ASTES

Advances in Science, Technology & Engineering Systems Journal

VOLUME 11-ISSUE 2|MAR-APR 2026

www.astesj.com
ISSN: 2415-6698

Editorial

Advances in Science, Technology and Engineering Journal (ASTESJ) remains committed to publishing high-quality interdisciplinary research that advances scientific understanding and addresses contemporary technological challenges. The papers presented in this issue demonstrate the transformative role of artificial intelligence, probabilistic computing, data science, and cybersecurity in solving complex real-world problems. From educational analytics and theoretical computational frameworks to intrusion detection and intelligent security operations, these contributions reflect the growing integration of innovative methodologies with practical applications across diverse domains.

Educational data analytics continues to benefit from advances in explainable artificial intelligence, as demonstrated by an interpretable gradient boosting framework for student performance prediction. By integrating SHAP-based feature selection, hybrid resampling, cost-sensitive decision thresholds, and optimized hyperparameter tuning, the proposed methodology achieves high predictive performance while maintaining model transparency and fairness. The ability to identify influential factors affecting academic achievement provides valuable support for evidence-based educational policies and personalized learning strategies, illustrating the increasing importance of interpretable machine learning in academic environments [1].

Novel computational paradigms are explored through the introduction of HLLSet, an enhanced probabilistic data structure capable of supporting complete set operations while representing semantic relationships without storing explicit elements. Built upon category-theoretic principles and supported by the Bell State Similarity metric, the proposed framework establishes a rigorous mathematical foundation for contextual representation, directed similarity, and self-regulating system dynamics. The demonstrated conservation properties and sheaf-theoretic formulation offer promising directions for the evolution of intelligent systems and probabilistic knowledge representation [2].

The increasing complexity of Internet of Things environments demands intrusion detection systems capable of handling high-dimensional, imbalanced, and heterogeneous network data with both accuracy and computational efficiency. A hybrid feature selection strategy combining filter and embedded methods with weighted feature fusion demonstrates exceptional detection performance across multiple benchmark IoT datasets. The integration of statistical feature ranking with Random Forest feature importance significantly enhances classification effectiveness, highlighting the value of optimized feature engineering for securing next-generation connected infrastructures [3].

Reliable cybersecurity operations require intrusion detection frameworks that extend beyond conventional threat recognition by addressing distribution shifts, operational reliability, and false-positive reduction. A decision-centric architecture integrating deep representation learning, graph neural networks, XGBoost classification, and meta-learning provides an effective solution for Security Operations Centers. The inclusion of SHAP-based interpretability further enhances transparency by offering meaningful explanations for detection decisions, while extensive evaluation demonstrates robustness across both time-series and cross-dataset scenarios. This contribution represents a significant advancement toward trustworthy and operationally effective intrusion detection systems for modern cyber defense environments [4].

The research contributions presented in this issue illustrate the continued convergence of artificial intelligence, mathematical modeling, data analytics, and cybersecurity in addressing complex

scientific and engineering challenges. Collectively, these studies advance theoretical knowledge while delivering practical solutions with significant societal and industrial relevance. It is anticipated that the innovative methodologies and insights presented will stimulate further interdisciplinary research and contribute to the ongoing development of intelligent, secure, and data-driven technologies.

References:

[1] A. El Arid, G. Nassreddine, L. Saleh, M. Al Majzoub, “An Ensemble Learning Approach for Student Performance Analysis of a Higher Educational Institute using a SHAP-Based Feature Selection and Optuna Optimization,” Advances in Science, Technology and Engineering Systems Journal, **11**(2), 1–11, 2026, doi:10.25046/aj110201.

[2] A. Mylnikov, “HLLSet Theory: A Unified Framework for Probabilistic Knowledge Representation,” Advances in Science, Technology and Engineering Systems Journal, **11**(2), 12–16, 2026, doi:10.25046/aj110202.

[3] M.S.S. Moe, W.M. Oo, “Hybrid Feature Selection for Anomaly Detection in IoT Network Intrusion Detection Systems,” Advances in Science, Technology and Engineering Systems Journal, **11**(2), 17–29, 2026, doi:10.25046/aj110203.

[4] I. Lotfi, M. Mandar, “TL-SOC: A Hybrid Decision-Centric Intrusion Detection Framework for Security Operations Centers,” Advances in Science, Technology and Engineering Systems Journal, **11**(2), 30–42, 2026, doi:10.25046/aj110204.

Editor-in-chief

Prof. Hamid Mattiello

ADVANCES IN SCIENCE, TECHNOLOGY AND ENGINEERING SYSTEMS JOURNAL

Volume 11 Issue 2 March-April 2026

CONTENTS

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 11, No. 2, 1-11 (2026)
www.astesj.com
Special Issue on Multidisciplinary Frontiers in Engineering, Computing and Applied Sciences

ASTES Journal
ISSN: 2415-6698

An Ensemble Learning Approach for Student Performance Analysis of a Higher Educational Institute using a SHAP-Based Feature Selection and Optuna Optimization

Amal El Arid[1], Ghalia Nassreddine[*2], Loubna Saleh[3], Mohamad Al Majzoub[4]

[1]*Electrical and Computer Engineering Department, Rafik Hariri University, Meshref, Lebanon*

[2]*Department of Financial Studies, Rafik Hariri University, Meshref, Lebanon*

[3]*Department of Human Resources, Saint Joseph University of Beirut, Beirut, Lebanon*

[4]*Department of Management and Marketing Studies, Rafik Hariri University, Meshref, Lebanon*

Email(s): aridaa@rhu.edu.lb (A. El Arid), nassreddinega@rhu.edu.lb (G. Nassreddine), loubna.saleh@net.usj.edu.lb (L. Saleh), majzoubm@rhu.edu.lb (M. Al Majzoub)

[*]Corresponding Author: Ghalia Nassreddine, Rafik Hariri University, Mechref, Lebanon, 00961-71-512012 & Email: nassreddinega@rhu.edu.lb

ARTICLE INFO

Article history:
Received: 21 January, 2026
Revised: 07 March, 2026
Accepted: 10 March 2026
Online: 19 March, 2026

Keywords:
Student performance
Higher education
Ensemble learning
AI-powered tools
Explanaible AI
SHAP feature selection
Optuna optimization

ABSTRACT

Forecasting and assessing student performance are crucial for allowing educators to pinpoint deficiencies and promote grade improvement. A thorough comprehension of feature contributions is crucial for improving model interpretability and facilitating informed decision-making in academic institutions. Explainable artificial intelligence encompasses methodologies and strategies designed to deliver transparent and accessible rationales for the decisions rendered by artificial intelligence and machine learning algorithms. In this research paper, an interpretable gradient boosting approach for predicting student performance is introduced, including both feature selection using SHapley Additive exPlanations (SHAP)-based features and cost-sensitive decision thresholds. The proposed methodology includes hybrid resampling with SMOTE-Tomek, Optuna hyperparameter optimization with stratified cross-validation, and SHAP-guided feature selection strategy. The proposed approach is tested using a dataset of a higher educational institute in the Middle East, including student information, learning management, and a video interaction system, to make an analysis and evaluate the performance of the students. The results show both improvement of the macro F1-score and the fail-class recall by achieving an accuracy of 94%, a weighted/macro F1-score of 0.9399, and a fail-class recall of 0.9619. The suggested method facilitates trade-offs among prediction accuracy, interpretability, and fairness, bridging the divide between high-performing machine learning models and practical educational applications, hence aiding in the formulation of data-driven policies and the customization of learning experiences.

1. Introduction

As of 2026, artificial intelligence (AI) has outgrown its status as an experimental quack and become a key component of modern education [1]. It is a powerful force multiplier to teachers because it automated administrative processes like grading and scheduling and provided students with learning paths that are highly customized to their individual pace and style in real-time [2].

