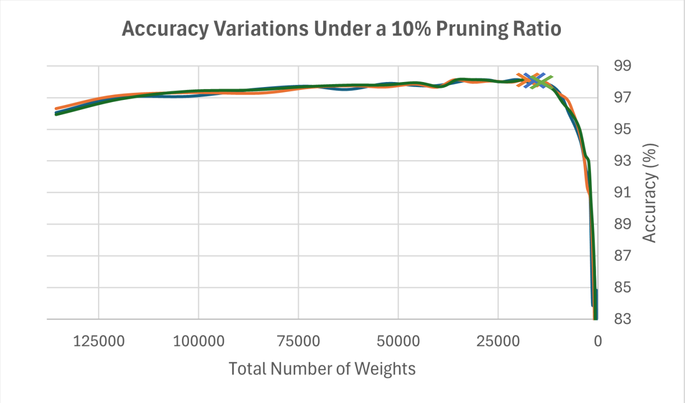


Figure 9: Accuracy vs. total number of weights with 10% pruning ratio

5.2 Neuron Allocation Between Hidden Layers

In conventional DNNs, the hidden layers often follow a pyramid configuration, in which each successive layer contains fewer neurons than the previous one [41]. For instance, in a network with two hidden layers, the size of the first layer is typically determined by factors such as dataset size, number of input features, number of output neurons, and task complexity. The second hidden layer often contains about half as many neurons as the first, reflecting its role in refining the learned representation into a form suitable for the output layer. Using this conventional architecture as a baseline, the present study investigates whether a specific relationship between the two hidden layers of a PC-DNN can achieve predictive performance comparable to the standard design. To evaluate this, the StradNet model was tested on three datasets, including *Adversarial Vessels*, *Oil Spill*, and *Sonar Classification*. The *Adversarial Vessels* dataset is described in Section 5.1. The *Oil Spill* dataset, derived from satellite imagery and reformatted into tabular form, is used to distinguish between spill and non-spill patches [42]. This dataset serves as our medium-scale application case and features a highly modified structure. It contains 60 attributes and 1,200 records and was augmented using the SMOTE technique to address class imbalance by generating additional samples for the underrepresented class. Controlled noise was also introduced as a form of data augmentation to encourage the network to learn meaningful patterns rather than memorize specific instances. Two augmentation strategies were considered: feature perturbation and output flipping [43], [44]. Feature perturbation involves applying small random variations to input feature values, which enhances the model's robustness and reflects natural variations observed in real-world conditions. Output flipping, on the other hand, simulates label noise by occasionally swapping class labels to mimic annotation errors. Together, the use of SMOTE and controlled noise increased the dataset's variability and complexity, thereby enhancing the network's capacity for robust pattern recognition.

The *Sonar Classification* dataset, which contains acoustic signals recorded at varying frequencies and angles, can be used to evaluate object detection performance in applications such as mine, vessel, and underwater structure identification [45]. Although the dataset documentation does not describe the specific data collection methodology, its close association with sonar technology, a common sensing modality used by marine vessels to detect underwater mines, debris, and other hazards, makes it relevant for our study. This small-scale dataset allows us to evaluate how well the network maintains generalization when trained with limited data and to demonstrate the robustness of the StradNet approach across different dataset sizes. It contains 60 input features and 200 samples. Using a small dataset also highlights the pruning scheme's behavior under a high risk of overfitting. If the pruned network trained on this dataset exhibits a substantial accuracy drop, it would indicate that the model depended heavily on specific features and was overfitted to the limited data. All input features were normalized using the min-max normalization scheme, which scales values into the [0, 1] range to ensure equal contribution of all variables. Each of the 60 features captures distinct signal characteristics, where different angles of incidence yield unique representations that enrich the overall input space.

In this case study, we examine the relationship between the two hidden layers, HL1 and HL2, in the StradNet models. Specifically, we investigate whether the pruning process affects the neuron ratio between these layers, as defined in (4).

$$neuronRatio = \frac{nNeu\_\mathrm{HL2}}{nNeu\_\mathrm{HL1}} \times 100\% \qquad (4)$$

where *nNeu*_HL1 and *nNeu*_HL2 are the numbers of neurons in hidden layers HL1 and HL2, respectively.

In the initial configuration, the neuron ratio between HL1 to HL2 is set to 50%. This setup allows us to determine whether a consistent pattern emerges across the resulting StradNet models. Table 2 presents both the initial and resulting configurations for the three datasets. The experimental results represent the averages of four independent runs, and the last column of the table reports the standard deviation of the accuracy values. Our findings indicate that a neuron ratio of 50% is not always optimal, particularly for large hidden layers, likely due to the increased complexity of the model and the associated tasks. Nevertheless, our approach effectively identifies dataset-specific optimal ratios of hidden neurons between HL1 and HL2.

Table 2: Average Changes in the ratio of neurons between HL1 and HL2

Neuron Ratio	***Total Neurons***	***HL1***	***HL2***	***Total Weights***	***Accuracy***	***Standard Deviation***
Adversarial Vessels						
50	768	512	256	139264	96.30	0.379
17.8	428	363	65	23537	97.41	0.252
Oil Spill						
50	768	512	256	156671	92.65	0.821
41.4	724	512	212	67609	91.03	0.543
Sonar Classification						
50	768	512	256	162304	90.11	0.904
75.3	384	219	165	16365	91.73	0.353

As shown in Table 2, for the *Adversarial Vessels* dataset, the baseline configuration used 512 neurons in HL1 and 256 in HL2. After applying the StradNet approach, the suitable HL2 size was found to be 17.8% of HL1, a much lower ratio than the standard 50%, determined dynamically through iterative structural adaptation. On average, this pruning removed about 340 neurons and approximately 83.1% of the weights, yet achieved slightly higher average accuracy than the baseline, suggesting that smaller, more targeted architectures may outperform larger ones when redundant nodes are removed. For the *Oil Spill* dataset, the suitable HL2 size was about 41.4% of HL1 compared to the baseline's 50%. Only 44 neurons, all from HL2, were pruned, and both configurations achieved similar accuracy, indicating that in some cases the 50% rule still yields desirable performance. For the *Sonar Classification* dataset, the model identified a increased ratio of 75.3%, removing nearly 50% of neurons and around 89.9% of the weights, which increased accuracy by more than 1.6% over the baseline. This suggests that the original network was over-parameterized, and that near-symmetrical layer sizes may benefit certain datasets. The standard deviation values of accuracy, also reported in Table 2, are consistently low, confirming that the results across the four independent runs are stable and reliable. Overall, these findings demonstrate that suitable neuron allocation between HL1 and HL2 is heavily dataset-dependent. The adaptability of the StradNet approach enables it to determine dataset-specific pruning schemes, reducing over-parameterization while improving efficiency and predictive accuracy. These results also highlight that fixed architectural heuristics, such as the common 50% HL1-to-HL2 ratio, are not universally optimal. Instead, effective layer sizing often emerges from data-driven adjustments that reflect the complexity and structure of the underlying task. By automatically discovering these patterns, StradNet reduces the need for manual trial-and-error design and produces compact architectures that match the requirements of each dataset.

5.3 A Comparative Study on Time Efficiency and Scalability

In most software systems, particularly real-time systems that handle large volumes of sensor data, rapid retraining and fast response times are critical for adapting to changing environments and preventing failures. In some scenarios, delays can be life-threatening. For example, in an industrial plant that relies on alarm algorithms to detect harmful substances, even a short delay of a few minutes could result in equipment damage or, in the worst case, casualties. Similarly, autonomous vehicles, medical monitoring devices, and maritime surveillance systems all depend on timely updates to remain effective in dynamic conditions. Faster retraining and more responsive software can therefore save both resources and human lives in situations where slower systems could lead to costly failures. The above examples highlight the practical importance of efficient model adaptation in real-world operational settings.

This section presents a case study comparing the time efficiency and scalability of StradNet with a conventional DNN pruning approach, in which 10% of the neurons are removed at each step until accuracy decreases by a specified amount (i.e., 1%). For a given dataset *DS*, we record the total time required to construct a suitable DNN architecture with two hidden layers using both approaches. As defined in (5), the total time, T_{DS_total}, is the sum of the times for each pruning round, consisting of three components: the training or retraining time T_{train_i}, the pruning time $T_{pruning\_i}$, and the testing time $T_{testing\_i}$, where $1 \leq i \leq k$ and k is the number of pruning rounds. The primary purpose of model testing is to determine when performance begins to degrade and when the pruning process should stop.

$$T_{DS\_total} = \sum_{i=1}^{k} (T_{train\_i} + T_{pruning\_i} + T_{testing\_i}) \qquad (5)$$

In the conventional approach, the structured-pruned DNNs are FC-DNNs, and at each step, the pruned networks are retrained from scratch. Each sample is trained and evaluated using the same model configuration, with hidden layer sizes of 512 and 256. The purpose of this setup is to examine the effects of dataset size reduction on each pruning algorithm. Modifying the neural network structure as the dataset shrinks would prevent a direct comparison between samples using the same algorithm. This study evaluates the time efficiency of StradNet models relative to the conventional pruning approach and demonstrates that StradNet exhibits greater scalability with increasing dataset size. As mentioned previously, our method removes weak connections and isolated neurons via magnitude-based unstructured global pruning, a directed methodology that eliminates neurons and parameters contributing least to overall performance. For this experiment, we selected the Adversarial Vessels dataset, which contains a large number of complex data points, to assess the efficiency of our automated approach.

Figure 10 compares time efficiency and scalability between the StradNet approach (blue line) and the conventional structured pruning approach (orange line) as the number of data points varies. From the figure, we observe a small gap between the two approaches when the number of data points is below approximately 50,000. As the dataset grows beyond 50,000 points, the time difference increases rapidly, reaching about 35 minutes for the conventional approach and 12.5 minutes for StradNet when the number of new data points is around 120,600. These results suggest that StradNet models are particularly well-suited for large-scale datasets, offering significant time savings and improved scalability compared to conventional pruning methods. This outcome is expected because the StradNet approach requires less time to identify a suitable network structure, preserving weight values across pruning iterations and dynamically shortening training epochs as the network achieves accurate predictions. In contrast, conventional networks reinitialize weights after each pruning step, making it harder to capture dataset patterns and effectively forcing the network to “start over” during each iteration. Additionally, as demonstrated in previous experiments, StradNet produces a more concise and efficient partially connected network structure, further contributing to faster response times and better scalability for growing datasets.

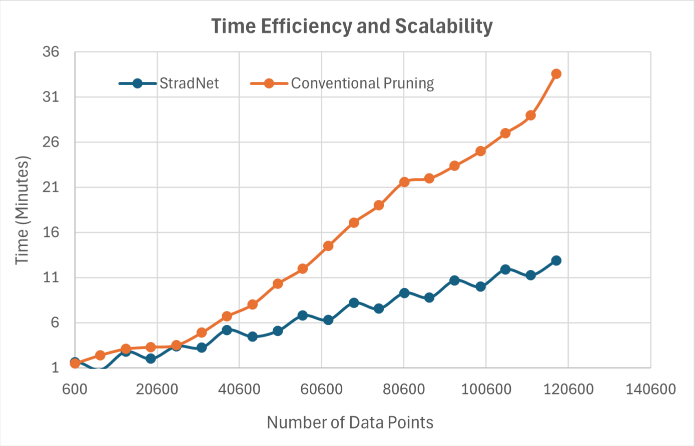


Figure 10: Time efficiency and scalability under varying dataset sizes

These improvements become increasingly important in real-time or continuously updated systems, where models must be retrained frequently to reflect new information. In such environments, even small reductions in retraining time accumulate into substantial long-term savings. Furthermore, the stability of the StradNet pruning strategy ensures that model performance does not degrade as data volume increases, enabling it to maintain both accuracy and responsiveness under demanding operational conditions.

6. Conclusions and Future Work

As DNNs continue to advance and achieve better performance on benchmark tests and practical applications, they also exhibit increasing complexity, storage requirements, and energy consumption. In this paper, we presented StradNet, an efficient and scalable framework for converting an FC-DNN into a reliable PC-DNN capable of generalizing and performing well on unseen data. This approach effectively reduces the resource demands of large DNNs while significantly lowering associated deployment costs. Experiments on dynamic marine datasets demonstrate that a multi-step pruning iteration combined with iterative fine-tuning produces PC-DNNs that outperform their FC-DNN counterparts in prediction accuracy, storage efficiency, and computation time. By leveraging this adaptive pruning strategy, StradNet can be extended to larger and deeper architectures with billions of parameters, such as LLMs [46], enabling them to potentially operate on resource-limited edge devices. Our experimental results also indicate that there is no universal pattern for initializing efficient DNNs; performance depends strongly on the specific application domain and dataset characteristics.

To further illustrate StradNet's efficiency and scalability, we compared it with a conventional pruning method and observed that it consistently identified effective network structures substantially faster across datasets of increasing size and complexity. Together, these findings provide compelling evidence that adaptive structural pruning is not merely an optimization technique but a robust design principle capable of guiding the development of streamlined neural architectures. These enhancements highlight the potential of StradNet as a practical framework for reducing model complexity while preserving predictive fidelity, supporting efficient retraining, and improving scalability in data-intensive or resource-constrained environments.

Future work in this area could focus on identifying effective values for additional hyperparameters or model parameters. In this paper, we primarily focus on compressing the number of hidden neurons while keeping other hyperparameters fixed to evaluate the benefits of hidden neuron reduction. Advanced hyperparameter tuning could involve finding optimal structures via dynamic learning rate adjustments or reducing the number of training epochs as accuracy improves. At a later stage, combining multiple automated structure-search algorithms could work interchangeably to generate the most effective network hyperparameter set. Other promising directions for structurally adaptive neural networks that could enhance StradNet include dynamic expansion, intelligent pruning schemes, and adaptive loop scheduling [47]-[49]. Dynamic expansion involves adding neurons to enable bidirectional adaptation; although StradNet already implements this in the fine-tuning step, multiple iterations may offer additional benefits. Currently, StradNet uses a magnitude-based pruning scheme, but a more sophisticated criterion could preserve important connections more reliably [37]. In addition, adaptive loop scheduling could accelerate convergence by varying the pruning ratio, enabling the network to prune more aggressively in certain iterations, achieve an effective structure more quickly, and gain greater flexibility. Finally, we plan to extend the StradNet framework to more complex DNNs, including CNN architectures [50] and transformer-based models [51], and to evaluate it on non-marine datasets from dynamic environments to further assess robustness and demonstrate broader generalization across domains.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

We thank the editors and the anonymous referees for their careful review of this paper and the many valuable suggestions they provided. This material is based on work supported by the Office of Naval Research (ONR) under the MUST IV Grant at the University of Massachusetts Dartmouth.

References

[1] S. Natarajan, S. A. R. Al-Haddad, F. A. Ahmad, R. Kamil, M. K. Hassan, S. Azrad, J. F. Macleans, S. H. Abdulhussain, B. M. Mahmmod, N. Saparkhojayev, et al., "Deep neural networks for speech enhancement and speech recognition: A systematic review," Ain Shams Engineering Journal, **16**(7), 103405, 2025, doi:10.1016/j.asej.2025.103405.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, **60**(6), 84-90, 2017, doi:10.1145/3065386.

[3] S. Bhat, H. Xu, and J. Carberry, "Automated medical coding using a hybrid decision tree with deep learning nodes," in Proceedings of the 11th IEEE International Conference on Big Data Computing Service and Machine Learning Applications (IEEE BigDataService 2025), 81-88, Tucson, Arizona, USA, 2025, doi:10.1109/BigDataService65758.2025.00017.

[4] M. Stinchcombe and H. White, "Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions," International 1989 Joint Conference on Neural Networks, **1**, 613-617, Washington, DC, USA, 1989, doi:10.1109/IJCNN.1989.118640.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, **60** (6), 84-90, May 2017, doi:10.1145/3065386.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), **2**, 3104-3112, December 2014, doi:10.5555/2969033.2969173.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., "Language models are few-shot learners," in Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS), Article No.: 159, 1877-1901, December 2020, doi:10.5555/3495724.3495883.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), 6000-6010, December 2017, doi:10.5555/3295222.3295349.

[9] D. Chenna, "Why the latest AI model isn't always best for edge AI," IEEE Spectrum, July 20, 2025. Retrieved on September 15, 2025 from: https://spectrum.ieee.org/edge-ai

[10] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, "Model complexity of deep learning: a survey," Knowledge and Information Systems, **63**, 2585-2619, 2021, doi:10.1007/s10115-021-01605-0.

[11] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," Journal of Machine Learning Research, **13**, 281-305, Feb. 2012, doi:10.5555/2188385.2188395.

[12] C. Guo, Y. Qiu, J. Leng, X. Gao, C. Zhang, Y. Liu, F. Yang, Y. Zhu, and M. Guo, "SQuant: on-the-fly data-free quantization via diagonal hessian approximation," in Proceedings of the 10th International Conference on Learning Representations (ICLR 2022), 2269-2286, 2023.

[13] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural network," in Proceedings of the 29th International Conference on Neural Information Processing Systems (NIPS' 15), **1**, 1135–1143, 2015, doi:10.5555/2969239.2969366.

[14] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), 2017, doi:10.48550/arXiv.1611.01578.

[15] W. Roth, G. Schindler, B. Klein, R. Peharz, S. Tschiatschek, H. Fröning, F. Pernkopf, and Z. Ghahramani, "Resource-efficient neural networks for embedded systems: a survey," The Journal of Machine Learning Research, **25** (1), 2506-2556, 2024, doi:10.5555/3722577.3722627.

[16] M. A. K. Raiaan, S. Sakib, N. M. Fahad, A. Al Mamun, Md. A. Rahman, S. Shatabda, and Md. S. H. Mukta, "A systematic review of hyperparameter optimization techniques in convolutional neural networks," Decision Analytics Journal, **11**, 100470, 2024, doi:10.1016/j.dajour.2024.100470.

[17] D. H. Shin, D. H. Ko, J. W. Han and T. E. Kam, "Evolutionary reinforcement learning for automated hyperparameter optimization in EEG classification," in Proceedings of the 2022 10th International Winter Conference on Brain-Computer Interface (BCI), 1-5, Gangwon-do, Korea, Republic of, 2022, doi:10.1109/BCI53720.2022.9734935.

[18] K. Khadka, J. Chandrasekaran, Y. Lei, R. N. Kacker, and D. R. Kuhn, "A combinatorial approach to hyperparameter optimization," in Proceedings of

the 2024 IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI (CAIN), 140-149, Lisbon, Portugal, 2024.

[19] P. Neary, “Automatic hyperparameter tuning in deep convolutional neural networks using asynchronous reinforcement learning,” in Proceedings of the 2018 IEEE International Conference on Cognitive Computing (ICCC), 73-77, San Francisco, CA, USA, 2018, doi:10.1109/ICCC.2018.00017.

[20] J. Qiu, Y. Zhao, and W. Li, “Neural architecture search method based on improved Monte Carlo tree search,” in Proceedings of the 2023 China Automation Congress (CAC), 9033-9037, Chongqing, China, 2023, doi:10.1109/CAC59555.2023.10451718.

[21] T. Kim, Y. Na, and S. Park, “Multi-head convolutional neural network compression based on high-order principal component analysis,” in Proceedings of the 2023 International Conference on Electronics, Information, and Communication (ICEIC), 1-4, Singapore, 2023, doi:10.1109/ICEIC57457.2023.10049909.

[22] H. Yang, Y. Liang, X. Guo, L. Wu, and Z. Wang, “Random pruning over-parameterized neural networks can improve generalization: a training dynamics analysis,” Journal of Machine Learning Research, **26**(84), 1-51, April 2025.

[23] I. Salehin, Md. S. Islam, P. Saha, S.M. Noman, A. Tuni, Md. M. Hasan, and Md. Abu Baten, “AutoML: a systematic review on automated machine learning with neural architecture search,” Journal of Information and Intelligence, **2**(1), 52-81, 2024, doi:10.1016/j.jiixd.2023.10.002

[24] J. Hu, P. Lin, H. Zhang, Z. Lan, W. Chen, and K. Xie, “A dynamic pruning method on multiple sparse structures in deep neural networks,” IEEE Access, **11**, 38448-38457, 2023, doi:10.1109/ACCESS.2023.3267469.

[25] X. Huang, J. Zhou, X. Yan, W. Ye, and X. He, “A pruning method for echo state network based on neuron importance and iterative fine-tuning,” in Proceedings of the 2023 China Automation Congress (CAC), 1034-1039, Chongqing, China, 2023, doi:10.1109/CAC59555.2023.10450422.

[26] L. K. Kalyanam and S. Katkoori, “Unstructured pruning for multi-layer perceptrons with tanh activation,” in Proceedings of the 2023 IEEE International Symposium on Smart Electronic Systems (iSES), 69-74, Ahmedabad, India, 2023, doi:10.1109/iSES58672.2023.00025.

[27] W. Na, K. Liu, W. Zhang, F. Feng, H. Xie, and D. Jin, “Automated multilayer neural network structure adaptation method with L1 regularization for microwave modeling,” IEEE Microwave and Wireless Components Letters, **32**(7), 815-818, July 2022, doi:10.1109/LMWC. 2022.3153058.

[28] L. Yang and D. Fan, “Dynamic neural network to enable run-time trade-off between accuracy and latency,” in Proceeding of the 2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC), 587-592, Tokyo, Japan, January 2021.

[29] R. Ming, H. Xu, S. E. Gibbs, D. Yan, and M. Shao, “A deep neural network based approach to building budget-constrained models for big data analysis,” in Proceedings of the 17th International Conference on Data Science (ICDATA’21), 1-8, Las Vegas, Nevada, USA, July 26-29, 2021, doi:arxiv.org/pdf/2302.11707.

[30] P. Wu, “Research on improved BP algorithm by computer simulation based on adaptive learning rate,” in Proceedings of the 2023 3rd Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), 9-12, Shenyang, China, 2023, doi:10.1109/ACCTCS58815. 2023.00138.

[31] B. Noordijk, M. L. G. Gomez, K. H. Tusscher, D. de Ridder, A. D. J. van Dijk, and R. W. Smith, “The rise of scientific machine learning: a perspective on combining mechanistic modelling with machine learning for systems biology,” Frontiers in Systems Biology, **4**, August 2024, doi:10.3389/fsysb.2024.1407994.

[32] G. Minai, D. Bobeldyk, and J. P. Leidig, “Evaluating the impact of diverse marine environments on image classification performance,” in Proceedings of the OCEANS 2024 - Halifax, 1-4, Halifax, NS, Canada, September 2024, doi: 10.1109/OCEANS55160.2024.10753966.

[33] A. A. Patil and K. Kulkarni, “A hybrid machine learning - numerical weather prediction approach for rainfall prediction,” in Proceedings of the 2023 IEEE India Geoscience and Remote Sensing Symposium (InGARSS), 1-4, Bangalore, India, 2023, doi:10.1109/InGARSS59135.2023.10490397.

[34] W. Lin, T. Li, and X. Li, “Deep learning-based object detection for environmental monitoring using big data,” Frontiers in Environmental Science, **13**, June 2025, doi:10.3389/fenvs.2025.1566224.

[35] H. Xu and A. Gade, “Smart real estate assessments using structured deep neural networks,” in Proceedings of the 2017 IEEE International Conference on Smart City Innovations (IEEE SCI 2017), 1126-1132, San Francisco, CA, USA, August 4-8, 2017, doi:10.1109/UIC-ATC.2017.8397560.

[36] G. J. Sawale and M. K. Rawat, “Stock market prediction using sentiment analysis and machine learning approach,” in Proceedings of the 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 1-6, Tirunelveli, India, 2022, doi:10.1109/ICSSIT53264. 2022.9716326.

[37] J. Frankle and M. Carbin, “The lottery ticket hypothesis: finding sparse, trainable neural networks,” in Proceeding of the Seventh International Conference on Learning Representations (ICLR 2019), New Orleans, USA, May 2019, doi:10.48550/arXiv.1803.03635.

[38] S. Liu, I. Ni'mah, V. Menkovski, D. Mocanu and M. Pechenizkiy, “Efficient and effective training of sparse recurrent neural networks,” Neural Computing and Applications, **33**, 9625-9636, 2021, doi:10.1007/s00521-021-05727-y.

[39] GFW, “Datasets and code,” Global Fishing Watch (GFW), 2024. Retrieved on December 1, 2024 from https://globalfishingwatch.org/data-download/datasets/public-training-data-v1

[40] D. Elreedy, A. Atiya, and Firuz Kamalov, “A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning,” Machine Learning, **113**, 4903-4923, 2024, doi:10.1007/s10994-022-06296-4.

[41] T. Masters, Practical Neural Network Recipes in C++. San Diego, CA, USA: Academic Press, 1993.

[42] A. Khan, “Oil spill - imbalanced classification,” Kaggle, 2024. Retrieved on December 1, 2024 from https://www.kaggle.com/code/ashrafkhan94/oil-spill-imbalanced-classification

[43] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Mądry, “Adversarial examples are not bugs, they are features,” in Proceedings of the 33rd International Conference on Neural Information Processing Systems, 125-136, Red Hook, NY, USA, 2019, doi:10.5555/3454287.3454299.

[44] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, “Making deep neural networks robust to label noise: a loss correction approach,” in Proceeding of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2233-2241, Honolulu, HI, USA, July 21-26, 2017, doi:10.1109/CVPR.2017.240.

[45] UC Irvine, “Connectionist bench (sonar, mines vs. rocks),” UCI Machine Learning Repository, Retrieved on December 1, 2024 from https://archive.ics.uci.edu/dataset/151/connectionist+bench+sonar+mines+vs+rocks

[46] M. Arya and Y. Simmhan, “Understanding the performance and power of LLM inferencing on edge accelerators,” in Proceedings of the 2025 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 1108-1111, Milano, Italy, 2025, doi:10.1109/IPDPSW66978. 2025.00173.

[47] J. Guo, C. L. P. Chen, Z. Liu and X. Yang, “Dynamic neural network structure: a review for its theories and applications,” IEEE Transactions on Neural Networks and Learning Systems, **36**(3), 4246-4266, March 2025, doi:10.1109/TNNLS.2024.3377194.

[48] A. Heyman and J. Zylberberg, “Fine granularity is critical for intelligent neural network pruning,” Neural Computation, **36**(12), 2677-2709, November 2024, doi:10.1162/neco_a_01717.

[49] F. Kasielke, R. Tschüter, C. Iwainsky, M. Velten, F. M. Ciorba and I. Banicescu, “Exploring loop scheduling enhancements in OpenMP: an LLVM case study,” in Proceedings of the 2019 18th International Symposium on Parallel and Distributed Computing (ISPDC), 131-138, Amsterdam, Netherlands, 2019, doi:10.1109/ISPDC.2019.00026.

[50] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” Artificial Intelligence Review, **53**(8), 5455-5516, December 2020, doi:10.1007/s10462-020-09825-6.

[51] P. Xu, X. Zhu, and D. A. Clifton, “Multimodal learning with transformers: a survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, **45**(10), 12113-12132, October 2023, doi:10.1109/TPAMI.2023.3275156.

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 10, No. 6, 42-54 (2025)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTESJ
ISSN: 2415-6698

Identifying Comprehension Faults Through Word Embedding and Multimodal Analysis

Kento Yasuda*, Hiromitsu Shimakawa, Fumiko Harada

Ritsumeikan University, Ibaraki, 567-8570, Japan

ARTICLE INFO

Article history:
Received: 14 October, 2025
Revised: 27 November, 2025
Accepted: 29 November, 2025
Online: 05 December, 2025

Keywords:
Data science
Writing questions
Open-ended questions
Text analysis
Electrodermal Activity (EDA)
Comprehension
Individual tutoring sessions
Multimodal

ABSTRACT

This study establishes a method for determining whether learners have an understanding of data science. Data science requires knowledge in various fields, which makes many learners give up. To prevent learners from being discouraged, it is necessary to judge the comprehension of the principles in each specified skill. It is important to assess not only learner's knowledge but also the extent to which they understand the underlying principles of data analysis methods. Open-ended questions effectively assess comprehensive understanding because they require learners to construct and articulate their understanding in their own words. This study analyzes teacher–learner interaction and electrodermal activity to examine the educational significance of open-ended responses. A random forest model identifies key behaviors, which indicate that articulating thoughts in one's own words is crucial for improving learner's understanding. For each specified skill, the proposed method provides us with a way to examine whether learner's answers to descriptive questions are close to the model answer. Learner's level of understanding is determined from the document vectors of their responses using the Word2vec method. In addition, important words for each skill are extracted using a Naive Bayes model. Furthermore, the method has identified words representing ununderstood concepts and unassociated procedures with a logistic regression model. Experimental results indicate that the proposed method effectively identifies learner's comprehension levels and extracts key linguistic features for distinguishing those with insufficient understanding. Using responses from 16 learners, the method achieves an F1-score of 0.824, although the small sample size limits generalizability. The word embeddings of learners with and without understanding show markedly different distributions. It suggests that we can identify the concepts and procedures that learners do not understand from their words. It enables us to offer suggestions to assist learners who are likely to be stuck.

1. Introduction

In recent years, the demand for data scientists has been increasing not only in IT companies but also in a wide range of industries, including manufacturing, logistics, healthcare, and so on. Data scientists need knowledge in a wide range of fields and problem-solving skills to apply them to real problems. However, what they need is not only the ability to understand field-specific terminologies and concepts. They must also understand the mathematical basis of the analytical methods they use. As noted in [1], data scientists must have the ability to select the most appropriate analytical method from various options and apply it to real data. To bring up learners to skillful data scientists, it is necessary to train mathematical ability as well as knowledge specific to each problem domain.

The utilization of data science technologies requires diverse mathematical knowledge and skills in probability, statistics, optimization, and programming. Those who try to master the knowledge often get stuck along the way. Many people fail to become data scientists because of repeated dead ends. They would repeat failures because there is no way of knowing what they understand and what they do not understand.

To uncover areas of incomprehension, it would be appropriate to analyze their free-text response to descriptive questions that test the mathematical knowledge regarding data analysis. However, it is difficult to automatically judge their understanding from the text they freely write. It is also important to identify which concepts learners with poor understanding do not understand. It needs manual tasks, which require huge costs. As shown in [2], [3], and [4], most previous studies on assessing comprehension have used multiple-choice questions. Very few studies assess comprehension from open-ended questions, as shown in [5] and

*Corresponding Author Kento Yasuda, Ritsumeikan University, is0584ih@ed.ritsumei.ac.jp

https://dx.doi.org/10.25046/aj100604

[6]. The experimental results have revealed that changes in electrodermal activity are observed when teachers encourage learners to express the correct answers in their own words. The experiments have also pointed out that learners who received such instruction exhibit higher levels of understanding. It is effective to assess learner's understanding using open-ended questions that allow them to express their comprehension in their own words.

This study assumes that learner's comprehension of data analysis methods can be estimated from the lexical features that appear in their free-text responses. Many of those who get stuck are unable to organize their knowledge, explain it in order, and express it in their own words. On the contrary, many people who understand without getting stuck can organize their knowledge to explain it in an orderly manner. Those learners answer questions with statements close to the sentences in the model answers. The study focuses on the difference.

Learners would get stuck because they have a misunderstanding of certain concepts. There is a high possibility that the words used in the correct descriptions explaining these concepts are not included in the learner's descriptions. The study examines words that are not in the sentences written by misunderstanding learners but are in the correct answer. It enables us to figure out what they do not understand. Based on the observation, the study analyzes how close learner answers are to the model answers for open-ended questions. The proposed method creates a distributed representation of words using Word2Vec. As described in [7], this approach uses a corpus of explanatory sentences for the relevant concepts presented in textbooks. The similarity between the learner's response and the model answer is measured using the cosine similarity between their document vectors, which are computed based on the words appearing in the response.

Experimental results show that the study can discriminate what is understood with the cosine similarity between the document vectors of the response sentences and those of the model answers. Recent advances in educational NLP have increasingly applied transformer-based models. In [8], the authors demonstrate that Sentence-BERT is effective for evaluating the semantic validity of short-answer responses. Building on these developments, the present study compares multiple representation methods, including Doc2Vec, Sentence-BERT, and TF-IDF combined with SVM, to examine their effectiveness in estimating learner's comprehension from free-text responses. The results indicate that the approach using Doc2Vec-based document vectors and cosine similarity achieves the highest F1-score, demonstrating its usefulness for comprehension estimation based on the lexical features of open-ended responses.

Furthermore, the distributed representations place words used by understanding learners in less overlap with those used by misunderstanding learners. The distribution of the two kinds of words turns out to be extremely far apart. Learners who cannot write the words appearing in the model answers are likely to get stuck soon. Early care of such learners will prevent them from running away from data science training. In [9], the authors focus exclusively on learners. The present paper extends this analysis by adding further analyses and incorporating teachers into the discussion. It also provides new insights by employing open-ended questions.

The structure of this paper is shown below. Section 2 describes the vectorization of documents by embedding. A measure of the closeness to the correct answers is proposed in Section 3. Section 4 presents an experiment to discuss its results. Section 5 concludes the study along with statements of future work.

2. Vectorization of Documents by Embedding

2.1. Dimensional Comprehension and Dimensional Selection

Information expressed in higher dimensions is complex to process and difficult to visualize. Therefore, it is necessary to reduce the dimensionality while losing as little of the necessary information as possible. The reduction enables graphical representation. It makes it easier to understand the distribution of the sample intuitively. Dimension selection can achieve dimension reduction through dimension compression.

Dimensional selection is the process of selecting dimensions that are easy for humans to understand in order to grasp the meaning of each dimension. Correct classification is possible by selecting appropriate dimensions. However, there is the problem that information is completely lost for unselected dimensions.

Dimensionality compression refers to representing high-dimensional data using a smaller set of dimensions while minimizing information loss. In [10], the authors explain this concept and demonstrate its effectiveness in practical applications. The interpretation of the dimension after reduction facilitates the classification of the information. In artificial dimension reduction, it is desirable to compress dimensions in order to estimate similarity and classify comprehension, because the discarded dimensions may contain important information. For example, when compressing a 100-dimensional object to one dimension, the scalar values are combined by taking the inner product of the 100-dimensional vector and the weight vector. The composite scalar value becomes the dimension after compression. Naturally, the method of deriving this weight vector differs according to the application.

PCA (Principal Component Analysis) and NMF (Nonnegative Matrix Factorization) are typical models for dimensional compression. In [11], the author explains that PCA selects the axis that minimizes the mean squared error. In other words, the maximum amount of information in the high-dimensional state is retained in a lower compressed dimension. This method compresses along the axis where the variance of each point is maximized. In [12], the authors explain that PCA is an unsupervised learning model because it does not require supervised data. It is considered a typical and basic model for dimensionality compression. Compared to PCA, NMF has a non-negative value constraint, which makes the results easier to interpret. In [13], [14] and [15], the authors describe NMF as an unsupervised learning model that serves as a typical and basic model for dimensionality compression.

2.2. Embedding and Cosine Similarity

The use of corpora yields a representation of the meaning of words. A corpus is a collection of example sentences that serves as a reference for machine learning models to understand a text. Words are polysemous. The meaning of the same word often changes from field to field. This makes it difficult to obtain a distributed representation that can be applied to all fields. To accurately represent the meanings of different fields, it is necessary to prepare a corpus with a rich collection of example sentences that are commonly used in a particular field. Embedding refers to the arrangement of natural language information, such as words and sentences, in a vector space that represents the meaning of the words and sentences. Vectorization of words and sentences is commonly used to calculate the similarity between words and sentences.

Word2Vec is commonly used for vectorizing words in natural languages, which vectorizes target words based on their co-occurrence probabilities. Word2Vec vectorizes target words from the probability of word co-occurrence. The input and output layers are neural networks with as many neurons as the number of words in the corpus and only one hidden layer. Hidden layers keep the number of neurons to a small number. When a set of words appearing in example sentences in a corpus is given to the input layer as a one-hot representation, this set of words is trained to be reproduced in the output layer.

Word2Vec considers the sequence of outputs of the hidden layer of the neural network after training as a vector representing the meaning of the words. This vector is called the distributed representation. The generated vector is calculated based on the co-occurrence probability of the words. The order relations of the words are not taken into account, so the context is ignored. Nevertheless, the advantage of being able to convert words into a distributed representation vector is significant.

One of the main advantages of vectorization by embedding is that the similarity of words can be calculated in terms of cosine similarity. Cosine similarity is calculated using the inner product of two vectors. Cosine similarity is an index that expresses how much the action of one of two vectors contributes to the action of the other vector. The possible values range from -1 to 1. The higher the value, the higher the similarity, while the lower the value, the lower the similarity. Cosine similarity can be calculated using the following formula.

$$sim = \cos\theta = \frac{\vec{a}\vec{b}}{|\vec{a}||\vec{b}|} = \frac{\vec{a}^{T}\vec{b}}{|\vec{a}||\vec{b}|} \qquad (1)$$

※Cosine similarity can be used to calculate vector similarity.