With the implementation of AI, the classroom will turn into an interactive environment in which the insights that are data-driven will determine gaps in learning before they grow, and AI-based co-pilots will tutor students round-the-clock [3]. To transform this technology into a beneficial rather than a distracting factor, it is necessary to focus on five core pillars by ensuring the schools have a robust digital infrastructure (high-speed connectivity and

hardware), thorough artificial intelligence literacy training of staff and pupils, clear ethical standards that guarantee the privacy of data and academic integrity, fair access to prevent socioeconomic imbalance, and a firm dedication to the human touch that is at the heart of the learning process [4, 5, 6]. Education has been radically transformed with the introduction of agentic AI systems, moving beyond the integration of limited digital tools. In the modern scholarly environment, AI is not perceived as an additional tool anymore; instead, it is a backbone of Education 4.0. This increase is defined by the one-size-fits-all approach to massive individualization. AIs are also finding their way into schools as a way of dealing with the twin problems of teacher shortages and classroom diversity [7]. Clearing cognitive processes (diagnostic testing and paperwork) for intelligent systems, educational establishments are trying to recapture the human aspect of teaching. However, this rapid scaling has outpaced the standardization of implementation protocols, resulting in a fragmented environment where the effectiveness of artificial intelligence varies significantly across different socioeconomic contexts.

There are two different and yet not completely separate spheres that form the technical foundation of this educational revolution: machine learning and deep learning [8].

1.1. Machine Learning and Deep Learning

Machine learning (ML) is the key force behind the learning analytics. Researchers can use previous student data to identify patterns which are not visible to the human eye through supervised learning algorithms, such as random forests, support machines or gradient boosting. For instance,

- Early warning systems [9]: ML models are intended to accurately identify the students who are at-risk of failure based on attendance, engagement metrics, and quiz scores.
- Adaptive sequencing [10]: In an unsupervised learning case, students are clustered together and the platforms is capable of dynamically updating the content difficulty on-the-fly.

ML concerns itself with the patterns, whereas deep learning (DL) concerns itself with the complexity of human interaction, including:

- Natural language processing [11]: Chatbots and virtual tutors based on the DL can provide more comprehensive feedback on open-ended essays and cease being based on multiple-choice evaluation.
- Computer vision [12]: Computer vision algorithms are now under study in physical or hybrid classrooms to both quantify student interest in the form of posture and eye-tracking and provide teachers with a heat map of classroom interest.
- Generative Architecture [13]: Using transformer-based architectures, educators can produce custom lesson plans and multi-sensory (text-to-video or text-to-image) content in a few seconds according to specific curriculum requirements.

Academic performance by students is one of the key measures of learning development, as it is influenced by such factors as gender, age, teaching staff, and learning conditions [14]. Academic achievement is a field of interest in the field of education that has been investigated through forecasting. Prediction and evaluation of student performance play an essential role in helping teachers clarify the deficiencies and improve the grades. On the same note, students are able to enrich their learning activities, and administrators improve their operations [15]. The recent advancements in educational data mining and artificial intelligence have made it much more accurate to predict the performance of students in respect to their academic results [16]. The studies have explored various machine learning and deep learning techniques, such as clustering-based feature detection and optimization of the classifier, convolutional neural networks with ensemble learning, generative models with deep support vectors, behavioral modeling (processes), and stacked ensemble approaches to failure prediction [17]. Different machine learning and deep learning systems were studied, such as clustering-based feature identification and optimization of the classifier; convolutional neural networks with ensemble learning; generative models with deep support vectors; process-based behavioral modeling; and stacked ensemble methods of failure prediction. Despite the achievements, there remain significant gaps in methodology that require filling. However, few studies have compared the effect of different features on the performance of machine learning models regarding predicting student performance. The overall perspective of the contributions of features is important to enhance the interpretability of models and make informed decisions in academic institutions. Explainable AI (XAI) is a collection of techniques and practices intended to provide easy-to-understand explanations of AI and ML models' decisions. An explainability layer can be added to these models, which allows building more reliable and open systems to support students and other educational stakeholders [18]. XAI has a couple of advantages in its implementation. It provides the decision-makers and stakeholders with a clear representation of the reasons that underlie the AI-driven decisions so that they are informed to make a decision. It is also useful in finding possible biases or limitations in the models, such as identifying skewed training data or algorithmic biases, and, as a result, provides more correct and equal results.

This paper presents a reliable, understandable, and data-driven framework that is both transparent and educational. The proposed approach will facilitate the informed academic decision-making process, as well as early detection of at-risk students, by combining the model interpretability and performance-aware decision-making strategies. On the whole, the article points out the value of predictive effectiveness, fairness, and explainability in using machine learning methods in education.

The remainder of this paper is organized as follows: Section 2 provides the background and related concepts of the study domain, Section 3 presents the proposed methodology in detail, and Section 4 reports and discusses the experimental results, followed by a results comparison with other methods. Finally, Section 5 concludes the paper and outlines future research directions.

2. Background

The utilization of AI-driven tools is rapidly changing the landscape of the educational process, redefining the approaches to the construction of the teaching, learning, and evaluation processes.

These technologies help educators to experience personalized learning and adaptive feedback and make decisions based on data. With their increasing adoption, it is necessary to know their impact and quality on education. The necessity to enhance the rankings of universities and employability of the students has led to the adoption of data mining and AI in learning to give a more accurate prediction of academic success [19]. The author used K-means clustering to identify the important factors of performance in his study and compared many classifiers. His findings indicated that a tuned support vector machine was one of the best alternatives, with the highest prediction accuracy of 96%. Conversely, the Naive Bayes classifier had the lowest accuracy, which is due to the fact that it provided an assumption about the independence of the feature. The study established that hyperparameter optimization greatly optimized the performance of extrapolative models in educational data mining.

In [20], the authors have introduced a machine learning structure that utilizes features extracted using the convolutional neural network to predict student academic achievement through the use of the "MOODLE" data. To overcome the imbalance of the population in the classes, the synthetic minority oversampling method was applied. The offered model had a classification accuracy of 99.9%. The variation in their approach was based on a combination of deep convoluted features and an additional tree classifier, which surpassed existing state-of-the-art methods.

In [21], the authors introduced a better model that integrated a conditional generative adversarial network and a deep support vector machine to predict the academic performance of students. The model overcame the limitations of small datasets by generating synthetic mockups and focused on features of home and school teaching. The findings showed that the integration of tutoring factors achieved the best results. In addition, the multi-kernel deep SVM also outperformed the existing models across some of its key metrics, such as accuracy and processing time, demonstrating its effectiveness in handling diverse educational data.

The authors of [22] provided a process-based behavior categorization and prediction model that could be used to predict e-learning performance. The estimated methodological split the learner interactions through process-behavior classification model and combined the features through a behavior classification-based e-learning performance framework. The empirical findings of their research on the dataset of OULAD showed more advanced projecting accuracy compared to traditional ones.

The authors of the study [16] proposed a study, which combined cluster analysis, discriminant analysis, and convolutional neural networks to estimate and predict academic results so that the possible errors of traditional approaches to the evaluation of students can be properly addressed. The innovative statistical metric was conducted to control the number of clusters in the K-means clustering in an accurate manner to enhance the consistency of prediction. The effectiveness of the proposed approach was determined with the help of model validation through the use of cross-validation methods.

A combination of an AI-based academic performance forecasting model enhanced collaborative online learning by using learning analytics, which was critical [23]. A quasi-experimental design was therefore designed to correlate student engagement, performance and satisfaction between supported and unsupported clusters. Findings indicated that the integrated model produced positive student engagement, improved learning, and satisfaction levels.

Furthermore, a stacked ensemble machine learning strategy, which combines the use of three models, namely, Random Forest, extreme gradient boosting (XGBoost), and feed-forward neural networks, was estimated to predict university student failure [24]. The ensemble was found to be more accurate and have the highest AUC than single models, experienced on university data. The method enabled initial detection of the vulnerable students, which enabled one to intervene in time by the education people.

The authors of [25] designed an ML and XAI framework on student career counseling that analyzed academic and employability data to maintain career judgments. The White Box as well as the Black Box prototypes have been trained in an educational dataset in their study. The greatest performance in the tested models was achieved with the Naive Bayes when recall was used and the scores of the F-measure were 91.2% and 90.7%, respectively.

A research in [26] inspected the recognition of the effect of AI and social media within the academic performance and mental health of university students, enabled through smart learning. Using the data collected in the course of the surveys of 401 students in China, the findings were clear that both AI and social media are positively associated with academic achievement and psychological well-being. In addition, the study also found that smart learning strengthened these beneficial relations to a great extent. These results highlighted the fruitful nature of digital technologies in modern education in the form of smart learning methods.