2.3. Vectorization of Documents

A textual description of a specialized field contains concepts and procedures from several more basic disciplines. Determining whether a given new description is correct is a matter of checking that the correct concepts and procedures are used for each basic field. In order to analyze the descriptions, a dimension is assigned to each basic field. Textual descriptions become representations in higher-dimensional spaces.

In [16], the authors explain that Doc2Vec distributed representations of sentences and documents, providing a compact vector representation that can be used for dimensionality compression. Doc2Vec vectorizes documents by embedding all words that occur in a document with Word2Vec and finding their average value. The analysis of large amounts of text data with Doc2Vec can convert sentences written in natural language into meaningful distributed representations. The transformation of the distributed representation measures the similarity between the vectors after analysis. It allows the sentence classification and the detection of similar sentences.

3. Determining Proximity to Correct Answers to Written Questions

3.1. Data Preprocessing and Signal Alignment

In the text preprocessing stage, the free-text responses are first normalized by unifying full-width and half-width characters. Noise such as symbols is then removed using regular expressions. Subsequently, morphological analysis is applied to extract nouns, verbs, adjectives, and adverbs. For vectorization, the extracted tokens are input into a Doc2Vec model to generate document embeddings. Given the limited amount of free-text data, the Doc2Vec parameters are configured to prioritize embedding stability, and the number of training epochs is increased until convergence is observed. To capture both contextual information and lexical distribution, the study employs both the PV-DM and PV-DBOW algorithms to obtain robust document representations.

To synchronize conversational data with physiological signals, the start and end times of each utterance in the conversation log are converted into seconds and mapped onto the same timeline as the EDA timestamps. This alignment allows the EDA values corresponding to each utterance to be extracted, enabling integrated analysis of linguistic behavior and physiological responses. In this study, “multimodal” refers to the integration of textual responses, conversational behaviors, and EDA signals, all aligned on a shared timeline for unified analysis.

To estimate how differences in cognitive load relate to learning performance, the ChangeFinder algorithm is applied to detect change points in the electrodermal activity. By identifying these change points, the study investigates situations in which cognitive load increases and examines how the magnitude and frequency of such changes differ between higher-performing and lower-performing learners.

In the ChangeFinder algorithm, a local autoregressive model is first constructed within a sliding window over the observed time series. The model is then used to generate a smoothed sequence, from which a second-stage AR model is learned. The change-point score is computed as the difference between the log-likelihood of the prediction model and that of the observation model, treating this difference as the loss that indicates abrupt changes in the signal.

In this study, time segments with high change-point scores are analyzed by examining teacher behaviors during the preceding one-minute interval. Furthermore, to identify more precise moments at which electrodermal activity changes occur within each one-minute window, a second pass of the ChangeFinder is applied to the EDA time series at a 0.25-second resolution. This finer-grained analysis enables the detection of specific instructional moments that induce changes in learner’s cognitive load. Based on the estimated change points, the study examines how teacher behaviors influence learning performance.

3.2. Estimation of Poorly Understood Concepts and Procedures

This paper proposes a method to evaluate whether the learner's answer is close to the correct answer in writing questions asked in order to check the learner's understanding. Furthermore, the study provides appropriate guidance for learners who do not have a good understanding. Textual analysis of the answers to the written questions determines which concepts and procedures in the specialized field are not understood. For this reason, the important words for each dimension are calculated after dimensionality reduction by dimensionality compression and dimensionality selection.

The calculation of words takes the difference between the answers of learners who understand and those who do not. The difference in answers estimates the concepts and procedures associated with the latter's lack of understanding. A schematic diagram of the proposed method is shown in Figure 1. Adaptation of this method can assist learners who are faltering if they identify concepts and procedures that they do not understand.

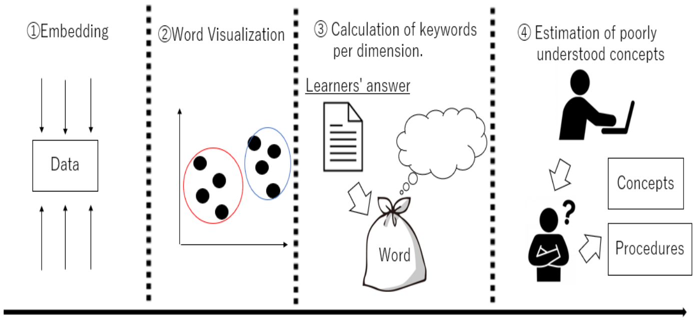

①Embedding
②Word Visualization
③ Calculation of keywords per dimension.
④ Estimation of poorly understood concepts
Learners' answer
Data
Word
?
Concepts
Procedures

Figure 1 : Methodological Overview Diagram.

This research requires the learner to master many concepts and procedures in a specialized field. Data analysis is taken as an example of a specialized field, but the proposed method is not specific to this field. The method gives the learner a task to analyze data and collects textual data from the learner's answers. The use of text analysis techniques determines whether an answer to a writing question is close to the correct answer.

The first step is dimensional compression and selection, followed by similarity estimation and comprehension classification. Similarity estimation calculates the cosine similarity by Doc2Vec. For comprehension classification, each word in the learner's answer and the model answer is converted into a distributed representation using Doc2Vec to visualize the similarity of the words. The visualization makes it easier to understand the classification of correct and incorrect answers. Next, after classification, this method calculates important words for each dimension. Naive Bayes lists the words that appear in the answers of correct and incorrect answers in descending order of frequency. Finally, this method extracts concepts that are poorly understood.

Logistic regression calculates the importance of important words in classifying those who answered correctly and those who answered incorrectly. Logistic regression also identifies procedures and concepts that are not understood. The evaluation of the Word2Vec and logistic regression models uses cross-validation to check the confusion matrix. The reliability and validity are ensured by calculating the F1-score from the reproducibility and fit rate, which is close to 0.8 without overfitting. Measures to reduce bias are taken to ensure that the actual level of understanding is adequately reflected in the test scores. Here, a certain number of people are ensured. The corpus is adjusted to reduce bias further.

3.3. Correct Answers and Solution Embedding

Model answers and learner answers in short answer questions are text data containing high dimensional information. The use of Doc2Vec allows the embedding of model answer examples and answer text into low-dimensional data. The compression of the dimensions allows all information to be taken into account.

First, morphological analysis using Mecab divides the learner's answer and the model example sentence into words to determine the degree of similarity. Next, Doc2vec is used to transform each sentence into a meaningful distributed representation. Finally, cosine similarity calculates the similarity between two sentence vectors obtained from Doc2Vec. The training of the Doc2Vec model requires a corpus, which is a collection of example sentences.

This research extracts multiple passages from textbooks that serve as answers to build a corpus. The way to improve the accuracy of the model is to increase the number of sentences used in the corpus. It also creates a corpus that matches the text content you want to analyze. The number of words is large when that of sentences used in a corpus is large. In addition, the amount of training for the model increases, resulting in higher accuracy. Furthermore, this method accurately distinguishes the intention of the questioner and the meaning of the question. In [17] and [18], the authors report that accuracy increases when the corpus is adapted to the content of each text so that it matches the subject matter being analyzed.

3.4. Word Visualization

When there are many dimensions, it is difficult to grasp the meaning of each dimension, so it is necessary to select dimensions that are easy for humans to understand. Doc2Vec visualizes the words that appear in the answer descriptions created by the learner, including correct and incorrect ones. The compression of high-dimensional vectors into two dimensions allows an understanding of the relationship between each text in the document vector.

The word groups are different for those who answered correctly and those who answered incorrectly. It is visualized by plotting document vectors on a two-dimensional plane. As an example, Figure 2 shows the words that appear in the learner’s answers and the model answers for a descriptive task asking why a conditional branch is necessary in a C programming exercise. This dataset contains data from 87 learners and is collected internally. The words "else" and "return" appear in the model answers, whereas the words "printf", "recognize", and "post" in the answers are words that are not necessary for a correct answer. This shows that there is a distance between words that are necessary and words that are not. Learners who do not understand answer the writing questions without using the necessary words.

They use the wrong words because they do not understand the syntax of if-else sentences.

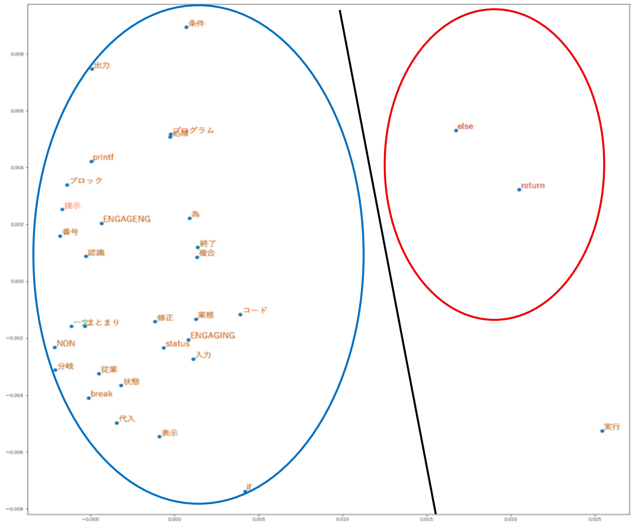


Figure 2: Similarity of Words Appearing in Example Answers and Model Answers.

3.5. Calculation of Keywords per Dimension

In [19], the authors describe Naive Bayes as a probabilistic classifier that identifies informative words based on their likelihood. Morphological analysis with MeCab is used to split sentences into words. For preprocessing, sentences are segmented into words using MeCab, which performs Japanese morphological analysis based on part-of-speech information, as explained in [20]. The result is a multinomial Naive Bayes model created from the generated word document matrix. Thus, creation involves calculating the probability of occurrence of words that appear in documents belonging to a specific field. The list of words in order of probability enables us to obtain the important words in the documents belonging to the field.

After acquiring important words, this method distinguishes the learner's writing into two classes: correct and incorrect. This method investigates what words are likely to be used in correct answers. It identifies words that are important in distinguishing between correct and incorrect answers. This important calculation uses a logistic regression model. Logistic regression is a regression model that uses qualitative variables (binary data) as objective variables.

This study uses the probability of the occurrence of words that appear in learner’s descriptions as an explanatory variable. Logistic Regression calculates the weights of the explanatory variables, so it is possible to find out which words are more likely to be correct if their probability of occurrence is high.

3.6. Identification of Poorly Understood Concepts

Learners who understand are likely to use the words they need and use keywords more frequently, as described in 3.4. In addition, learners who do not understand do not use the necessary words and use important words less frequently. The logistic regression model finds the weights of words that are particularly important in separating correct from incorrect answers among the multiple important words used by the correct and incorrect respondents.

This study allows the extraction of poorly understood concepts by determining the weights of particularly important words. The calculation of word importance allows us to discover which concepts and procedures have a significant influence on the classification of correct and incorrect answers.

4. Experiments, Experimental Results and Discussion

4.1. Experimental Overview

The task is to analyze specific data with a specified algorithm and to collect textual data from the subject's answers to written questions. This dataset is collected internally. The task involves decomposing MNIST image data into its components using NMF and PCA. The tasks include fill-in-the-blank questions, discussion questions, comprehension verification questions and a confirmation test.

This research particularly focuses on comprehension verification questions and confirmation tests. The fill-in-the-blanks question involves creating multiple blanks in a Python code that uses NMF and PCA to decompose the image data of a human face into its constituent elements. The fill-in-the-blank questions check whether the learner can correctly fill in the blanks. This question estimates whether the learner correctly understands the knowledge about the code for analysis. The discussion question asks learners to describe what the numerical values output as a result of applying the above Python code to the given data mean.

The comprehension verification questions, and confirmation tests ask questions about concepts and procedures in the field of dimensional compression, to which the two specified machine learning algorithms belong. The 40-point understanding verification questions include 16 true/false questions and 4 descriptive questions that require explanations using 30 characters or more. The scores are shown in Table 1.

Therefore, setting a minimum number of characters for written questions prevents learners from answering with fewer characters. Table 2 shows the comprehension verification questions used in the experiment. In addition, the confirmation test consisted of two writing questions of 100 characters or more, as shown in Table 3. These questions test the learner's understanding of lectures that explain the specified methods for analyzing a given problem. These also estimate whether the learner can correctly interpret the results of applying the method.

In the first experiment, subjects work on the NMF (non-negative matrix factorization). The learner and the teacher solve the problem together in a one-to-one format. The learner’s role solves machine learning fill-in-the-blanks, discussion, and comprehension verification problems. On the other hand, the teacher’s role helps the learners with exercises so that they can solve the problems.

There should be a difference in proficiency between the learner’s role and the teacher’s role. Therefore, the learner’s role is a learner who has been studying data science for a few months, and the teacher’s role is a learner who has been studying data science for several years.

The teacher's role is assumed to be a learner with a deep understanding of data science. There are 12 subjects in the learner's role. Regarding the experimental procedure, learners learn about NMF in advance in class. Next, the learner's role solves the fill-in-the-blank questions and discussion questions, and the teacher's role provides support. Finally, learners complete comprehension verification questions to check whether they understand NMF.

The second experiment involves PCA (Principal Component Analysis). As in the first experiment, the learner's and teacher's roles work one-to-one. The subjects are the 12 learners who participated in the first experiment and four learners who did not participate in Experiment 1.

The learners first attend a lesson, then complete a fill-in-the-blank question, a reflective question and finally a writing test. In the second experiment, the learners do not have to answer questions to verify their understanding of the first experiment, but rather they have to answer open-ended questions of 100 words or more. This is the purpose of this research, which is to find a method to determine whether the answer to the descriptive question is close to the correct answer.

The reason for using NMF and PCA tasks is that NMF and PCA have a similar function of extracting common components. Working on both NMF and PCA may deepen the understanding of both NMF and PCA.

Table 1: Scores on the 40-points Comprehension Verification Question in Experiment 1

※Table 1 shows the results for the 12 subjects of the 40-point comprehension verification and written questions in Experiment 1.

Learner	1	2	3	4	5	6	7	8	9	10	11	12
Score	34	34	28	24	35	33	30	26	27	25	33	24

Table 2: Comprehension Verification Questions to be Used in Experiment 1

※It contains both a writing question and a correct answer question. If you consider a statement to be incorrect in a correct or incorrect question, please provide a correct explanation in the description box.

Q 1	Non-negative matrix factorization is not non-negative for all features.
Q 2	Non-negative matrix factorization decomposes the original matrix into a matrix multiplication form.
Q 3	For the non-negative matrix factorization, the product of the original matrix and the two non-negative-valued matrices after decomposition is exactly equal without error.
Q 4	Features can be extracted by decomposing the matrix in the NMF.
Q 5	Answer in at least 30 words what scaling is.
Q 6	Answer the need for scaling in at least 30 words.
Q 7	Name one scaling method.
Q 8	Non-negative matrix factorization expresses a non-negative matrix as the product of two non-negative matrices.
Q 9	A horizontal sequence is called a row vertical sequence is called a column.
Q 10	Non-negative matrix factorization features can estimate missing values and can be classified using hidden features.
Q 11	Non-negative matrix factorization is supervised learning.
Q 12	The non-negative matrix factorization is the ability to estimate and complete missing values.
Q 13	One of the features of non-negative matrix factorization is the ability to estimate and complete missing values.
Q 14	Non-negative matrix factorization has been applied in various fields such as text data, sound separation, and automatic musical notation.
Q 15	Why must the non-negative matrix factorization target non-negative values ?
Q 16	In this exercise, we extracted features from facial images, but non-negative matrix factorization can also be applied to other applications such as Amazone product recommendations. Answer why it can be applied to such a variety of examples.
Q 17	Non-negative matrix factorization is not non-negative for all features.
Q 18	Non-negative matrix factorization decomposes the original matrix into a matrix multiplication form.
Q 19	For the non-negative matrix factorization, the product of the original matrix and the two non-negative-valued matrices after decomposition is exactly equal without error.
Q 20	Dimensional compression of non-negative-valued matrix factorization is the transformation of low-dimensional data into higher dimensions.

Table 3: Questions for the Confirmation Test to be Used in Experiment 2

※Please answer at least 100 words.

Q 1	Answer why common features can be taken out by multiplying the NMF matrices.
Q 2	Answer how the NMF gradually modifies the randomly set weight and feature matrices to reduce the error.

4.2. Optimal Method for Measuring Comprehension

In this experiment, open-ended questions are employed as a means of assessing learner's understanding. Moreover, the reason for setting open-ended questions is that learners must express correct answers in their own words.

In the first experiment, conversations between teachers and learners, which had a positive effect on learners, and electrodermal activity are recorded. In [21], the authors explain that electrodermal activity represents changes in the skin's electrical properties caused by sweat gland activity. In addition, the teacher's behavior toward the learner is quantified based on the content of the conversation.

Specifically, features that can be expressed as quantities, such as the number of characters spoken by the teacher and the number of questions asked by the teacher, are expressed as actual numerical values. For subjective evaluation items (e.g., whether the teacher guides the learner), we manually recorded whether the behavior appeared during that period. If it exists, it is represented as 1; otherwise, it is represented as 0.

Therefore, we present the results of random forest discrimination to identify behaviors that cause cognitive load to learners by teachers. In [22], the authors explain that random forests can estimate the importance of each predictor variable for classification and often achieve better predictive performance than linear regression.

Furthermore, because cognitive load causes stress in people, changes in cognitive load affect galvanic skin response through changes in skin conductance. In [23], the authors explain that higher cognitive load leads to stronger electrodermal responses. In addition, several studies have reported a positive correlation between cognitive load and electrodermal activity, as shown in [24], [25] and [26].

Although electrodermal activity is influenced by emotional changes, research shows that it can still objectively distinguish between different levels of cognitive load. In [27], the authors demonstrate this robustness even in the presence of emotional variation. Therefore, it can be said that skin potential activity can be used to measure cognitive load significantly even in individual instruction.

The objective variable is whether the teacher caused a change in the learner's skin potential activity during the one minute, and the explanatory variable is the teacher's behavior during the one-minute exercise. The data used in this study are collected from two learner participants during an individual tutoring experiment, where the same teacher participant conducted one-on-one instructional sessions with each learner. A total of 61 data samples collected from the two learner participants are divided into training and test datasets in an 80:20 ratio. Cross-validation is performed by alternately switching the training and test datasets 12 times to evaluate prediction accuracy. The average accuracy across the 12 iterations is approximately 79%. The variable importance derived from the random forest model is presented in Figure 3.

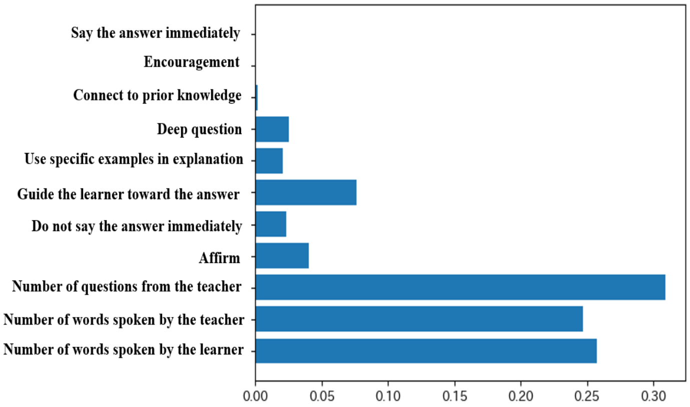


Figure 3: Variable Importance in the Random Forest Model

Among the variables with high importance, the most influential explanatory variable is the number of questions posed by the teacher to the learner during the individual tutoring sessions. When the learner is questioned by the teacher, it is assumed that cognitive load is induced as the learner attempts to formulate a response.

Subsequently, the variables that exhibited high importance included the number of words spoken by the learner, the number of words spoken by the teacher, and whether the teacher's utterances are intended to guide the learner toward the answer. It is assumed that a higher cognitive load is imposed when the learner produces a greater number of utterances. This is because the learner formulates and articulates responses while engaging in cognitive processing.

It is also observed that electrodermal activity increased when the teacher produced a larger number of utterances. This may be because the learner is cognitively processing the teacher's utterances to comprehend them.

In a similar manner to the teacher participant, additional analyses are performed for Participant B, who also acted as a teacher and provided one-on-one instruction to two learners. A total of 47 one-minute data samples obtained from the two learner participants are aggregated and split into training and test datasets in an 80:20 ratio.

A total of 47 one-minute data samples obtained from the two learner participants are aggregated and split into training and test datasets in an 80:20 ratio. Cross-validation is conducted by repeatedly exchanging the training and test datasets 12 times to evaluate the prediction accuracy. As a result, the mean prediction accuracy is approximately 40%.

The relatively low prediction accuracy may be caused by the fact that Participant B engaged in fewer instructional interactions with the learners compared to Participant A. Since the teacher's interventions are limited, when the explanatory variables are defined according to the teacher's behaviors during one-minute instructional segments, several factors other than the teacher's interventions appeared to contribute to elevated electrodermal activity. Therefore, this may have resulted in low prediction accuracy.

To identify the factors that influence performance, we first compute the correlation coefficient between EDA and the learner's scores. In this analysis, the learner's EDA during approximately 30 minutes of one-to-one tutoring is examined at 0.25-second intervals. For each time point, we compare the EDA value with that of the immediately preceding interval, and the proportion of intervals in which the EDA increases is treated as the overall rate of EDA elevation.

The learner's performance is defined as the score obtained on the forty-point test completed after the one-to-one tutoring session. The correlation between test scores and the proportion of increases in electrodermal activity is then computed. The result shows a correlation coefficient of minus zero point two four two, indicating that the frequency of increases in EDA, interpreted as increased cognitive load, does not correlate with learner's test performance. Figure 4 presents the relationship between the learner's test scores and the proportion of increases in electrodermal activity.

Even when electrodermal activity increases, this does not imply that the learner's performance improves. One possible explanation is that electrodermal activity also fluctuates in response to emotional changes. It is highly likely that the learners experienced emotional variations during their interactions with the teaching assistant, which may have contributed to changes in electrodermal activity independent of cognitive load.

Another possible explanation is that an increase in cognitive load does not necessarily lead to improved understanding. Cognitive load can have both beneficial and non-beneficial effects on learning outcomes.

According to cognitive load theory, there are three types of cognitive load, among which only germane cognitive load is considered to enhance learning. The other two types, intrinsic cognitive load and extraneous cognitive load, are regarded as unrelated to learning. During one-on-one tutoring, it is likely that not only germane load but also intrinsic load, caused by the inherent difficulty of the task, and extraneous load, arising from unclear or confusing instructions, are imposed on learners.

Furthermore, even when a teacher attempts to induce germane cognitive load, performance does not improve unless the learner's actual understanding deepens. These considerations indicate that simply increasing cognitive load does not enhance performance, and that unnecessary cognitive load should be avoided to support effective learning.

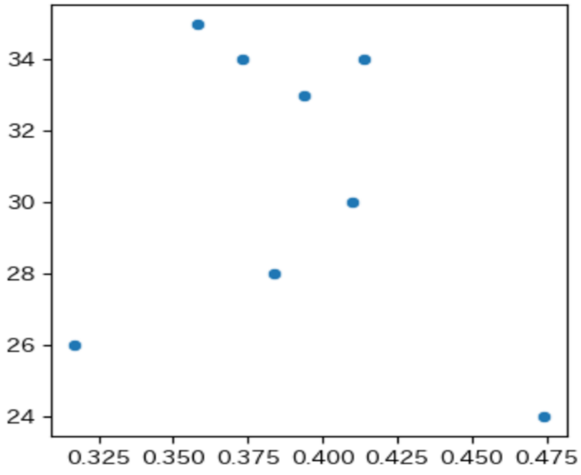


Figure 4: Scatter Plot of Scores and EDA

Next, to identify the types of cognitive load that positively influence performance, we use change point detection to extract the moments in which cognitive load increases and analyze the corresponding segments of conversation.

This study uses a total of four datasets collected from two teacher participants, Participant A and Participant B, each of whom provided instruction to two learners. Both learners taught by Participant A achieved higher test scores following the individual tutoring sessions, whereas both learners taught by Participant B showed lower test performance.

Therefore, Participant A is classified as a teacher who positively influenced learner's performance, whereas Participant B is classified as a teacher who did not exert a positive influence on learner's performance. The learner's test scores and standard scores are presented in Table 4.

Table 4: Test Scores and Standard Scores of Four Learners

Learner	Teacher	Score	Standard Score
1	Participant A	34	59.1
2	Participant A	34	59.1
3	Participant B	28	43.5
4	Participant B	24	33.2

In order to identify teacher behaviors that positively affect learner's performance, the behaviors of the two teacher participants are analyzed at the moments when electrodermal activity increased. To achieve this, change-point detection of electrodermal activity is performed using the ChangeFinder algorithm.

The teacher's behaviors in the one minute preceding each detected change point are compared to identify behavioral differences. To aid in understanding the above procedure, Figure 5 presents a graph illustrating the original electrodermal activity data and the scores generated by applying the ChangeFinder algorithm.

The left vertical axis represents the ChangeFinder change score, the right vertical axis represents electrodermal activity, and the horizontal axis represents time in 50-second intervals. Furthermore, the red line indicates the original data output, whereas the blue line indicates the change scores obtained from the ChangeFinder algorithm.

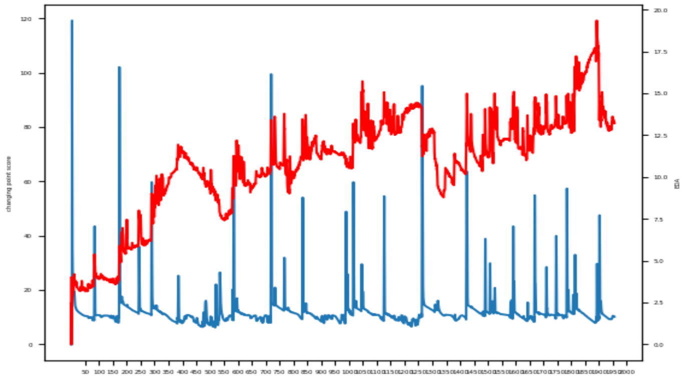


Figure 5: Electrodermal Activity and Change Scores

For the high-performing learners, the periods of increased electrodermal activity corresponded to times when the teacher asked a greater number of questions to the learners. In particular, this teacher frequently posed questions intended to guide the learners toward the correct answers.

For the low-performing learners, the periods of elevated electrodermal activity corresponded to times when the learners asked questions to the teacher or engaged in self-initiated reflection. These results are consistent with the hypothesis that teachers who positively affect learner's performance encourage learners to articulate correct answers in their own words.

These results are derived from the analysis of data obtained from four learners. To further test the hypothesis, data obtained from eight learners will be utilized. Table 5 presents the performance group classifications, test scores, and standard scores for the eight learners.

The data from the eight learners are divided into two groups, with the top four performers in one group and the bottom four performers in the other, and the behaviors observed during the one minute preceding each moment of increased electrodermal activity are analyzed.

The analyzed items include the following: (1) the proportion of learner utterances in the total teacher–learner conversation during one minute, (2) the number of words spoken by the learner per minute, (3) the number of words spoken by the teacher per minute, (4) the total number of words spoken by both the learner and the teacher per minute, (5) the number of questions asked by the teacher per minute, (6) the number of guiding questions asked by the teacher per minute, (7) the number of questions asked by the learner per minute, and (8) whether the teacher immediately provided the answer within one minute.

Table 5: Test Scores and Standard Scores of Eight Learners

Learner	Performance Group	Score	Standard Score
1	High performing	34	59.1
2	High performing	34	59.1
3	Low performing	28	43.5
4	Low performing	24	33.2
5	High performing	35	61.6
6	High performing	33	56.5
7	Low performing	30	48.7
8	Low performing	26	38.4

The analysis is conducted for each of the eight learners. The averages are calculated separately for the top four and bottom four performers. The results, rounded to four decimal places, are shown in Table 6. Compared with the low-performing group, the high-performing group showed a lower proportion of learner utterances and a higher total number of words spoken by the teacher, while the number of words spoken by the learners is nearly the same.

These findings suggest that, in the high-performing group, the teacher engaged in more frequent verbal interactions with the learners. Furthermore, compared with the low-performing group, the high-performing group exhibited a greater total number of words spoken by both the learners and the teacher, suggesting that both participants spent less time in silence.

Compared with the low-performing group, the high-performing group shows more teacher-initiated questions and fewer learner-initiated questions, indicating stronger instructional control.

The high-performing group also exhibited a greater number of guiding questions posed by the teacher. Therefore, for the high-performing group, higher levels of cognitive load are likely to occur during periods when the teacher is asking questions or engaging in verbal instruction.

In contrast, for the low-performing group, higher levels of cognitive load are more likely to occur when learners are engaged in independent thinking or when they ask questions to the teacher after a period of individual reflection.

Taken together, these findings support the hypothesis that encouraging learners to articulate correct answers in their own words is valid, and they underscore the importance of using open-ended questions to evaluate learner's understanding.

Table 6: Mean Values of the High and Low Performing Groups

Item	High Performing	Low Performing
Proportion of learner utterances	0.177	0.306
Total number of words spoken by learners	40	41.75
Total number of words spoken by the teacher	229.167	141
Total number of words spoken by both learner and Teacher	268.917	182.75
Number of questions asked by the teacher	1.125	0.15
Number of guiding questions asked by the teacher	0.75	0.05
Number of questions asked by the learner	0.083	0.25
Whether the teacher immediately gave the answer	0	0.25

4.3. Estimation of Similarity by Doc2Vec

In order to learn, the text data needs to be pre-processed so that unnecessary parts are removed from each text. This research uses processed text data from sentences as learning data and creates a learning model. The parameters and their values during training are shown in Table 7.

Table 7: Parameters during Learning

Parameter	size	window	min	workers	epochs
Setpoint	10	5	1	4	100

・size：dimensions of distributed representation

・window：number of surrounding words in context

(Decide how many words to consider at the same time.)

・min：minimum number of occurrences of words used for learning (Discard words with fewer occurrences than this value.)

・workers：number of threads in learning

・epochs：number of epochs

Doc2Vec checks the accuracy of the distributed representation of sentences. The similarity is calculated between the text written by the learner in the two confirmation test questions in Experiment 2 and the text of the correct answers. The use of Doc2Vec yields a vector representing each text. The cosine similarity checks how close the learner's text is to the correct text.

The individual learner scores are calculated by computing the cosine similarity of the distributed representations via Doc2Vec, then taking the average cosine similarity between Confirmation Tests 1 and 2. There are 12 learners in Experiment 1 and Experiment 2. Four more learners are added to the eight listed in Table 5. There are also four learners who participate in Experiment 2 only.

We analyze four learners who participated in Experiment 1 and also attended PCA classes outside the experiment. A significant difference is observed in the mean cosine similarity scores between these learners and those who participated only in Experiment 2.

Furthermore, Learners who participated in both Experiment 1 and 2 show higher cosine similarity scores than those who joined only Experiment 2. These results suggest that experiencing both NMF and PCA helps learners consolidate and deepen their understanding of NMF.

To address the need for quantitative evaluation, we conducted a performance analysis using 32 free-text responses collected from 16 learners. The proposed Doc2Vec-based cosine similarity approach is evaluated in terms of accuracy, precision, recall, and F1-score.

To provide a rigorous comparison, three baseline models are included: (1) TF-IDF combined with SVM, (2) Sentence-BERT, and (3) cosine similarity without embedding. Table 8 summarizes the results. The Doc2Vec cosine similarity with embeddings achieved the highest overall performance (F1-score = 0.824), outperforming TF-IDF + SVM (F1 = 0.800), Sentence-BERT (F1 = 0.818), and cosine similarity without embedding (F1 = 0.698). These results indicate that Doc2Vec embeddings capture lexical and semantic features that are not effectively represented by traditional bag-of-words models or simple similarity measures.

To further evaluate statistical significance, we calculated the p-value, confidence interval, and effect size for the Doc2Vec-based classifier. The method shows a statistically significant difference ($p = 0.0096$), a large effect size (Cohen's d = 1.286), and a 95% CI [0.500, 0.833]. These findings demonstrate that the Doc2Vec approach discriminates reliably between learners with sufficient and insufficient understanding.

Taken together, the quantitative comparisons confirm that Doc2Vec embeddings provide the most robust representation for comprehension estimation in this dataset. Additionally, the superior performance relative to Sentence-BERT suggests that transformer-based scoring models do not necessarily generalize well in low-data settings such as ours, supporting the use of lightweight embedding models for small-scale educational assessment tasks.

Table 8: Baseline Results

Baseline	TF-IDF ＋ SVM	Sentence Bert	Doc2Vec (No Embedding)	Doc2Vec (with Embedding)
Accuracy	0.667	0.733	0.667	0.700
Precision	0.690	0.783	0.850	0.700
Recall	0.952	0.857	0.650	1.000
F1-score	0.800	0.818	0.698	0.824

4.4. Extracting Important Words Using Naive Bayes

Experiment 2 consisted of two descriptive questions as a confirmation test. Doc2Vec converts the similarity of words that appear between correct and incorrect answers into a distributed representation and visualizes it. The visualization confirms whether it is possible to classify those who answer correctly and those who answer incorrectly. Figure 6 shows the classification results for Confirmation Test 1.

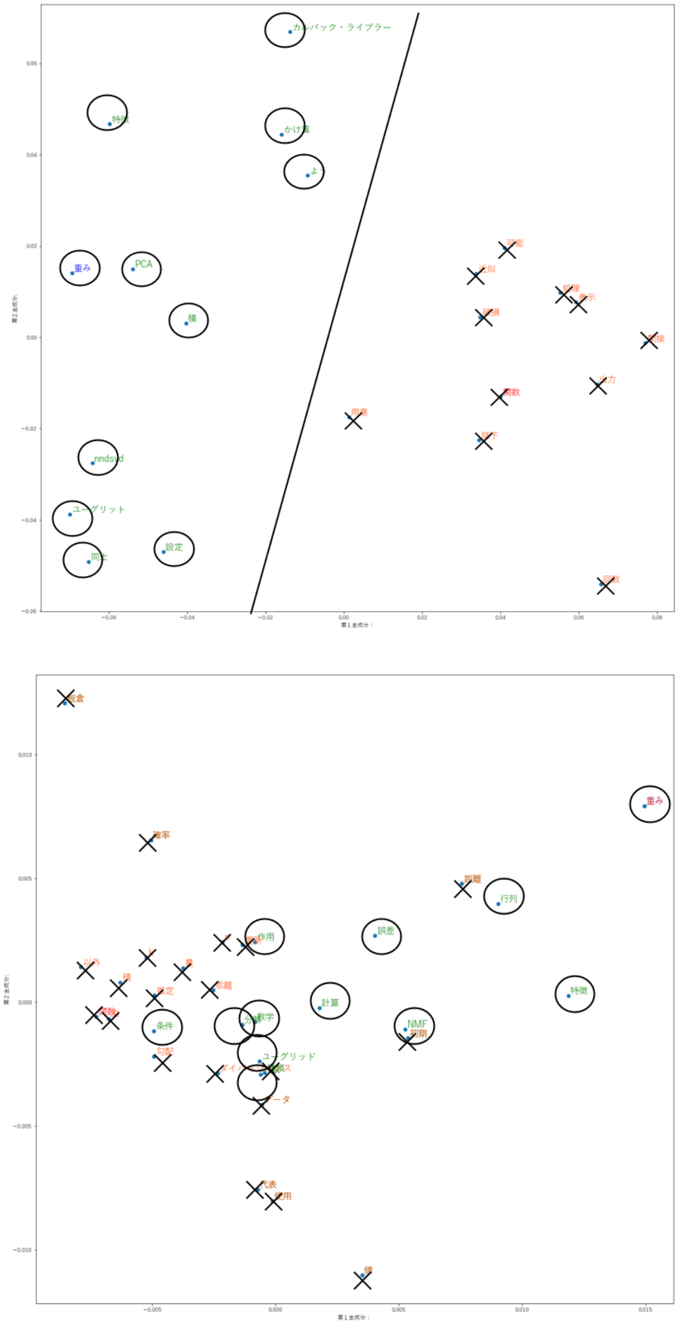


Figure 6: Distributed Representation of the Words in Each of the Subjects' Answers and Correct Answers in Confirmation Test 1(with and without Corpus)

In Confirmation Test 1, words frequently used by correct answers are marked with black circles, and words frequently used by incorrect answers are marked with black crosses. The left-hand diagram shows the case where no corpus is provided for each question, and the right-hand diagram shows the case where a corpus is provided.

Figure 6 shows that with a question-by-question corpus, it is possible to use a straight line to classify the words most frequently used by both correct and incorrect answers. This shows the necessity of having a corpus for each question. Consistent classification patterns are also observed in Confirmation Test 2.

4.5. Extracting Important Words Using Naive Bayes

Words whose appearance probability exceeds 0.02 are considered important words. Learners with high cosine similarity to the correct answer have high comprehension, while learners with low cosine similarity have poor comprehension. In both Confirmation Test 1 and Confirmation Test 2, the important words used by learners with high cosine similarity are positioned on the left side. For Confirmation Test 1, these words include "non-negative", "data", "factorization", "weight", and "multiplication". For Confirmation Test 2, the important words are "matrix", "feature", "use", "weight", "data", "be", "modification", and "calculation", "multiplication." In contrast, the important words used by learners with low cosine similarity are placed on the right side. For Confirmation Test 1, these words are "base" and "vector", and for Confirmation Test 2, the words are "function" and "minimum".

Although some words overlap between learners with high and low cosine similarity, the non-overlapping words represent the truly important terms. Accordingly, learners with high cosine similarity to the correct answers use a larger number of important words, whereas learners with low cosine similarity use far fewer of these important terms.

4.6. Calculation of importance by Logistic Regression

We examine the words with the largest absolute weights in the logistic regression model used to classify correct and incorrect answers. These are important words for classifying correct and incorrect answers. The words 'non-negative' and 'factorization' are more important in Confirmation Test 1, and the words 'error', 'random', and 'use' are more important in Confirmation Test 2. The important words in Confirmation Test 1 and Confirmation Test 2 relate to the features and mechanisms of the NMF. The difficult concepts for learners are the core properties of NMF, such as low dimensionality, non-negativity, and the mechanisms for exploring feature and weight matrices. These concepts are found by identifying several key terms. Upper-division university learners studying data science are selected for the analysis.

4.7. Significance of Results

The results of all the analyses allow us to identify the words that are important for classifying correct and incorrect answers. This identification compares the words used by learners with a high cosine similarity to the correct answer with those used by learners with a low cosine similarity. This can identify areas and concepts that learners with low cosine similarity do not understand.

The method uses the results of responses to writing questions from previous learners who have experienced the same lesson and task. Examining new learner's writing can find those who seem to be faltering. Learners with a poor understanding do not use key vocabulary. They do not know what the key concepts are that they need to understand in the area.

In [28], the authors introduce transformer-based NLP models such as BERT for the automatic scoring of short-answer responses. However, we do not adopt BERT-based approaches because lighter methods—especially Doc2Vec—show higher performance while requiring fewer computational resources.

In [29], the authors explain that ChatGPT, which has recently received significant attention, can generate fluent and high-quality responses to user queries. However, in [30], the authors report that ChatGPT shows limitations in paraphrasing and in handling tasks that require processing semantically similar expressions. Because of these limitations, obtaining an accurate response often requires that the questioner clearly identify what is not understood and explicitly specify the relevant concept when asking a question. Mastering ChatGPT therefore requires understanding one's own points of confusion and recognizing the key concepts within each content area. In this respect, the responses of past learners to writing questions provide valuable insight into which keywords are essential for understanding.

In [31], [32] and [33], the authors indicate that many prior studies on comprehension measurement rely on mark-test formats. This is to minimize the degree of freedom in the notation of the analyzed subject. This method analyzes answers to descriptive questions with a high degree of freedom. Open-ended questions are advantageous because they reduce bias and allow the discovery of unanticipated learner ideas. For learners, the open-ended nature of the questions means that they have to express themselves in their own words and explain logically, which is thought to improve their understanding.

For data science educators, the results can be used to create practice questions to check learner's understanding, for example, by using the questions in the examinations as questions on points that many learners did not understand.

For data science learning, especially when it comes to the motivation and engagement of learners who have difficulty with the material, the open-ended questions in this study reveal key points in those who do and do not understand the material. Therefore, it can be assumed that teaching these points will have a more positive impact on motivation and commitment.

On the other hand, subjects could be asked to write Python code that solves an appropriate data analysis problem. With Python code, there are limited ways to express it, and the characteristics become clearer when it comes to understanding and not understanding. The analysis of the code allows for a more accurate analysis than analyzing descriptions in natural language. Therefore, this study also found it necessary to analyze the codes.

Although this study is able to identify the concepts needed for understanding, it is possible that it may not be applicable to a diverse range of learners at different levels. Therefore, as a future research project, although this experiment is conducted with about 15 learners, it may be necessary to conduct the experiment with a separate group of learners, e.g., 100 learners.

Given the work required to analyze the open-ended responses, it is clear that the method can be applied to around 15 upper-year university learners. However, it is not known whether this would make the possible analysis more accurate in larger classes or in an online environment. This method does not require training, so real-time evaluation is possible. However, large corpora are needed beforehand.

In this study, all participants are fully informed about the purpose of the research, the use of EDA, conversational data, and free-text response data, as well as the measures taken to protect their privacy. Written informed consent is obtained prior to participation. All collected data are anonymized and securely stored to ensure that no individual can be identified.

5. Conclusion

This study proposed a method for estimating learner's understanding of data science by analyzing their open-ended responses using document embeddings.

By transforming free-text answers into distributed representations and evaluating their cosine similarity to model answers, the method successfully identifies learners who demonstrate correct conceptual understanding. The quantitative evaluation shows that the Doc2Vec-based approach achieves the highest performance among multiple baselines, with an F1-score of 0.824, statistically significant improvement ($p = 0.0096$), and a large effect size (Cohen's $d = 1.286$; 95% CI [0.500, 0.833]). These results indicate that the embedding-based similarity measure provides a reliable metric for discriminating between adequate and insufficient comprehension.

The analysis further reveals that learners with low similarity scores tend to omit key conceptual vocabulary used in correct answers, allowing the method to automatically identify specific concepts or terminology that learners misunderstand or fail to recall. This demonstrates the utility of open-ended responses for diagnosing misconceptions in ways that traditional multiple-choice assessments cannot capture.

From an educational perspective, the proposed method offers actionable benefits for instructors. By automatically extracting missing key terms and detecting conceptual gaps from learner's written explanations, instructors can more efficiently pinpoint topics that require reinforcement and design targeted feedback or supplementary exercises.

The approach also provides an immediate diagnostic tool for monitoring learner's comprehension during data science instruction, helping educators intervene before misunderstandings become entrenched. Overall, the findings support the effectiveness of embedding-based analysis for assessing conceptual understanding from free-text answers and highlight the pedagogical value of incorporating open-ended questions into data science education. Future research will focus on extending the proposed method to larger and more diverse learner populations and exploring its integration into real-time educational support systems.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

We would like to thank many people for their guidance and encouragement in the preparation of this paper. We would like to thank everyone who willingly participated in the experiments during the execution of this paper. Finally, we would like to thank the members of the Data Engineering Laboratory for their invaluable advice and cooperation.

References

[1] R. D. De Veaux et al., "Curriculum guidelines for undergraduate programs in data science," Annual Review of Statistics and Its Application, 4, 15–30, 2017, doi:10.1146/annurev-statistics-060116-053930.

[2] H. Hedges and K. Given, "Addressing confirmation bias in middle school data science education," Foundations of Data Science, **5**(2), 2023, doi: 10.3934/fods.2021035.

[3] C. E. Brassil and B. A. Couch, "Multiple-true-false questions reveal more thoroughly the complexity of student thinking than multiple-choice questions:A Bayesian item response model comparison," International Journal of STEM Education, 6, 2019, doi: 10.1186/s40594-019-0169-0.

[4] K. Inohara, C. Matsuno, M. Furuya, and I. Kutsuzawa, "Differences between yes/no and multiple-choice vocabulary tests: Examination from the perspective of familiarity with reading," Japanese. Journal of Psychology, 91(6), 367–377, 2021, (in Japanese), doi: 10.4992/jjpsy.91.19028.

[5] R. Azuma, "Analysis of relationship between learner's characteristics and level of understanding using text-mining," Journal of Japan Society for Information and Systems in Education, 2017, (in Japanese).

[6] B. R. Shapiro, A. Meng, C. O'Donnell, C. Lou, E. Zhao, B. Dankwa, and A. Hostetler, "Re-Shape: A method to teach data ethics for data science education," in Proc. ACM CHI Conference on Human Factors in Computing Systems, 1–13, 2020, doi: 10.1145/3313831.3376251.

[7] C. Xing, D. Wang, X. Zhang, and C. Liu, "Document classification with distributions of word vectors," in Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, Siem Reap, Cambodia, 1–5, 2014.

[8] A. Condor, M. Litster, and Z. Pardos, "Automatic short answer grading with SBERT on out-of-sample questions," in Proc. 14th International Conference on Educational Data Mining, 345–352, 2021.

[9] K. Yasuda, H. Shimakawa, and F. Harada, "Identifying comprehension fault from word occurrences in writing questions," in Proc. Int. Conf. Frontiers of Signal Processing, Paris, France, 133–141, 2024, doi: 10.1109/ICFSP62546.2024.10785436.

[10] A. Géron, Practical Machine Learning with Scikit-learn, Keras, and TensorFlow, 2nd ed., M. Shimoda, Supervis., T. Nagao, Transl., O'Reilly Japan, pp. 215–235, 2020.

[11] D. Hachiya, Basic Python Learning from 0, Kodansha, 2020, (in Japanese).

[12] H. Zhang, "The optimality of naive bayes," in Proc. 17th International FLAIRS Conference, 562–567, 2004.

[13] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," Computational Statistics and Data Analysis, 52, 155–173, 2007, doi: 10.1016/j.csda.2006.11.006.

[14] M. W. Gills and F. Glineur, "Document classification using nonnegative matrix factorization and underapproximation," in Proc. IEEE International Symposium on Circuits and Systems, 2782–2785, 2009, doi: 10.1109/ISCAS.2009.5118379.

[15] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, **401**, 788–791, 1999, doi: 10.1038/44565.

[16] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in Proc. 31st International Conference on Machine Learning, 32, 1188–1196, 2014, doi: 10.5555/3044805.3045025.

[17] T. Sasada, S. Mori, Y. Yamagata, H. Maeda, and T. Kawahara, "Definition of recipe terms and automatic construction of a tagging corpus for automatic recognition," Journal of Natural Language Processing, **22**, 107–131, 2015, (in Japanese), doi: 10.5715/jnlp.22.107.

[18] S. K. Safa and D. R. CH, "Development of a practical system for computerized evaluation of descriptive answers of middle school level students," Interactive Learning Environments, **30**(2), 215–228, 2019, doi: 10.1080/10494820.2019.1651743.

[19] A. McCallum and K. Nigam, "A comparison of event models for naïve Bayes text classification," in Proc. AAAI Conference Artificial Intelligence, 1998.

[20] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Japanese morphological analysis using conditional random fields," Journal of Natural Language Processing, 161, 89–96, 2004, (in Japanese).

[21] M. Benedek et al., "A continuous measure of phasic electrodermal activity," J. Neurosci. Methods, **190**(1), 80–91, 2010, doi: 10.1016/j.jneumeth.2010.04.028.

[22] K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne, "A unifying framework for detecting outliers and change points from time series," IEEE Transactions on Knowledge and Data Engineering, **18**(4), 482–492, 2006, doi: 10.1109/TKDE.2006.1599387.

[23] P. Ayres, J. Y. Lee, F. Paas, and J. J. G. van Merriënboer, "The validity of physiological measures to identify differences in intrinsic cognitive load," Frontiers Psychology, 12, 2021, doi: 10.3389/fpsyg.2021.643265.

[24] N. Nourbakhsh et al., "Using galvanic skin response for cognitive load measurement in arithmetic and reading tasks," in Proc. 24th Australasian Computer–Human Interaction Conference, 420–423, 2012, doi: 10.1145/2414536.2414602.

[25] C. Setz et al., "Discriminating stress from cognitive load using a wearable EDA device," IEEE Transactions on Information Technology in Biomedicine, 14, 410–417, 2009, doi: 10.1109/TITB.2009.2036164.

[26] B. Mehler et al., "Sensitivity of physiological measures for detecting systematic variations in cognitive demand from a working memory task: An on-road study across three age groups," Human Factors, 54, 396–412, 2012, doi: 10.1177/0018720812442086.

[27] N. Nourbakhsh, F. Chen, Y. Wang, and R. Calvo, "Detecting users' cognitive load by galvanic skin response with affective interference," ACM Transactions on Interactive Intelligent Systems, **7**(3), 1–20, 2017, doi: 10.1145/2960413.

[28] P. Ghavidel, S. Zargari, and A. Mohammadi, "Using BERT and XLNet for the Automatic Short Answer Grading Task," in Proc. International Conference on Artificial Intelligence in Education, 125–136, 2020, doi: 10.5220/0009422400580067.

[29] G. Zuccon and B. Koopman, "Dr ChatGPT, tell me what I want to hear: How prompt knowledge impacts health answer correctness," arXiv preprint arXiv:2304.10017, 2023, doi: 10.18653/v1/2023.emnlp-main.928.

[30] Q. Zhong, L. Ding, J. Liu, B. Du, and D. Tao, "Can ChatGPT understand too? A comparative study on ChatGPT and fine-tuned BERT," 2020, doi: 10.48550/arXiv.2302.10198.

[31] S. Ashraf, S. Saleem, T. Ahmed, Z. Aslam, and M. Shuaeeb, "Iris and foot based sustainable biometrics identification approach," in Proc. International Conference on Software, Telecommunications and Computer Networks, Split, Croatia, 1–6, 2020, doi: 10.23919/SoftCOM50211.2020.9238333.

[32] S. Saleem, S. Ashraf, and M. K. Basit, "CMBA – A candid multi-purpose biometric approach," ICTACT Journal of Image and Video Processing, 2211–2216, 2020, doi: 10.21917/ijivp.2020.0317.

[33] Y. Fujita, Y. Hino, and A. Akazawa, "Multigigabit optical interconnection LSTIs," in Proc. Symposium on VLSI Circuits, Kyoto, Japan, 69–70, 1993, doi: 10.1109/VLSIC.1993.920541.

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 10, No. 6, 55-64 (2025)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTESJ
ISSN: 2415-6698

System-Level Test Case Design for Field Reliability Alignment in Complex Products

Robinson Lawrance [1], Nishith Kumar Reddy Gorla[*2]

[1]*Reliability Engineering, HCL America Inc, Sunnyvale, CA, 94085, USA*

[2]*Reliability Engineering, CORE ITS LLC, South Plainfield, NJ, 07080, USA*

ARTICLE INFO

Article history:
Received: 15 October, 2025
Revised: 14 November, 2025
Accepted: 16 November, 2025
Online: 05 December, 2025

Keywords:
Test Case Design
Field Reliability
Complex Product
Reliability Analysis
Reliability Testing
Reliability Growth Testing
Medical and Consumer Product
Test to Field Correlation
Integration Testing
Closed Loop Approach
Customer Satisfaction

ABSTRACT

Achieving targeted reliability for complex products in real-world field environments remains a persistent challenge, even when laboratory validation suggests high performance. A significant reliability gap often emerges during the initial deployment phase, typically within the first one to five years where field failure rates can be up to twice those predicted in controlled settings. Compounding this issue is the limited correlation between failure modes observed in the field and those anticipated during lab testing, with studies indicating only 50–60% alignment. These discrepancies result in unforeseen operational costs, elevated warranty claims, and reduced customer satisfaction.

This paper investigates the root causes of the disconnect between laboratory predictions and field performance, proposing a comprehensive framework to improve reliability demonstration and failure mode correlation. The framework introduces a closed-loop reliability correlation system that integrates diverse data sources and feedback mechanisms to achieve up to 95% alignment between lab and field failure modes.

The proposed methodology builds upon traditional DFMEA practices by incorporating Function Block Diagrams (FBD), Interface Matrix (IM), Parameter (P-) Diagrams, and field failure trend analysis. It expands the scope of reliability assessment to include actual usage conditions, patterns, and stakeholder interactions shifting from an engineer-centric view to a holistic, user-centered approach. Internal component-level data remains consistent, but the enriched context enables deeper insights into real-world performance.

By embedding these multidimensional analyses into system-level test case design, the framework ensures comprehensive coverage of critical variables, noise factors, and interaction effects. This results in more representative simulations, improved predictive accuracy, and early identification of latent failure modes. Ultimately, the proposed approach bridges the gap between laboratory and field environments, enhancing reliability metrics, and *enabling proactive mitigation strategies that align with operational realities.*

1. Introduction

Demonstrating the targeted reliability of highly complex products in real-world field environments remains a significant challenge, even when such reliability has been successfully validated under controlled laboratory conditions. A noticeable gap often emerges during the initial deployment phase, typically within the first one to five years where the actual field reliability consistently falls short of expectations. Empirical observations across various product categories reveal that failure rates in the field can be approximately twice as high as those predicted in laboratory settings. For instance, a product designed to achieve a 1% failure rate may exhibit a field failure rate ranging from 2% to 3% during early usage.

A second critical challenge lies in the correlation between failure modes observed in the field and those identified during laboratory testing. Studies indicate that this correlation typically ranges between 50% and 60%, implying that nearly half of the failures encountered in the field are previously unobserved or unanticipated. These unexpected failure modes contributed to increased operational costs, including higher warranty claims, spare parts inventory management, and diminished customer satisfaction. The field correlation gaps for various products are shown in Figure 1.

[*] Corresponding Author: Nishith Kumar Reddy Gorla, Ph.D., nishithgorla@gmail.com

https://dx.doi.org/10.25046/aj100605

These findings collectively suggest that conventional laboratory testing methodologies are insufficient in capturing the full spectrum of failure mechanisms that manifest in real-world conditions. This paper investigates the underlying causes of this disconnect and proposes a comprehensive framework to enhance field reliability demonstration and improve failure mode correlation to a target of 95%.

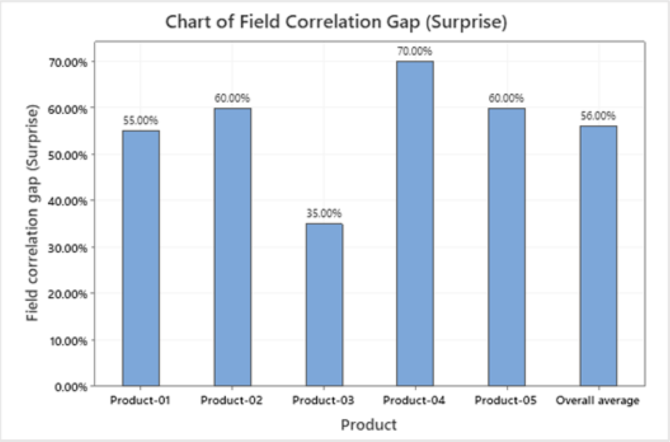


Figure 1: Gap in Field Correlation [1]

The focus of this study is twofold: (1) to establish robust strategies for demonstrating reliability in field environments, and (2) to develop a closed-loop reliability correlation system. This system integrates diverse data inputs and feedback mechanisms to bridge the gap between laboratory predictions and field performance, thereby enabling more accurate reliability assessments and proactive mitigation strategies.

This methodology emphasizes system-level validation rather than component-level or subsystem-level testing, as those are prerequisite activities for comprehensive system verification. The focus is on evaluating end-to-end user interactions and real-world operational scenarios, rather than isolated component demonstrations or accelerated stress screening techniques such as HALT (Highly Accelerated Life Testing) or HASS (Highly Accelerated Stress Screening).

2. Approach and Methodology

2.1. Existing Approach

In most cases, test designers prioritize the engineering perspective to identify the dominant failure modes revealed through analytical methods, typically via Design Failure Mode and Effects Analysis (DFMEA). DFMEA systematically enumerates potential failure modes that a product may encounter throughout its lifecycle, along with associated severity and occurrence ratings. This information enables engineers to target specific failure modes during the test design phase.

This existing approach in Figure 2 is particularly effective for systems with minimal external interactions and limited interdependencies among subsystems.

A primary limitation of this methodology is its insufficient integration of existing empirical data, as well as its lack of consideration for noise factors and variable conditions encountered in real-world environments. Consequently, while the approach enables engineers to effectively demonstrate reliability within controlled laboratory settings, it falls short in accurately representing field-level performance and operational variability.

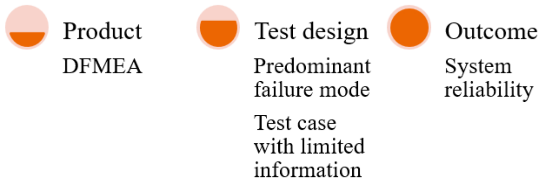


Figure 2: Existing Approach

2.2. Proposed Approach

Field reliability is influenced by multiple interdependent factors, primarily supplier reliability, manufacturing process reliability, and design reliability. Among these, supplier and process-related issues typically manifest as infant mortality failures occurring within the first 0 to 2 years of field deployment. These early-life failures, although impactful, are generally not classified as reliability failures in the context of long-term product performance. Consequently, this study places greater emphasis on design reliability and its associated failure mechanisms at the system level.

Design reliability is traditionally assessed through a sequential process that begins with reliability analysis and culminates in reliability testing. The analysis phase encompasses several key activities: feasibility studies, component selection guided by design-for-reliability principles, risk assessment via Design Failure Mode and Effects Analysis (DFMEA), and reliability prediction using either Physics of Failure (PoF) models or handbook-based estimation techniques. These analytical methods primarily focus on design characterization and theoretical reliability estimation and are often insulated from the influence of real-world usage conditions and environmental interactions.

Reliability testing, on the other hand, is typically conducted at the component or sub-assembly level with the objective of validating system-level reliability. While this approach is effective for relatively simple systems, it becomes inadequate as system complexity increases. In such cases, component-level testing fails to capture the intricate interactions, noise factors, and variable conditions that influence system behavior over time. Therefore, system-level testing becomes essential not merely to validate performance, but to replicate the multifaceted conditions of the field environment and expose latent failure modes.

Designing effective system-level tests involves a multi-layered approach, beginning with sample selection and extending through detailed failure analysis. At the core of this process lies the test case, the fundamental unit of reliability testing. A test case is defined as a structured sequence of actions applied to the system under test, intended to simulate real-world operational scenarios and evaluate system behavior and reliability over time. The design of test cases is critical, as it serves as the primary mechanism for embedding field-representative conditions into laboratory testing environments.

Given these considerations, reliability testing particularly at the system level emerges as a focal point for improvement. A deeper examination reveals that the effectiveness of the entire testing framework hinges on the quality and relevance of the test cases employed. Enhancing test case design is thus pivotal to bridging the gap between laboratory predictions and field performance, and to achieving more accurate and comprehensive reliability assessments.

2.3. Methodology

The flowchart in Figure 3 illustrates the foundational structure of field reliability by categorizing it into three primary domains: supplier reliability, manufacturing process reliability, and design reliability. While supplier and process reliability are typically associated with early-life failures which often occur within the first two years of product deployment, design reliability plays a more critical role in long-term performance. Design reliability is further divided into analysis and testing. The analysis phase includes component selection, risk analysis (such as DFMEA), and reliability estimation using models like Physics of Failure or handbook-based approaches. The testing phase is differentiated by system complexity: for simple systems, component and sub-assembly level testing may suffice, but for highly complex systems, system-level testing becomes essential. This level of testing aims to replicate real-world conditions by accounting for interactions, environmental noise, and variable factors that influence system behavior. At the heart of system-level testing is the test case, a structured sequence of actions designed to simulate field scenarios and evaluate reliability over time. The image underscores that effective test case design is pivotal for bridging the gap between laboratory predictions and actual field performance.

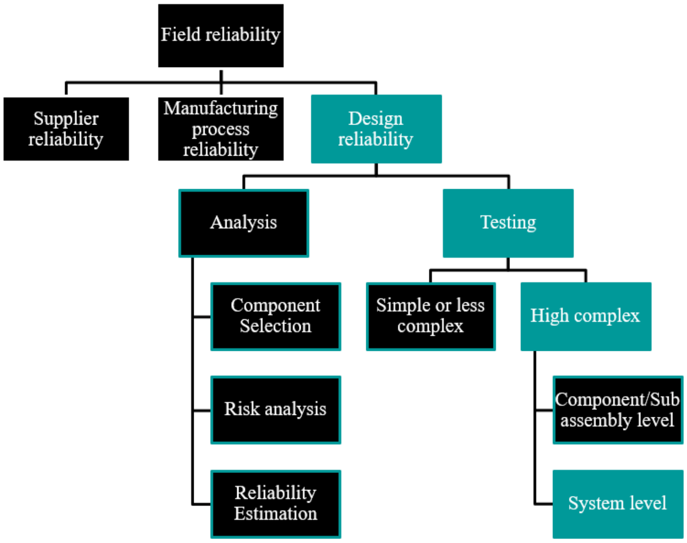


Figure 3: Field Reliability Flowchart

The overview of system level testing is shown in Figure 4. As highlighted in the figure, test case design is a critical and foundational element of system-level reliability testing. Its role in accurately simulating real-world field conditions makes it indispensable for capturing complex failure modes. The subsequent sections of this paper will explore test case design in greater detail, emphasizing its structure, implementation, and impact on improving reliability outcomes.

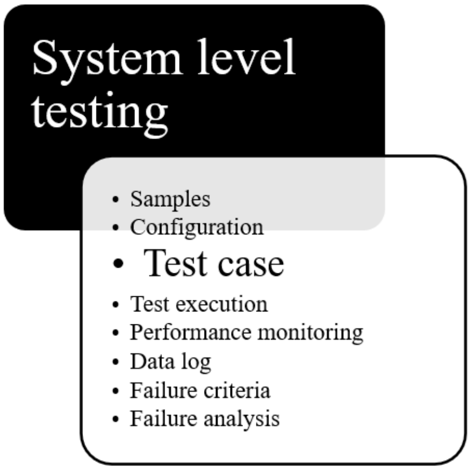


Figure 4: System Level Testing Overview

3. Test Case Design Inputs

Test case design is a structured process aimed at integrating all relevant inputs that can effectively simulate real-world failure modes observed in the field. These inputs are derived from multiple sources, including product reliability analysis, historical field performance data, internal study findings, and interaction parameters that influence system behavior under operational conditions. The overview of inputs for the test case design is shown in Figure 5.

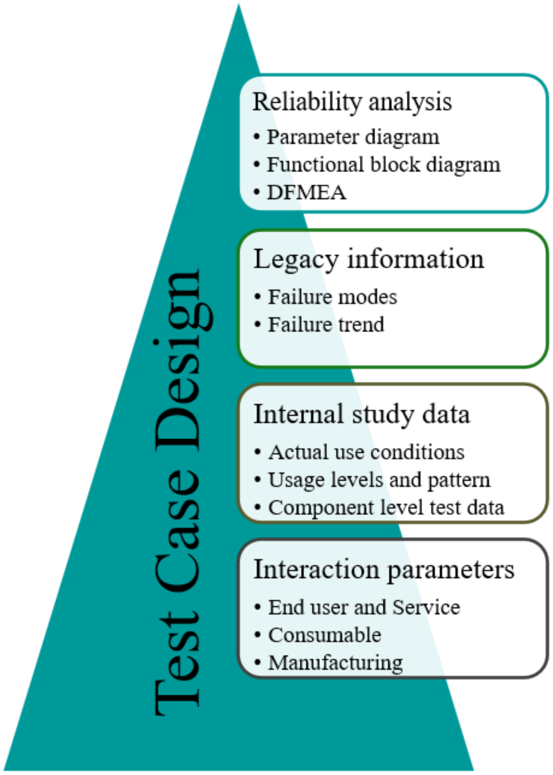


Figure 5: Test Case Design Overview

As shown in Figure 5, inputs include outputs from reliability analysis such as parameter diagrams, functional block diagrams, and Design Failure Mode and Effects Analysis (DFMEA) which help identify potential risks and failure mechanisms during the design phase. Legacy information, including historical failure modes and failure trends, provides valuable insights into recurring issues and long-term reliability challenges. Internal study data contributes further depth by capturing actual use conditions, usage levels and patterns, and component-level test results, all of which reflect how the product behaves under operational stress. Additionally, interaction parameters spanning end-user behavior, service environments, and manufacturing influences play a crucial role in shaping system performance and reliability outcomes.

By synthesizing these inputs, test case design becomes a powerful tool for replicating field conditions in laboratory settings, enabling the identification and mitigation of complex failure modes that might otherwise go undetected. More details on each input parameter will be discussed in the next section of the paper.

3.1. Reliability Analysis

3.1.1. Parameter (P-) Diagram

A major source of unexpected failures in the field stems from the underestimation of noise factors present in real-world operating environments, which contribute to approximately 28% of these surprises [1]. To address this, it is essential that test case design explicitly incorporates noise parameters and potential error states that reflect actual field conditions. One effective tool for this purpose is the Parameter Diagram (P-diagram), which helps in developing robust products that are less sensitive to external variability. The P-diagram outlines the relationship between input factors and the desired functional output, while also identifying sources of noise that may disrupt performance.

The process of generating test cases from a P-diagram involves two key steps. First, noise parameters often linked to dominant failure modes are identified and translated into actionable testing elements. These parameters typically include piece-to-piece variation, customer usage behavior, aging effects, system-level interactions, and environmental influences. Second, each of these actions is transformed into specific test cases designed to simulate the corresponding noise effects during reliability testing.

3.1.2. Functional Block Diagram (FBD)

In the context of reliability engineering, the Functional Block Diagram (FBD) serves as a foundational input for Design Failure Mode and Effects Analysis (DFMEA), where each function identified in the diagram is systematically linked to potential failure modes and their associated effects. When used in test case design, the FBD helps ensure that critical functional paths and interface interactions are captured, enabling the development of test scenarios that more accurately reflect real-world operating conditions and potential failure mechanisms.

So, these insights are then translated into targeted test actions, which form the basis of test cases. In this way, the FBD not only supports functional understanding but also strengthens the test case design process by ensuring that critical interactions and dependencies are captured and evaluated under simulated field conditions.

3.1.3. Design Failure Mode and Effects Analysis (DFMEA)

Ensuring high product reliability begins with identifying and validating risks throughout the Product Development Lifecycle (PDLC), with Design Failure Mode and Effects Analysis (DFMEA) serving as the foundational step in this process.

Incorporating DFMEA into test case design is essential for effective risk validation, as studies show that DFMEA-related factors contribute to over 24% of field reliability issues [1]. As a result, DFMEA is considered a critical input for any reliability test planning effort. It involves a systematic evaluation of potential failure modes, their root causes, associated effects, and corresponding mitigation strategies [2,3].

Translating DFMEA findings into test cases allows for the assessment of nearly all identified failure modes within a controlled lab environment. However, it is important to note that certain failure modes or causes may require dedicated testing at the component or sub-assembly level. These tests are considered prerequisites and fall outside the scope of system-level test case conversion. By integrating DFMEA into the test design process, organizations can significantly enhance the accuracy and completeness of reliability validation.

3.2. Legacy Information

3.2.1. Failure Modes

To develop effective and representative test cases, it is essential to analyze field failure data to identify predominant failure modes. This analysis provides direct insight into real-world reliability issues and highlights recurring patterns that may not be evident through design analysis alone. By extracting and categorizing failure modes from field data, engineers can prioritize the most critical risks and ensure that these are systematically incorporated into test case design. This approach strengthens the relevance of system-level testing by aligning laboratory evaluations with actual field performance, thereby improving the accuracy of reliability predictions and reducing the likelihood of unanticipated failures.

In some cases, detailed product-specific data may not be readily available for reliability analysis or test case development. When this occurs, alternative approaches must be employed to ensure comprehensive failure mode coverage. One method involves examining similar or related products to identify common failure modes and patterns that may be applicable. Another approach is to consult established reliability standards and handbooks, such as the NSWC Mechanical Handbook [4], MIL-HDBK-217F [5], and Telcordia SR-332 [6], which provide extensive failure mode data for a wide range of components and systems. These resources serve as valuable references for estimating reliability and constructing representative test cases when direct product data is limited or unavailable.

3.2.2. Failure Trends

When analyzing field failure data for a product, it is essential to evaluate key reliability metrics such as failure occurrence rates, Mean Time to Failure (MTTF), and other performance indicators to inform the development of test case designs. These metrics provide quantitative insight into the frequency and timing of failures, helping to prioritize which failure modes should be addressed in laboratory testing. By incorporating this data into the test case design process, engineers can ensure that the most critical and impactful failure scenarios are accurately represented, thereby enhancing the relevance and effectiveness of system-level reliability validation.

3.3. Internal Study Data

3.3.1. Actual Use Conditions

Actual product use conditions refer to the real-world environments, behaviors, and operational patterns under which a product is deployed and utilized by end users. These conditions encompass a wide range of variables, including usage frequency, load levels, duty cycles, environmental exposure (e.g., temperature, humidity, vibration), and user interaction styles.

Incorporating actual use conditions into test case design is essential for accurately replicating field scenarios and uncovering failure modes that may not surface under ideal or controlled laboratory settings. Aligning test parameters with observed usage data often gathered through field studies, customer feedback, or telemetry engineers can create more representative and effective reliability tests. This approach helps ensure that the product is validated not just for theoretical performance, but for its robustness in the diverse and unpredictable contexts in which it will operate.

3.3.2. Usage Levels and Pattern

Traditional engineering tests conducted at the component and subassembly levels often fail to account for customer usage patterns. These tests typically rely on standardized samples qualified for global platforms, aiming for broad compatibility across multiple product lines. While efficient, this approach frequently overlooks critical aspects of real-world usage, contributing to field reliability issues estimated to account for approximately 12% of failures [1]. To address this gap, customer usage pattern-based testing is essential. It ensures that products are evaluated under conditions that closely mirror actual operational scenarios, thereby enhancing both safety and reliability.

The process of creating test cases based on customer usage patterns as per the paper [1].

When designing test cases, it is crucial to adopt the perspective of the end user rather than that of the engineer. This shift in viewpoint helps ensure that the test cases reflect practical, everyday interactions with the product.

3.3.3. Component Level Test Data

Component test data provides critical insights into the performance, durability, and failure characteristics of individual parts within a system. This data is typically gathered through internal or supplier qualification tests, accelerated life testing, environmental stress screening, and other validation methods conducted at the component level. Incorporating component test data into system-level test case design helps ensure that known weaknesses, tolerance limits, and degradation patterns are accounted for in broader reliability evaluations.

By analyzing metrics such as failure rates, wear-out mechanisms, thermal limits, and electrical thresholds, engineers can design targeted test cases that simulate realistic stress conditions and verify whether the system can tolerate component-level variability. Additionally, component test data supports the identification of high-risk interfaces and dependencies, enabling more precise fault injection and robustness testing. This layered approach strengthens the overall reliability strategy by bridging the gap between isolated component behavior and integrated system performance.

3.4. Interaction Parameters

3.4.1. End User (Operator's Manual)

Developing test actions based on a product's operation manual is a key strategy to ensure that the product can be used safely and effectively by its intended users. These test cases should reflect all essential functions, user interactions, troubleshooting steps, safety procedures, and typical usage scenarios as outlined in the manual. This approach helps validate that the product performs reliably under expected conditions and aligns with user expectations.

To create test cases from an operation manual, follow steps presented in the paper [1].

By grounding test case design in documented operational guidance and adopting the user's perspective, engineers can ensure that reliability testing reflects actual usage and uncovers potential issues that may arise during day-to-day operation.

3.4.2. Service Manual

Designing test actions based on a service manual is essential to validate that maintenance, calibration, repair, and troubleshooting procedures are effective and can be reliably executed by service personnel. This ensures that the product remains functional, safe, and serviceable throughout its lifecycle [1].

To develop such test cases, it is important to thoroughly review the service manual, focusing on the accuracy and completeness of each procedure. This includes verifying that all service tasks can be performed as described, confirming that calibration steps yield correct and consistent results, ensuring that troubleshooting methods lead to accurate fault identification, and validating that repair instructions successfully restore the product to its intended operational state. By incorporating these elements into test case design, engineers can assess the serviceability of the product and reduce the risk of post-deployment failures due to maintenance errors or incomplete procedures.

3.4.3. Consumables

Consumables refer to materials or components that are used up, replaced, or replenished during the normal operation or maintenance of a product such as filters, batteries, lubricants, inks, or cleaning agents. Their performance, compatibility, and replacement cycles can significantly influence overall product reliability. When designing test cases, it is important to account for the role of consumables by simulating their typical usage, degradation over time, and potential variability in quality or sourcing.

Test scenarios should evaluate how the product performs with both standard and suboptimal consumables, assess the impact of delayed or improper replacement, and verify that the product provides clear guidance or safeguards for consumable-related maintenance. Including consumables in reliability testing helps ensure that the product remains functional and safe under realistic operating conditions, especially in cases where end-user behavior or service practices may vary.