Moreover, scholars also tend to embrace SHAP (SHapley Additive exPlanations), a game-theoretical model created in to offer an analytically sound method of describing individual forecasts [27]. SHAP allocates the payout (the prediction of the model) fairly among all the input features (e.g. the previous grades of a student, his or her socioeconomic status, or engagement measures) by the marginal contribution to the end result. In contrast to other global metrics of importance, SHAP provides both global interpretability (what factors contributed to the overall performance of the school) and local interpretability (why was a particular student named as being at-risk). This two-fold power is the key to educational equity, as it will enable teachers to audit models and learn the logic of AI-based interventions in specific detail, thus building trust and responsibility in an online classroom [28].

2.1. Gaps and Motivations

Recent developments in educational data mining and artificial intelligence have significantly increased the accuracy of projecting the academic performance of students as discussed above. A number of studies have looked into different ML and DL approaches, including, but not limited to, clustering-based feature discovery, optimized classifiers [19], convolutional neural networks with ensemble learning [20], generative models with deep support vector machines [21], process-based behavioral modeling [22], and stacked ensemble classifiers in failure prediction [24]. Despite these contributions, some significant gaps in the methodology remain that should still be addressed. Nonetheless, there still remain certain gaps in the latest studies, including:

- Researchers do not pay much attention to model explainabil-

ity. The majority of the studies in this area focus on predictive performance measures, such as accuracy and AUC, yet they fail to indicate the influence of individual features on the model decision making. Even Guleria and Sood [25] proposed explainable AI methods in educational data mining; but they analyzed them primarily in terms of conventional classifiers and did not introduce explainability in high-performing ensemble architectures.

- Feature selection techniques can be heuristic or unspecified. Earlier works are based on feature engineering, clustering, or deep feature extraction [19, 22, 16]. None of the studies examined applies SHAP values as a systematic selection feature of features, which might contribute to better generalization, reduce redundancy, and simplify the analysis of the results.

- Techniques of managing class imbalance are not discussed. Although techniques such as synthetic minority over-sampling technique (SMOTE) and conditional generative adversarial networks have been employed to construct synthetic data have been applied [20, 21], these techniques do not entirely remove bias in favor of the majority class. In the majority of studies, they apply set decision thresholds, which may render it extremely difficult to identify minority classes. This is particularly significant in the school environment where it is necessary to know at a quick time those students who are likely to fail or are failing in order to be able to assist them.

- Most of the evaluation methods are accuracy based. Reported accuracies can also mask poor performance on minority classes, especially where there is an imbalanced dataset. Optimization goals are rarely used (such as metrics such as macro F1-score and the fail-class recall, which are more useful to predict educational risk), and threshold-aware validation is generally absent.

To overcome these shortcomings, the present paper recommends an explainable gradient boosting model of SHAP-based feature selection and cost-sensitive decision thresholding. The proposed method includes hybrid resampling using SMOTE-Tomek, hyperparameter optimization using Optuna and stratified cross-validation, and a SHAP-based feature selection method. Additionally, there is a threshold optimization approach; which is precision-recall based to maximize the macro F1-score and improve fail-class recall. The given approach ensures the equilibrium of predictive performance, transparency, and practical educational information due to the combination of explainability and cost-aware decision-making into the learning pipeline, which contributes to the state of the art in student performance prediction.

3. Methodology

The proposed approach is presented in Algorithm 1, composed of five main steps: Data loading and preprocessing, feature enhancement and hybrid resampling strategy, hyper-parameter optimization via Optuna with stratified cross-validation, model training and SHAP-based feature selection, and model training, threshold-aware validation, testing and explainability.

Algorithm 1: An Explainable Gradient Boosting Algorithm with SHAP-Based Feature Selection and Cost-Sensitive Decision Thresholding

Result: Optimized XGBoost classifier, selected feature subset, optimal decision threshold, and validated performance metrics.

START

STEP 1: Data Loading and Preprocessing
- Load dataset D;
- Remove duplicates and missing values;
- Encode target labels $y \epsilon 0, 1$;
- Label-encode categorical features;
- Standardize numerical features;
- Split D: 70% stratified training and 30% testing sets;

STEP 2: Feature Enhancement and Hybrid Resampling Strategy
- Apply feature engineering on training and testing sets;
- Apply SMOTE–Tomek on training data;
- Obtain balanced set D_{train}^{res};

STEP 3: Hyperparameter Optimization via Optuna with Stratified Cross-Validation
- Initialize Optuna study;
- while Optuna trial $t \leq T$ do
  - Sample XGBoost hyperparameters;
  - Initialize Stratified K-Fold cross-validation;
- for each fold $k = 1toK$ do
  - Train XGBoost model on training fold;
  - Predict labels on validation fold;
  - Compute Macro F1-score;
- end
- Return mean Macro F1-score;
- Select optimal hyper-parameters θ^*

STEP 4: Model Training and SHAP-Based Feature Selection
- Train XGBoost model using θ^* on D_{train}^{res};
- Compute SHAP values for all features;
- Rank features by mean absolute SHAP importance;
- Select top M features;
- Predict class probabilities on holdout test set;
- Compute Precision–Recall curve;
- Determine optimal threshold τ^* maximizing F1-score;

STEP 5: Model Training, Threshold-Aware Validation, testing and Explainability
- Initialize Stratified K-Fold CV;
- for each fold $k = 1toK$ do
  - Train XGBoost on selected features;
  - Predict probabilities;
  - Apply threshold τ^*
  - Compute $Accuracy$, $MacroF1$, $Recall$;
- end
- Train final model on full resampled training set;
- Predict on holdout test set;
- Generate SHAP summary and bar plots;

END

3.1. Data Loading and Preprocessing

After collecting data from different sources, a preprocessing step should be performed to clean and standardize the dataset. In this study, the data go through two steps of processing that consist of removing duplicate records and samples that contained missing values to achieve both data purity and model training bias protection.

The target variable, denoted as y, represents student performance results and should be encoded using a label encoder to give 1 for passed students and 0 for failed ones. The label encoding technique encodes the categorical labels (string labels) into numeric values, allowing ML algorithms to use them. The different categories of a feature are allocated different integers, and the uniqueness of the classes is maintained without suggesting any rank.

The feature set, called X, was divided into categorical and numerical features. Categorical features are transformed into numeric using the label encoding technique. However, the numerical features are standardized using z-score normalization that produced features with equal scales and better model training results [29]. The z-score normalization equation is given by:

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

Where x represents the original feature value, μ is the mean of the feature, σ represents the standard deviation of the feature, and x' is the standardized feature value.

After that, the dataset goes through its last preparation stage by applying a 70/30 stratified split, which maintains the original class distribution between training and testing sets. The training set was used for model development, training, feature engineering, feature selection, and cross-validation. The testing set is employed later for model performance evaluation.

3.2. Feature Enhancement and Hybrid Resampling Strategy

To enhance the predictive power of the model, feature engineering was performed on the training and test sets. In both cases, more numerical characteristics were calculated by taking the row-wise mean and standard deviation of the initial numerical characteristics. In addition, the interaction terms between the three number features that have the strongest relationship with the target variable were also generated to capture possible nonlinear relationships. Formally, if x_j and x_k are two chosen numerical features; the interaction feature is the one that is determined as [30]:

$$x_{jk}^{interaction} = x_j \times x_k \tag{2}$$

Once the feature engineering had been completed, a hybrid SMOTE-Tomek resampling strategy was employed to handle the class imbalance in the training set. The Synthetic Minority Oversampling Technique (SMOTE) was used to create more samples of the minority class, and then Tomek links were post-removal to remove borderline majority samples. The result of this process was a balanced set of training, called D_trainers, with a better modelization of the minority class of "Fails," the future training of models becomes less biased.

3.3. Hyperparameter Optimization via Optuna with Stratified Cross-Validation

The XGBoost classifier is used in this study to train the model with a hyperparameter tuning step performed using the Optuna tool to reach the highest macro F1-score. XGBoost is a technique of ensemble learning, a subset of ML. It consists of constructing a set of trees, each tree correcting the mistakes of the former ones. It solves a regularized goal function that is a loss term and a complexity term to enhance predictive accuracy and avoid overfitting. Optuna is an automatic hyperparameter tuning tool that proves its efficiency in various applications. It searches for the best parameters based on techniques like the tree-structured Parzen estimator. Using cross-validation, Optuna can produce the hyperparameter set θ^* according to a user-specified objective, like the F1-score.