3.4.4. Manufacturing Process

The manufacturing process plays a critical role in product reliability and must be considered when designing test cases. Variations introduced during production such as material inconsistencies, assembly tolerances, process deviations, and operator-induced errors can significantly impact product performance and lead to field failures. To account for these risks, test case design should incorporate scenarios that simulate manufacturing-induced variability. This includes evaluating products built under different process conditions, assessing the impact of known production challenges, and validating that the product maintains its intended function across acceptable manufacturing tolerances.

Additionally, insights from process control data, quality audits, and yield trends can help identify high-risk areas that warrant focused testing. By integrating manufacturing process considerations into reliability test planning, engineers can ensure that the product is robust not only in design but also in how it is built ultimately reducing the likelihood of defects and improving field performance.

3.5. Test Case Design Process Flow

The test case design has a five-step process to convert each input parameter to final executable test cases and test matrixes as shown in Figure 6.

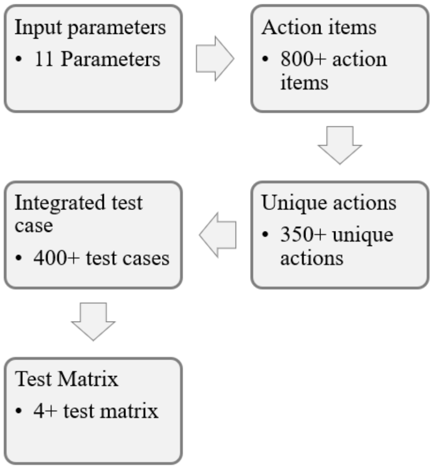


Figure 6: Test Case Design Flowchart

3.5.1. Input Parameters

In the first step, the engineers collect all relevant input parameters available for the product line. The approach is not limited to predefined parameters; it can include any additional inputs that help create test cases representing real-world usage conditions.

Key sources for input parameters include:

- Parameter (P-) Diagram
- Functional Block Diagram (FBD)
- Design Failure Mode and Effects Analysis (DFMEA)
- Customer Usage Patterns and Actual Use Conditions

These are some of the mandatory inputs to create better test case design.

3.5.2. Action Items

Action items serve as the first transition point from input parameters to executable test inputs. In this step, each input parameter is translated into actionable items.

For example, every line item from the DFMEA is analyzed from a testing perspective to identify potential failure modes that can be replicated and validated in a controlled lab environment.

3.5.3. Unique Actions

In this step, all action items are compared to identify uniqueness. Redundant items are removed to create an optimized list of distinct actions. This ensures that only non-overlapping actions progress to the next stage for test case development.

3.5.4. Integrating into Test Cases

Integrating unique action items into test cases involves a two-step process:

- Assigning weightage to input parameters
- Assigning weightage to individual unique actions

Weightage factors are determined by the test case designer based on the problem statement and the product development phase. Typical considerations include:

- If the test focuses on the end-user perspective, higher weightage is given to usage levels, operating conditions, and service scenarios.
- If there is limited information on factor influence, a flat model (equal weightage for all parameters and actions) can be applied.

However, analyzing the problem statement and applying an appropriate weightage model is strongly recommended for accurate and meaningful test design. The weightage model for various input parameters is shown in Table 1.

Table 1: Input Parameters and Weightage Factors

Input Parameter	**Weightage Factor**	**Individual Unique Actions**
Parameter diagram (P-diagram)	15%	Based on mode or cause and its occurrence
Functional Block Diagram (FBD)/Interface matrix (IM)	30%	Based on primary function to secondary function
Design failure mode effective analysis (DFMEA)	30%	Based on occurrence
Failure mode	10%	Based on failure that can happen early

Input Parameter	Weightage Factor	Individual Unique Actions
Failure trend	8%	Based on field failures
Actual use conditions	15%	Higher weightage to most used condition
Usage levels and pattern	15%	Based on higher usage to lower usage
Component level test data	5%	Weightage to early failure one
End user and service	15%	Based on occurrence of event
Consumables	5%	Based on usage pattern
Manufacturing	5%	Based on failure observed

3.5.5. *Test Case and Test Matrix*

Based on the integrated inputs and the required number of test cases, a test matrix is created. The matrix organizes test cases according to the weightage assigned to each input parameter.

For example:

- If the test matrix consists of 200 test cases, input parameters are distributed proportionally based on their weightage.
- Approximately 60 test cases may include DFMEA-derived inputs, while the remaining cases incorporate other parameters such as functional diagrams, usage patterns, and service conditions.
- Many test cases will include multiple input parameters to ensure comprehensive coverage and realistic simulation of usage conditions.

4. Test Case Design

The test case design methodology is particularly effective for system-level testing when employing Reliability Growth Testing (RGT). This approach emphasizes the exploration of input variations, noise factors, and complex interactions among system components. By systematically incorporating these elements, RGT enhances the likelihood of uncovering hidden failure modes and ensures comprehensive validation of system behavior under diverse conditions.

Each test case should be constructed with a minimum of two and a maximum of five distinct objectives defined as discrete input actions or conditions. This constraint balances thoroughness with efficiency. Test cases with fewer than two objectives may lack sufficient complexity to expose interaction-based failure modes, while those exceeding five objectives risk becoming overly convoluted, leading to an exponential increase in the number of test cases required. This not only prolongs the testing cycle but may also dilute the focus, making it harder to isolate and diagnose failure modes.

While multi-objective test cases are beneficial for capturing intricate system behaviors, excessive objectives can introduce noise and reduce the clarity of test outcomes. Therefore, careful calibration of test case complexity is essential to optimize fault detection without compromising test manageability.

The number of test cases and test matrix based on the inputs parameters for multi-functional printer is shown below in Table 2.

Table 2 illustrates structured progression in the system-level test design process, beginning with diverse input parameters such as DFMEA, P-Diagram, Interface Matrix, and others. Each of these inputs contributes to a set of actionable items, which are then distilled into unique actions to each action reflecting a meaningful aspect of system behavior, risk, or usage condition. Once the unique actions are defined, they serve as the foundation for generating the total test case count. These test cases are designed to validate the system against each unique action under various conditions and scenarios. Finally, the test cases are grouped into test matrices by ensuring comprehensive coverage, traceability, and efficiency in system-level validation. Example of test case inputs from the input parameters for a multi-functional printer as shown in Table 3.

Table 2: Test Case Count from Test Inputs for a Multi-Function Printer

Input Parameters	Actionable Items Count	Unique Actions Count	Test Case Count	Test Matrix Count
P-Diagram	100	30	500 to 700	3 to 5 contain 50 to 200 jobs range
FBD/Interface Matrix (IM)	50	20		
DFMEA	500	150		
Failure Mode	22	15		
Failure Trend	5	5		
Actual Use Conditions	12	10		
Usage Level and Pattern	16	16		
Component Level Test Data	5	3		
End User and Service	8	5		
Consumables	20	20		
Manufacturing	4	3		

Table 3: Test Case Input Examples

S. No	Input Parameters	Input from Input Parameters	Action Example	Test Case Input
P-01	P-Diagram	Calibrate variation	Perform calibration during testing by service person	Add one or two test cases in the matrix to include this based on field usage
P-02	FBD/IM	Harness connection to receptacle	Remove and reinsert connector	Test cases based on usage
P-03	DFMEA	Toner spillage during replacement	Create a monitor point to view toner spillage	Test case to view and record this information
P-04	Failure Mode	Two sheet feed by feeder	Test case with A3 paper	Use A3 print or copy jobs
P-05	Failure Trend	Fax job failed during print job	Add concurrent jobs in a matrix	Simulate concurrent jobs condition
P-06	Actual Use Conditions	School facility, Office facility, Commercial	Test case with representing these environment	School conditions are more with one page with more than 30 to 100 copies
P-07	Usage Level and Pattern	1 page print 80%	Use more one-page print job	Create 80% print jobs with one page
P-08	Component Level Test Data	Clutch failure after 50,000 cycles	Record usage condition and perform Preventive Maintenance (PM)	Include Preventive Maintenance (PM) test case
P-09	End User and Service	Cancel print while printing	Give 100-page print and cancel	Add 3 jobs in 300 jobs matrix
P-10	Consumables	Plain, Recycle, color papers	Use different types of consumables	Incorporate
P-11	Manufacturing	Enclosed poor finishing.	Check point during installation	Create installation test case to capture this information

The example test case from the input parameters in Table 3 is illustrated in Table 4.

Table 4: Test Case Design Example

Parameters	Input
Type of job	Print
# of pages	1 (P-07)
Copies	50
Image or text	Text
Original Size	A4
Paper size	A3 (P-04)
Paper type	Recycle (P-10)
Paper tray	Tray 1
Remark	Cancel job after printing 10 pages. (P09)

This example in Table 4 covers four input parameters: P-04, P-07, P-09, and P-10. P-04 primarily addresses multi-feed issues involving A3 paper, which has been identified as a common failure mode in legacy input systems. P-07 focuses on usage levels and patterns, noting that nearly 80% of print jobs are single-page tasks. P-09 considers the end-user perspective, particularly situations where users perceive the output as incorrect and cancel the ongoing job. Lastly, P-10 evaluates consumables from the standpoint of using recycled paper.

4.1. Test Case and Test Matrix Execution Approach

Test execution will be performed using a round-robin method. Each test matrix will be executed a minimum of 3 to 5 times before moving to the next set of test cases. This ensures that all failure modes are identified, and all input parameters are tested multiple times to build confidence in the results. Recommendations for changes to the test matrix should be based on failure modes identified during testing and new learnings from each run. If a test case is not yielding any new insights, it is better to move to the next set of test cases. This saturation point typically occurs after 3 to 5 full runs of each test matrix.

5. Test Case Design Improvement Across Product with Closed Loop to Field failure and Scenarios.

This entire methodology is data-driven and relies on multiple input sources. Each input is associated with several parameters, which are continuously updated based on insights from various sources, such as: Field failure data, Internal test data, Manufacturing data, and Legacy product data.

Initially, test cases are derived from the parameters available at the start of the process. Over time, additional parameters are incorporated as new data becomes available. For future products, all learnings from predecessor products are integrated into the test case design.

When test cases span multiple products, the correlation with actual field performance increases significantly often reaching up to 95%. This iterative refinement ensures that test cases remain relevant, comprehensive, and predictive of real-world conditions.

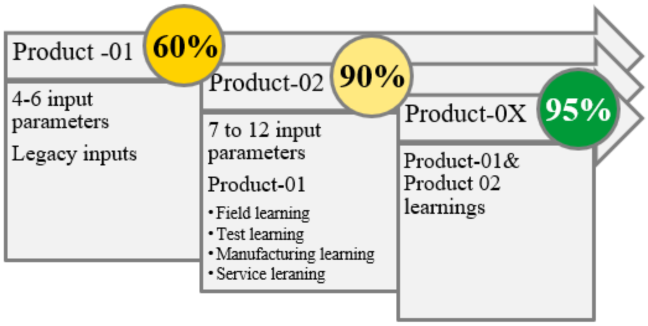


Figure 7: Test Case Design Flowchart

As per Figure 7, In the initial product cycle, testing with limited input parameters achieved approximately 60% correlation between lab-detected failures and field-reported failures. As the number of input parameters increased and learnings from predecessor products were incorporated, the correlation improved significantly reaching up to 95%.

These correlation values are calculated by comparing:

- Number of failures reported in the field

  vs.

- Number of failures reproduced in the test environment

This iterative enhancement demonstrates the value of expanding input sources and leveraging historical data to improve predictive accuracy and test coverage.

Table 5: Field to Test Result Correlation for Product-01 with limited Input Parameters

Failure Mode	Lab testing	Field Failure
Paper jam with Fault code - 153	Yes	Yes
Multi-feed papers	Yes	Yes
Torn paper from feeder	No	Yes
Hole punch option not working	Yes	Yes
Noise from feeder	No	Yes
Door misalignment	Yes	No
Door scratch marks	No	Yes
Toner spillage	Yes	Yes
Failures identified when compared to field	4	7
Overall correlation %	4/7= **57%**	

Table 6: Field to Test Result Correlation for Product-02 with all Input Parameters

Failure Mode	Lab testing	Field Failure
Paper jam with Fault code – 330	Yes	Yes
Paper jam with Fault code – 958	Yes	Yes
Scanner fault	Yes	Yes
User Interface Grey out	Yes	Yes
Toner spillage	Yes	Yes
Hard wrinkles on copies	Yes	Yes
Multi-feed papers	Yes	Yes
Fault code -721	Yes	Yes
Fault-03	No	Yes
Paper jam with Fault code - 153	Yes	Yes
Paper jam with Fault code - 751	Yes	Yes
No fault code paper jam	Yes	Yes
Failures identified when compared to field	11	12
Overall correlation %	11/12= **92%**	

6. Results

The proposed method (Bolded) in Table 7 introduces a more comprehensive and user-centered approach compared to the existing method. While both rely on DFMEA, the proposed method enhances reliability analysis by incorporating FBD/IM, P-Diagram and field failure trends, offering a broader view of real-world issues. It expands legacy information beyond traditional failure modes to include actual usage conditions, patterns, and levels, enriching the data context. Internal study data remains consistent at the component level, but the proposed method adds depth by considering how components perform under real use.

Table 7: Existing Method vs Proposed Method

Method	Reliability analysis	Legacy information	Internal study data	Interaction parameters
Existing method	DFMEA	Failure modes (DFMEA)	Component level test data (DFMEA)	Focused on engineer's point of view
Proposed method	DFMEA **FBD/IM, P-Diagram**	Field failure mode and **trends**	**Actual use conditions Usage levels and patterns.** Component level test data.	**End user and service. Consumable. Manufacturing**

As per the results from Table 5 and Table 6, the data from two different product lines using the existing method and the proposed method indicates that when relevant parameters are incorporated as inputs to test case design, the correction rate increased from 57% to 92%.

If one can go deeper into each failure mode that is not identified in the lab environment, it provides more insight into what is missing in the lab setup. For example, "torn paper from feeder" could occur due to various types of paper entering the feeder types not considered during lab testing, leading to this failure appearing in the field.

Additionally, this approach creates a closed-loop system for improvement, as outlined in Section 5, ensuring continual enhancement by adding new parameters or adjusting existing ones based on information from various sources.

Most notably, interaction parameters shift from an engineer-centric view to a more holistic perspective that includes end users, service teams, consumables, and manufacturing, making the proposed method more aligned with practical, operational realities.

7. Discussion

In the context of system validation and reliability assessment, several challenges arise during the conversion of action items into executable test cases. Not all action items are directly translatable into test scenarios due to their abstract nature or lack of measurable parameters. Furthermore, certain actions demand extensive testing efforts, which can be resource-intensive and time-consuming, especially when replicating complex operational conditions. A significant limitation is the unavailability or insufficiency of input data, which hampers the ability to construct meaningful and representative test cases. This data gap can lead to incomplete coverage and reduced confidence in the test outcomes. This approach will work well for complex systems where have more interaction between systems, subsystems, and end-user.

From a disadvantage standpoint, the overall cost of testing escalates due to the need for specialized setups, prolonged test durations, and iterative validation cycles. Additionally, overly constrained test cases designed with rigid assumptions or narrow boundaries may inadvertently mask or fail to expose latent failure modes, thereby compromising the robustness of the reliability analysis. Corner cases, while essential for stress testing, can sometimes produce failures that distort reliability metrics and trend analyses, leading to misleading interpretations of system performance under typical operating conditions. Although it has some constraints, Return on Investment (ROI) remains key: the high upfront cost of testing is offset by a significant reduction in warranty costs, recalls, and reputation damage.

8. Conclusion

To enhance the robustness and relevance of system-level testing, it is essential to integrate all valuable analysis results into the test case design. The main contribution of this work is not a single tool, but an integrative framework that defines the relationships and information flow between classical reliability tools that are often used in silos. This framework transforms a collection of static analyses into a dynamic process for test design.

This approach ensures comprehensive coverage of critical variables, noise factors, and interaction effects that influence real-world performance. By embedding these multidimensional insights, the test cases become more representative of actual operating conditions, thereby improving the fidelity of simulations and predictive accuracy. Such enriched test case design facilitates strong field-to-test correlation, with potential alignment reaching up to 95%, significantly reducing discrepancies between lab-based evaluations and field behavior. Moreover, this methodology effectively bridges the gap between controlled laboratory environments and dynamic field conditions, enabling the early identification of latent failure modes and enhancing overall reliability metrics.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] R. Lawrance and N. K. R. Gorla, "Test Case Design for the System Level Reliability Testing of a Complex Electro-Mechanical Product+," 2025 Annual Reliability and Maintainability Symposium (RAMS), Destin, FL, USA, 2025, pp. 1-6, https://doi.org/10.1109/RAMS48127.2025.10935027

[2] International Organization for Standardization, "Medical devices – Application of risk management to medical devices", ISO-14971, Geneva, Switzerland: ISO, 2019

[3] AIAG & VDA, "Failure Mode and Effects Analysis", Southfield, MI, USA: Automotive Industry Action Group, 2019.

[4] Naval Surface Warfare Center, "NSWC Handbook of Reliability Prediction Procedures for Mechanical Equipment", Bethesda, MD, USA: Naval Surface Warfare Center, May 2011.

[5] Department of Defense, "MIL-HDBK-217F: Reliability Prediction Procedures of Electronic Equipment", Washington, DC, USA: Department of Defense, Dec. 1991.

[6] Telcordia Technologies Inc., "Telcordia SR-332: Reliability Prediction Procedure for Electronic Equipment", Piscataway, NJ, USA: Telcordia Technologies Inc., Feb. 2016.

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 10, No. 6, 65-76 (2025)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTESJ
ISSN: 2415-6698

Federated Learning with Differential Privacy and Blockchain for Security and Privacy in IoMT A Theoretical Comparison and Review

Shaista Ashraf Farooqi*[1], Aedah Abd Rahman[1], Amna Saad[2]

[1]*Asia e University (AeU) Wisma Subang Jaya, Jalan SS 15/4, Subang Jaya, Malaysia*

[2]*Universiti Kuala Lumpur, Malaysian Institute of Information Technology, 1016 Jalan Sultan Ismail, 50250 Kuala Lumpur, Malaysia*

Email(s): aedah.abdrahman@aeu.edu.my (A. A. Rahman), amna@unikl.edu.my (A. Saad)

*Corresponding Author: Shaista Ashraf Farooqi, Asia e University (AeU) Wisma Subang Jaya, Jalan SS 15/4, Subang Jaya, Malaysia. shaista_ashraf@yahoo.com

ARTICLE INFO

Article history:
Received: 11October, 2025
Revised: 27 November, 2025
Accepted: 29 November, 2025
Online: 15 December, 2025

Keywords:
Internet of Medical Things IoMT
Federated Learning
Differential Privacy
Blockchain
Scalability
Security and Privacy
Decentralized Systems

ABSTRACT

The growing integration of the Internet of Medical Things (IoMT) into healthcare has amplified the need for secure and privacy-preserving artificial intelligence. Federated Learning (FL) has emerged as a pivotal paradigm for decentralized medical data processing; however, it still faces challenges concerning data confidentiality, trust management, and scalability. This review presents an extended theoretical comparison of two prominent privacy-preserving frameworks—Federated Learning with Differential Privacy (FL-DP) and Federated Learning with Blockchain (FL-BC)—to assess their suitability for ensuring data security, transparency, and regulatory compliance in IoMT environments. The FL-DP framework safeguards patient data through noise injection during model updates, offering mathematically proven privacy guarantees. Conversely, the FL-BC framework reinforces trust and integrity via immutable ledgers and consensus mechanisms such as Proof of Stake (PoS) and Byzantine Fault Tolerance (BFT). Reviewing literature published between 2021 and 2025, this study examines trade-offs in privacy, scalability, latency, and energy efficiency, while highlighting emerging hybrid architectures that integrate both approaches. The findings reveal that FL-DP provides stronger privacy control, whereas FL-BC ensures verifiable trust and traceability—together forming the foundation for next-generation secure and trustworthy federated learning systems in IoMT-driven healthcare.

1. Introduction

This paper is an extended version of our earlier work, "*A Theoretical Comparison of Federated Learning with Differential Privacy and Blockchain for Security and Privacy in IoMT,*" originally presented in [1] The conference paper established a foundational comparison between Federated Learning with Differential Privacy (FL-DP) and Federated Learning with Blockchain (FL-BC), emphasizing their roles in improving privacy, trust, and scalability in decentralized healthcare systems.

Building upon that initial study, the present journal version provides a broader theoretical framework, a comprehensive literature review (2021–2025), and a deeper analysis of the privacy–utility–trust trade-offs, scalability, and compliance implications associated with both frameworks. Furthermore, this paper introduces an extended hybrid conceptual model integrating

DP and Blockchain for holistic privacy preservation and distributed trust management in Federated Learning environments.

The growing network of smart medical devices is reshaping healthcare globally by enabling continuous remote monitoring and intelligent diagnostics through connected medical devices. It plays a critical role in predictive medicine by facilitating earlier disease detection, personalized treatments, and improved clinical outcomes. Wearable sensors and embedded devices continuously track vital signs, enabling healthcare providers to make real-time, data-driven interventions.

Additionally, IoMT promotes remote healthcare delivery for patients in underserved or rural areas, reducing hospitalization

https://dx.doi.org/10.25046/aj100606

rates and enhancing access to medical services. According to recent industry projections, the IoMT market size is anticipated to reach USD 188 billion by 2028, highlighting the growing interconnection between clinical systems and intelligent medical devices [2].

Despite its revolutionary potential, IoMT's distributed nature exposes it to severe privacy, security, and scalability challenges. Sensitive patient information traversing heterogeneous networks can be intercepted, manipulated, or exploited, resulting in data breaches and cyberattacks.

In addition, the US Health Insurance Portability and Accountability Act (HIPAA) and the European Union General Data Protection Regulation (EU GDPR) pose significant compliance challenges, particularly concerning the storage, processing, and sharing of healthcare data [3] These challenges underscore the need for advanced, privacy-preserving computational paradigms that can analyze distributed data without compromising confidentiality.

In modern healthcare systems, collaboration across hospitals, laboratories, and wearable devices has become essential for building intelligent models. Through the use of federated learning, these entities can jointly train a shared global model without transferring raw data. This approach keeps sensitive information local, ensuring both data privacy and patient confidentiality [4]. Although federated learning reduces the risks that come with centralized data storage, it also introduces new security challenges. Among these are inference attacks, data poisoning, and model inversion threats. Such attacks may reveal sensitive data hidden in the model updates shared between clients and the server.

To address these limitations, researchers have proposed two major enhancement frameworks:

i. *Federated Learning with Differential Privacy (FL-DP),* which injects mathematically calibrated noise into model gradients or parameters to ensure formal privacy guarantees (ε, δ); and

ii. *Federated Learning with Blockchain (FL-BC),* which leverages distributed consensus, immutability, and cryptographic integrity to eliminate single points of failure and enhance trust among participants [5], [6].

While both frameworks contribute toward improving privacy and security, they differ fundamentally in design, scope, and scalability. FL-DP emphasizes data confidentiality through controlled noise and privacy budgets, whereas FL-BC ensures system integrity and auditability through decentralized verification mechanisms. The selection of one framework over the other depends on application-specific requirements such as latency, computational resources, trust models, and compliance obligations.

This extended paper systematically reviews, compares, and synthesizes these two approaches, presenting a multi-dimensional theoretical analysis that spans privacy guarantees, trust mechanisms, computational efficiency, scalability, and compliance readiness. The main objectives of this extended version are to:

i. Provide a comprehensive review (2021–2025) of existing FL-DP and FL-Blockchain research;

ii. Theoretically evaluate their security, privacy, and scalability trade-offs in distributed learning;

iii. Propose an integrated hybrid architecture that unifies DP's formal privacy mechanisms with Blockchain's decentralized trust model; and

iv. Identify open challenges and research directions for future federated systems in data-sensitive domains.

The subsequent sections are arranged in the following manner. Section II presents a detailed review of literature, summarizing key advances in FL-DP and FL-Blockchain frameworks. Section III outlines the theoretical foundations and comparative analysis model. Section IV discusses hybrid integration opportunities and open research challenges. Section V concludes with future perspectives on developing scalable, compliant, and privacy-preserving federated learning ecosystems.

2. Related Work

The advancement of connected medical technologies has given rise to the Internet of Medical Things (IoMT)—a network of interconnected healthcare devices that continuously collect, transmit, and analyze patient data in real time. This paradigm shift has accelerated innovations in personalized healthcare, remote monitoring, and predictive diagnostics, yet it has also introduced serious challenges in data privacy, system security, and computational scalability [7] The sensitive nature of medical data and the distributed operation of IoMT systems make them particularly vulnerable to data breaches, model poisoning, and inference attacks, prompting extensive research into privacy-preserving and trustworthy machine learning frameworks [8]-[9].

Recent studies have emphasized the need for decentralized learning models that eliminate the reliance on centralized data repositories. To address these issues, federated learning (FL) offers a promising way for multiple participants, such as hospitals, laboratories, and wearable devices—to collaboratively train models while retaining data locally.

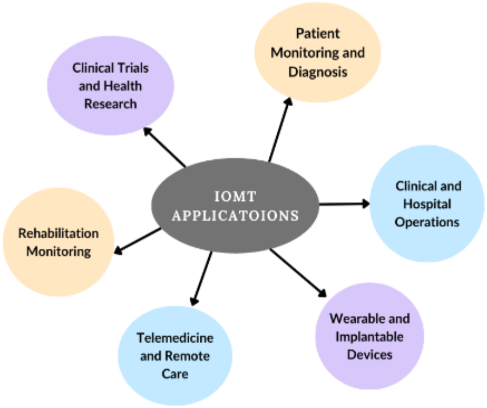

IOMT APPLICATOIONS
Patient Monitoring and Diagnosis
Clinical Trials and Health Research
Clinical and Hospital Operations
Rehabilitation Monitoring
Wearable and Implantable Devices
Telemedicine and Remote Care

Figure 1: Various IoMT Applications

However, despite its potential to enhance privacy and scalability, FL remains vulnerable to gradient leakage, adversarial updates, and trust issues among participating entities [10]-[11]. To address these gaps, advanced frameworks combining Differential Privacy (DP) and Blockchain have been proposed, offering complementary protection mechanisms for secure collaborative learning.

Figure 1 provides an overview of key applications of the Internet of Medical Things (IoMT) across modern healthcare systems.

2.1. Federated Learning (FL)

Federated Learning decentralizes the conventional machine-learning pipeline by allowing multiple clients—such as hospitals, laboratories, or edge gateways—to train local models on private datasets while sharing only encrypted or aggregated parameter updates with a central coordinator [12]. The classical workflow relies on periodic communication rounds in which each participant performs local training, transmits model gradients or weights, and receives the globally aggregated model computed through algorithms such as Federated Averaging (FedAvg) [13].

Although FL mitigates direct data exposure, several vulnerabilities remain. Gradient leakage and model inversion attacks can reconstruct sensitive features from transmitted updates, whereas data poisoning and backdoor manipulation threaten model integrity [14], [15]. Furthermore, heterogeneity in data distribution (non-IID conditions), computational power, and network bandwidth across IoMT devices leads to bias and instability in global convergence [16]. Communication inefficiency is another constraint; frequent synchronization between hundreds of clients increases latency and energy consumption, especially in bandwidth-limited healthcare networks [17]. To alleviate these issues, research trends emphasize secure aggregation protocols, adaptive client participation, and compression-based communication schemes, yet the fundamental trust and confidentiality gap persists—motivating the incorporation of additional cryptographic and decentralized components.

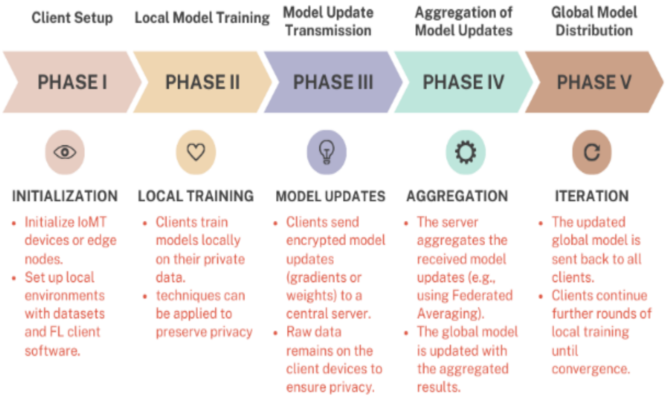

Client Setup
Local Model Training
Model Update Transmission
Aggregation of Model Updates
Global Model Distribution
PHASE I
PHASE II
PHASE III
PHASE IV
PHASE V
INITIALIZATION
• Initialize IoMT devices or edge nodes.
• Set up local environments with datasets and FL client software.
LOCAL TRAINING
• Clients train models locally on their private data.
• techniques can be applied to preserve privacy
MODEL UPDATES
• Clients send encrypted model updates (gradients or weights) to a central server.
• Raw data remains on the client devices to ensure privacy.
AGGREGATION
• The server aggregates the received model updates (e.g., using Federated Averaging).
• The global model is updated with the aggregated results.
ITERATION
• The updated global model is sent back to all clients.
• Clients continue further rounds of local training until convergence.

Figure 2. Phases of the Federated Learning process

Adding Differential Privacy (DP) to the FL framework by injecting controlled noise into model updates protects the anonymity of individual data contributions. The architecture must consider the differences among client devices, which vary in computational power, network bandwidth, and data types, leading to non-IID data and potential delays. Reducing communication overhead is vital for large-scale FL systems in bandwidth-limited IoMT environments, requiring optimized protocols and protection against attacks through methods like Byzantine fault tolerance and possible blockchain integration [18]. Figure 2. illustrates the different phases of the Federated Learning process, from local model training to global aggregation and model redistribution.

2.2. Integration of Differential Privacy within Federated Learning

Differential Privacy (DP) protects sensitive information by injecting carefully calibrated statistical noise into model calculations. A randomized mechanism M satisfies (ε, δ)-DP if for any neighboring datasets D and D′, differing by one record, and for any possible output S,

$$\Pr\,[M(D) \in S] \leq e\ \varepsilon\ \Pr\,[M(D') \in S] + \delta \qquad (1)$$

where ε quantifies the privacy loss and δ bounds the probability of exceeding it [18]. In FL, DP can be applied either locally—each client perturbs its updates before transmission—or globally—noise is added by the aggregator. The local variant offers stronger confidentiality but degrades accuracy more severely.

Between 2021 and 2025, research has focused on optimizing the privacy–utility balance by adaptive noise scheduling, gradient clipping, and sensitivity-based budget allocation [19]. Some works employ input-discriminative local DP to allocate smaller budgets to less-sensitive features and larger budgets to high-risk attributes [20]. Others combine DP with Secure Multi-Party Computation (SMPC) or Homomorphic Encryption (HE) to protect updates during aggregation [21], [22]. Despite these advances, practical deployment faces obstacles:

i. Excessive noise in high-dimensional medical datasets diminishes diagnostic accuracy;
ii. Cumulative privacy loss across multiple rounds complicates budget management; and
iii. Resource-constrained IoMT nodes struggle to perform the additional arithmetic required for DP perturbation.

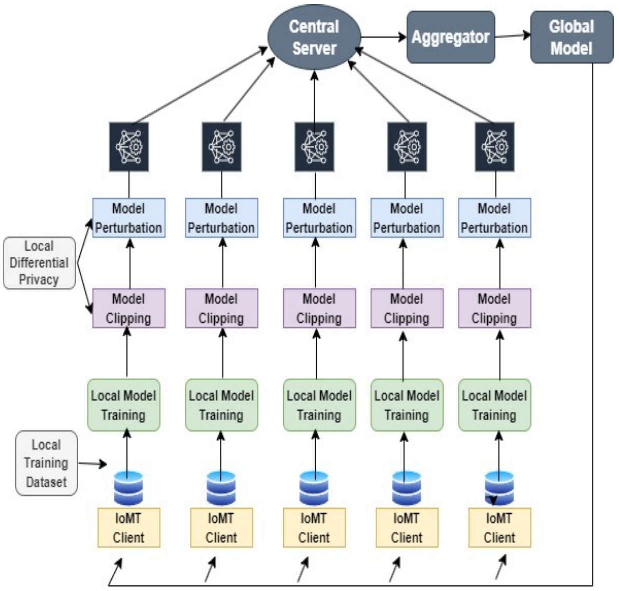


Figure 3. Federated Learning with Differential Privacy (FL-DP) framework

Consequently, while DP provides mathematical confidentiality guarantees, it lacks mechanisms for auditability and distributed consensus, making it complementary—but not sufficient—for end-to-end trust establishment. Figure 3 presents the architectural design of the Federated Learning with Differential Privacy (FL-DP) framework, demonstrating how client-level noise addition and secure aggregation jointly ensure data confidentiality and model utility.

2.3. Blockchain for Secure Federated Learning

Blockchain technology introduces decentralization, transparency, and immutability into collaborative learning environments [23]. By storing each model update as a cryptographically linked transaction, Blockchain eliminates reliance on a single central server and enables tamper-proof logging. Consensus algorithms such as PoW (Proof of Work), PoS (Proof of Stake), DPoS (Delegated Proof of Stake), and BFT (Byzantine Fault Tolerance) validate contributions and ensure ledger consistency [24].

Applied to FL, Blockchain offers multiple security benefits:

i. Integrity Assurance —each update is verifiable and immutable;
ii. Trust Establishment —participants can confirm authenticity without a central authority;
iii. Traceability and Accountability —complete histories of model contributions are maintained; and
iv. Resilience to Single-Point Failures —distributed validation enhances fault tolerance.

Empirical studies have demonstrated that blockchain-enhanced FL architectures can detect tampering and improve reliability in distributed healthcare and industrial networks [25], [26], [27], [28]. However, these benefits are accompanied by significant trade-offs. Consensus formation increases computational cost, latency, and energy demand, which are particularly problematic for lightweight edge or IoMT devices. Moreover, the immutability of Blockchain conflicts with regulatory requirements such as the GDPR Articles 16 and 17 (“right to rectification and erasure”), necessitating auxiliary designs like off-chain storage or ZKPs (zero-knowledge proofs) [29]. Therefore, while Blockchain strengthens integrity and transparency, its direct adoption in real-time federated environments remains constrained by scalability and compliance limitations. Figure 4 provides an overview of the Federated Learning framework integrated with Blockchain technology.

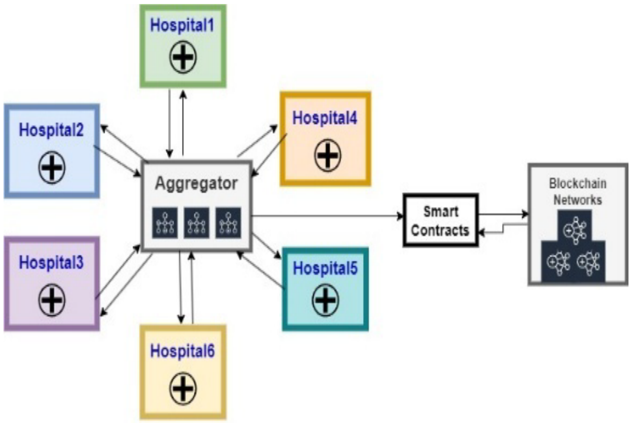


Figure 4. Federated Learning with Blockchain Framework

2.4. Comparative and Hybrid Frameworks

To overcome the individual shortcomings of DP and Blockchain, recent research trends favor hybrid frameworks that exploit their complementary strengths [30], [31], [32]. In these designs, DP provides quantifiable privacy protection at the data or gradient level, whereas Blockchain ensures global trust and tamper resistance at the system level. Model updates are first sanitized through DP mechanisms and then validated and recorded on a Blockchain ledger, establishing dual-layer protection. Such integration enhances resilience against both inference attacks and malicious parameter manipulation.

Nevertheless, existing hybrid approaches often emphasize prototype implementation rather than rigorous theoretical evaluation. Comprehensive comparisons that analyze privacy-utility trade-offs, energy consumption, latency, and regulatory compliance across the two paradigms remain limited. In particular, few studies address how privacy budgets (ε, δ) interact with consensus latency and block-generation frequency—an essential consideration for time-critical medical applications.

2.5. Summary of Research Gaps

The state of the art reveals three persistent deficiencies:

i. Absence of integrated theoretical models that holistically compare FL-DP and FL-Blockchain across multidimensional evaluation metrics including privacy strength, scalability, communication overhead, and compliance readiness.
ii. Limited discussion of governance and regulation, particularly the reconciliation of formal privacy frameworks with legal standards such as HIPAA and GDPR within decentralized architectures.
iii. Inadequate hybridization strategies capable of simultaneously delivering strong privacy guarantees, verifiable trust, and real-time efficiency for heterogeneous IoMT and edge-intelligence ecosystems.