To manage possible over-fitting risks and to guarantee model generalization, a number of validation strategies were incorporated into the suggested technique. First, in the hyper-parameter optimization, a stratified cross-validation with K folds was used so that the model would not be fit to a particular subset of training data and provide a more accurate estimate of the performance on the different cross-validation folds. Furthermore, the data was divided into 70/30 stratified train test split whereby the test was entirely concealed during the training and model selection. Furthermore, bias reduction is ensured solely through resampling (SMOTE-Tomek) applied to the training data only without information leakage into the test set.

3.4. Model Training and SHAP-Based Feature Selection

XGBoost is retrained based on τ^* on the training set. Let $D_{train}^{res} = (X_i, y_i)_{i=1}^{N}$ represents the resampled training set. The XGBoost model predicts the probability of class c for sample i as:

$$\hat{y}_i^c = softmax(\Sigma_{k=1}^{K} f_k(X_i)), f_k \epsilon F \tag{3}$$

where F denotes the space of regression trees, and K is the number of boosting iterations [31]. The final predicted class is obtained as:

$$\hat{y}_i = \underset{c}{\operatorname{argmax}}\, \hat{y}_i^c \tag{4}$$

To understand better the contributions of each feature, SHAP values were calculated for all features in X_{train}^{res} [32].

For feature j and instance i, the SHAP value ϕ_{ij} describes the contribution of feature j to the model output. It is given by:

$$\hat{y}_i = \phi_0 + \Sigma_{j=1}^{M} \phi i j \tag{5}$$

where ϕ_0 is the expected model output and M is the total number of features. Features were ranked by the mean absolute SHAP value in all instances:

$$SHAP_j^{mean} = \frac{1}{N} \Sigma_{i=1}^{N} |\phi i j| \tag{6}$$

The top M features with the highest mean SHAP values were selected for used later to retrained the XGBoost classifier while reducing dimensionality and conserving the most influential predictors. The number of selected features M is computed based on

the ranking of mean absolute SHAP importance values to retain only the most influential variable contributing to the model output.

After training, the model was applied to the holdout test set. To enhance the performance of the minority "Fail" class, the decision threshold τ^* was optimized by maximizing the F1-score, using:

$$\tau^* = \underset{\tau}{\text{argmax}}\, F1 - score(\tau) \tag{7}$$

where F1-score is given later in this section. The final binary predictions were calculated by:

$$\hat{y}_i = \begin{cases} 1, & if\ \hat{p}_i > \tau^* \\ 0, & otherwise \end{cases} \tag{8}$$

This structure guarantees the maximum prediction accuracy and interpretability, permitting the identification of the most influential features while adapting the classification threshold for imbalanced classes.

3.5. Model Training, Threshold-Aware Validation, Testing, and Explainability

To assess the performance of the proposed model, a stratified K-fold cross-validation CV was conducted on the resampled training set D^{res}_{train}, with the top M features provided by SHAP [33].

XGBoost was trained on the training partition in each fold ($k = 1, etc., K$) and applied to the classification probabilities of the validation partition. The minimized threshold (τ^*) and Equation (8) were used to convert the projected probabilities $\hat{p}_i$ into binary class predictions. A binary classification problem can be defined in the form of a confusion matrix (CM) as shown in Table 1, where TP is true positives, TN is true negatives, FP is false positives, and FN is false negatives [34].

Table 1: Confusion Matrix

Actual \ Predicted	Positive (1)	Negative (0)
Positive (1)	TP	FN
Negative (0)	FP	TN

From CM, the following metrics are calculated:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

Accuracy defines the ratio of the rightly classified instances of all the instances and the overall correctness of the model. Precision is the ratio of the number of true positive instances among the total number of cases that are predicted to be positive, and this measures how well positive predictions are made. Recall (or sensitivity) is a display of the percentage of the real positive instances identified, which indicates the capability of the model to identify the positive category. The F1-score is the harmonic average of the precision and the recall measures, which will give a balance between the two measures.

Averages of these metrics that are cross-validated provide a good estimate of expected performance on unseen data. After that, the holdout test set predictions were conducted to check the generalization abilities of the model. For Explainability, SHAP values were calculated on the test set in order to measure how much each feature contributed to the model predictions. SHAP values quantify the contribution that each feature has towards the prediction of an instance with a positive or negative contribution. The values of the SHAP were visualized by creating summary and bar plots to demonstrate the most significant features and understand the decision-making process of the model and have interpretable predictions both of the majority of the class, Pass, and the minority, Fail.

4. Results and Discussion

4.1. Dataset Description

The dataset used in this paper is collected from [35]. This dataset includes information from 326 students at Middle East College in Oman, gathered between Spring 2017 and Spring 2021. It gathers data from three different educational platforms, including Student Information System (SIS), Moodle, and eDify. The dataset encompasses 23 features, including CGPA, coursework scores, time spent on Moodle on and off campus, and video engagement metrics like play, pause, and likes. All data were anonymous and standardized. Dataset features are illustrated in Table 2, where 'Result' is the target variable.

Table 2: Features of the adopted dataset

Feature	Description	Type
ModuleCode	Unique code identifying the academic module	Categorical (Nominal)
ModuleTitle	Official title of the academic module	Categorical (Nominal)
SessionName	Academic session or semester in which the module is offered	Categorical (Nominal)
ApplicantName	Anonymized identifier of the student	Categorical (Nominal)
CGPA	Cumulative Grade Point Average of the student prior to the module	Numerical (Continuous)
AttemptCount	Number of attempts taken by the student to pass the module	Numerical (Discrete)

Table 2: Features of the adopted dataset

Feature	Description	Type
RemoteStudent	Indicates whether the student is enrolled as a remote learner (Yes/No)	Categorical (Binary)
Probation	Indicates whether the student is on academic probation	Categorical (Binary)
HighRisk	Indicates whether the student is classified as high academic risk	Categorical (Binary)
TermExceeded	Indicates whether the student exceeded the expected program duration	Categorical (Binary)
AtRisk	Indicates prior failure in two or more modules	Categorical (Binary)
AtRiskSSC	Indicates whether the student is registered with the Student Success Center	Categorical (Binary)
OtherModules	Number of other modules taken by the student in the same semester	Numerical (Discrete)
PlagiarismHistory	Level or count of previous plagiarism-related incidents	Numerical (Discrete)
CW1	Marks obtained in Coursework 1	Numerical (Continuous)
CW2	Marks obtained in Coursework 2	Numerical (Continuous)
ESE	Marks obtained in the End-Semester Examination	Numerical (Continuous)
Online C	Time spent on Moodle activities within campus (minutes)	Numerical (Continuous)
Online O	Time spent on Moodle activities outside campus (minutes)	Numerical (Continuous)
Played	Number of times video learning content was played	Numerical (Discrete)
Paused	Number of times video playback was paused	Numerical (Discrete)
Likes	Number of likes given to video learning content	Numerical (Discrete)
Segment	Number of replayed or segmented video interactions	Numerical (Discrete)
Result	Final academic outcome of the module (Pass/Fail)	Categorical (Target Variable)

4.2. Results

The performance of the proposed XGBoost-based approach based on hybrid resampling, SHAP-based feature selection, and threshold-based validation will be presented. First, the hyperparameter tuning step with Optuna led to the optimization of the XGBoost configuration balanced between learning capacity and regularization. The resulting model uses 500 estimators with a depth of 4 and a low learning rate (0.019), allowing the model to converge consistently without overfitting, as illustrated in Table 3.

The subsampling value further enhances the generalization. Additionally, the selected values of gamma and minimum child weight could limit the unnecessary growth of trees.

Table 3: XGBoost Hyper-parameters resulting from Optuna step

Hyperparameter	Value
n_estimators	500
max_depth	4
learning_rate	0.0190
subsample	0.8716
colsample_bytree	0.8325
gamma	0.6009
min_child_weight	2

To increase the minority-class detection, the precision-recall curve was used to optimize the decision threshold. The best value of the threshold τ in achieving failing students is 0.60. This threshold value is delivered from deep analysis of PR. Indeed, by increasing the threshold, the model shifts toward higher sensitivity for the "Fail" class, enhancing recall metric while maintaining balanced precision metric.