The present paper addresses these gaps by conducting a comprehensive theoretical comparison and a synthesized review of contemporary developments (2021–2025). It further introduces a hybrid conceptual architecture that combines Differential Privacy’s mathematical confidentiality with Blockchain’s decentralized trust management to create secure, transparent, and regulation-compliant federated learning systems.

3. Theoretical Framework and Comparative Analysis

The integration of Differential Privacy (DP) and Blockchain into Federated Learning (FL) creates two distinct but conceptually overlapping frameworks for secure, privacy-preserving distributed intelligence. Both frameworks address critical vulnerabilities of conventional FL but differ in their underlying security models, communication architecture, and computational dynamics. This section presents the theoretical basis of each framework, followed by a multi-dimensional comparison across key metrics such as privacy strength, trust assurance, scalability, latency, and compliance readiness.

3.1. Federated Learning Framework

In a typical Federated Learning system [33], a set of N clients $\{C_1, C_2, \dots, C_N\}$ collaboratively train a global model w coordinated by a central aggregator. Each client C_i holds a local dataset $D_i$and updates the model parameters using gradient descent. The local objective function is:

$$F_i(w) = \frac{1}{|D_i|} \sum_{x_j \in D_i} \mathcal{L}(w; x_j) \tag{2}$$

where $\mathcal{L}(\cdot)$is the loss function.

The global objective across all clients is minimized as:

$$F(w) = \sum_{i=1}^{N} \frac{|D_i|}{|D|} F_i(w) \tag{3}$$

During each communication round t, local clients compute updates:

$$w_i^{t+1} = w^t - \eta \nabla F_i(w^t) \tag{4}$$

where η is the learning rate. The server aggregates these updates via Federated Averaging (FedAvg):

$$w^{t+1} = \sum_{i=1}^{N} \frac{|D_i|}{|D|} w_i^{t+1} \tag{5}$$

While this aggregation preserves data locality, it does not inherently prevent gradient leakage or ensure integrity of updates. To mitigate these risks, privacy and security enhancements through DP and Blockchain have been proposed.

3.2. Differential Privacy-Enhanced Federated Learning (FL-DP)

The FL-DP framework strengthens client privacy by applying differential privacy during gradient transmission or aggregation. The formal definition of (ε, δ)-Differential Privacy [34] ensures that the inclusion or exclusion of any data point minimally affects the model output:

$$\mathrm{Pr}\,[\mathrm{M(D)} \in \mathrm{S}] \leq \mathrm{E}^{\mathrm{E}} \mathrm{Pr}\,[\mathrm{M(D')} \in \mathrm{S}] + \Delta \tag{6}$$

where D and D' differ by one record, M is the randomized mechanism, and $S$is the subset of possible outputs.

In FL-DP, local updates are first clipped to a predefined sensitivity bound $C$to control gradient magnitude:

$$\tilde{g}_i = \frac{g_i}{\max\left(1, \frac{\|g_i\|_2}{C}\right)} \tag{7}$$

and Gaussian noise $\mathcal{N}(0, \sigma^2 C^2)$is added before aggregation:

$$\hat{g}_i = \tilde{g}_i + \mathcal{N}(0, \sigma^2 C^2) \tag{8}$$

The privacy budget ε defines the trade-off between privacy and accuracy — smaller ε yields higher privacy but introduces more noise, reducing model utility [35].

The cumulative privacy loss after Trounds is bounded using composition theorems:

$$\varepsilon_{total} \leq \sqrt{2Tln\,(1/\delta)} \cdot \varepsilon + T\varepsilon(e^{\varepsilon} - 1) \tag{9}$$

Advantages of FL-DP

i. Formal mathematical guarantees of individual data protection.
ii. Compatibility with GDPR and HIPAA standards.
iii. Lightweight computational cost suitable for IoMT and edge devices.

Limitations

i. Gradual degradation of model accuracy with increasing noise.
ii. Privacy budget exhaustion over multiple communication rounds.
iii. Centralized aggregator remains a potential single point of trust failure.

In summary, Differential Privacy fortifies Federated Learning with formal, quantifiable confidentiality guarantees, enabling secure model updates without direct data exposure; however, its reliance on centralized aggregation and sensitivity to noise-induced utility loss necessitate complementary mechanisms—such as Blockchain—to achieve end-to-end trust and integrity in decentralized learning environments.

3.3. Blockchain-Integrated Federated Learning (FL-BC)

The FL-Blockchain framework decentralizes model aggregation by recording each client's model update on a distributed ledger. Rather than relying on a single trusted server, participating nodes validate and reach consensus on the authenticity of model updates before integration [36].

Each local model update w_i^t is converted into a transaction $T_i^t$and broadcast to the network. The transaction includes:

$$T_i^t = (ID_i, h(w_i^t), Sig_i, Time_t) \tag{10}$$

where $h(\cdot)$is the hash function ensuring immutability and $Sig_i$is the client's digital signature.

Blocks $B_k$consist of validated transactions $\{T_i^t\}$and are chained via cryptographic hashes:

$$B_k = \{T_i^t, h(B_{k-1}), Time_k, Sign_{miner}\} \tag{11}$$

Consensus protocols such as Proof of Stake (PoS) or Byzantine Fault Tolerance (BFT) determine the validity of each block. The expected consensus delay $D_c$is approximately proportional to the number of participating nodes $N$and the time per validation τ_v:

$$D_c \propto N \times \tau_v \tag{12}$$

This relationship indicates that as the number of participating nodes or the validation time increases, the overall consensus latency rises proportionally, highlighting the scalability-performance trade-off inherent in blockchain-based federated learning systems [37].

Advantages of FL-Blockchain

i. Decentralized trust model eliminates the single point of failure.
ii. Immutable and auditable transaction history ensures data integrity.
iii. Enables traceability and accountability across federated participants.

Limitations

i. High computational and communication overhead due to consensus.
ii. Energy consumption unsuitable for resource-limited IoMT devices.
iii. Immutability complicates GDPR compliance ("right to erasure").

In conclusion, Blockchain integration enhances Federated Learning by introducing decentralized consensus, immutable record-keeping, and verifiable trust among participants; however, its computational complexity, energy cost, and potential regulatory conflicts underscore the need for hybrid designs that combine Blockchain's integrity assurance with Differential Privacy's formal confidentiality guarantees.

3.4. Layered Architectural Analysis

Since Differential Privacy and Blockchain operate through very different mechanisms within the Federated Learning ecosystem, it is important to analyze how each one contributes to overall system security and efficiency. FL-DP focuses on protecting data confidentiality by adding controlled noise and managing privacy budgets [38], whereas FL-BC ensures decentralized trust and data integrity through consensus validation and immutable ledgers [39]. This analysis examines both frameworks across three key dimensions — the data layer, model layer, and trust layer — to highlight their strengths, limitations, and potential integration points in building a unified privacy-preserving federated learning architecture. Table 1 provides a comparative representation of FL-DP and FL-BC across three key operational layers.

Table 1. Three-layer comparison of FL-DP and FL-BC frameworks

Layer	FL with Differential Privacy (FL-DP)	FL with Blockchain (FL-BC)
Data Layer	Raw data remains localized; Gaussian or Laplace noise is injected into model updates to prevent re-identification.	Raw data remains local; encrypted model updates are converted into transactions recorded on-chain.
Model Layer	Centralized aggregation of noisy gradients using FedAvg; privacy controlled by (ε, δ).	Decentralized aggregation via consensus mechanisms (PoS, BFT); verified by all nodes.
Trust Layer	Relies on trusted central server for update aggregation; vulnerable to insider threats.	Establishes distributed trust through immutable ledgers; resistant to tampering or forgery.

The layered architectural analysis provides a structured view of how Differential Privacy and Blockchain strengthen different components of the Federated Learning pipeline. While the data layer focuses on protecting sensitive information, the model layer ensures collaborative training, and the trust layer governs transparency and accountability. Understanding the distinct functions of each layer helps identify both overlaps and gaps between the two frameworks. Building on this structural perspective, the next section presents a comparative theoretical analysis that quantitatively and qualitatively evaluates FL-DP and FL-BC across multiple performance dimensions, including privacy strength, scalability, communication cost, energy efficiency, and regulatory compliance.

3.5. Comparative Theoretical Analysis

To evaluate the relative merits of Federated Learning with Differential Privacy (FL-DP) and Federated Learning with Blockchain (FL-BC), this subsection presents a systematic theoretical comparison across multiple performance dimensions—including privacy assurance, trust management, scalability, communication efficiency, energy utilization, and regulatory compliance. The analysis integrates mathematical formulations, architectural characteristics, and operational trade-offs to establish a comprehensive understanding of how each framework contributes to secure a privacy-preserving decentralized learning. Table 2 outlines the evaluation criteria employed to distinguish FL-DP from FL-BC across core operational aspects.

Table 2. Comparison criteria between FL-DP and FL-BC

Criterion	Federated Learning with Differential Privacy (FL-DP)	Federated Learning with Blockchain (FL-BC)
Privacy Mechanism	ε-DP with Gaussian or Laplace noise injection; controls exposure through noise scaling.	Cryptographic hashing, digital signatures, and distributed consensus ensure immutability.
Trust Model	Centralized; requires trust in the aggregation server.	Fully decentralized; trust distributed among nodes.
Data Integrity	Protected by central server; vulnerable to tampering if compromised.	Immutable record of updates across all participants.
Scalability	High; lightweight computation, minimal bandwidth requirement.	Limited by consensus latency and block size.
Energy Efficiency	Suitable for IoMT and edge devices.	High energy cost due to validation and block mining.

Compliance	Fully aligned with GDPR and HIPAA through formal privacy guarantees.	Conflicts with GDPR erasure rules; mitigated by off-chain storage.
Latency	Low; depends on communication rounds and privacy noise.	High; determined by consensus time and network delay.
Use Case Suitability	Privacy-critical domains (e.g., personalized healthcare, finance).	Multi-party collaborations requiring verifiable audit trails.

The comparative results highlight that each framework addresses distinct yet complementary aspects of federated system security. FL-DP excels in safeguarding individual-level data through mathematically verifiable privacy guarantees and lightweight implementation, making it ideal for latency-sensitive and resource-limited environments [40]. FL-BC, in contrast, strengthens system-level integrity and accountability through decentralized consensus and immutable audit trails, effectively mitigating insider threats and tampering risks. However, its computational overhead and regulatory challenges limit its scalability in high-frequency or energy-constrained networks [41]. Collectively, these observations suggest that neither framework alone provides a complete solution for secure federated intelligence. Instead, their integration into a hybrid FL–DP–Blockchain architecture can yield a balanced trade-off between privacy, trust, and operational efficiency—laying the foundation for the comprehensive discussion presented in the following section.

4. Discussion

The theoretical comparison of Federated Learning with Differential Privacy (FL-DP) and Federated Learning with Blockchain (FL-BC) frameworks reveals two complementary paradigms, addressing privacy, security, and trust in decentralized machine learning environments. Both frameworks aim to overcome the limitations of conventional centralized training, yet their core operational principles and design objectives differ fundamentally. This section critically discusses their comparative strengths and weaknesses, practical implications, and the prospects of hybrid integration for building trustworthy, scalable federated ecosystems.

4.1. Privacy Protection and Data Confidentiality

The foremost objective of FL-DP is to ensure formal and quantifiable privacy. By injecting calibrated Gaussian or Laplacian noise into gradients, FL-DP guarantees that individual data records remain indistinguishable, even if an adversary has auxiliary information. This mathematical assurance, expressed through the privacy budget (ε, δ), provides a clear theoretical foundation for privacy measurement [42].

However, the noise-utility dilemma persists. Excessive noise reduces model accuracy and can slow convergence, especially in high-dimensional healthcare datasets. Furthermore, managing cumulative privacy loss across multiple communication rounds requires privacy accountants and budget rescaling mechanisms to avoid privacy exhaustion. Despite these limitations, FL-DP aligns naturally with data protection laws such as GDPR and HIPAA, as it inherently supports the “data minimization” and “privacy by design” principles.

In contrast, FL-Blockchain protects privacy indirectly. It ensures transaction-level confidentiality through cryptographic techniques such as hashing, encryption, and zero-knowledge proofs, but it does not conceal the metadata of transactions [43]. Hence, while Blockchain prevents tampering and falsification, it may inadvertently expose usage patterns, communication frequencies, or timestamps. Thus, Blockchain emphasizes system-level trust and transparency, whereas DP provides individual-level privacy guarantees. Both mechanisms are therefore orthogonal yet complementary—one safeguards the “what” (data), the other secures the “how” (process).

4.2. Security, Integrity, and Trust Mechanisms

Security in FL extends beyond data confidentiality—it encompasses model integrity, authentication, and auditability. In FL-DP, security primarily depends on the central server’s ability to enforce privacy budgets and resist aggregation-level attacks. The introduction of secure multi-party computation (SMPC) or homomorphic encryption (HE) can mitigate these threats but increases computational overhead. Moreover, the centralized trust model still presents a single point of vulnerability; if the aggregator is compromised, all participating clients may be exposed [44].

FL-Blockchain, in contrast, transforms this trust paradigm by decentralizing authority. Consensus mechanisms such as Proof of Stake (PoS) and Byzantine Fault Tolerance (BFT) collectively validate model updates, eliminating the need for a trusted intermediary. Each transaction is cryptographically signed, time-stamped, and permanently recorded, making it tamper-evident and auditable [45]. The immutable nature of Blockchain also prevents rollback or version manipulation attacks.

However, this decentralized robustness comes at a cost. Consensus protocols significantly increase latency, energy consumption, and communication complexity, particularly when scaling to hundreds of participating nodes. Moreover, the immutable ledger conflicts with the “right to be forgotten” stipulated in GDPR, necessitating hybrid off-chain or privacy-enhancing designs to maintain legal compliance.

Therefore, while FL-DP ensures controlled privacy leakage, FL-Blockchain guarantees trustless collaboration and tamper-proof data integrity—two distinct security frontiers essential for distributed intelligence [46].

4.3. Scalability and Communication Efficiency

Scalability remains a crucial determinant of framework suitability for real-world deployment. FL-DP exhibits comparatively high scalability because the differential noise addition is a lightweight operation, introducing minimal communication overhead. Clients transmit only perturbed gradients, and the central server performs a simple aggregation step. This efficiency makes FL-DP ideal for IoMT, mobile edge computing, and low-power environments [46].

Conversely, FL-Blockchain's performance degrades as network size increases. Every new model update must be validated by multiple peers, serialized into a block, and propagated through the network.

The consensus delay ($D_c \propto N \times \tau_v$) grows linearly with the number of validators, making it unsuitable for time-critical applications such as emergency monitoring or remote surgery. Proposed optimizations, such as sharding, Layer-2 scaling, and side-chain protocols, alleviate some overhead but add architectural complexity [47].

Hybrid approaches that apply Blockchain selectively—for example, logging only final global updates or model checkpoints—can balance transparency with speed. Additionally, asynchronous aggregation combined with DP-based local privacy can further enhance throughput while maintaining compliance and verifiability.

4.4. Latency, Energy Consumption, and Real-Time Responsiveness

Latency and energy efficiency directly affect the feasibility of FL frameworks in large-scale medical or industrial networks. FL-DP offers lower latency because its operations primarily involve local computation and simple message passing. The most time-consuming process—noise addition—is independent of the number of participating clients. As a result, FL-DP supports real-time applications, such as continuous glucose monitoring or anomaly detection in wearable devices [48].

In contrast, Blockchain's consensus formation introduces significant latency. For example, Proof of Work (PoW)–based networks suffer from mining delays, while PoS and BFT require multiple rounds of communication to reach agreement [49]. This delay not only affects responsiveness but also increases energy consumption, rendering FL-BC less suitable for low-power IoT nodes. Alternative lightweight consensus mechanisms—Proof of Authority (PoA) or Practical Byzantine Fault Tolerance (PBFT)—offer faster confirmation times but may reduce decentralization. Therefore, energy-aware hybrid configurations, where Blockchain operations are delegated to edge gateways or cloud nodes, present a viable compromise.

4.5. Compliance, Auditability, and Governance

Regulatory compliance has become a defining constraint in the deployment of data-driven systems, especially in healthcare and finance. FL-DP aligns naturally with legal frameworks such as HIPAA and GDPR, since differential privacy explicitly prevents re-identification and enables formal privacy accounting. Each operation can be logged and audited using the privacy budget ε, creating a verifiable record of information exposure [50].

FL-Blockchain introduces auditability by design through immutable records. Every model update and transaction is permanently logged, supporting forensic investigation and operational transparency. However, this same immutability challenges compliance with data-subject rights under GDPR Articles 16 and 17. Researchers have proposed using off-chain storage, zero-knowledge proofs (ZKPs), and private or permissioned blockchains to reconcile these conflicts. While these solutions improve compliance, they reduce decentralization, underscoring the trade-off between privacy flexibility and trust transparency [51].

4.6. Hybrid Integration: Toward Unified Privacy and Trust

The limitations of individual frameworks have motivated the emergence of hybrid FL–DP–Blockchain architectures. Such designs seek to achieve dual-layer protection by combining DP's statistical privacy with Blockchain's decentralized auditability.

In a hybrid configuration, each client first applies local differential privacy to sanitize its gradient update. The perturbed model is then encrypted and broadcast as a Blockchain transaction. Consensus nodes validate updates before inclusion in a block, ensuring authenticity and eliminating malicious contributions. Once a sufficient number of updates are validated, the aggregated model is globally updated and redistributed [52], [53], [54].

This integration produces several advantages:

- Formal privacy guarantees at the client level.
- Tamper-proof audit trails across the learning network.
- Elimination of centralized trust dependencies.
- Improved accountability and traceability for compliance audits.

However, hybridization introduces new design challenges. The combination of DP noise, encryption, and consensus overhead can increase computation and communication complexity. Effective deployment thus requires adaptive privacy budgeting, energy-efficient consensus algorithms, and off-chain storage mechanisms to maintain scalability.

4.7. Practical Implications and Domain Suitability

The findings indicate that FL-DP is most effective for privacy-sensitive and latency-critical applications such as personalized healthcare, finance, and telemedicine, where accuracy and confidentiality must coexist [55]. FL-Blockchain, on the other hand, is best suited for multi-institutional collaboration, auditable research, and public data registries, where transparency and tamper resistance outweigh real-time constraints [56].

Hybrid FL–DP–Blockchain models show potential for national health data exchanges, clinical trial collaboration, and inter-hospital machine learning initiatives, where both privacy and decentralized governance are required. By integrating lightweight Blockchain consensus with dynamic DP noise allocation, such systems could deliver trustworthy AI at scale.

4.8. Challenges and Open Research Directions

Despite the significant advancements achieved through Federated Learning with Differential Privacy (FL-DP) and Blockchain-based Federated Learning (FL-BC) frameworks, several unresolved challenges continue to hinder their large-scale adoption in IoMT environments. These open issues highlight the need for further research and technological innovation in the following areas:

i. *Dynamic Privacy Accounting:* Ensuring real-time recalibration of privacy budgets remains a major challenge. As model updates occur across multiple communication rounds, maintaining differential privacy guarantees without causing excessive accuracy loss requires adaptive noise calibration and continuous privacy tracking mechanisms.

ii. *Lightweight Consensus Protocols:* Blockchain integration in IoMT introduces high computational and energy demands. Developing lightweight, energy-efficient consensus mechanisms—optimized for edge and resource-constrained medical devices—is essential to sustain scalability and reduce latency while preserving network integrity.

iii. *Cross-Regulatory Compliance:* Global healthcare systems operate under diverse privacy regulations such as GDPR, HIPAA, and PDPA. Achieving seamless compliance within immutable blockchain frameworks demands interoperable policy layers capable of translating regulatory requirements into auditable smart contracts and metadata governance models.

iv. *Scalable Hybrid Architectures:* Designing scalable hybrid architectures is challenging because FL workflows must integrate with blockchain layers while handling high-volume, distributed healthcare data. Balancing efficient off-chain training with on-chain verification, and ensuring seamless scaling and interoperability across hospitals, edge devices, and cloud systems, adds significant complexity.

v. *Benchmarking and Standardization:* The absence of unified benchmarks and evaluation standards limits cross-framework comparison. Establishing standardized metrics for privacy loss, trust evaluation, latency, and energy consumption would enable consistent assessment and accelerate adoption in real-world healthcare systems.

Addressing these gaps will determine the feasibility of integrating FL-DP and FL-Blockchain into mainstream AI infrastructure for healthcare and other critical domains.

4.9. Summary

The discussion highlights that Federated Learning with Differential Privacy ensures mathematical privacy but lacks distributed verifiability, whereas Federated Learning with Blockchain provides trust and transparency at the expense of scalability and compliance flexibility. Integrating both paradigms within a hybrid FL–DP–Blockchain framework can reconcile these tensions by uniting formal privacy guarantees with distributed trust assurance. Such convergence represents a promising direction for developing next-generation, privacy-preserving, and regulation-aware federated learning architectures capable of powering secure, intelligent systems across diverse application domains.

5. Conclusion and Future Scope

The extended analysis, presented in this paper, provides a comprehensive theoretical comparison of two leading paradigms for secure and privacy-preserving federated learning — Federated Learning with Differential Privacy (FL-DP) and Federated Learning with Blockchain (FL-BC). Both frameworks were evaluated across multiple dimensions including privacy protection, data integrity, scalability, latency, energy efficiency, and regulatory compliance. The findings reveal that while each approach contributes significantly to the security of decentralized learning, their design philosophies and operational priorities differ.

FL-DP offers formal mathematical privacy guarantees through the addition of calibrated noise, ensuring that sensitive individual information cannot be reconstructed or inferred. Its lightweight computational footprint and alignment with GDPR and HIPAA make it highly suitable for latency-sensitive and privacy-critical domains such as personalized healthcare, financial analytics, and mobile edge computing. However, the effectiveness of FL-DP is bounded by the privacy-utility trade-off and the cumulative privacy loss that arises over multiple training rounds.

In contrast, FL-Blockchain emphasizes distributed trust, transparency, and immutability. By replacing centralized aggregation with decentralized consensus, it ensures auditability and tamper resistance across all model updates. Nevertheless, high energy consumption, communication overhead, and conflicts with “right-to-erasure” provisions in regulatory frameworks restrict its scalability and practical deployment in resource-constrained environments.

The comparative synthesis suggests that these two paradigms are complementary rather than competitive. A hybrid FL–DP–Blockchain architecture can unite the statistical rigor of differential privacy with the decentralized trust of blockchain, producing an adaptive, end-to-end secure learning environment. Such integration would enable privacy-preserving model training with verifiable integrity, transparent accountability, and auditable compliance.

5.1. Research Contributions

This extended work makes the following key contributions:

i. Presents a comprehensive theoretical framework unifying privacy, trust, and scalability analysis of FL-DP and FL-BC.
ii. Expands the literature coverage (2021–2025) with systematic evaluation of privacy, communication, and compliance dimensions.
iii. Proposes a layered comparative model (data, model, and trust layers) outlining where DP and Blockchain differ or intersect in the FL ecosystem.
iv. Introduces the conceptual foundation for a hybrid FL–DP–Blockchain architecture, emphasizing privacy-trust co-optimization and regulatory conformity

5.2. Future Research Directions

While the comparative analysis highlights the promising potential of both Federated Learning with Differential Privacy (FL-DP) and Federated Learning with Blockchain (FL-BC) frameworks, several research gaps remain before these paradigms can achieve widespread, real-world implementation in IoMT-driven systems. The following directions outline potential pathways for future exploration:

i. Adaptive Privacy Budgeting

Future research should focus on developing dynamic differential privacy mechanisms that can intelligently adjust the noise scale based on model convergence rates, data sensitivity, and contextual risk factors. Such adaptive strategies would enable a more optimal balance between privacy preservation and model accuracy during iterative training.

ii. Energy-Efficient Consensus Mechanisms

Current blockchain-based consensus algorithms often incur significant energy and latency costs, making them unsuitable for IoMT and edge environments. Designing lightweight consensus mechanisms—such as Proof of Authority (PoA), Directed Acyclic Graph (DAG)-based approaches, or optimized Practical Byzantine Fault Tolerance (PBFT) variants—could drastically improve scalability and operational sustainability in constrained devices.

iii. Hybrid System Prototyping

Empirical evaluation remains limited. Building and benchmarking hybrid FL–DP–Blockchain prototypes using real-world datasets from domains such as healthcare and finance will be essential for quantifying privacy-utility-latency trade-offs, communication overhead, and energy performance under realistic conditions.

iv. Regulatory Alignment

Integrating privacy-enhancing cryptographic primitives—such as Zero-Knowledge Proofs (ZKPs) and Secure Multi-Party Computation (SMPC)—can bridge the gap between blockchain immutability and compliance with privacy regulations like GDPR and HIPAA. Research in this area should focus on developing regulatory-aware frameworks that automate compliance verification within decentralized learning systems.

v. Cross-Domain Interoperability

Ensuring seamless communication between federated ecosystems across domains (e.g., hospitals, smart cities, and autonomous systems) requires the development of standardized APIs, metadata schemas, and secure communication protocols. This would facilitate interoperability and data exchange without compromising privacy guarantees.

vi. Security Auditing and Explainability

The future of privacy-preserving AI depends not only on protection but also on trust. Embedding explainable AI (XAI) mechanisms within federated architectures will allow transparency in privacy-preserving decisions, support security auditing, and enhance user trust by making model behavior interpretable to stakeholders.

5.3. Final Remarks

In conclusion, Federated Learning augmented by Differential Privacy and Blockchain represents a transformative approach to building trustworthy, privacy-preserving, and regulation-aware AI ecosystems. As data sensitivity and regulatory scrutiny continue to rise, integrating these two paradigms offers a pragmatic pathway toward sustainable decentralized intelligence. Future research that focuses on scalable hybrid architectures, adaptive privacy accounting, and lightweight consensus design, will play a pivotal role in realizing next-generation secure Federated Learning frameworks, capable of empowering Healthcare 5.0, smart industries, and beyond.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] S.A. Farooqi, A.A. Rahman, A. Saad, “A Theoretical Comparison of Federated Learning with Differential Privacy and Blockchain for Security and Privacy in IoMT,” in Proceedings of the 2025 19th International Conference on Ubiquitous Information Management and Communication, IMCOM 2025, Institute of Electrical and Electronics Engineers Inc., 2025, doi:10.1109/IMCOM64595.2025.10857505.

[2] A. Said, A. Yahyaoui, T. Abdellatif, HIPAA and GDPR Compliance in IoT Healthcare Systems, 198–209, 2024, doi:10.1007/978-3-031-55729-3_16.

[3] S. Ashraf Farooqi, A. Memon, S. Zamir, K. Malik, W. Batool, H. Zahid, NAVIGATING AI IN THE REAL WORLD: TRANSFORMATIONS, REGULATIONS, AND CHALLENGES.

[4] J. Liu, J. Zhang, M.A. Jan, R. Sun, L. Liu, S. Verma, P. Chatterjee, “A Comprehensive Privacy-Preserving Federated Learning Scheme with Secure Authentication and Aggregation for Internet of Medical Things,” IEEE Journal of Biomedical and Health Informatics, **28**(6), 3282–3292, 2024, doi:10.1109/JBHI.2023.3304361.

[5] M. Ali, H. Karimipour, M. Tariq, “Integration of blockchain and federated learning for Internet of Things: Recent advances and future challenges,” Computers and Security, 108, 102355, 2021, doi: 10.1016/j.cose.2021.102355.

[6] K. Begum, M.A.I. Mozumder, M. Il Joo, H.C. Kim, “BFLIDS: Blockchain-Driven Federated Learning for Intrusion Detection in IoMT Networks,” Sensors, **24**(14), 2024, doi:10.3390/s24144591.

[7] B. Bhushan, A. Kumar, A.K. Agarwal, A. Kumar, P. Bhattacharya, A. Kumar, Towards a Secure and Sustainable Internet of Medical Things (IoMT): Requirements, Design Challenges, Security Techniques, and Future Trends, Sustainability (Switzerland), **15**(7), 2023, doi:10.3390/su15076177.

[8] S.A. Farooqi, Federated Learning for Secure and Resilient AI Systems, IGI Global Scientific Publishing: 307–344, 2025, doi:10.4018/979-8-3373-2200-1.ch010.

[9] M. Rahman, H. Jahankhani, Security Vulnerabilities in Existing Security Mechanisms for IoMT and Potential Solutions for Mitigating Cyber-Attacks, 307–334, 2021, doi:10.1007/978-3-030-72120-6_12.

[10] M. Ali, F. Naeem, M. Tariq, G. Kaddoum, “Federated Learning for Privacy Preservation in Smart Healthcare Systems: A Comprehensive Survey,” IEEE Journal of Biomedical and Health Informatics, **27**(2), 778–789, 2023, doi:10.1109/JBHI.2022.3181823.

[11] S. Rani, A. Kataria, S. Kumar, P. Tiwari, “Federated learning for secure IoMT-applications in smart healthcare systems: A comprehensive review,” Knowledge-Based Systems, 274, 2023, doi:10.1016/j.knosys.2023.110658.

[12] P.M. Mammen, “Federated Learning: Opportunities and Challenges,” 2021.

[13] L. Collins, H. Hassani, A. Mokhtari, S. Shakkottai, FedAvg with Fine Tuning: Local Updates Lead to Representation Learning, 2022.

[14] G. Xia, J. Chen, C. Yu, J. Ma, “Poisoning Attacks in Federated Learning: A Survey,” IEEE Access, 11, 10708–10722, 2023, doi:10.1109/ACCESS.2023.3.

[15] Y. Wan, Y. Qu, W. Ni, Y. Xiang, L. Gao, E. Hossain, “Data and Model Poisoning Backdoor Attacks on Wireless Federated Learning, and the Defense Mechanisms: A Comprehensive Survey,” IEEE Communications

Surveys & Tutorials, **26**(3), 1861–1897, 2024, doi:10.1109/COMST.2024.3361451.

[16] R. Somasundaram, M. Thirugnanam, "Review of security challenges in healthcare internet of things," Wireless Networks, **27**(8), 5503–5509, 2021, doi:10.1007/s11276-020-02340-0.

[17] A.A. El-Saleh, A.M. Sheikh, M.A.M. Albreem, M.S. Honnurvali, "The Internet of Medical Things (IoMT): opportunities and challenges," Wireless Networks, **31**(1), 327–344, 2025, doi:10.1007/s11276-024-03764-8.

[18] Y. Zhao, J. Chen, "A Survey on Differential Privacy for Unstructured Data Content," ACM Computing Surveys, **54**(10 s), 2022, doi:10.1145/3490237.

[19] J. Shi, W. Wan, S. Hu, J. Lu, L. Yu Zhang, "Challenges and Approaches for Mitigating Byzantine Attacks in Federated Learning," in 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE: 139–146, 2022, doi:10.1109/TrustCom56396.2022.00030.

[20] S.A. Farooqi, A.A. Rahman, A. Saad, "Differential Privacy Based Federated Learning Techniques in IoMT: A Review," in 2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM), IEEE: 1–7, 2024, doi:10.1109/IMCOM60618.2024.10418361.

[21] S.P. Sanon, R. Reddy, C. Lipps, H.D. Schotten, "Secure Federated Learning: An Evaluation of Homomorphic Encrypted Network Traffic Prediction," in 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), IEEE: 1–6, 2023, doi:10.1109/CCNC51644.2023.10060116.

[22] O. Dib, S. Li, Z. Li, R. Abdallah, E. hacen Diallo, "FL-SMPC++: A robust framework for privacy-preserving federated learning," Results in Engineering, 28, 107380, 2025, doi:10.1016/J.RINENG.2025.107380.

[23] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, Z. Tari, "Blockchain-Based Federated Learning for Securing Internet of Things: A Comprehensive Survey," ACM Computing Surveys, **55**(9), 2023, doi:10.1145/3560816.

[24] H. Xiong, M. Chen, C. Wu, Y. Zhao, W. Yi, Research on Progress of Blockchain Consensus Algorithm: A Review on Recent Progress of Blockchain Consensus Algorithms, Future Internet, **14**(2), 2022, doi:10.3390/fi14020047.

[25] C. Ma, J. Li, L. Shi, M. Ding, T. Wang, Z. Han, H.V. Poor, "When Federated Learning Meets Blockchain: A New Distributed Learning Paradigm," IEEE Computational Intelligence Magazine, **17**(3), 26–33, 2022, doi:10.1109/MCI.2022.3180932.

[26] Y. Shahsavari, O.A. Dambri, Y. Baseri, A.S. Hafid, D. Makrakis, "Integration of Federated Learning and Blockchain in Healthcare: A Tutorial," 2024.

[27] S.K. Singh, L.T. Yang, J.H. Park, "FusionFedBlock: Fusion of blockchain and federated learning to preserve privacy in industry 5.0," Information Fusion, 90, 233–240, 2023, doi:10.1016/J.INFFUS.2022.09.027.

[28] F. Sun, Z. Diao, "Federated Learning and Blockchain-Enabled Intelligent Manufacturing for Sustainable Energy Production in Industry 4.0," Processes, **11**(5), 2023, doi:10.3390/pr11051482.

[29] S. Gupta, "Zero-Knowledge Proofs For Privacy-Preserving Systems: A Survey Across Blockchain, Identity, And Beyond," Engineering and Technology Journal, **10**(07), 2025, doi:10.47191/etj/v10i07.23.

[30] H.B. Desai, M.S. Ozdayi, M. Kantarcioglu, "BlockFLA: Accountable Federated Learning via Hybrid Blockchain Architecture," in CODASPY 2021 - Proceedings of the 11th ACM Conference on Data and Application Security and Privacy, Association for Computing Machinery, Inc: 101–112, 2021, doi:10.1145/3422337.3447837.

[31] R. Anitha, M. Murugan, "Privacy-preserving collaboration in blockchain-enabled IoT: The synergy of modified homomorphic encryption and federated learning," International Journal of Communication Systems, **37**(18), 2024, doi:10.1002/dac.5955.

[32] H. Xiong, Y. Zhao, Y. Xia, M. Zhang, K.-H. Yeh, "DA-FL: Blockchain Empowered Secure and Private Federated Learning With Anonymous Authentication," IEEE Transactions on Reliability, **74**(4), 5133–5146, 2025, doi:10.1109/TR.2025.3587088.

[33] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, H.V. Poor, "Federated Learning Over Wireless IoT Networks With Optimized Communication and Resources," IEEE Internet of Things Journal, 9(17), 16592–16605, 2022, doi:10.1109/JIOT.2022.3151193.

[34] C. Dwork, Differential Privacy, 1–12, 2006, doi:10.1007/11787006_1.

[35] T. Fukami, T. Murata, K. Niwa, I. Tyou, "DP-Norm: Differential Privacy Primal-Dual Algorithm for Decentralized Federated Learning," IEEE Transactions on Information Forensics and Security, 19, 5783–5797, 2024, doi:10.1109/TIFS.2024.3390993.

[36] F. Ayaz, Z. Sheng, D. Tian, Y.L. Guan, "A Blockchain Based Federated Learning for Message Dissemination in Vehicular Networks," IEEE Transactions on Vehicular Technology, **71**(2), 1927–1940, 2022, doi:10.1109/TVT.2021.3132226.

[37] C. Ying, F. Xia, D.S.L. Wei, X. Yu, Y. Xu, W. Zhang, X. Jiang, H. Jin, Y. Luo, T. Zhang, D. Tao, "BIT-FL: Blockchain-Enabled Incentivized and Secure Federated Learning Framework," IEEE Transactions on Mobile Computing, **24**(2), 1212–1229, 2025, doi:10.1109/TMC.2024.3477616.

[38] S. Feng, M. Mohammady, H. Hong, S. Yan, A. Kundu, B. Wang, Y. Hong, "Harmonizing Differential Privacy Mechanisms for Federated Learning: Boosting Accuracy and Convergence," CODASPY 2025 - Proceedings of the 15th ACM Conference on Data and Application Security and Privacy, 60–71, 2025, doi:10.1145/3714393.3726517.

[39] F. Yu, H. Lin, X. Wang, A. Yassine, M.S. Hossain, "Blockchain-empowered secure federated learning system: Architecture and applications," Computer Communications, 196, 55–65, 2022, doi:10.1016/J.COMCOM.2022.09.008.

[40] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, W. Zhang, "A survey on federated learning: challenges and applications," International Journal of Machine Learning and Cybernetics, **14**(2), 513–535, 2023, doi:10.1007/s13042-022-01647-y.