Table 4 illustrates the performance of the optimized model on the holdout test set.

Table 4: Performance metrics of the proposed approach

Class	Precision	Recall	F1-score	Support
Fail (0)	0.87	0.68	0.76	19
Pass (1)	0.93	0.97	0.95	79
Macro Avg	0.90	0.83	0.86	98
Weighted Avg	0.92	0.92	0.91	98
Accuracy	—	—	0.9184	98

The model attained a satisfactory accuracy of 91.84%. The macro F1-score of 0.86 means that both classes have shown an equal performance. Although the Pass class was forecasted with significant reliability, the Fail class was forecasted with a recall of 0.68, indicating the natural challenge of minority-class forecasting even with aggressive oversampling and threshold optimization.

To further measure robustness, threshold-dependent stratified 5-fold cross-validation was performed in Table 5 that shows a high level of performance in every fold with a high accuracy of over 89% and a 100% fail-class recall. Also, it summarizes the results with a mean accuracy of 94.00%, a mean macro F1-score of 0.9399, and a mean fail-class recall of 0.9619, validating the high and consistent minority-class detection.

Table 5: Stratified 5-Fold Cross-Validation Results and Mean Performance

Fold \ Metric	Accuracy	Macro F1-score	Fail-class Recall
Fold 1	0.8955	0.8954	0.9355
Fold 2	0.9701	0.9701	0.9688
Fold 3	0.9403	0.9402	0.9375
Fold 4	0.9242	0.9242	0.9677
Fold 5	0.9697	0.9697	1.0000
Mean	0.9400	0.9399	0.9619

Accordingly, these findings show that the proposed framework

has high predictive accuracy with a significantly higher Fail-class recall, which is an important factor in early academic risk identification. The consistency in the cross-validation folds validates the strength of the threshold-conscious and explainable learning approach.

4.3. Explainability SHAP Analysis

SHAP was used to examine the role of each feature in the predictions of the proposed XGBoost model to provide the transparency and interpretability of the suggested predictive framework. As shown in Figure 1, the SHAP summary bar plot ranks features by their mean absolute SHAP values, that is, by their overall average effect on the model output. It can be seen in the analysis that 'ESE' is the most powerful characteristic, and this means that the performance of the final examination is the most powerful determinant in the decision-making process of the model. This fact aligns with the academic assessment systems in which the end-term tests play an important role in determining the overall results. The large value of SHAP in 'ESE' proves that the model predictions are heavily informed by indicators that have an important academic meaning.

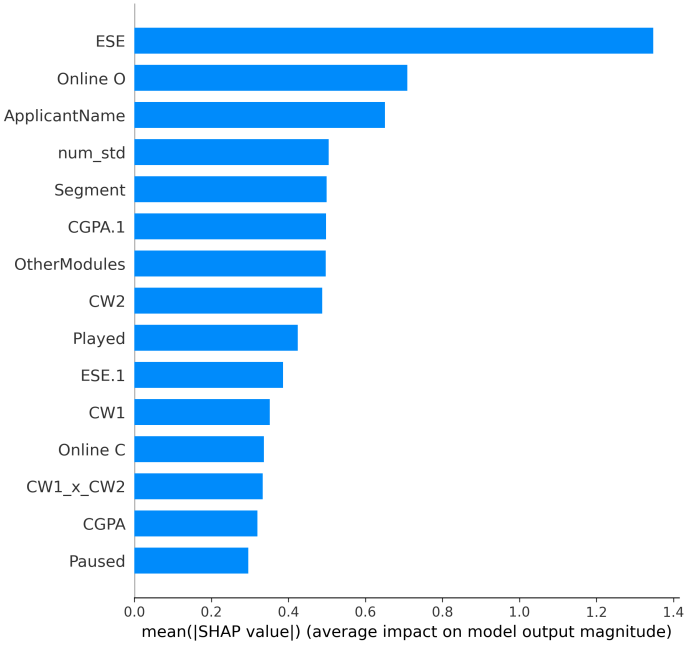

Figure 1: SHAP summary bar plot ranks

The variables of online engagement, specifically 'Online O' (off-campus online activity), also show significant significance. This underlines the importance of prolonged learning activities beyond the classroom system and implies that vigorous interaction with the learning platforms is one of the main predictors of student success. These findings highlight the applicability of online learning behavior in the current educational analytics.

Coursework-related features ('CW1' and 'CW2') help improve the predictive power, which supports the significance of continuous assessment in the context of predicting the initial performance trends. In addition, the interaction feature ('CW1' × 'CW2') reveals that the interaction of the elements of coursework will give complementary information over the scores of both categories, and this proves the efficacy of feature interaction engineering. Video learning activities such as 'Played', 'Paused', and 'Segment' are behavioral features with moderate and consistent effects. These characteristics mirror the consumption patterns and the level of engagement of students, which provide highly informative behavioral data that add to the traditional academic data. 'OtherModules' as a feature and indicator of concurrent academic workload also adds to the interpretability of the model by reflecting the influence of course load performance. Built-in statistical characteristics like 'num_mean' and 'num_std' describe the trends and variability of academic performance on a global scale, which improves the capacity of the model to describe latent patterns in a model with many attributes.

It is noteworthy that 'ApplicantName', even though it was anonymous and label-encoded, is one of the influential features. This shows an admitted weakness of the encoding of identifiers in tree-based models: that encoded representations can induce unwanted patterns. Although such an aspect enhances predictive accuracy, it is crucial to exercise caution in interpretability, which underscores the significance of responsible feature selection in explainable ML.

Figure 2 shows a global response of feature contributions to the predictive outcome, both regarding the importance of these contributions and the direction of the same.

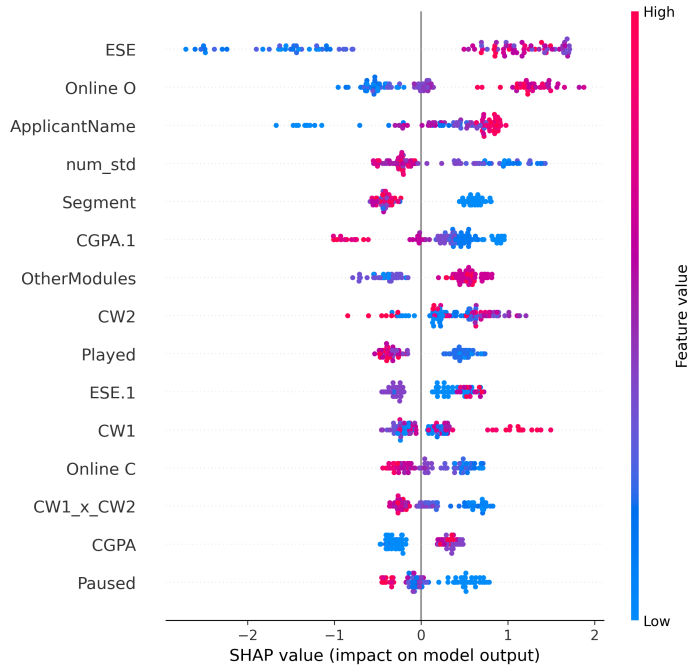

Figure 2: SHAP summary plot demonstrating the overall significance and directional influence of the features of input used on the model to predict student performance. The points depict values of SHAP of each feature-instance combination, where color denotes feature value (low to high).

It shows that the End-Semester Evaluation - 'ESE' - plays the greatest role in predicting the performance of students. The increase in 'ESE' values results in increases in positive SHAP value, which implies a large contribution to desirable academic perfor-

mance, and the decrease in 'ESE' score significantly decreases the predicted performance. This observation is in line with the theory of educational assessment, where cumulative assessment includes the learning consistency and the acquisition of knowledge.

'Online O', 'Played', and 'Online C' seem to have some considerable impact on engagement features as well. The beneficial impact on the model predictions of high levels of online communication and content interaction underlines the importance of online interaction in the contemporary learning settings. Conversely, characteristics relating to disengagement mechanisms ('Paused' and increased variability in engagement 'num_std') have a negative impact, which indicates that disproportionate and/or discontinuous learning behaviors are linked to worse academic performance. The continuous assessment elements, such as 'CW1', 'CW2', and 'OtherModules', present strong positive contributions to their values in the case of high values. The interaction aspect 'CW1' × 'CW2' also illustrates that the balanced performance in the coursework assessments is synergistic in nature, and hence the value of long-term academic performance over short-term performance. The findings support the argument that the model successfully identifies the nonlinear associations between assessment elements.