[41] Z. Cai, J. Chen, Y. Fan, Z. Zheng, K. Li, "Blockchain-Empowered Federated Learning: Benefits, Challenges, and Solutions," IEEE Transactions on Big Data, **11**(5), 2244–2263, 2025, doi:10.1109/TBDATA.2025.3541560.

[42] A. El Ouadrhiri, A. Abdelhadi, "Differential Privacy for Deep and Federated Learning: A Survey," IEEE Access, 10, 22359–22380, 2022, doi:10.1109/ACCESS.2022.3151670.

[43] S. Singh, S. Rathore, O. Alfarraj, A. Tolba, B. Yoon, "A framework for privacy-preservation of IoT healthcare data using Federated Learning and blockchain technology," Future Generation Computer Systems, 129, 380–388, 2022, doi:10.1016/J.FUTURE.2021.11.028.

[44] C. Chen, J. Liu, H. Tan, X. Li, K.I.K. Wang, P. Li, K. Sakurai, D. Dou, "Trustworthy federated learning: privacy, security, and beyond," Knowledge and Information Systems, **67**(3), 2321–2356, 2025, doi:10.1007/s10115-024-02285-2.

[45] J. Liu, C. Chen, Y. Li, L. Sun, Y. Song, J. Zhou, B. Jing, D. Dou, Enhancing trust and privacy in distributed networks: a comprehensive survey on blockchain-based federated learning, Knowledge and Information Systems, **66**(8), 4377–4403, 2024, doi:10.1007/s10115-024-02117-3.

[46] B. Soudan, S. Abbas, A. Kubba, O. Abu Waraga, M. Abu Talib, Q. Nasir, "Scalability and performance evaluation of federated learning frameworks: a comparative analysis," International Journal of Machine Learning and Cybernetics, **16**(5), 3329–3343, 2025, doi:10.1007/s13042-024-02453-4.

[47] A.A. Ahmed, O.O. Alabi, "Secure and Scalable Blockchain-Based Federated Learning for Cryptocurrency Fraud Detection: A Systematic Review," IEEE Access, 12, 102219–102241, 2024, doi:10.1109/ACCESS.2024.3429205.

[48] K. Wei, J. Li, C. Ma, M. Ding, C. Chen, S. Jin, Z. Han, H. Vincent Poor, "Low-Latency Federated Learning over Wireless Channels with Differential Privacy," IEEE Journal on Selected Areas in Communications, **40**(1), 290–307, 2022, doi:10.1109/JSAC.2021.3126052.

[49] S. Otoum, I. Al Ridhawi, H. Mouftah, "A Federated Learning and Blockchain-Enabled Sustainable Energy Trade at the Edge: A Framework for Industry 4.0," IEEE Internet of Things Journal, **10**(4), 3018–3026, 2023, doi:10.1109/JIOT.2022.3140430.

[50] S.R. Chalamala, N.K. Kummari, A.K. Singh, A. Saibewar, K.M. Chalavadi, "Federated learning to comply with data protection regulations," CSI Transactions on ICT, **10**(1), 47–60, 2022, doi:10.1007/s40012-022-00351-0.

[51] M. S, J. K R, "Blockchain-enabled federated learning with edge analytics for secure and efficient electronic health records management," Scientific Reports, **15**(1), 1–20, 2025, doi:10.1038/s41598-025-12225-x.

[52] J. Liu, C. Chen, Y. Li, L. Sun, Y. Song, J. Zhou, B. Jing, D. Dou, Enhancing trust and privacy in distributed networks: a comprehensive survey on blockchain-based federated learning, Knowledge and Information Systems, **66**(8), 4377–4403, 2024, doi:10.1007/s10115-024-02117-3.

[53] A. Hussain, W. Akbar, T. Hussain, A. Kashif Bashir, M.M. Al Dabel, F. Ali, B. Yang, "Ensuring Zero Trust IoT Data Privacy: Differential Privacy in Blockchain Using Federated Learning," IEEE Transactions on Consumer Electronics, **71**(1), 1167–1179, 2025, doi:10.1109/TCE.2024.3444824.

[54] J. Liu, J. Zhang, M.A. Jan, R. Sun, L. Liu, S. Verma, P. Chatterjee, "A Comprehensive Privacy-Preserving Federated Learning Scheme with Secure Authentication and Aggregation for Internet of Medical Things," IEEE Journal of Biomedical and Health Informatics, **28**(6), 3282–3292, 2024, doi:10.1109/JBHI.2023.3304361.

[55] M. Abaoud, M.A. Almuqrin, M.F. Khan, "Advancing Federated Learning Through Novel Mechanism for Privacy Preservation in Healthcare Applications," IEEE Access, 11(August), 83562–83579, 2023, doi:10.1109/ACCESS.2023.3301162.

[56] N.A. Hussein, H.K. Ben Ayed, Blockchain for Smart Cities Management and Security: A Review, 13–40, 2025, doi:10.1007/978-3-031-69441-7_2.

Advances in Science, Technology and Engineering Systems Journal
Vol. 10, No. 6, 77-87 (2025)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTES Journal
ISSN: 2415-6698

PerfVis+: From Timestamps to Insight through Integration of Visual and Statistical Analysis

Marlene Böhmer[*], Thorsten Herfet

Saarland University, Saarland Informatics Campus, Telecommunications Lab, Saarbrücken, 66123, Germany

ARTICLE INFO

Article history:
Received: 15 November, 2025
Revised: 04 December, 2025
Accepted: 06 December, 2025
Online: 23 December, 2025

Keywords:
Network Measurement
Visualization Tools
Timestamp Visualization
Statistical Network Analysis
Protocol Evaluation
5G Performance

ABSTRACT

Complex networked systems provide a cornucopia of network statistics, many of which relate to the temporal behaviour of subsystems, devices, or even individual protocol layers. Different and flexible visualizations can play a crucial role in discovering and making patterns, relations, and trends tangible. We developed PerfVis, a tool that visualizes timestamp data to aid in detecting patterns and changes in the data. Although its measurement capabilities are designed for network transmission measurements, we believe that its analysis part can also be used for other research fields. This paper extends PerfVis from a visualization tool focused on a single type of plot to a comprehensive analyser for timestamp data. It is suitable for evaluating complex systems and analysing protocols. In particular, we integrate statistical output, customizable sending patterns, and inclusion of external data sources, broadening PerfVis's applicability in research and operations. Case studies highlight the benefits these extensions bring and how the tool can help analyse the system behaviour of a real 5G network and the protocol behaviour of the Two-Way Active Measurement Protocol, another established network measurement tool.

1. Introduction

Effective analysis tools accelerate research and development cycles by providing examination from different perspectives. Especially, proper visualizations of data help humans quickly recognise patterns, anomalies, and correlations, and grasp the underlying trends and relationships within the data. PerfVis[1] has been developed as a visualization tool for timing information, initially presented in the 1st International Workshop on Empirical Network Measurements and Methodologies (ENMETH'25) at the 22nd IEEE Consumer Communications & Networking Conference (CCNC) in 2025 [1]. This paper extends this work and presents new features for the tool, supported by case studies from network transmission analysis.

Most network measurement tools primarily provide numerical statistics and lack capabilities to present data visually in a customisable manner, limiting detailed analysis and interpretation. As a visualization tool, PerfVis provides animations of an intuitive 2D timeline visualization presenting timing data. The following recapitulates the visualization idea behind this 2D timeline visualization.

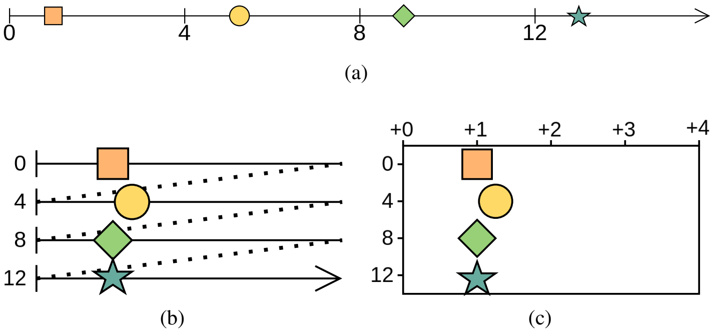
The visualization is based on timestamps and utilises some periodicity i to construct the visual representation. The fundamental visualization concept sorts all timestamps along a timeline (Fig. 1a), segments this timeline (Fig. 1b), and arranges these segments in two dimensions as consecutive rows within images (Fig. 1c). Playing these images in sequence forms a video. The dimensions of each image are determined by the time span of a single row t_{row} and the number of rows per image n_{rows}.

To exploit the full potential of the visualization, it is beneficial to assign a row duration t_{row} to the dominant periodicity, which could correspond to the sending schedule interval or to another periodicity introduced by the underlying system. As PerfVis, by default, uses the sending schedule interval i, the dominant periodicity can be expressed as $t_{row} = ipr \cdot i$, where ipr is the intervals per row, which relates the schedule interval to the dominant periodicity. For short measurements the number of rows n_{rows} can be chosen to fit the full measurement in one 2D timeline visualization, for longer measurements with live visualization it is beneficial to choose it for a comfortable viewing frame rate, otherwise it can be chosen freely according to preference or support of highlighting of evolvements in the measurement.

[*] Corresponding Author: Marlene Böhmer, Telecommunications Lab, Saarland University, Campus Building C6 3, 66123 Saarbrücken, Germany, boehmer@cs.uni-saarland.de

[1] https://git.nt.uni-saarland.de/research-projects/independent/performance-visualization/-/tree/astesj25?ref_type=tags

https://dx.doi.org/10.25046/aj100607



(a)

(b)

(c)

Figure 1: Construction principle of a PerfVis 2D timeline visualization. A one-dimensional timeline (a) is split into parts and arranged as rows in the second dimension (b). The final plot (c) omits the timeline itself and instead shows row time on the x axis, adding to the accumulated time on the y axis. This visualization highlights periodic events and deviations from the periodicity. Combining a sequence of 2D timeline plots (c) to a video adds another dimension. The different time scales represented in rows, 2D timeline plots, and a sequence of those plots allow observations over multiple orders of magnitude.

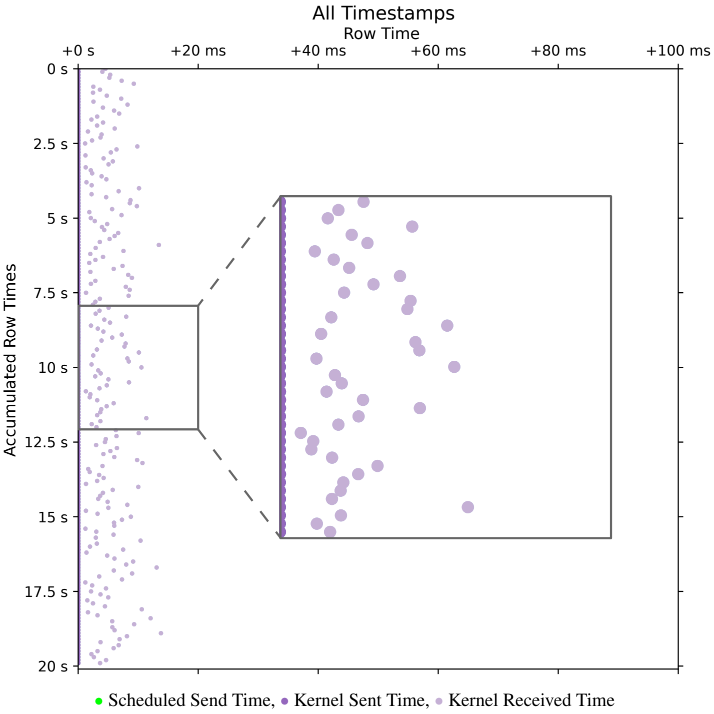


Figure 2: PerfVis 2D timeline visualization of a PerfVis measurement configured with a 100 ms packet interval over a Wi-Fi 5 (802.11ac) link on which the sending device was contending with one other device. The plot uses 100 ms as row time and presents timestamps captured by the Linux kernel on the sender and receiver sides. The equality of packet interval and row time results in a scheduled event at the beginning of each row. The sending side timestamps are so close to their scheduled times that they overlap. This plot allows judgment that sending adheres closely to the 100 ms schedule and provides an impression of the delay and jitter by considering the distance between sent and received events and the variance of the received events.

Fig. 2 shows an example of a PerfVis 2D timeline visualization of a network transmission measurement with timestamps collected at the sender and receiver sides. The row time is chosen to be equal to the packet interval of the measurement, which is 100 ms. As the sending timestamps closely adhere to the proposed sending schedule, they lie on a vertical line at the very left of the plot and cover up the green scheduled send timestamps. The receiving timestamps, however, are influenced by the delay and jitter of the transmission scatter to the right of the sending timestamps. Very beneficial for interpretability is the fact that the packet interval and correspondingly the row time are much larger than the actual delay between the sending and receiving timestamps. This means that all timestamps belonging to a single measurement packet fall into the same row, and no timestamps from other packets appear in that row. It should be noted that this assumption breaks when timestamps from the sending and receiving sides are taken from unsynchronised clocks.

The strength of the PerfVis 2D timeline visualization lies in its ability to exploit periodicity, making it easy to identify deviations from this periodicity visually. However, PerfVis does not care where this periodicity originates. It might originate from the sending pattern, but it might also stem from an underlying system, such as the fixed time structure of a 5G system. PerfVis distinguishes itself from the most commonly used network measurement tools, such as Ping or the Two-Way Active Measurement Protocol, by considering individual timestamps directly, rather than relying solely on aggregated metrics like delay. This granular approach enables detailed visualization and analysis of the temporal behaviour of network events, enhancing insight into network performance beyond conventional summary statistics.

Nevertheless, while having a view of the individual timestamps is important, retaining the statistics remains essential. This is already demonstrated with the example in Fig. 2, where knowing the delay is an important ingredient for interpreting the visualization. Therefore, PerfVis has been extended with an additional analysis module. The statistics module now complements the existing visualization module, which produces the aforementioned visualization. The summary statistics of delay and jitter metrics generated by the statistics module independently provide valuable qualitative and quantitative insights for assessment, reporting, and comparison. However, when combined with the visualization module, these statistics enable even more profound understanding by allowing users to explore data comprehensively and adding information beyond the limits of the visualization. The statistics module is the key stepping stone to gaining more profound insights.

To summarise, PerfVis introduces a novel analysis capability by directly considering individual timestamps of events and flexibly visualising them, in contrast to established tools such as Ping, TWAMP, and OWAMP, which report aggregated metrics like RTT, one-way delay, and jitter and thereby constrain the observable phenomena. The 2D timeline visualization innovatively combines linear and circular timeline layouts, preserving the ability to judge absolute time while emphasising periodic structures that become immediately visible as geometric shapes. This periodic folding of time enables a unique multi-scale perspective that supports concurrent inspection of behaviour over multiple orders of magnitude, from fine-grained jitter up to long-term trends. The extended statistics module further strengthens the approach by providing established delay and jitter statistics that enrich the visualization with quantitative context, forming an analysis loop in which statistics guide how to configure and inspect the visualization. At the same time, visually discovered patterns can be quantified and validated within the same tool, thus enabling advanced diagnostic tasks that are impractical with existing tools, such as visually locating rare effects tied to specific phases of a periodic schedule and immediately assessing their quantitative impact on delay and jitter. Further, PerfVis has been extended for more flexibility in three other aspects: (1) It provides scripts to include timestamp data from specific external timestamp sources to be integrated into the PerfVis analysis modules. This allows for new viewing points, and the scripts serve as examples for including even more sources. (2) The PerfVis measurement modules support more flexible and customisable sending patterns for their network transmission measurements. Given the diversity of application transmission patterns, accurately modelling sending patterns in measurement tools is critical when examining specific network behaviours of traffic. (3) The PerfVis visualizer module has been equipped with the ability to align timestamps in the visualization to focus the analysis on relations between timestamps that belong together. Taken together, these properties make PerfVis not just another latency-plotting tool but a flexible and extensible analysis environment that delivers insights into timing behaviour and periodic structures that prior network measurement tools cannot.

2. Related Work

A rich variety of research exists on tools and methodologies for measuring network transmissions. Ping is probably the most established and fundamental network diagnostic tool, measuring round-trip time to provide baseline latency and packet loss metrics and is widely used for network performance assessment. Although it has been around for a long time, it is still widely used, not only for debugging but also for latency measurement purposes, e.g. in [2]. Other established measurement protocols, such as the Two-Way Active Measurement Protocol (TWAMP) [3], provide round-trip time (RTT) metrics, while the One-Way Active Measurement Protocol (OWAMP) [4] focuses on one-way latency measurements. These protocols serve as foundational latency measurement tools but do not inherently support comprehensive visualization or multi-point timing analysis, which are required for in-depth analysis of complex modern networks. The insights from longitudinal 5G performance studies [5] motivate our approach and raise awareness that the methodology and tool for measurement itself have a significant impact on the measurement result.

Moreover, with some of the new generation networks being time-slotted systems, there are even more challenges for network measurements to gain a comprehensive understanding of their behaviour [6]. Although dedicated measurement equipment and tools are available for some technologies, such as 5G, they are often too expensive for small labs or have limited and non-extensible visualization and analysis capabilities. PerfVis provides an open-source implementation and includes the inclusion of external data captured by tools existing at the lab or other open-source tools, e.g., packet captures that contain timestamp information.

Increasing the number of measurement points along a transmission path enables finer-grained latency and jitter characterization [7, 8], supporting cross-layer analyses that are essential for modern heterogeneous networks. Our tool's measurement does not capture more than one interface and not more than two layers of timestamps, but our extension integrates these ideas by enabling the ingestion of external data for analysis, thus allowing for both multi-layer and multi-point views. Hardware-based fine-grained timestamping solutions, such as P4STA [9], could complement visualization tools by improving the accuracy of timestamp capture, thereby providing a more precise foundation for analysis.

The extension of flexible sending patterns also for measurement tools is not solely motivated by static effects like buffers or scheduling algorithms that show different performance for distinct traffic patterns, but also by the fact that the implementation of fulfilling quality of service requirements is becoming more dynamic when conducted with the help of traffic classification [10, 11].

Visualization of timing data has been explored extensively in various domains. Foundational concepts in *Visualization of Time-Oriented Data* [12] emphasise the importance of multi-scale temporal perspectives and dynamic event representation. These principles are extended in our work to enable intuitive visualization, which can be categorised as a combination of linear and cyclic timeline arrangements, and the animation embeds the time aspect into the physical time as an additional dimension. Effective visualization of network timings facilitates more efficient analysis by leveraging human pattern recognition capabilities [13], enabling rapid identification of anomalies and correlation of temporal behaviours across protocol layers. Different visualizations for timing and latencies are imaginable. For timing and timestamps, timelines are a straightforward approach also used for PerfVis and examined in [14]. For latencies, PerfVis relies on a combination of conservative plots. An interesting approach, utilising heatmaps, is presented in [15]. Live visualization and interaction further aid exploration of complex timing relations, as demonstrated in prior work [16, 12].

The metrics used by our statistics module are basic delay and jitter metrics. RFC 7679 [17] introduces the one-way delay metric with general issues regarding time and corresponding definitions of clock uncertainties. RFC 5481 [18] defines the packet delay variation metrics, and the jitter metric used is defined in RFC 3550 [19]. While [20] argues that aggregate metrics in a coarser resolution might be sufficient for an ISP, deep inspection of single traces requires a more fine-grained approach.

3. Tool Extensions

PerfVis is designed as a flexible and extensible platform for providing intuitive visual and, by now, also statistical insights into timing structures. By supporting the visualization of timestamps, delays, and jitter, PerfVis enables comprehensive analysis. Coming from the field of network measurements, PerfVis provides means to capture network transmission measurements across protocol layers, but the visualization can also be fed with other timestamp data. Before detailing the extensions to the PerfVis modules, the following subsection recaps the data that PerfVis works with and introduces the additional data sources that can be included.

3.1. External Timestamp Sources

3.1.1. Timestamp Data

PerfVis uses timestamps to create its visualization and statistics. In principle, timestamps can originate from different layers and devices and may have distinct timing references. More timestamps collected along the packet's path provide a more comprehensive view of its network transmission in terms of timing. Timestamps should be consistent so that computing differences between them is valid. To calculate delay and jitter metrics, at least two timestamps, thus a pair, per packet have to be available. Nevertheless, the base data remains the individual timestamps, not the aggregated metrics.

PerfVis measurements record timestamps from both the sending and receiving ends of transmissions, thus collecting pairs of timestamps. The PerfVis visualization and statistics capabilities currently operate on a set of up to three pairs of timestamps. Provided that all timestamps are mutually consistent, external timestamps can be used freely, either alone or alongside natively captured ones. When this requirement cannot be guaranteed, subsets of timestamps that fulfil the consistency criterion may be used for separate visualization.

3.1.2. Integration Concept

External timestamp sources enable the analysis of PerfVis measurements from a different perspective, such as different nodes than the end nodes that run the PerfVis sender and receiver. Timestamp sources might be intermediate nodes on the transmission path that create packet captures; thus, with one measurement, different sections of the path can be investigated. In case measurements from a different network layer are available from an external tool, these could enable further cross-layer diagnosis. Another, maybe even more interesting, external timestamp source is traffic from a different network measurement tool or application.

The investigation of external timestamp sources with the PerfVis analysis modules relies on transforming the external data into a format that they can process. To support this integration, external data must have a periodic structure to exploit the PerfVis visualizer's potential fully and contain identifiable pairs of related timestamps, typically identified by sequence numbers, to calculate the statistics.

3.1.3. Integration Implementation

The PerfVis analysis modules read timestamps from NPZ files, which are simply NumPy[2] structured arrays saved using the corresponding save function. This means that simple Python scripts that arrange external data as NumPy structured arrays and save them for PerfVis are sufficient to integrate external timestamp sources. The structure that the arrays need to have can be derived from the PerfVis implementation and the example conversion scripts provided.

At the time of writing, two transformation scripts are provided. The first transforms PCAP files from PerfVis measurements into input files for the visualizer. The packet capture that produces the PCAP file can originate from any intermediate node along the path of the PerfVis measurement. The capture does not have to contain all timestamps that form the timestamp pairs; it can also contain only one set of timestamps to be finally combined with the original output of the PerfVis sender or receiver.

The second script transforms a PCAP file of a TWAMP measurement into input files for the visualizer. As TWAMP does round-trip measurements, an intuitive approach is to take the sending and receiving timestamps on the device that runs the TWAMP sender. If only one way should be analysed, this is also possible by taking packet captures on the device of the TWAMP sender and the TWAMP reflector. These are two examples, but also other external data, possibly not even stemming from network measurements, can be transformed similarly.

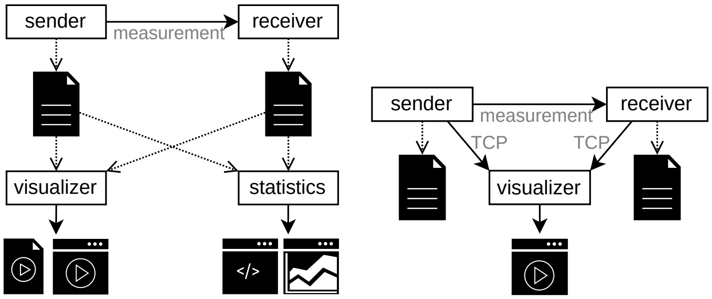
3.2. PerfVis Modules

PerfVis is implemented in Python and is organised into modular components supporting multiple operational modes. The measurement modules, sender and receiver, are responsible for collecting timestamps through one-way network transmissions. The analysis modules include the visualizer and the statistics module. PerfVis supports both intermittent measurement mode (Fig. 3a) and live measurement mode (Fig. 3b), enabling a range of experimental and analytical workflows. The module updates were implemented as part of a Bachelor thesis, demonstrating PerfVis's extensibility for research projects[3].

3.2.1. Sending Schedules in Measurement Modules

The sender and receiver modules send measurement packets according to some predefined schedule and record timestamps for each packet at their side. The most straightforward schedule is the periodic sending pattern, which uniformly spaces packets by a configured interval, as demonstrated in the WiFi example in Fig. 2. It is important to note that executing a schedule should not depend on the processing time. Thus, events should be triggered by a clock, not just by the time between events, as this will nearly always accumulate errors and cause increasing deviation from the schedule.

The PerfVis sender module has been extended to provide a broader range of predefined schedules for transmitting measurement packets. Because the 2D timeline visualization in PerfVis relies on periodicity to develop its full potential, all supported sending patterns are repeated periodically. The new sending schedules allow the adaptation of sending patterns to simulate application traffic or test network behaviour under different burst sizes within a single measurement. The following introduces the new sending patterns and illustrates each with an example.



(a)

(b)

Figure 3: Operation modes of PerfVis. (a) In intermitted mode, sender and receiver save their measurement data to a file; afterwards, the visualizer and statistics modules can read the files. The visualizer can provide a video or an animation; the statistics module can provide numerical commandline output or statistics graphs. (b) In live mode, sender and receiver send the measurement data directly to the visualizer, which views an animation.

[2] https://numpy.org

[3] Bachelor thesis supervised by the authors: Felix Weber, 'Enhancing PerfVis for Comprehensive Network Analysis', Saarland University, 2025

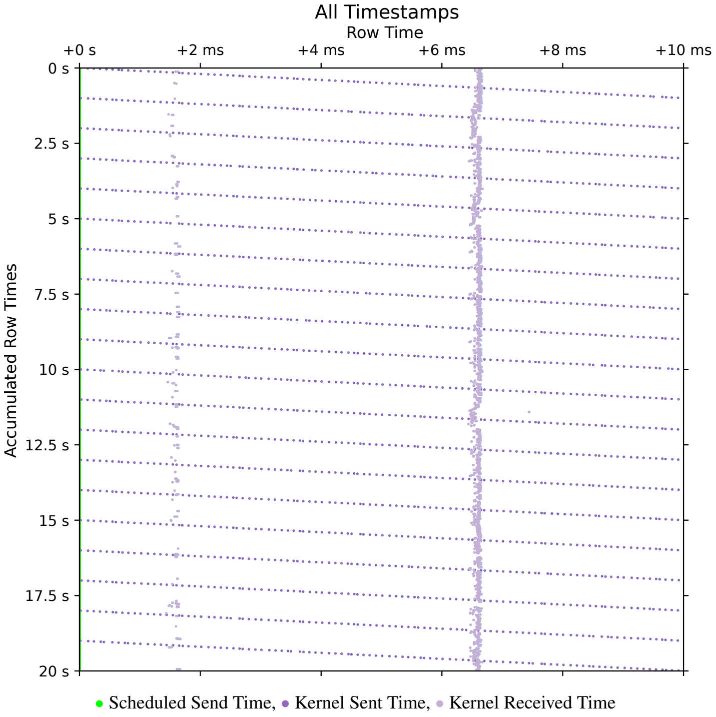


Figure 4: PerfVis 2D timeline visualization of a PerfVis measurement over a 5G link and configured with a sending pattern of 10 ms periodicity and an increasing shift of 0.1 ms. The sending pattern systematically sweeps 100 points in a row, which makes it evident that the receiving side timestamps, accumulating into vertical lines, clearly show the influence of the underlying 5G link with its 10 ms frame structure.

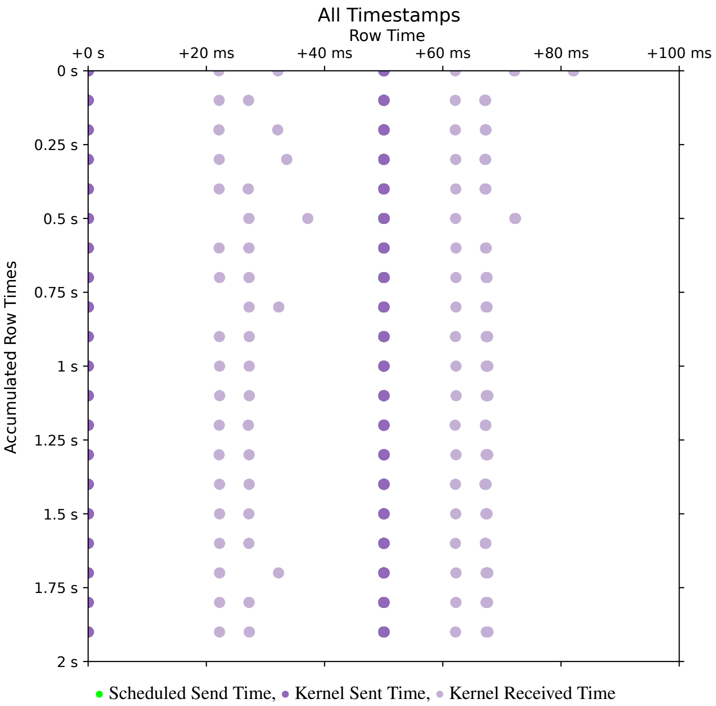


Figure 5: PerfVis 2D timeline visualization of a PerfVis measurement over a 5G link and configured with a sending pattern of one burst every 50 ms alternating between 5 and 20 packets per burst. The visualization with 100 ms row time shows two bursts per row: 5 packet bursts on the left and 20 on the right. It reveals that the 5-packet bursts experience a larger delay than the 20-packet bursts, and that both are transmitted using two transmission opportunities as they accumulate at two receive points per burst.

Modified periodic schedule The simple periodic sending pattern can be modified by adding linearly increasing or random temporal shifts within the configured packet interval. The linear shifting implementation is motivated by the need to emulate precise schedules with small packet intervals (e.g., Figure 8 in [1]) that may otherwise overwhelm network paths or packet capture capabilities. Linear shifts allow hitting specific schedule points at larger intervals, bypassing these problems. Random shifts might be helpful if random behaviour is to be tested or structures are not yet fully known.

Fig. 4 shows an example where sending timestamps are linearly shifted by 0.1 ms per 10 ms interval. The display of scheduled send times in green at the very left of the plot does not account for the shift but only the interval. The actual send times, however, are shifted and thus appear as tilted lines. The actual shift can also be derived from the angle of the tilt. A new tilted line starts every 1 s, which can be seen on the y-ordinate. With the packet interval of 10 ms, this leads to 100 measurement packets per second and as the shift ranges over the 10 ms on the y-ordinate in that time of 1 s, the shift can be calculated as the range devided by the number of measurement points, which results in $\frac{10\text{ms}}{100} = 0.1\text{ms}$. Due to the 5G TDD pattern, packet reception clusters into two occasions every 10 ms, appearing as two vertical lines in the visualization.

Bursty schedule Bursty sending patterns alter the simple periodic pattern by transmitting different numbers of packets in the configured interval. It is implemented by accepting arrays indicating packet counts for succeeding intervals, including the possibility of 0 packets in an interval. With this, quite complex sending patterns can be realised.

Fig. 5 illustrates a burst pattern with two bursts every 50 ms over the same 5G link as the measurement for Fig. 4. The first burst contains 5 packets and appears at the left of the plot, the second burst contains 20 packets and appears in the middle of the plot. With this, the network behaviour under bursts can be tested. In this example, the 5G network uses two transmission opportunities to transmit the 5 packet burst, but also requires only two transmission opportunities for the 20 packet burst, as indicated by the two points at which the receive timestamps accumulate after each burst transmission. To extract this conclusion from Fig. 5, the maximum delay being less than 50 ms is required as additional information. Actually, it is 37.1050 ms here, which can be extracted from the statistics module introduced in Section 3.2.3.

3.2.2. Alignment in the Visualizer Module

The task of the visualizer module is to create animations or videos of the 2D timeline visualization following the visualization concept illustrated in Fig. 1c. The visualizer module has been extended by an alignment functionality that enables artificially shifting some timestamps to a fixed schedule. If different timestamps have been collected for each packet, a single collection point is chosen for alignment. The timestamps from this chosen collection point then determine the shift for all timestamps of the corresponding packet, ensuring that the relations between timestamps for one packet remain unchanged after the alignment. Alignment helps in judging relations between timestamps and is showcased in Section 4.2.

3.2.3. Statistics Module

Although visualization of timestamps provides valuable insights, combining statistics with the visualization empowers users to move beyond pattern recognition and gain even deeper insights. The capability of providing statistical output also makes PerfVis suitable for a broader range of use cases, including protocol evaluation, performance comparisons, and operational troubleshooting. The statistics output allows PerfVis to be used as a simple alternative to other statistical analysis tools, but offers the option to visualize results, e.g., in cases of unexpected statistical values. The statistics output further provides a quantitative basis for result reporting and enables objective comparisons both across different measurements and between distinct measurement tools.

The new PerfVis statistics module calculates key metrics for each packet, namely delay, packet delay variation, inter-packet delay variation, and jitter. The calculation is based on a pair of timestamps for each packet and yields a stream of values that can be further processed. For a measurement, multiple streams can be calculated for different metrics and different pairs of timestamps, if more than one pair is available.

The statistics module provides two options for processing streams: printing a numerical summary of streams to the command line or presenting a stream graphically in a plot showing the metrics' progression over the course of the measurement and the distribution. At the time of writing, statistics output is not implemented for the live visualization of ongoing measurements, but only for completed measurements. The following provides a brief introduction to the metrics and the module's two outputs, each with an example.

Metrics As expected, the delay is the time between two timestamps. In our case, most of the time, that will be the time it takes for a transmission to travel between the sending and receiving nodes on the same network layer. When t_1 and t_2 are the first and second timestamps captured for a packet, then the delay is the difference between these two timestamps.

$$D = t_2 - t_1$$

For further analysis, the delay is not restricted to the timestamps on the sending and receiving sides at the same layer. However, it can also be any combination of two timestamps for a packet, thus providing insight into processing delays from one network layer to another, as long as respective timestamps are available.

The packet delay variation (PDV) is a measure of how much the delay differs from a reference delay D_{ref}.

$$PDV(i) = D(i) - D_{ref}$$

Usually, the reference delay is chosen as the minimum delay that occurs in a measurement $D_{ref} = D_{min}$ as defined in [18]. This is also the default case for PerfVis, although the reference delay can be set differently if desired.

The inter-packet delay variation (IPDV) is a measure of the difference between the delay of a packet and the delay of the previous packet in a sequence.

$$IPDV(i) = D(i) - D(i-1)$$

As noted in [18], the mean should usually be zero, but could deviate from zero in case of delay change over time or clock drift in case timestamps stem from different system clocks.

Different formulas are specified under the jitter term; we are using the definition of the Real-time Transport Protocol (RTP) [19].

$$J(i) = \alpha \cdot |IPDV(i)| + (1 - \alpha) \cdot J(i-1)$$

The RTP RFC sets $\alpha = \frac{1}{16}$. PerfVis also uses this as a default, but allows customisation of both α and the jitter value to start with for the first packet.

Commanline Statistics Summary To summarise a set of statistics streams for a measurement, a set of the following functions can be chosen: min, max, mean, median, 95th percentile, 99th percentile, standard deviation, span, and last value, and the results are printed to the command line in a table.

Commandline Output 1 shows the statistics of the Wi-Fi example from Fig. 2. In addition to the statistics for the kernel layer, it lists the statistics for the Python application layer that are not visible in Fig. 2 because the kernel layer timestamps occlude the application layer timestamps, as their time difference is too small to be visible.

Statistics Plots The statistics plots that visualize a single stream of a specific metric. These plots can provide further insights into development over time or distribution. Apart from the value plot in the centre, a histogram on the right illustrates the distribution, and boxplots at the bottom give insight into how the distribution changes over time.

```
Statistics for the full transmission in ms
200 packets sent at 160.8 Bytes/s - 200 received at 160.8 Bytes/s
with 0.0 percent loss (0 lost).
+---------------------+---------------------+---------------------------------------+
|        sort0        |        sort1        |                 DELAY                 |
|                     |                     |   MIN   |  MEAN   |   MAX   |   STD   |
+---------------------+---------------------+---------------------------------------+
|Python Sent          |Python Received      | 1.1349  | 5.0051  | 13.7789 | 2.7554  |
|Kernel Sent          |Kernel Received      | 1.1287  | 4.9972  | 13.7723 | 2.7555  |
+---------------------+---------------------+---------------------------------------+
|        sort0        |        sort1        |                  PDV                  |
|                     |                     |   MIN   |  MEAN   |   MAX   |  LAST   |
+---------------------+---------------------+---------------------------------------+
|Python Sent          |Python Received      | 0.0000  | 3.8701  | 12.6439 | 2.4969  |
|Kernel Sent          |Kernel Received      | 0.0000  | 3.8684  | 12.6436 | 2.4948  |
+---------------------+---------------------+---------------------------------------+
|        sort0        |        sort1        |                 IPDV                  |
|                     |                     |   MIN   |  MEAN   |   MAX   |  LAST   |
+---------------------+---------------------+---------------------------------------+
|Python Sent          |Python Received      |-10.5464 | -0.0045 | 9.8375  | -0.9985 |
|Kernel Sent          |Kernel Received      |-10.5455 | -0.0045 | 9.8362  | -1.0008 |
+---------------------+---------------------+---------------------------------------+
|        sort0        |        sort1        |                JITTER                 |
|                     |                     |   MIN   |  MEAN   |   MAX   |  LAST   |
+---------------------+---------------------+---------------------------------------+
|Python Sent          |Python Received      | 0.0000  | 2.7118  | 4.0393  | 2.9834  |
|Kernel Sent          |Kernel Received      | 0.0000  | 2.7121  | 4.0389  | 2.9835  |
+---------------------+---------------------+---------------------------------------+
```

Commandline Output 1: PerfVis statistics output for a PerfVis measurement configured with a 100 ms packet interval over a Wi-Fi 5 (802.11ac) link on which the sending device was contending with one other device (same measurement presented in Fig. 2). The commandline output reports the data rate and loss of the measurement above the table, and the delay and jitter metrics for different timestamp pairs in the table. In this case, the timestamp pairs are the sending and receiving side timestamps from the Python application layer, as well as the kernel timestamps.