Variables in student profiling - 'Segment' - have mixed SHAP, which implies that there is heterogeneity in student learning patterns across different groups. This result justifies the necessity of individual learning interventions and accommodating academic support systems. Interestingly, although 'CGPA.1' is historically a very powerful predictor, its impact is relatively weak, implying that historical academic performance is not as effective as the impact may have in the situations when more behavioral and assessment data is present. The fact that 'ApplicantName' is one of the crucial features indicates the existence of latent institutional records or structural or contextual information. Although this can enhance predictive accuracy, it also introduces both ethical and fairness concerns, which underscores the need to be cautious about which features are introduced and who controls education AI systems. In case with the prediction regarding the performance of students, names can be used as the proxies of the sensitive variables such as ethnicity, gender, or regionality. To make the research and testing efficient, the 'ApplicantName' field was retained to evaluate and learn more about how the model is sensitive to such possible biases. Therefore, the measures of fairness can be evaluated to ensure that the algorithm can be deployed as an instrument of equitable educational assistance.

The explainability analysis with the use of SHAP proves that the suggested model is based on academic and behaviorally relevant characteristics mainly, which makes the prediction readable and consistent with the knowledge in the educational domain. SHAP is able to offer global feature importance as well as transparent rationale of predictions, increasing the confidence in the model and facilitating its use in educational decision-support systems, especially to provide early warning of at-risk learners.

4.4. Comparison and Discussion

Table 6 shows a comparison between the proposed approach and existing systems for the proposed method (PM) in this study vs. other methods: random forest (RF), extra trees (ET), gradient boosting (GB), gaussian Naïve Bayes (GNB), stochastic gradient descent (SGD), and logistic regression (LR). As illustrated, the proposed explainable gradient boosting model has the best performance in all the evaluation metrics with an accuracy of 94%, a weighted macro F1-score of 0.9399, and a fail-class recall of 0.9619, which are better than all the baseline models.

Table 6 also illustrates the 95% confidence intervals (CI) delivered by all ML models to provide an estimate of the variability of models' performance for the dataset. The proposed method achieves very narrow CI values, indicating stable and consistent predictive performance. This small CI confirms the robustness and generalization capability of the proposed framework compared with the other ML methods.

The more conventional ensemble approaches (RF, ET, GB) are equally competitive but with much lower fail-class recall, suggesting that they are not as sensitive to at-risk students. Four main reasons may be associated with the better performance of the proposed approach, namely:

1. XGBoost optimized with Optuna that is capable of capturing nonlinear interactions between features;

2. a SMOTE-Tomek hybrid resampling mechanism that controls class imbalance;

3. SHAP-based feature selection, which leads to efficient generalization;

4. optimization of the decision threshold by costs, which focuses on the detection of the fail class.

Combining these elements allows predicting student performance accurately, balancing it, and explaining the results, which is why the proposed approach can be highly effective when applied to early warning and academic decision-support systems.

In addition to the performance benefits, the given framework has high applicability to the educational decision-making process. The high fail-class recall means that most of the students at risk are properly identified, and this is essential in timely academic interventions and early warning systems. Furthermore, integration of SHAP-based explainability enables educators and administrators to know why a student is predicted to fail, allowing them to support him/her in specific ways and not in general ways.

The proposed method allows trade-offs between predictive accuracy, interpretability, and fairness to fill the gap between high-performing machine learning models and real-world educational application, helping to formulate data-driven policies and the personalization of learning journeys. This method requires approximately 14.5 million operations to learn 333 samples each with 37 features with 392 trees of maximum depth 3, which demonstrates its efficiency in computing with this sized dataset. Such findings supported the fact that the suggested model could deliver high predictive performance and be scaled-up and implemented on regular educational data. Besides, the implementation of the model requires approximately 0.54 seconds, which is very computationally efficient. Therefore, it is the right solution, which is scalable and applicable in real-time or near-real-time applications in education.

Table 6: Performance Comparison of the proposed approach and existing systems, where PM – Proposed Model; RF – Random Forest; ETC – Extra Trees Classifier; GBM – Gradient Boosting Machine; GNB – Gaussian Naïve Bayes; SGD – Stochastic Gradient Descent classifier; LR – Logistic Regression.

Method \ Metric	Accuracy (Mean ± CI)	Macro F1-score (Mean ± CI)	Fail-class Recall (Mean ± CI)
PM	0.9400 ± 0.025	0.9399 ± 0.025	0.9619 ± 0.021
RF	0.9050 ± 0.030	0.9100 ± 0.028	0.8800 ± 0.035
ETC	0.8990 ± 0.031	0.9000 ± 0.030	0.8800 ± 0.034
GBM	0.8990 ± 0.032	0.9000 ± 0.031	0.8700 ± 0.036
GNB	0.7350 ± 0.041	0.7400 ± 0.040	0.7300 ± 0.043
SGD	0.6910 ± 0.045	0.6900 ± 0.046	0.6000 ± 0.050
LR	0.6720 ± 0.047	0.6700 ± 0.048	0.6700 ± 0.046

5. Conclusions

Forecasting and evaluating the performances of the students is a process that is necessary since it helps the teachers to realize the areas they have weaknesses and motivate their students to work hard to elevate their grades. In the context of making the models more interpretable and simplifying the process of academic institutions making knowledgeable decisions, the full picture of the contributions made by features is a must. XAI is used to refer to a collection of methods and strategies that are aimed at offering rationales that are both clear and easily available to the judgments that are made by the machine learning algorithms and artificial intelligence. In this research study, a gradient boosting strategy that is interpretable is introduced with the aim of predicting the performance of students. The strategy integrates the feature selection by using SHAP-based features, as well as the decision thresholds that are sensitive to the possible costs. The methodology that has been proposed includes hybrid resampling using SMOTE-Tomek, Optuna-hyperparameter optimization using stratified cross-validation, and SHAP-guided feature selection method. The proposed approach is tested with the help of a dataset of a higher educational establishment, a situated area in the Middle East. This data includes data about students, learning management, as well as a video interaction system. This testing is meant to carry out an analysis and test the performance of the students. Due to the achievement of the accuracy of 94%, the weighted/macro F1-score of 0.9399, and the fail-class recall of 0.9619, the results indicate that the macro F1-score and the fail-class recall have improved. Due to the fact that the proposed approach allows establishing trade-offs between the prediction accuracy, interpretability, and fairness, it allows closing the gap between machine learning models with high performance and practical educational outcomes. Consequently, it assists in the development of data-based policies and the personalization of learning. Although the proposed approach demonstrates high predictive capability, the sample size is not large; as well as the dataset only covers one higher education institution in the Middle East region. This weakness may influence the externalization of the results. Thus, prospective research must be conducted in larger and more varied datasets to test the model on the dissimilar institutional contexts and remove the possible bias, which the current dataset can have.

Conflict of Interest The authors declare no conflict of interest.

References

[1] C. Koukaras, S. G. Stavrinides, E. Hatzikraniotis, M. Mitsiaki, P. Koukaras, C. Tjortjis, "Navigating the future of education: A review on telecommunications and AI technologies, ethical implications, and equity challenges," in Telecom, volume 7, 2, MDPI, 2026, doi:10.3390/telecom7010002.

[2] P. Naayini, "AI and the future of education: Advancing personalized learning and intelligent tutoring systems," Frontiers in Educational Innovation and Research, **1**(1), 29–39, 2025, doi:10.62762/FEIR.2025.332098.

[3] S. Arora, A. Rajesh, R. Misra, G. Singh, "Bridging technology and trust: the role of AI-driven robo-advisors in middle-class financial management," Management Decision, 1–24, 2025, doi:10.1108/MD-01-2025-0093.

[4] T. Heafner, D. Maxwell, "CIVIC: Five pillars for using artificial intelligence in social studies education," in Society for Information Technology & Teacher Education International Conference, 2581–2589, Association for the Advancement of Computing in Education (AACE), 2025.

[5] R. Daher, "Integrating AI literacy into teacher education: a critical perspective paper," Discover Artificial Intelligence, **5**(1), 217, 2025, doi:10.1007/s44163-025-00475-7.

[6] G. Nassreddine, L. Saleh, M. Al Majzoub, A. El Arid, "SHAP explainability: An ensemble learning approach for student performance prediction," in 2025 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), 432–438, IEEE, 2025, doi:10.1109/IAICT65714.2025.11100707.

[7] Y. Luo, G. Zhou, Y. Cui, "Understanding generative artificial intelligence adoption in higher education faculty: evidence from Chinese Universities and technical and vocational colleges," Education and Information Technologies, 1–41, 2026, doi:10.1007/s10639-025-13885-y.

[8] S. Sengupta, S. Chakrabarti, "Towards a smarter education system: an investigation into ML and DL for information retrieval," Multimedia Tools and Applications, 1–34, 2025, doi:10.1007/s11042-025-20656-x.

[9] W. Cao, N. Mai, "Predictive analytics for student success: AI-driven early warning systems and intervention strategies for educational risk management," Educational Research and Human Development, **2**(2), 36–48, 2025.

[10] I. Gligorea, M. Cioca, R. Oancea, A.-T. Gorski, H. Gorski, P. Tudorache, "Adaptive learning using artificial intelligence in e-learning: A literature review," Education Sciences, **13**(12), 1216, 2023, doi:10.3390/educsci13121216.

[11] Hariyanto, F. X. D. Kristianingsih, R. Maharani, "Artificial intelligence in adaptive education: a systematic review of techniques for personalized learning," Discover Education, **4**(1), 458, 2025, doi:10.1007/s44217-025-00908-6.

[12] S. Rawat, M. Rodrigues, P. Sheregar, K. A. Wagaskar, A. K. Tripathy, "Computer vision based hybrid classroom attention monitoring," in 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), 1–6, IEEE, 2024, doi:10.1109/ICITEICS61368.2024.10624965.

[13] G. Zhang, "Transformer-based AI framework for optimising English teaching evaluation strategies: a data-driven and explainable approach," International Journal of Information and Communication Technology, **26**(9), 107–127, 2025, doi:10.1504/IJICT.2025.145828.

[14] G. Gokmen, T. Ç. Akinci, M. Tektaş, N. Onat, G. Kocyigit, N. Tektaş, "Evaluation of student performance in laboratory applications using fuzzy logic," Procedia-Social and Behavioral Sciences, **2**(2), 902–909, 2010, doi:10.1016/j.sbspro.2010.03.124.

[15] Y. A. Alsariera, Y. Baashar, G. Alkawsi, A. Mustafa, A. A. Alkahtani, N. Ali, "Assessment and evaluation of different machine learning algorithms for predicting student performance," Computational intelligence and neuroscience, **2022**(1), 4151487, 2022, doi:10.1155/2022/4151487.

[16] G. Feng, M. Fan, Y. Chen, "Analysis and prediction of students' academic performance based on educational data mining," IEEE Access, **10**, 19558–19571, 2022, doi:10.1109/ACCESS.2022.3151652.

[17] M. Abdasalam, A. Alzubi, K. Iyiola, "Student grade prediction for effective learning approaches using the optimized ensemble deep neural network," Education and Information Technologies, **30**(8), 10159–10183, 2025, doi:10.1007/s10639-024-13224-7.

[18] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, R. Ranjan, "Explainable AI (XAI): Core Ideas, Techniques, and Solutions," ACM Comput. Surv., **55**(9), 2023, doi:10.1145/3561048.

[19] E. Ahmed, "Student performance prediction using machine learning algorithms," Applied computational intelligence and soft computing, **2024**(1), 4067721, 2024, doi:10.1155/2024/4067721.

[20] N. Abuzinadah, M. Umer, A. Ishaq, A. Al Hejaili, S. Alsubai, A. Eshmawi, A. Mohamed, I. Ashraf, "Role of convolutional features and machine learning for predicting student academic performance from MOODLE data," Plos one, **18**(11), e0293061, 2023, doi:10.1371/journal.pone.0293061.

[21] S. Sarwat, N. Ullah, S. Sadiq, R. Saleem, M. Umer, A. Eshmawi, A. Mohamed, I. Ashraf, "Predicting students' academic performance with conditional generative adversarial network and deep SVM," Sensors, **22**(13), 4834, 2022, doi:10.3390/s22134834.

[22] F. Qiu, G. Zhang, X. Sheng, L. Jiang, L. Zhu, Q. Xiang, B. Jiang, P.-k. Chen, "Predicting students' performance in e-learning using learning process and behaviour data," Scientific Reports, **12**(1), 453, 2022, doi:10.1038/s41598-021-03867-8.

[23] F. Ouyang, M. Wu, L. Zheng, L. Zhang, P. Jiao, "Integration of artificial intelligence performance prediction and learning analytics to improve student learning in online engineering course," International Journal of Educational Technology in Higher Education, **20**(1), 4, 2023.

[24] J. Niyogisubizo, L. Liao, E. Nziyumva, E. Murwanashyaka, P. C. Nshimyumukiza, "Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization," Computers and Education: Artificial Intelligence, **3**, 100066, 2022, doi:10.1016/j.caeai.2022.100066.

[25] P. Guleria, M. Sood, "Explainable AI and machine learning: performance evaluation and explainability of classifiers on educational data mining inspired career counseling," Education and Information Technologies, **28**(1), 1081–1116, 2023, doi:10.1007/s10639-022-11221-2.

[26] M. F. Shahzad, S. Xu, W. M. Lim, X. Yang, Q. R. Khan, "Artificial intelligence and social media on academic performance and mental well-being: Student perceptions of positive impact in the age of smart learning," Heliyon, **10**(8), 2024, doi:10.1016/j.heliyon.2024.e29523.

[27] S. M. Lundberg, S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017.

[28] S. R. Rebeka, R. Thomas, "Fostering Engagement and Trust in E-Learning Communities Through Social Media Platforms," in Innovative Approaches to Social Media in Education, chapter 11, 241–255, IGI Global, 2025, doi:10.4018/979-8-3693-3868-1.ch011.

[29] P. Muthulakshmi, M. Parveen, "Z-score normalized feature selection and iterative African buffalo optimization for effective heart disease prediction," International Journal of Intelligent Engineering & Systems, **16**(1), 2023.

[30] T. Yin, H. Chen, Z. Yuan, J. Wan, K. Liu, S.-J. Horng, T. Li, "A robust multilabel feature selection approach based on graph structure considering Fuzzy dependency and feature interaction," IEEE Transactions on Fuzzy Systems, **31**(12), 4516–4528, 2023, doi:10.1109/TFUZZ.2023.3287193.

[31] I. L. Cherif, A. Kortebi, "On using extreme gradient boosting (XGBoost) machine learning algorithm for home network traffic classification," in 2019 Wireless Days (WD), 1–6, 2019, doi:10.1109/WD.2019.8734193.

[32] H. Wang, Q. Liang, J. T. Hancock, T. M. Khoshgoftaar, "Feature selection strategies: a comparative analysis of SHAP-value and importance-based methods," Journal of Big Data, **11**(1), 44, 2024, doi:10.1186/s40537-024-00905-w.

[33] G. Nassreddine, A. El Arid, M. Nassereddine, O. Al Khatib, "Fault detection and classification for photovoltaic panel system using machine learning techniques," Applied AI Letters, **6**(2), e115, 2025, doi:10.1002/ail2.115.

[34] G. Nassreddine, A. El Arid, M. Nasseredine, "Solar PV Power Prediction System Based on Machine Learning Approach," in 2023 IEEE International Conference on Energy Technologies for Future Grids (ETFG), 1–7, 2023, doi:10.1109/ETFG55873.2023.10407291.

[35] R. Hasan, S. Palaniappan, S. Mahmood, A. Abbas, K. U. Sarker, "Dataset of Students' Performance Using Student Information System, Moodle and the Mobile Application "eDify"," Data, **6**(11), 2021, doi:10.3390/data6110110.