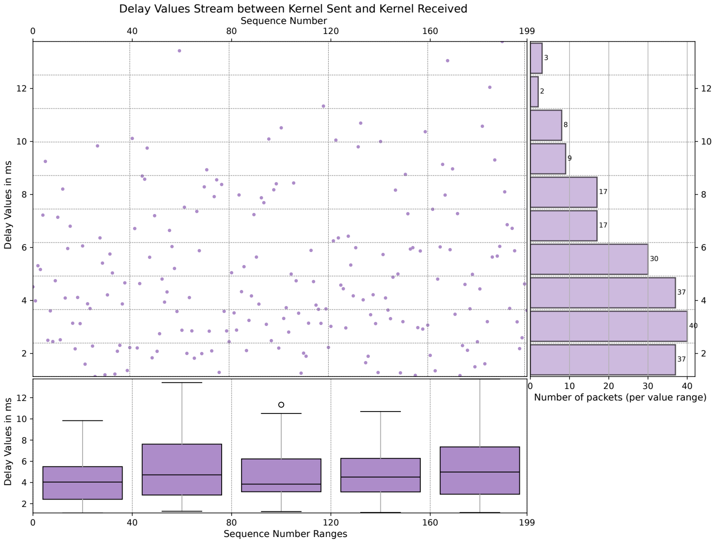


Figure 6: PerfVis delay plot of a PerfVis measurement configured with a 100 ms packet interval over a Wi-Fi 5 (802.11ac) link on which the sending device was contending with one other device (same measurement presented in Fig. 2 and Commandline Output 1). The PerfVis statistics plot consists of three combined plots. The centre plot shows individual delays; the bottom plot allows judging the evolution over time; and the right plot provides insight into the overall distribution of delays.

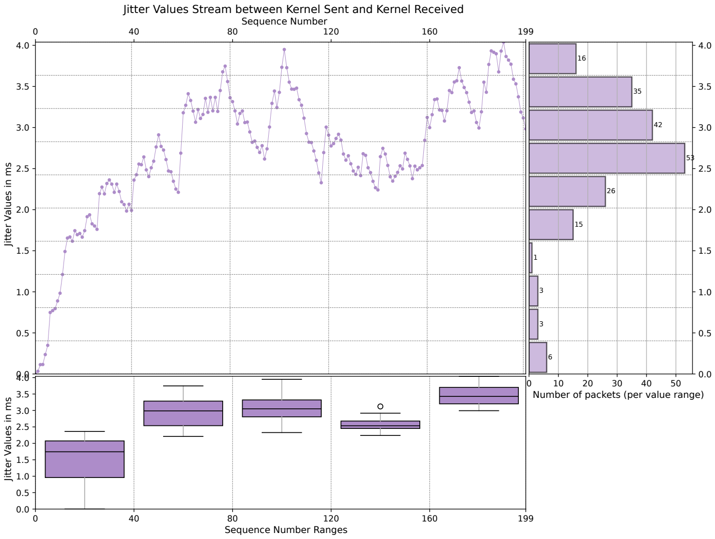


Figure 7: PerfVis jitter plot of a PerfVis measurement configured with 100 ms packet interval over a Wi-Fi 5 (802.11ac) link on which the sending device was contending with one other device (same measurement presented in Figs. 2 and 6 and Commandline Output 1). PerfVis provides the statistics plot for all delay and jitter metrics. This jitter plot highlights the visualization of a metric evolving over time.

The centre plots report the values of the metric over the sequence number of the packets, which usually increases over the course of the measurement and thus corresponds to the course of time. As this scatter plot can be challenging to grasp, summaries for bins in both ordinates are provided, making it more accessible.

Boxplots on the bottom complement the course of time on the x-axis. The boxes themselves provide the 25th and 75th percentiles, and the line in the middle is the median. The whiskers are based on the 1.5 interquartile range, and circles denote outliers.

The bar chart on the right gives the number of values that fall into the specific bin on the y-axis. This provides some more information about the distribution of the metric.

The Figs. 6 and 7 show the kernel layer delay and jitter plots of the Wi-Fi example, respectively, corresponding to Fig. 2 and Commandline Output 1. The boxplots in Fig. 7 show that the jitter first has to approach some meaningful value from the starting value of 0. While the statistics in Commandline Output 1 indicate that the kernel delay mean is approximately 5 ms, the bar chart in Fig. 6 shows that the bin centred around 3 ms contains the most delay values.

4. Case Studies

The following two use cases highlight how the presented extensions can provide further insights into network or protocol behaviour. The first case sheds light on 5G scheduling and utilises the statistics output and plots in combination with the PerfVis 2D timeline visualization. The second case illustrates some key characteristics of the TWAMP protocol and demonstrates the inclusion of TWAMP data into PerfVis, highlighting the applications of the visualization alignment and the usage of summary statistics.

4.1. 5G Scheduling

The measurement over a 5G link presented in Fig. 4, combined with the statistics output and plots, provides additional insights. The 5G system underlying the 5G measurements in this paper is a 5G campus network that consists of a dedicated hardware radio access network and a research and development core in software, developed in Europe. The 5G link in this measurement is configured with a 7:3 dual period TDD slot pattern ’DDDSUDDSUU’ of downlink slots ’D’, uplink slots ’U’ and special slots ’S’. The special slots contain another 10 downlink OFDM symbols, 2 guard period OFDM symbols and 2 uplink OFDM symbols in that order. The TDD slot pattern is repeated twice in a 10 ms 5G frame, which leads to 4 sections in a 5G frame where uplink is possible. When reviewing Fig. 4 with this knowledge that the measurement is performed in the uplink direction, it is surprising that only two of the uplink sections available in a 10 ms 5G frame are used, and one of them even quite rarely. When opting for low network delays, this PerfVis measurement clearly shows that the 5G system does not operate ideally to accomplish low delays.

With the statistical information, even more insights into the scheduling of the 5G system can be gained. The PerfVis 2D timeline visualization does not give information about the delay. The statistics commandline output provides a minimum of 1.3164 ms, a mean of 11.8884 ms and a maximum of 32.3745 ms. With the maximum delay, it can be inferred that the packets are transmitted within the first 3 5G frames after their transmission, and an IPDV mean of 0 indicates that the delay is not generally increasing or decreasing over time.

Having a closer look into the PerfVis PDV plot in Fig. 8, and assuming that the minimal delay in the measurement is roughly the minimum delay that can be acheived over the 5G link in this configuration, it becomes clear that the vast majority of packets is transmitted in the following 2 5G frames after sending, as their delay differes up to 20 ms from the minimal delay. The PDV values exhibit a structure of descending lines, which aligns with the fact that a shifting sending pattern is employed (see Fig. 4).

The PerfVis IPDV plot in Fig. 9, however, reveals another interesting insight. The IPDV values are mainly 10 ms and −10 ms and only very rarely 0 ms. This means that packets that follow each other are nearly never transmitted in the same 5G frame, but rather alternately in the next or the second next 5G frame. This unexpected scheduling behaviour is an interesting subject for further research.

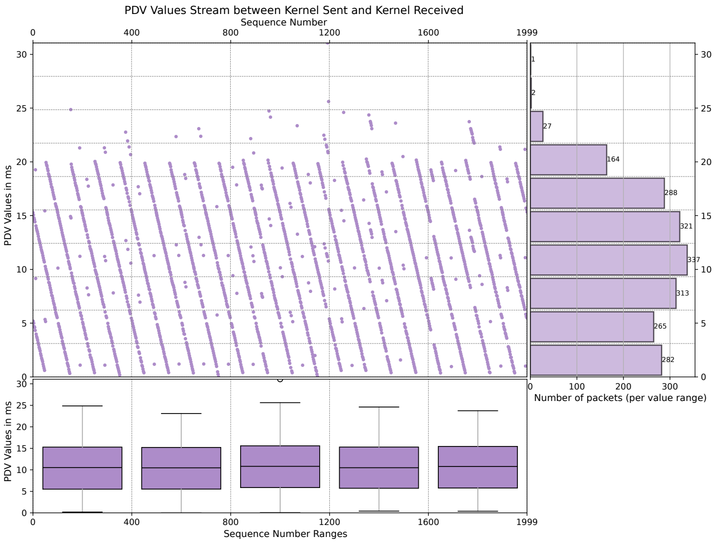


Figure 8: PerfVis PDV plot PerfVis measurement over a 5G link and configured with a sending pattern of 10 ms periodicity and an increasing shift of 0.1 ms (same measurement presented in Fig. 4). The shift of the sending time and the 5G link that accumulates the receive times causes a tilted line pattern in the PDV plot. The PDV being mainly in the range of 0 ms to 20 ms means that the sending usually varies over two 5G frames.

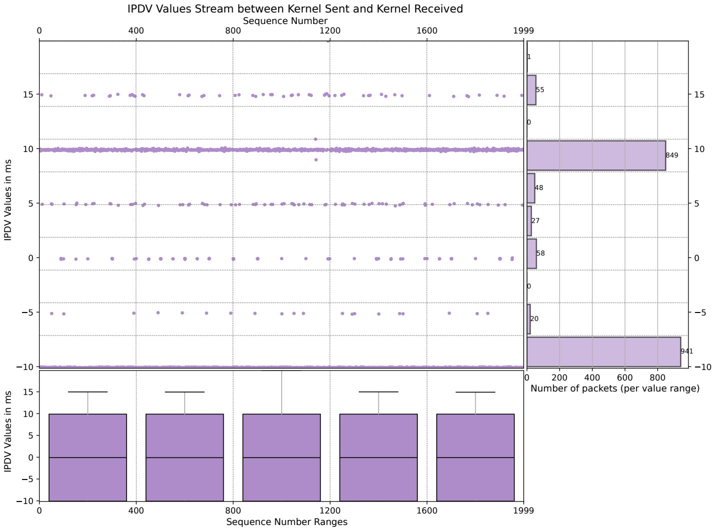


Figure 9: PerfVis IPDV plot PerfVis measurement over a 5G link and configured with a sending pattern of 10 ms periodicity and an increasing shift of 0.1 ms (same measurement presented in Figs. 4 and 8). The IPDV reveals that our 5G radio access network alternately schedules packets for the next and second next 5G frames.

4.2. TWAMP Measurement

This example demonstrates the use of TWAMP measurement packet capture to provide PerfVis visualizations and statistics, which requires an external timestamp source. The Two-Way Active Measurement Protocol (TWAMP) [3] is an open protocol for round-trip network measurements. The measurement is conducted between two hosts, with one assigned the role of the session sender and the other assigned the role of the session reflector. TWAMP builds on the One-Way Active Measurement Protocol (OWAMP) [4] but offers to account for processing delays at the session reflector. The configuration of the TWAMP measurement presented here is as closely comparable to the PerfVis measurement in the Section 3.2.3. This means that it is configured with a packet interval of 100 ms and is conducted over the same WIFI link with one contending device.

When comparing the PerfVis visualization of the PerfVis measurement in Fig. 2 with the PerfVis visualization of the TWAMP measurement in Fig. 10, it is evident that the TWAMP measurement shows a drift for the sending timestamps already. The artificial scheduled send times are just an assumption due to the configuration of a 100 ms interval for the TWAMP measurement. The reason is that TWAMP does not adhere to a fixed schedule and presumably does not account for its own computation time at the session sender. The TWAMP RFC even mentions that the packet timing is not important: "Since the send schedule is not communicated to the Session-Reflector, there is no need for a standardised computation of packet timing" [3]. With time-slotted systems like 5G in mind, a fixed sending schedule, however, makes it much easier to investigate system effects in a controlled manner.

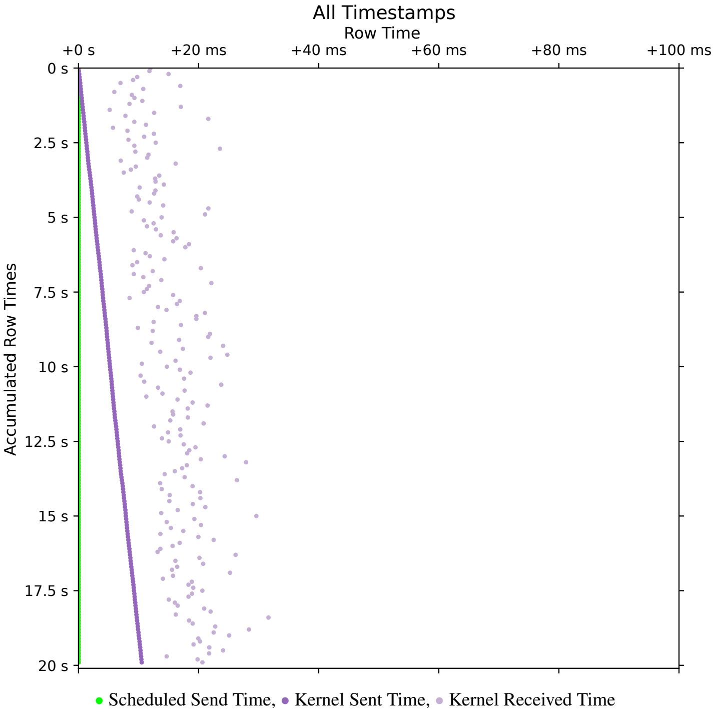


Figure 10: PerfVis 2D timeline visualization of a TWAMP measurement configured with 100 ms packet interval over a Wi-Fi 5 (802.11ac) link on which the sending device was contending with one other device. The link is the same as the Wi-Fi example from before, but the measurement protocol differs. TWAMP does not adhere to a fixed schedule, and thus the line of the send timestamps drifts away from the expected sending times of the configured 100 ms.

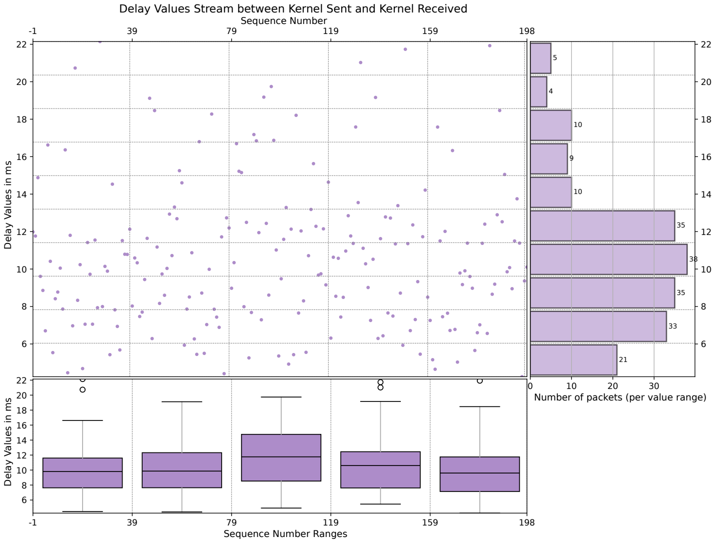


Figure 11: PerfVis delay plot of a TWAMP measurement configured with 100 ms packet interval over a Wi-Fi 5 (802.11ac) link on which the sending device was contending with one other device (same measurement presented in Fig. 10). The drift from the expected schedule is not observable in the delays, which are the basis of most network measurement tools.

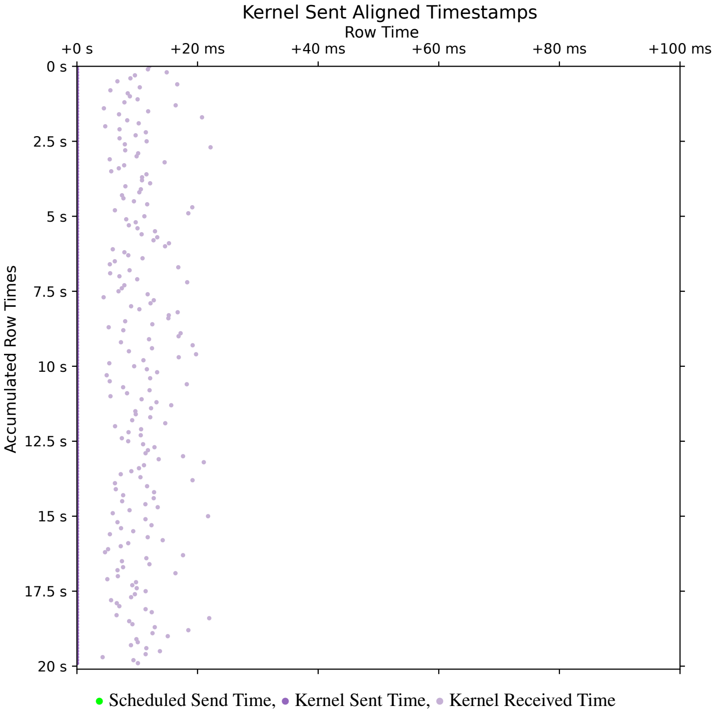


Figure 12: PerfVis aligned 2D timeline visualization of a TWAMP measurement configured with 100 ms packet interval over a Wi-Fi 5 (802.11ac) link on which the sending device was contending with one other device (same measurement presented in Figs. 10 and 11 and Commandline Output 2). This visualization aligned the kernel send timestamps in Fig. 10 with the expected schedule, namely the beginning of the row. After alignment, the absolute time of events is no longer correct. Still, the relative time between events that belong together, i.e., to the same packet, is kept constant to make it easier to judge delay and jitter metrics from the 2D timeline plot.

From Fig. 10 it is also easy to see that the drift is roughly 10.5 ms over the course of the measurement, and it is easy to calculate that it shifts about 50 µs for each of the 200 packets. In fact, it is not observable through the delay and jitter metrics that the measurement does not adhere to a fixed schedule, as these metrics are independent of the specific sending time (see Fig. 11). Also, the mean of the IPDV is roughly 0. A non-zero mean of the IPDV could indicate shifts in the delay but not in the sending pattern.

If the external tool, whose data is used for PerfVis, also provides statistical output, PerfVis can be used to provide additional metrics and to cross-check the statistical values with each other. The output from TWAMP itself (see Commandline Output 2) shows very similar values for delay and jitter compared to the PerfVis statistics output (Commandline Output 3). Slight differences exist between them due to the different origins of the timestamps, specifically from the TWAMP application and the kernel-layer packet capture used for the PerfVis statistics.

In this example, where the TWAMP measurement shows a drift in the PerfVis 2D Timeline visualization, the new alignment functionality can help in judging delays and comparing them to results without drifts. Fig. 12 shows the aligned version of Fig. 10. The alignment shifted all send timestamps to the supposed send times of the artificial schedule. With this alignment, the relationship between the timestamp pairs is more obvious and can be compared to Fig. 2, which contains the PerfVis measurement, which adheres to a fixed sending schedule on its own. Due to TWAMP being a two-way measurement protocol, the delays between send and receive events in Fig. 10 are further apart than in Fig. 2 with the PerfVis one-way measurement.

```
Round time delay jitter: 3.897239ms
Average round trip time: 10.591315ms
RT Lost packets: 0/200,  RT Loss Ratio: 0.00%
```

Commandline Output 2: TWAMP statistics output for a TWAMP measurement configured with 100 ms packet interval over a Wi-Fi 5 (802.11ac) link on which the sending device was contending with one other device (same measurement presented in Figs. 10 and 11). The measurement tool's output can serve as a cross-check for the later PerfVis analysis.

```
Statistics for the full transmission in ms
200 packets sent at 1.6 KB/s - 200 received at 1.6 KB/s
with 0.0 percent loss (0 lost).
+-------------------+-------------------+-----------------------------------------+
|       sort0       |       sort1       |                  DELAY                  |
|                   |                   |   MIN   |  MEAN   |   MAX   |   STD     |
+-------------------+-------------------+-----------------------------------------+
|Kernel Sent        |Kernel Received    | 4.2517  | 10.5711 | 22.1464 | 3.8976    |
+-------------------+-------------------+-----------------------------------------+
|       sort0       |       sort1       |                  PDV                    |
|                   |                   |   MIN   |  MEAN   |   MAX   |  LAST     |
+-------------------+-------------------+-----------------------------------------+
|Kernel Sent        |Kernel Received    | 0.0000  | 6.3194  | 17.8947 | 5.8513    |
+-------------------+-------------------+-----------------------------------------+
|       sort0       |       sort1       |                 IPDV                    |
|                   |                   |   MIN   |  MEAN   |   MAX   |  LAST     |
+-------------------+-------------------+-----------------------------------------+
|Kernel Sent        |Kernel Received    |-14.1508 | -0.0095 | 15.8029 | 0.7409    |
+-------------------+-------------------+-----------------------------------------+
|       sort0       |       sort1       |                 JITTER                  |
|                   |                   |   MIN   |  MEAN   |   MAX   |  LAST     |
+-------------------+-------------------+-----------------------------------------+
|Kernel Sent        |Kernel Received    | 0.0000  | 4.1356  | 5.9353  | 3.8720    |
+-------------------+-------------------+-----------------------------------------+
```

Commandline Output 3: PerfVis statistics output for a TWAMP measurement configured with 100 ms packet interval over a Wi-Fi 5 (802.11ac) link on which the sending device was contending with one other device (same measurement presented in Figs. 10 to 12 and Commandline Output 2). The PerfVis output provides an extensive set of statistics complementing the tool's output.

4.3. Reproducability Details

To facilitate reproduction of the measurements presented in this paper, the following details specify the hardware, software, and measurement setup used in our case studies.

The measurement setup comprises a server, called LabServer, connected to a 5G router that acts as a client for 5G or Wi-Fi. For 5G measurements, the 5G router connects through the 5G RAN to the 5G Core running on a server, called CoreServer, which is connected back to the LabServer via Ethernet. For Wi-Fi measurements, the 5G router instead connects to another Wi-Fi router connected to the LabServer via Ethernet. For Wi-Fi contention, a laptop also connects to the Wi-Fi router and generates traffic by executing Iperf3. The measurement commands are executed in two separate LXC containers on the LabServer for the PerfVis sender and receiver or the TWAMP client and reflector. This way, the two containers share the same hardware clock, and clock synchronisation is not an issue.

All measurement commands, raw NPZ files, and PCAP captures, as well as 5G and Wi-Fi link configuration details, are available in the PerfVis Git Repository.[4]

Table 1: Reproducibility Hardware and Software Specifications

Component	Specification/Details
	Hardware
LabServer	CPU AMD Ryzen 9 9950X; 60GiB RAM; 2 Ethernet on ASUS ProArt B650-Creator; Proxmox version 8.4.14 (running kernel: 6.8.12-10-pve)
LXC Containers	Ubuntu 24.04.3 LTS (GNU/Linux 6.8.12-10-pve x86_64); 4 CPUs; 2GiB RAM
5G Router	Milesight UR75-504AE-P-W2 V1.2; Firmware 78.0.0.4
5G RAN	Huawei BBU5900; Huawei RRU5836E
5G Core	Fraunhofer Fokus Open5GCore; deployed in LXC containers
CoreServer	DELL PowerEdge R640Intel Xeon Gold 6246; 46GiB RAM; Intel Network Adapter X710-DA4; Proxmox version 8.4.14 (running kernel: 6.8.12-10-pve)
Wi-Fi Router	D-Link DIR-X1560
Laptop	Lenovo L13 Yoga; MediaTek MT7921 Wireless Network Adapter; Manjaro Linux (kernel 6.6.107-1-MANJARO)
	Software
Python	version 3.12.3
PerfVis	Git tag 'astesj25'
TWAMP	implementation by Emma Mirica[5](adapted for additional commandline output)
TShark	version 4.2.2
Iperf3	version 3.19.1

5. Conclusion

With its visualization, PerfVis provides means to view timestamp data in new ways, and the extensions presented in this paper make it a dynamic tool for experimental network analysis or protocol evaluation. Scripts for incorporating data from external timestamp sources into the visualization offer new perspectives. The enhanced visualization gives more degrees of freedom in customising the visual representation, and the new statistics module closes the gap to traditional tools that primarily provide output in numbers. The case studies demonstrated that combining PerfVis 2D Timeline visualisation with statistics as summaries and graphs can provide deep insights into otherwise hidden structures.

We plan to develop the tool further, adding more options on how the animation or video fills up frames with data, e.g. a sliding mode that we might associate with other monitoring tasks, as well as usability in terms of user interface and interactivity and supporting a more modular approach for including a larger number of timestamps from multiple layers and sources at the same time.

PerfVis, with its underlying visualization and analysis ideas, provides help and inspiration for network engineers and researchers to deep dive into specific network behaviours. The analysis part is also open for timing data from other areas not related to transmission network analysis.

Conflict of Interest The authors declare no conflict of interest.

Acknowledgment We thank our students Felix Weber and Julius Herrmann for their contributions and inspiring discussions. The 5G Campus Network has been financed by Saarland State Major Instrumentation (Art. 143c GG).

References

[1] M. Böhmer, T. Herfet, "PerfVis: Visualization of Timings in Network Transmission," in 2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC), 1–6, IEEE, 2025, doi:10.1109/ccnc54725.2025.10976200.

[2] A. Bhat, V. Ganatra, A. Shaha, B. Chandrasekaran, V. Naik, "On the Constancy of Latency at the Internet's Edge," in 2025 9th Network Traffic Measurement and Analysis Conference (TMA), 1–10, IEEE, 2025, doi:10.23919/tma66427.2025.11096966.

[3] K. Hedayat, R. Krzanowski, A. Morton, K. Yum, J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)," RFC 5357, RFC Editor, 2008, doi:10.17487/RFC5357.

[4] S. Shalunov, B. Teitelbaum, A. Karp, J. Boote, M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)," RFC 4656, RFC Editor, 2006, doi:10.17487/rfc4656.

[5] O. Basit, I. Khan, M. Ghoshal, Y. C. Hu, D. Koutsonikolas, "5G Metamorphosis: A Longitudinal Study of 5G Performance from the Beginning," in Proceedings of the 2025 ACM Internet Measurement Conference, 17–31, ACM, 2025, doi:10.1145/3730567.3732914.

[6] J. Fabini, M. Abmayer, "Delay Measurement Methodology Revisited: Time-Slotted Randomness Cancellation," IEEE Transactions on Instrumentation and Measurement, **62**(10), 2013, doi:10.1109/tim.2013.2263914.

[4] https://git.nt.uni-saarland.de/research-projects/independent/performance-visualization/-/tree/astesj25/publications/astesj25?ref_type=tags

[5] https://github.com/emirica/twamp-protocol/tree/d682c7d40d67362edc112d68e3ecbd4a84ed654a

[7] P. Orosz, T. Skopko, J. Imrek, "A NetFPGA-based network monitoring system with multi-layer timestamping: Rnetprobe," in 2012 15th International Telecommunications Network Strategy and Planning Symposium (NETWORKS), IEEE, 2012, doi:10.1109/netwks.2012.6381709.

[8] S. Reif, A. Schmidt, T. Hönig, T. Herfet, W. Schröder-Preikschat, "X-LAP: A systems approach for cross-layer profiling and latency analysis for cyber-physical networks," ACM SIGBED Review, **15**(3), 19–24, 2018, doi:10.1145/3267419.3267422.

[9] R. Kundel, F. Siegmund, J. Blendin, A. Rizk, B. Koldehofe, "P4STA: High performance packet timestamping with programmable packet processors," in NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2020, doi:10.1109/NOMS47738.2020.9110290.

[10] M. A. Aleisa, "Traffic classification in SDN-based IoT network using two-level fused network with self-adaptive manta ray foraging," Scientific Reports, **15**(1), 2025, doi:10.1038/s41598-024-84775-5.

[11] A. Azab, M. Khasawneh, S. Alrabaee, K.-K. R. Choo, M. Sarsour, "Network traffic classification: Techniques, datasets, and challenges," Digital Communications and Networks, **10**(3), 676–692, 2024, doi:10.1016/j.dcan.2022.09.009.

[12] W. Aigner, S. Miksch, H. Schumann, C. Tominski, Visualization of time-oriented data, Springer London, 2023, doi:10.1007/978-1-4471-7527-8.

[13] A. Protopsaltis, P. Sarigiannidis, D. Margounakis, A. Lytos, "Data visualization in internet of things: tools, methodologies, and challenges," in Proceedings of the 15th International Conference on Availability, Reliability and Security, ARES 2020, ACM, 2020, doi:10.1145/3407023.3409228.

[14] S. Di Bartolomeo, A. Pandey, A. Leventidis, D. Saffo, U. H. Syeda, E. Carstensdottir, M. Seif El-Nasr, M. A. Borkin, C. Dunne, "Evaluating the Effect of Timeline Shape on Visualization Task Performance," in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20, ACM, 2020, doi:10.1145/3313831.3376237.

[15] B. Gregg, "Visualizing system latency," Communications of the ACM, **53**(7), 48–54, 2010, doi:10.1145/1785414.1785435.

[16] Z. Liu, J. Heer, "The Effects of Interactive Latency on Exploratory Visual Analysis," IEEE Transactions on Visualization and Computer Graphics, **20**(12), 2122–2131, 2014, doi:10.1109/tvcg.2014.2346452.

[17] G. Almes, S. Kalidindi, M. Zekauskas, "A One-Way Delay Metric for IP Performance Metrics (IPPM)," RFC 7679, RFC Editor, 2016, doi:10.17487/rfc7679.

[18] A. Morton, B. Claise, "Packet Delay Variation Applicability Statement," RFC 5481, RFC Editor, 2009, doi:10.17487/rfc5481.

[19] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, RFC Editor, 2003, doi:10.17487/rfc3550.

[20] S. Sundberg, A. Brunstrom, S. Ferlin-Reiter, T. Høiland-Jørgensen, R. Chacón, "Measuring Network Latency from a Wireless ISP: Variations Within and Across Subnets," in Proceedings of the 2024 ACM on Internet Measurement Conference, IMC '24, 29–43, ACM, 2024, doi:10.1145/3646547.3688438.

Advances in Science, Technology and Engineering Systems Journal
Vol. 10, No. 6, 88-96 (2025)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTES Journal
ISSN: 2415-6698

A Multi-class Acoustic Dataset and Interactive Tool for Analyzing Drone Signatures in Real-World Environments

Mia Yaqin Wang[1], Mackenzie Linn[1], Andrew Patrick Berg[1], Qian Zhang*[2]

[1]*College of Charleston, Department of Computer Science, Charleston, 29424, USA*

[2]*College of Charleston, Department of Engineering, Charleston, 29424, USA*

Email(s): wangy5@cofc.edu (M. Wang), linnmj@g.cofc.edu (M. Linn), berga2@g.cofc.edu (A. Berg)

*Corresponding Author: Qian Zhang, 66 George Street, Charleston, SC 29424: zhangq@cofc.edu

ARTICLE INFO

Article history:
Received: 11 September, 2025
Revised: 11 December, 2025
Accepted: 13 December, 2025
Online: 24 December, 2025

Keywords:
Drone Audio Dataset
Drone Acoustics
Spectrogram
MFCC
Data Visualization
Web Application

ABSTRACT

The rapid proliferation of drones across various industries has introduced significant challenges related to privacy, security, and noise pollution. Current drone detection systems, primarily based on visual and radar technologies, face limitations under certain conditions, highlighting the need for effective acoustic-based detection methods. This paper presents a unique and comprehensive dataset of drone acoustic signatures, encompassing 32 different categories differentiated by brand and model. The dataset includes raw audio recordings, spectrogram plots, and Mel-frequency Cepstral Coefficient (MFCC) plots for each drone. Additionally, we introduce an interactive web application that allows users to explore this dataset by selecting specific drone categories, listening to the associated audio, and viewing the corresponding spectrogram and MFCC plots. This tool aims to facilitate research in drone detection, classification, and acoustic analysis, supporting both technological advancements and educational initiatives. The paper details the dataset creation process, the design and implementation of the web application, and provides experimental results and user feedback. Finally, we discuss potential applications and future work to expand and enhance the project.

1. Introduction

The rapid proliferation of drones in various industries such as delivery, surveillance, agriculture, and entertainment has introduced significant challenges and opportunities. While drones offer numerous benefits, their widespread use has also raised concerns regarding privacy, security, and noise pollution. Unauthorized drone activity can lead to breaches of privacy and potential security threats, while drones contribute to environmental noise pollution, affecting human health and wildlife. Current drone detection systems primarily rely on visual and radar-based technologies, which face limitations under poor visibility or in cluttered environments. Acoustic-based detection presents a promising complementary approach, but there is a notable lack of comprehensive acoustic datasets encompassing a wide range of drone models and operational conditions. This gap hinders the development of robust detection algorithms and effective noise mitigation strategies. Additionally, there is a need for interactive tools to facilitate research and education in drone acoustics.

This journal paper extends our previous work on drone visualization, which was presented in the 2024 Artificial Intelligence x Humanities, Education, and Art (AIxHeart 2024) Conference [1]. To address these challenges, we present a novel dataset comprising audio recordings, spectrograms, and Mel-frequency Cepstral Coefficient (MFCC) plots for 32 different drone categories, differentiated by brand and model. Alongside this dataset, we introduce an interactive web application designed to allow users to explore the data intuitively. Users can select specific drone categories, listen to the associated audio recordings, and view the corresponding spectrogram and MFCC plots. This tool aims to enhance research capabilities in drone detection, classification, and acoustic analysis, support noise mitigation efforts, and serve as an educational resource. The publicly available website can be found online [1].

The rest of this paper is organized as follows: Section 2 provides a detailed review of the literature related to drone acoustic detection, classification, and noise pollution, as well as existing datasets and interactive tools. Section 3 describes the dataset, in-

[1] https://mackenzie-jane.github.io/drone-visualization/

https://dx.doi.org/10.25046/aj100608

cluding methods of data collection and the formats of the audio recordings, spectrograms, and MFCC plots. Section 4 discusses the design and implementation of the visualization web application, outlining its user interface and backend architecture. Section 5 presents experimental results, including an analysis of the dataset and user feedback on the web application. In Section 6, we explore potential applications of our dataset and tool, and outline future work to expand and enhance the project. Finally, Section 7 concludes the paper, summarizing our contributions and the impact of our research.

2. Literature Review

The rapid growth of drone technologies and acoustic sensing capabilities has sparked a diverse body of research spanning detection techniques, sensing modalities, dataset development, and interactive tools for education and exploration. This section provides a structured review of prior work in seven key areas. We begin by summarizing general audio-based detection methods, followed by vision-based and radar-based approaches, each offering unique benefits and limitations. We then review detection systems based on radio frequency signatures and explore available datasets designed to support classification and benchmarking tasks. Finally, we highlight interactive tools that promote hands-on learning and simulation-based exploration in both formal education and public-facing research platforms.

2.1. Audio-Based Methods for UAV Detection

Acoustic sensing offers a unique, low-cost, and passive modality for unmanned aerial vehicle (UAV) detection, particularly effective in scenarios where visual or radio frequency based systems may be limited by occlusion, range, or signal interference. When drones operate, their motors and rotors emit characteristic sounds that vary across models, providing an opportunity to capture distinctive audio fingerprints. These acoustic signatures can be used not only for detection but also for identification and classification, especially when represented using time-frequency features such as spectrograms and Mel Frequency Cepstral Coefficients (MFCCs) [2, 3].

Several studies have leveraged these acoustic characteristics to build UAV detection datasets and evaluate various signal processing pipelines. In [2], the authors compared five feature extraction techniques available in the Librosa Python library—MFCCs [4], chroma, Mel spectrograms, spectral contrast, and tonnetz—applied to audio recordings collected from DJI Phantom 4 and EVO 2 Pro drones, along with environmental noise samples. Their analysis revealed that combining multiple acoustic features significantly enhanced the discriminative capacity of the data.

Other researchers have explored real-time detection potential and robustness to noise. In [5], the authors studied UAV detection within a 150 meter range using Gaussian Mixture Models (GMM), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). They augmented drone recordings with environmental sounds and evaluated MFCC and Mel spectrogram features. Their results confirmed that acoustic signal fidelity degrades beyond 150 meters, reinforcing the importance of proximity in audio based systems.

In [6], the researchers used Short Time Fourier Transform (STFT) features with CNNs to evaluate detection accuracy and false alarm rates in open air environments. Their dataset, built from hovering Phantom 3 and Phantom 4 drones, achieved high detection rates exceeding 98% and a low false alarm rate of 1.28%, showcasing the utility of normalized time-frequency features for clean drone recordings.

To address data scarcity, [7] generated synthetic UAV audio using Generative Adversarial Networks (GANs). Their hybrid dataset supported both binary classification (drone versus noise) and multiclass identification tasks. CNN, RNN, and CRNN models trained on this augmented corpus demonstrated improved generalizability, highlighting the potential of synthetic data for expanding UAV sound libraries.

More recently, [8] introduced a self-supervised learning framework to mitigate the limitations of label dependent models. Their approach transforms drone audio into MFCC-based image representations and applies SimCLR, a contrastive learning algorithm, to learn generalized latent features. Without requiring labeled data, their model achieved classification accuracy on par with supervised baselines—reaching a top-1 accuracy of 87.91%. Notably, the system generalized to unseen drone types, demonstrating strong potential for scalable and adaptive drone detection in real world settings.

Collectively, these studies emphasize the need for diverse, well annotated, and context rich UAV acoustic datasets to support future research in drone detection and sound based classification. Our work builds upon this foundation by providing a comprehensive dataset of 32 UAV classes with corresponding MFCC and Mel spectrogram visualizations, aimed at enabling both analytical exploration and educational applications through an interactive web tool.

2.2. Vision Based Methods for UAV Detection

Computer vision has emerged as one of the most widely adopted approaches for UAV detection and classification due to the proliferation of cameras and advances in machine learning based image analysis. Vision based systems rely on optical or infrared imagery to locate and identify drones based on their shape, motion patterns, and appearance. However, they face challenges in low light conditions, visual occlusion, and adverse weather.

In [9], the authors developed an end to end object detection framework using the You Only Look Once (YOLO) architecture, a real-time single shot detector built on convolutional neural networks (CNNs). Their system was trained using a dataset composed of bird and drone images embedded in varied backgrounds to simulate real world variability. The trained model demonstrated high precision and recall (both at 0.9), confirming the feasibility of rapid detection from video frames.

In [10], the authors proposed motion stabilization techniques to enhance the visual classification of UAVs from moving cameras. By extracting spatio temporal features from image cubes and applying regression based stabilization, their system improved object detection in dynamic scenes. They evaluated boosted tree and CNN based classifiers on custom collected datasets comprising UAVs and aircraft, reporting average precision scores above 0.84 for UAV categories.

In [11], the researchers proposed a two module system that combines a drone detection module and an identification module. The detection module used Haar like features and OpenCV's object detection pipeline, while the identification module applied a simple CNN with two convolutional and two fully connected layers. Their dataset included 7,000 drone and 3,000 non-drone images. The overall system achieved 89% detection accuracy and 91.6% identification accuracy.

These vision based techniques offer robust performance in clear visual environments and are well suited for medium to long range UAV detection. However, they are less effective under occlusion, poor lighting, or fast drone maneuvers, motivating the need for complementary modalities such as acoustics or radar.

2.3. Radar Based Methods for UAV Detection

Radar based detection methods provide an effective alternative for identifying UAVs, especially in scenarios involving poor visibility or long distance operation. These systems detect objects by emitting radio waves and analyzing the reflected signals, offering advantages in range, reliability, and performance under challenging environmental conditions.

In [12], the authors developed a radar detection framework using an S band continuous wave radar coupled with a deep belief network (DBN). They extracted micro Doppler signatures from three different UAV types, including a helicopter, an artificial bird, and a quadcopter. By using spectral correlation functions as inputs, their DBN based classifier achieved over 90% classification accuracy. Gaussian noise was added for data augmentation, which helped evaluate system performance under various signal conditions.

In [13], the authors proposed a system using a frequency modulated continuous wave radar along with a pre trained convolutional neural network, specifically GoogleNet. Their dataset included over 66,000 micro Doppler and cadence velocity diagram images collected both indoors in an anechoic chamber and outdoors. By simulating different motor types and observation angles, they assessed model robustness and achieved 94.7% classification accuracy across varied scenarios.

Radar based approaches demonstrate strong performance in tracking UAV movement across large distances and in detecting small objects under low light or visually obstructed environments. However, these systems typically require more specialized hardware and signal processing expertise, which may limit widespread adoption compared to vision or acoustic based systems.

2.4. Radio Frequency Based Methods for UAV Detection

Radio frequency (RF) based detection systems leverage the electromagnetic emissions produced by the communication link between UAVs and their controllers. These systems can detect and identify drones based on signal characteristics such as transmission frequency, modulation patterns, and signal strength, making them particularly useful for detecting both the drone and the pilot's control source.

In [14], the authors presented a detection framework that uses radio frequency transmissions between UAVs and their controllers to distinguish UAV related activity from background communication signals. Their approach employed a Bayesian model derived from Markov processes to perform binary detection and multiclass classification. Extracted features included spectral entropy, skewness, variance, and kurtosis of the signal, followed by feature selection using neighborhood component analysis. The processed features were input into several machine learning classifiers, including support vector machines and neural networks. Their model achieved over 96% classification accuracy.

In [15], the authors proposed a detection method based on Gaussian Mixture Models and an adaptive thresholding mechanism to determine UAV activity. They collected data from seven UAV models, extracting signal onset points through statistical analysis of Gaussian distributions. Their method achieved 97% detection accuracy, demonstrating strong performance in identifying the beginning of UAV control signal activity.

In [16], the researchers conducted a comprehensive review of radio frequency based UAV detection and classification methods. Their study categorized techniques into classical, advanced, and hybrid engineering approaches, highlighting the strengths and limitations of each in relation to input data types such as MAC addresses, communication protocols, extracted features, and raw I/Q signals. They emphasized that deep learning based methods, particularly those leveraging raw RF signals and time frequency representations, have shown superior performance in recent literature. Furthermore, they discussed key publicly available RF drone datasets, such as the DroneRF and VTI RF datasets, and noted the scarcity of large scale open access RF datasets. Their comparative analysis showed that deep learning models trained on the DroneRF and VTI RF datasets consistently achieved high classification performance. For example, CNN-based models on the DroneRF dataset reported up to 100% detection accuracy and over 94% type identification accuracy [16]. Similarly, models evaluated on the VTI_RF_Dataset achieved over 99% in drone detection and 97% in multiple drone identification scenarios. These results underscore the effectiveness of modern deep learning techniques when paired with well-curated RF signal datasets.

Radio frequency based techniques provide the advantage of long range and real-time detection capabilities, particularly in open environments where signal propagation is reliable. However, these methods can be challenged by encrypted communication, frequency hopping protocols, and legal restrictions on RF signal monitoring. As such, they are often used in conjunction with other sensing modalities for comprehensive UAV detection solutions.

2.5. Audio-Derived Visual Methods

Recent work has explored the transformation of acoustic data into visual formats to leverage the strength of computer vision models in drone classification. In [17], the authors introduced a novel methodology that converts audio recordings into Mel Frequency Cepstral Coefficient (MFCC) plots, enabling the use of image-based deep learning models. Their dataset included 32 UAV categories, each with 100 five-second audio recordings, from which MFCC visualizations were generated.

The authors trained and compared three vision architectures—EfficientNet, ResNet50, and Vision Transformer—on these MFCC images. Among them, EfficientNet achieved the highest classification accuracy of 96.31%, followed by ResNet50 at 94.22%, and Vision Transformer at 73.69%. These results highlight the promise of combining auditory signals with visual model pipelines and underscore the discriminative power of MFCCs when processed through image-based frameworks.

This audio-to-visual strategy provides a compelling hybrid solution that bridges the strengths of acoustic data and modern vision models. It also opens new opportunities for multi-modal fusion in UAV detection, particularly in settings where raw audio may be harder to interpret directly.

2.6. Existing UAV Audio Datasets

Despite promising advancements in UAV detection through acoustic sensing, the progress of audio-based systems has been constrained by the limited availability of high-quality, publicly accessible datasets. Most existing efforts in this domain focus on creating targeted corpora tailored to specific drone models or environmental conditions, which restricts their utility for broader model generalization and benchmarking.

In [18], the authors introduced one of the largest open-access UAV audio datasets to date, containing recordings from 15 different drones—including both small toy models and larger Class I UAVs—totaling 8,120 seconds of annotated audio. The dataset captures diverse operational sounds and was used to train a convolutional neural network for 15-class classification, achieving an average test accuracy of 98.7% and a test loss of 0.076. These results underscore the dataset's value for supporting robust, real-world classification systems.

Building on this work, [19] further investigated the impact of feature design on UAV classification performance. Specifically, they evaluated various quantities of Mel Frequency Cepstral Coefficients (MFCCs) and determined that using 30 coefficients provided an optimal balance between feature richness and noise resilience. This study also introduced a companion image-based dataset derived from the original audio, consisting of waveform, spectrogram, Mel filter bank, and MFCC plots across 26 UAV categories. With 100 audio samples per category, the dataset supports both visual and audio modality exploration and facilitates the development of multimodal detection systems.

Together, these datasets offer a foundation for standardized evaluation in acoustic-based UAV detection and serve as critical resources for researchers aiming to improve generalization, scalability, and interpretability of drone classification models.

2.7. Interactive Tools for Research and Education

Interactive tools for audio-based exploration and simulation are increasingly used to enhance understanding of acoustic principles, foster student engagement, and support open-ended inquiry in both research and classroom settings.

Interactive tools and platforms for exploring acoustic data are crucial for advancing research and education. Projects like BirdVox [20], which provides interactive access to bird sound datasets, illustrate the benefits of such tools. However, similar resources for drone acoustics are notably lacking. This gap hampers the ability of researchers to conduct in-depth analyses and limits the educational potential of these datasets.

In [21], the authors introduced the Acoustics Apps platform, a browser-based e-learning environment that uses high-fidelity simulations powered by COMSOL Server technology. These apps support interactive exploration of complex wave phenomena, musical instrument behavior, and room acoustics without requiring access to physical lab equipment. Designed to be intuitive and device-independent, Acoustics Apps have been successfully used in both high school and university settings to visualize invisible acoustic behaviors and engage students through virtual experiments and self-guided exploration.

In the context of sonic interaction design, [22] presented the Sound Design Toolkit (SDT), a modular software environment for real-time, physics-based sound synthesis. SDT includes a library of sound models—such as impacts, friction, and fluid sounds—that can be interactively controlled using sensors or mapped to MIDI/OSC inputs. Developed with education and prototyping in mind, SDT facilitates experiential learning by enabling users to sketch, manipulate, and evaluate sonic feedback in design scenarios. Its taxonomy of everyday sounds also makes it suitable for classroom demonstrations of sound physics and design aesthetics.

In [23], the authors designed a simple, interactive GUI tool in MATLAB that allows students to record and visualize sound waveforms and their corresponding frequency spectra. Primarily used to teach the Fourier transform concept in introductory engineering courses, the tool lets users experiment with different input sounds—including their own voice—and immediately observe how spectral components vary. This low-cost and hands-on approach is particularly effective in demystifying frequency-domain analysis for first-year students.

In [24], the researchers examined the role of Interactive Audio Visual (IAV) media in improving creative thinking among science students. Their mixed-methods study found that students who engaged with IAV materials demonstrated significantly higher creative thinking scores compared to those using conventional PowerPoint media. Students also expressed high levels of interest and ease in using the interactive content, noting improved comprehension and increased motivation during tasks involving simulation-based learning. These findings reinforce the educational value of interactive multimedia in facilitating critical and creative thinking skills.

In [1], the authors developed a web-based drone audio visualization tool that enables users to explore the unique acoustic signatures of drones by listening to recordings and examining their associated spectrogram and MFCC plots. This platform laid the groundwork for the current journal study by providing an initial system for drone sound data visualization. The updated version enhances this foundation with expanded features, improved interactivity, and integration of a broader UAV dataset. The platform presents a browsable interface with drone images and playback controls, designed to make drone acoustics accessible for students, researchers, and hobbyists.

Collectively, these platforms reflect a growing emphasis on accessible, engaging, and interactive resources in acoustics education

Table 1: UAV Audio Dataset: 32 Classes with Collection Sites

Manufacturer	**Model**	**Drone Type**	**Number of Files**	**Duration (sec)**	**Collection Site**
Self-build	David Tricopter	Outdoor	100	500	Columbus, IN
Self-build	PhenoBee	Outdoor	100	500	West Lafayette, IN
Autel	Evo 2 Pro	Outdoor	100	500	New Richmond, IN
DJI	Avata	Outdoor	100	500	Charleston, SC
DJI	FPV	Outdoor	100	500	Charleston, SC
DJI	Matrice 200	Outdoor	100	500	West Lafayette, IN
DJI	Matrice 200 V2	Outdoor	100	500	New Richmond, IN
DJI	Matrice 600p	Outdoor	100	500	New Richmond, IN
DJI	Mavic Air 2	Outdoor	100	500	New Richmond, IN
DJI	Mavic Mini 1	Outdoor	100	500	New Richmond, IN
DJI	Mini 2	Outdoor	100	500	New Richmond, IN
DJI	Mini 3	Outdoor	100	500	Charleston, SC
DJI	Mini 3 Pro	Outdoor	100	500	Charleston, SC
DJI	Mavic 2 Pro	Outdoor	100	500	New Richmond, IN
DJI	Neo	Outdoor	100	500	Charleston, SC
DJI	Mavic 2s	Outdoor	100	500	New Richmond, IN
DJI	Phantom 2	Outdoor	100	500	New Richmond, IN
DJI	Phantom 4	Outdoor	100	500	New Richmond, IN
DJI	Tello	Indoor	100	500	Charleston, SC
DJI	RoboMaster TT Tello	Indoor	100	500	New Richmond, IN
Hasakee	Q11	Indoor	100	500	West Lafayette, IN
Holystone	HS210	Indoor	100	500	Charleston, SC
Hover	X1	Outdoor	100	500	Charleston, SC
Syma	X5SW	Indoor	100	500	West Lafayette, IN
Syma	X5UW	Indoor	100	500	West Lafayette, IN
Syma	X8SW	Indoor	100	500	West Lafayette, IN
Syma	X20	Indoor	100	500	West Lafayette, IN
Syma	X20P	Indoor	100	500	West Lafayette, IN
Syma	X26	Indoor	100	500	West Lafayette, IN
Swellpro	Splash 3 plus	Outdoor	100	500	New Richmond, IN
Yuneec	Typhoon H Plus	Outdoor	100	500	New Richmond, IN
UDI RC	U46	Outdoor	100	500	West Lafayette, IN
		Total	3,200	16,000	

and sonic research. They demonstrate how interactivity—whether through sound manipulation, simulation, or visualization—can significantly deepen conceptual understanding and foster interdisciplinary learning.

3. Methodology

3.1. Data Collection

The drone data collection is an ongoing multi-year effort aimed at building a large-scale, diverse dataset of UAV acoustic signatures [1, 18, 19, 25]. As of 2025, the dataset comprises 3,200 audio recordings captured from 32 distinct unmanned aerial vehicles (UAVs), totaling 16,000 seconds of raw flight audio, as shown in Table 1. Each UAV contributed 100 five-second audio clips. These recordings span a wide range of drone types and environments and serve as the foundation for acoustic analysis, feature extraction, and educational visualization.

Drone Overview: The collection includes 28 quadcopters, one tricopter, two hexacopters, and one tail-sitter UAV. The majority feature standard X-frame quadrotor configurations. Drone platforms include commercial and consumer models from DJI, Autel, Syma, Yuneec, UDI, Hasakee, Holystone, and Hover, as well as two custom-built designs. Notable entries include the *David Tricopter*, a custom-built tricopter with a 34-inch diameter and AfroFlight Naze32 flight controller, and *PhenoBee*, a large-scale hexacopter weighing 23 kg, designed by Ziling Chen and built on the Ardupilot Cube Orange platform.

Recording Sites: Audio recordings were collected in diverse indoor and outdoor environments across three U.S. locations: West Lafayette, Indiana; New Richmond, Indiana; and Charleston, South Carolina. Indoor data from Indiana were acquired in university laboratories, while outdoor recordings were made on a private farm in New Richmond. Charleston-based data were collected in the College of Charleston's Drone Lab at the Harbor Walk Campus (indoor) and from the rooftop of the South Carolina Aquarium parking garage (outdoor). Recordings captured natural environmental noise such as wind, birdsong, and traffic, contributing to a realistic audio corpus.

Recording Equipment: From 2021 to 2023, data were recorded using a MacBook Air (1.1GHz quad-core Intel Core i5, 8GB RAM) with the system's internal microphone. Beginning in 2024, recordings were made with an updated MacBook Air featuring an Apple M3 chip and 16GB of memory. No external microphones or post-processing techniques were used, preserving the raw acoustic characteristics of each drone.

This dataset underpins the visual and analytical tools presented in this study, including the expanded web-based interface for exploring drone-specific acoustic features such as MFCCs and spectrograms.

3.2. Visualization Dataset Creation

The project's implementation uses Librosa [4] to compute the Mel Frequency Cepstral Coefficient (MFCC). Our number of mfcc were set to 20 (n-mfcc, FFT window size to 2048 (n-fft), overlap between frames 512 (hop length), and the number of mels to 128 (n-mels). The mathematical descriptions below reflect what is abstracted in the Librosa package.

Extracting MFCCs from an audio dataset involves several steps. The process begins with digital audio files (i.e. .wav, .mp3, .ogg, etc), representing the raw audio signal. The audio is segmented into short overlapping windows ranging from 20 to 40 milliseconds. To reduce signal noise, the Hanning window function is applied, it is mathematically given as used by Harris [26]:

$$w[n] = \frac{1}{2}[1 - cos(\frac{2\pi n}{N})], \text{ for } 0 \le n \le N - 1.$$

where $w[n]$ is the Hanning window function and N is the total number of windows to be computed. Note that we are using zeroth indexing in the function above. This improves the accuracy of the following feature representations, by smoothing the signal with the $1 - cos(\frac{2\pi n}{N})$ term.

Next, the Short-Time Fourier Transform is applied (STFT). Specifically, the continuous-time STFT is applied. It can mathematically be given as:

$x(t, \omega)$:

$$\text{STFT}_x(t, \omega) = \int_{-\infty}^{\infty} x(\tau)\, w(\tau - t)\, e^{-j\omega\tau}\, d\tau$$

The STFT is computed for each windowed frame. Using the $w(\tau - t)$ time-centered windowing function segments the raw signal $x(\tau)$ onto the STFT's sinusoidal basis function $e^{-j\omega\tau}$. the $\text{STFT}_x(t, \omega)$. If computation stopped at this step, the plot would be a spectrogram.

Next the mel-scale is applied. Which was developed to mimic the human perception of hearing. It isn't critical to model performance, but is consistent in related literature [2], [6]. Mathematically the mel-scale can be written as:

$$m(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right)$$

The mel-scale transform converts frequencies from Hertz (Hz) to those in the mel-scale (mels). If computation stopped, the plot would be a mel-spectrogram

From here the approach uses triangular filter banks to calculate the relative amplitude of the frequencies. Shown is the piecewise definition as used in Hang Xu et al. [27]:

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \frac{k - f[m-1]}{f[m] - f[m-1]} & f[m-1] \le k \le f[m] \\ \frac{f[m+1] - k}{f[m+1] - f[m]} & f[m] \le k \le f[m+1] \\ 0 & k > f[m+1] \end{cases}$$

m is the mel filter index and k is the index for the frequency bin. If computation stopped here, the plot would be considered a mel-filterbank.

In the final step, the processed signal is log-scaled and then passed through the Discrete Cosine Transform (DCT) of the mel log signal. Specifically the DCT-II formalization, which is standard for audio processing. Mathematically it is defined as:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \qquad \text{for } k = 0, \ldots N - 1 .$$

The formulation of the augmented base cosine function allows the cosine function $\cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]$ to give unique information for each frequency component, lending itself for an efficient orthogonal representation without waste.

The DCT transform converts N time/spatial samples into N frequency coefficients: $[x_0, x_1, \ldots, x_{N-1}] \rightarrow [X_0, X_1, \ldots, X_{N-1}]$.

After applying the DCT, the MFCC is computationally complete, resulting in a compact and rich representation of an original audio signal.

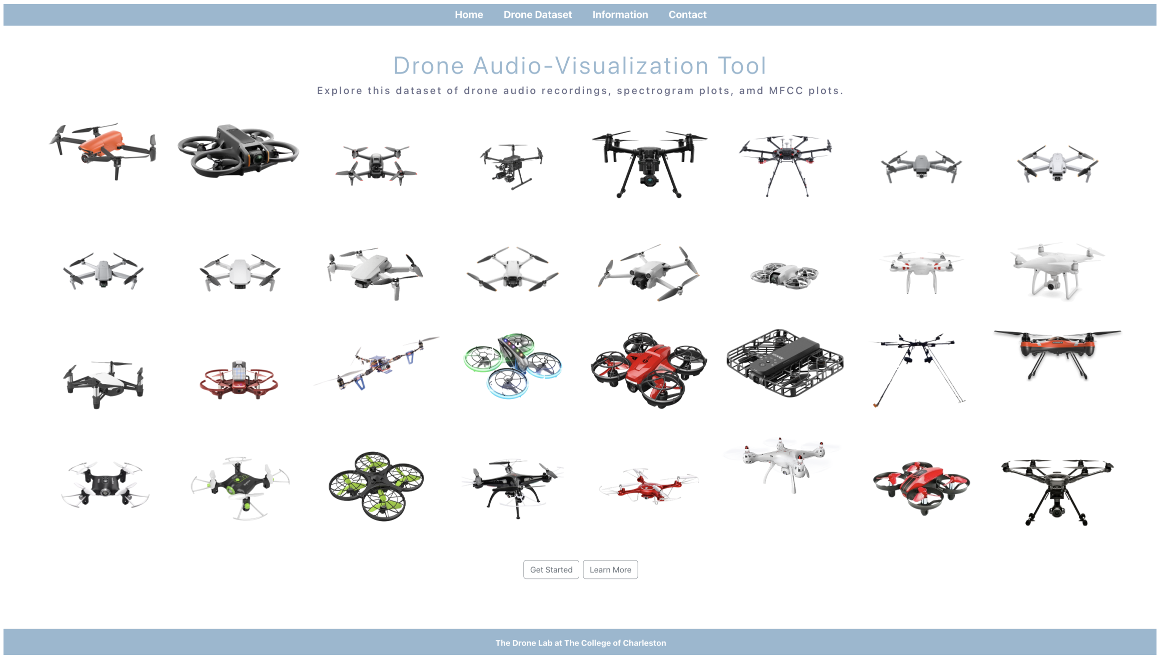
We have generated a total of 3,200 MFCC plots extracted from audio recordings across 32 categories, with each category containing 100 audio files. These MFCC plots serve as feature-rich representations of the acoustic characteristics captured from the audio data; essential for further analysis and classification tasks, reflecting the unique acoustic signatures of various UAV drone audio recordings.

3.3. Web Application Development

The Drone Audio Visualization Tool is an interactive web application designed to enhance the exploration and analysis of a drone audio dataset. Its user interface (UI) is designed so that users can intuitively navigate the application, explore the dataset, and gain meaningful insight into drone audio patterns. The publicly available website can be accessed at: https://mackenzie-jane.github.io/drone-visualization/#/.

Users begin by opening the homepage, which provides an overview of the project and a selection of all 32 drone images. This serves as a visual entry point into the dataset and facilitates quick orientation between pages. Figure 1 illustrates the layout of the home page, which shows the drone images and basic information.

From the homepage, users can navigate to the drone dataset page, which features a responsive grid layout of 32 cards, one for each drone in the dataset. Each card includes the drone's name and image, enabling quick identification and selection. This intuitive and visually appealing layout helps users quickly identify and select their drone of interest.



Drone Audio-Visualization Tool

Explore this dataset of drone audio recordings, spectrogram plots, and MFCC plots.

Figure 1: Audio Visualization Tool Web Application Home Page

Upon selecting a drone, users are redirected to the drone detail page. This page presents a cohesive view of the attributes of the selected drone, including its name, image, an audio recording sample, and two visualization: a Mel-Frequency Cepstral Coefficient (MFCC) plot and a spectrogram plot. These visualizations are generated at a randomly selected frame, offering a snapshot of the drone's acoustic signature. Figure 2 shows an example of the drone information and plots. The layout ensures that the visualizations, drone image, and audio information are presented in a cohesive way. This enables users to simultaneously see and hear key characteristics of each drone, supporting both qualitative and quantitative analysis of drone sound profiles.

This website is built using the React framework to structure and render dynamic components for each drone. In addition, CSS is used to style the interface and ensure responsiveness across devices. JavaScript enables interactivity, such as page transitions and dynamic content loading. Media files, drone images, MFCC plots, spectrograms, and audio files are stored in organized subdirectories

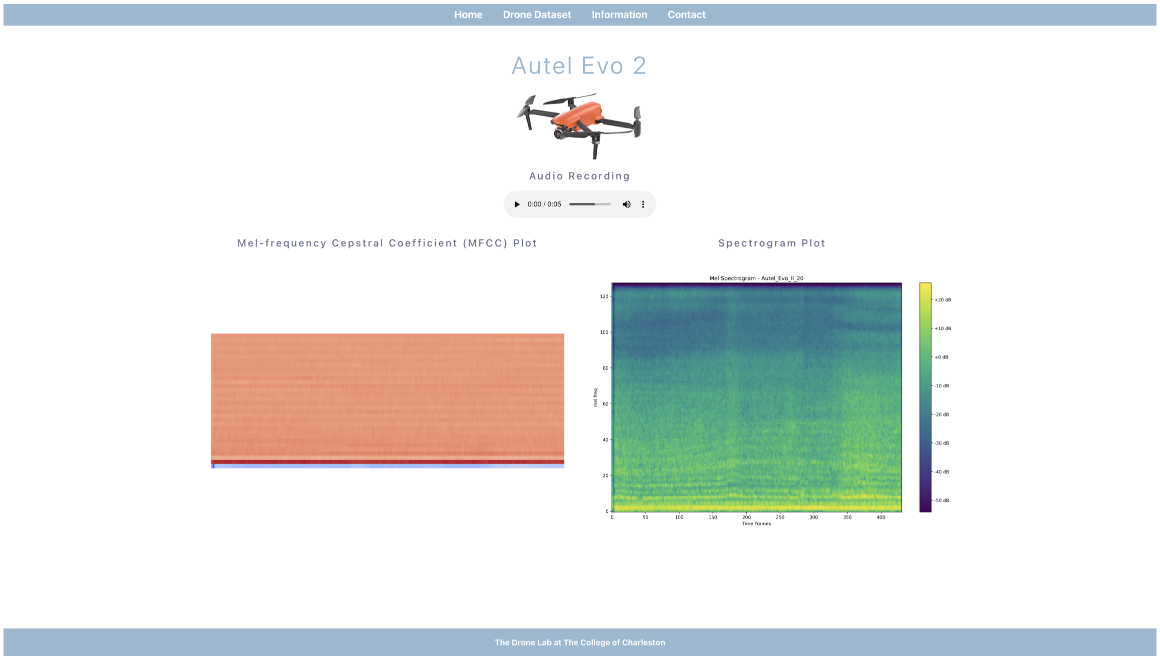


Figure 2: Audio Visualization Tool Drone Display Page

within the public folder. A structured JSON file maps drone identifiers to their corresponding media and metadata. This architecture supports efficient and dynamic content loading.

The dataset includes audio recordings, spectrogram images, and MFCC plots for various drones at specific frames ranging from 0 to 100. A Python script was used to randomly select frame numbers between 0 and 100 for each drone. These random frame indices are used to extract specific audio segments and the corresponding MFCC and spectrogram plots for each drone. This approach ensures an unbiased and varied sampling, which is useful for identifying distinguishing acoustic features across different drones.

Using static file storage and dynamic content loading, the web application provides an efficient and user-friendly platform for drone audio visualization. This website improves accessibility to the dataset and supports further research in drone classification and analysis. The source code for the Drone Audio Visualization Tool is publicly available on GitHub at: `https://github.com/mackenzie-jane/drone-visualization`.

4. Conclusion

The rapid expansion of UAV usage across commercial, industrial, and public domains has intensified concerns related to privacy, security, and noise pollution, exposing limitations in existing visual- and radar-based detection systems. This work contributes to addressing these challenges by advancing acoustic-based drone analysis through the development of a large-scale, multi-class UAV audio dataset and an accompanying interactive visualization platform. Together, these contributions provide a foundation for systematic study of drone acoustic signatures and support the development of robust detection and classification methods.

Beyond the dataset itself, the interactive web application enables intuitive exploration of drone acoustic characteristics by integrating audio playback with MFCC and spectrogram visualizations. This platform lowers the barrier to entry for researchers, educators, and students, supporting both analytical investigation and educational use. The experimental results and positive user feedback demonstrate the utility of acoustic representations for distinguishing drone models and highlight the effectiveness of visualization-driven analysis in understanding complex audio patterns.

Future work will focus on expanding the dataset to include additional drone models and operating conditions, enhancing the web application's functionality, and integrating more advanced machine learning techniques for automated detection and classification. Comparative studies across acoustic, visual, and multimodal datasets will also be explored to better understand the strengths and limitations of each sensing modality. Through these continued efforts, this research aims to support scalable, reliable, and accessible solutions for addressing the growing challenges associated with widespread UAV deployment.

References

[1] M. Y. Wang, D. C. Ramirez, E. Noonan, M. Linn, Q. Zhang, "A Comprehensive Dataset and Visualization Tool for Drone Acoustic Signatures," in 2024 Artificial Intelligence x Humanities, Education, and Art (AIxHEART), 13–17, IEEE, 2024, doi:10.1109/AIxHeart62327.2024.00009.

[2] Y. Wang, F. E. Fagian, K. E. Ho, E. T. Matson, "A feature engineering focused system for acoustic uav detection," in 2021 Fifth IEEE International Conference on Robotic Computing (IRC), 125–130, IEEE, 2021, doi:10.1109/IRC52146.2021.00031.

[3] Y. Wang, F. E. Fagiani, K. E. Ho, E. T. Matson, "A Feature Engineering Focused System for Acoustic UAV Payload Detection," in ICAART (3), 470–475, 2022, doi:10.5220/0010843800003116.

[4] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, O. Nieto, "librosa: Audio and music signal analysis in python," in Proceedings of the 14th python in science conference, volume 8, 18–25, Citeseer, 2015, doi:10.25080/Majora-7b98e3ed-003.

[5] S. Jeon, J.-W. Shin, Y.-J. Lee, W.-H. Kim, Y. Kwon, H.-Y. Yang, "Empirical study of drone sound detection in real-life environment with deep neural networks," in 2017 25th European Signal Processing Conference (EUSIPCO), 1858–1862, IEEE, 2017, doi:10.23919/EUSIPCO.2017.8081531.

[6] Y. Seo, B. Jang, S. Im, "Drone detection using convolutional neural networks with acoustic stft features," in 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 1–6, IEEE, 2018, doi:10.1109/AVSS.2018.8639425.

[7] S. Al-Emadi, A. Al-Ali, A. Al-Ali, "Audio-Based Drone Detection and Identification Using Deep Learning Techniques with Dataset Enhancement through Generative Adversarial Networks," volume 21, 4953, Multidisciplinary Digital Publishing Institute, 2021, doi:10.3390/s21154953.

[8] J. Kim, M. Y. Wang, E. T. Matson, "Self-supervised drone detection using acoustic data," in 2023 Seventh IEEE International Conference on Robotic Computing (IRC), 67–70, IEEE, 2023, doi:10.1109/IRC59093.2023.00018.

[9] C. Aker, S. Kalkan, "Using deep networks for drone detection," in 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 1–6, IEEE, 2017, doi:10.1109/AVSS.2017.8078539.

[10] A. Rozantsev, V. Lepetit, P. Fua, "Detecting flying objects using a single moving camera," volume 39, 879–892, IEEE, 2016, doi:10.1109/TPAMI.2016.2564408.

[11] D. Lee, W. G. La, H. Kim, "Drone detection and identification system using artificial intelligence," in 2018 International Conference on Information and Communication Technology Convergence (ICTC), 1131–1133, IEEE, 2018, doi:10.1109/ICTC.2018.8539442.

[12] G. J. Mendis, T. Randeny, J. Wei, A. Madanayake, "Deep learning based doppler radar for micro UAS detection and classification," in MILCOM 2016-2016 IEEE Military Communications Conference, 924–929, IEEE, 2016, doi:10.1109/MILCOM.2016.7795448.

[13] B. K. Kim, H.-S. Kang, S.-O. Park, "Drone classification using convolutional neural networks with merged Doppler images," volume 14, 38–42, IEEE, 2016, doi:10.1109/LGRS.2016.2624820.

[14] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, I. Guvenc, "Micro-UAV detection and classification from RF fingerprints using machine learning techniques," in 2019 IEEE Aerospace Conference, 1–13, IEEE, 2019, doi:10.1109/AERO.2019.8741970.

[15] C. Zhao, M. Shi, Z. Cai, C. Chen, "Detection of unmanned aerial vehicle signal based on Gaussian mixture model," in 2017 12th International Conference on Computer Science and Education (ICCSE), 289–293, IEEE, 2017, doi:10.1109/ICCSE.2017.8085504.

[16] B. Sazdić-Jotić, B. Bondžulić, I. Pokrajac, J. Bajčetić, M. Mohammed, "Drone classification based on radio frequency: techniques, datasets, and challenges," in Conference papers, 10th International Scientific Conference on Defensive Technologies (OTEH 2022), 2022.

[17] J. Kim, Q. Zhang, E. T. Matson, M. Y. Wang, "Improving Drone Classification with Audio-Derived Visual Features: A Vision Model Comparison," in 2024 Eighth IEEE International Conference on Robotic Computing (IRC), 41–45, IEEE, 2024, doi:10.1109/IRC63610.2024.00013.

[18] M. Y. Wang, Z. Chu, I. Ku, E. Cho Smith, E. T. Matson, "A 15-Category Audio Dataset for Drones and an Audio-Based UAV Classification Using Machine Learning," International Journal of Semantic Computing, 1–16, 2024, doi:10.1142/S1793351X24300048.

[19] M. Wang, Z. Chu, C. Entzminger, Y. Ding, Q. Zhang, "Visualization and Interpretation of Mel-Frequency Cepstral Coefficients for UAV Drone Audio Data," in Proceedings of the 13th International Conference on Data Science, Technology and Applications, 528–534, 2024, doi:10.5220/0012827400003756.

[20] V. Lostanlen, A. Cramer, J. Salamon, A. Farnsworth, B. M. Van Doren, S. Kelling, J. P. Bello, "BirdVox: Machine listening for bird migration monitoring," bioRxiv, 2022–05, 2022, doi:10.1101/2022.05.31.494155.

[21] L. Moheit, J. D. Schmid, J. M. Schmid, M. Eser, S. Marburg, "Acoustics Apps: Interactive simulations for digital teaching and learning of acoustics," The Journal of the Acoustical Society of America, **149**(2), 1175–1182, 2021, doi:10.1121/10.0003438.

[22] S. D. Monache, P. Polotti, D. Rocchesso, "A toolkit for explorations in sonic interaction design," in Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound, AM '10, Association for Computing Machinery, New York, NY, USA, 2010, doi:10.1145/1859799.1859800.

[23] S. Arabasi, H. Al-Taani, D. Ü. Kapanadze, "A visual and interactive learning tool: frequency content of sound waves," in EDULEARN18 Proceedings, 10719–10724, IATED, 2018, doi:10.21125/edulearn.2018.2627.

[24] M. Tawil, A. Dahlan, "Application of interactive audio visual media to improve students' creative thinking skill," in Journal of Physics: Conference Series, volume 1752, 012076, IOP Publishing, 2021, doi:10.1088/1742-6596/1752/1/012076.

[25] Y. Wang, Z. Chu, I. Ku, E. C. Smith, E. T. Matson, "A Large-Scale UAV Audio Dataset and Audio-Based UAV Classification Using CNN," in 2022 Sixth IEEE International Conference on Robotic Computing (IRC), 186–189, 2022, doi:10.1109/IRC55401.2022.00039.

[26] Harris, "On the Use of the Windows fro Harmonic Analysis with the Discrete Fourier Transform," 1978.

[27] A. Huang, Hon, Spoken Language Processing: A guide to Theory, Algorithm, and System Development, Prentic Hall, 2001.