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 11, No. 2, 12-16 (2026)
www.astesj.com

ASTES Journal
ISSN: 2415-6698

HLLSet Theory: A Unified Framework for Probabilistic Knowledge Representation

Alex Mylnikov*

Independent Researcher, South Amboy, NJ USA

*Corresponding Author: Alex Mylnikov, South Amboy, NJ USA, Email: alexmy@lisa-park.com

ARTICLE INFO

Article history:
Received: 29 January, 2026
Revised: 26 March, 2026
Accepted: 28 March, 2026
Online: 04 April, 2026

Keywords:
HyperLogLog
Probabilistic Data Structures
Category Theory
Noether's Theorem
Knowledge Representation

ABSTRACT

This paper introduces HLLSet (HyperLogLog Set), a probabilistic data structure that behaves like a set under all standard operations while containing no explicit elements. Unlike traditional HyperLogLog, which only estimates cardinality, HLLSets support full set operations (union, intersection, difference) through enhanced register structures and provide a principled framework for representing semantic relationships. We establish a category-theoretic foundation for HLLSets, where objects are contextual representations and morphisms are directed similarity relations defined by a dual-threshold system (τ for inclusion tolerance, ρ for exclusion intolerance). We introduce Bell State Similarity (BSS), a directed similarity metric that measures the overlap between probabilistic representations. The framework demonstrates that balanced addition and deletion operations on HLLSet-based representations give rise to a discrete conservation law analogous to Noether's theorem, providing a principled steering mechanism for AI system evolution. We formalize HLLSets within a categorical framework and establish that HLLSet collections form sheaves over ϵ-isometry categories, with the condition $|N| - |D| = 0$ serving as a stability criterion that enables self-regulating system dynamics.

1. Introduction

1.1. The Core Challenge

Traditional probabilistic data structures, particularly HyperLogLog [1], excel at cardinality estimation for massive datasets but provide no support for set operations or semantic interpretation. When multiple AI systems need to share knowledge, represent relationships, or evolve over time, the limitations of such structures become acute:

1. **Lack of set operations**: HyperLogLog cannot compute intersections, differences, or unions while maintaining probabilistic guarantees.

2. **Hash incompatibility**: Systems using different hash functions cannot directly compare or combine their representations.

3. **No structural interpretation**: There is no principled way to extract semantic relationships from the underlying hash-based representations.

4. **No evolutionary dynamics**: Traditional structures are static; they cannot adapt or evolve with changing knowledge.

1.2. The HLLSet Solution

We introduce **HLLSet** (HyperLogLog Set), a probabilistic data structure that extends HyperLogLog to support full set operations while maintaining the original's memory efficiency. Unlike traditional HyperLogLog, which stores only the maximum trailing-zero count per register, HLLSets use bit-vectors that enable exact Boolean operations.

The key insight is that HLLSets function as **anti-sets**: they behave like sets under all standard operations yet contain no explicit elements. Instead, each token contributes exactly one bit to a fixed-size fingerprint, and this fingerprint is the object of all subsequent operations. The token-to-bit mapping is deterministic and idempotent: the same token always maps to the same bit, and adding it twice yields no change.

This inversion—from "set of elements" to "element of a space of fingerprints"—enables us to forget tokens entirely and work purely in the algebra of fingerprints. All set operations become bitwise Boolean operations on register vectors, and semantic relationships emerge from the lattice structure of these fingerprints.

1.3. Contributions

This paper makes the following contributions:

1. **HLLSet Definition and Algebra**: Formal definition of HLLSets as fixed-size bit-vectors with idempotent inscription, including rigorous definitions of set operations via Boolean logic.

2. **Bell State Similarity (BSS)**: A directed similarity metric that measures inclusion (τ) and exclusion (ρ) between HLLSets, providing the foundation for categorical morphisms.

3. **Categorical Foundation**: Formalization of HLLSets as objects in a category HLL with morphisms defined by BSS thresholds.

4. **Noether Steering Law**: Proof that balanced addition and deletion operations on HLLSet-based representations yield a discrete conservation law, providing a principled mechanism for system self-regulation.

5. **Ambiguity Resolution Framework**: Multi-seed triangulation and cohomological disambiguation methods for handling hash collisions.

2. HLLSet: Definition and Algebra

2.1. Basic Definition

Definition 1 (HLLSet)**.** *An HLLSet is determined by two global parameters:*

- *m: number of registers (typically a power of two)*
- *b: width of each register in bits (e.g., 8, 16, 32)*

Its state is a vector $\mathbf{R} = (R_1, R_2, \ldots, R_m)$*, where each* $R_i \in \{0,1\}^b$ *is a b-bit integer. Equivalently,* $\mathbf{R}$ *is an* $m \times b$ *bit matrix. The empty HLLSet* $\emptyset$ *has* $\mathbf{R} = \mathbf{0}$*.*

2.2. Token Inscription

Let $\mathcal{T}$ be the universe of tokens (byte strings). For each token $t \in \mathcal{T}$ we compute a hash $h = \mathsf{hash}(t)$ of length at least $\log_2 m + b$ bits.

1. **Register index**: $i = h \bmod m$
2. **Bit position**: Take the remaining bits $q = h \gg \lceil \log_2 m \rceil$ and compute $z = \nu(q)$, the number of trailing zeros in the binary representation of q, capped at $b-1$ (i.e., $z = \min(\nu(q), b-1)$)

The **singleton HLLSet** $\{\!|t|\!\}$ is defined as the HLLSet having a single 1-bit at register i, position z, and zeros elsewhere:

$$(\{\!|t|\!\})_i = 1 \ll z, \qquad (\{\!|t|\!\})_{j \neq i} = 0.$$

Key property: one token, one bit. Every token occupies exactly one bit in the fingerprint. Different tokens may map to the same (i, z) pair—this is a **collision**. Collisions are intrinsic and desirable; they create the controlled ambiguity that enables generalization and the directed similarity measure (BSS).

2.3. Set Operations

HLLSets support all standard set operations via bitwise Boolean logic:

Definition 2 (Set Operations)**.** *For HLLSets A and B with register vectors* $\mathbf{R}_A$ *and* $\mathbf{R}_B$*:*

$$\mathbf{R}_{A \cup B} = \mathbf{R}_A \vee \mathbf{R}_B \quad \textit{(union)} \tag{1}$$

$$\mathbf{R}_{A \cap B} = \mathbf{R}_A \wedge \mathbf{R}_B \quad \textit{(intersection)} \tag{2}$$

$$\mathbf{R}_{A \setminus B} = \mathbf{R}_A \wedge \neg(\mathbf{R}_B) \quad \textit{(difference)} \tag{3}$$

These operations are associative, commutative, and idempotent; they satisfy all usual identities of set algebra.

2.4. Cardinality Estimation

The true cardinality $|A|$ (number of distinct tokens inscribed) can be estimated from the fingerprint. Using the classic HyperLogLog estimator [1]:

$$\widehat{|A|} = \alpha_m m^2 \left(\sum_{i=1}^{m} 2^{-R_i} \right)^{-1},$$

where α_m is a bias correction factor. The estimator is unbiased with relative error $O(1/\sqrt{m})$.

2.5. Tokenization Functor

Let $\mathsf{Seq}(\mathcal{T})$ be the free monoid over tokens (finite sequences). Define $\phi : \mathsf{Seq}(\mathcal{T}) \to \mathbf{HLLSet}$ by:

$$\phi(\varepsilon) = \emptyset, \qquad \phi(t_1 t_2 \ldots t_k) = \{\!|t_1|\!\} \cup \{\!|t_2|\!\} \cup \cdots \cup \{\!|t_k|\!\}.$$

Proposition 1 (Functoriality)**.** *ϕ is a monoid homomorphism:*

$$\phi(s \cdot t) = \phi(s) \cup \phi(t), \qquad \phi(\varepsilon) = \emptyset.$$

3. Bell State Similarity (BSS)

3.1. Definition and Motivation

We introduce a **directed** similarity metric between HLLSets. The name "Bell State Similarity" reflects its role in measuring the "state overlap" between probabilistic fingerprints of sets, quantifying how much one HLLSet's representation is contained within another.

Definition 3 (Bell State Similarity)**.** *For HLLSets A, B with* $|B| > 0$*, define:*

$$\mathrm{BSS}_\tau(A \to B) = \frac{\widehat{|A \cap B|}}{\widehat{|B|}}, \qquad \mathrm{BSS}_\rho(A \to B) = \frac{\widehat{|A \setminus B|}}{\widehat{|B|}}.$$

If $\widehat{|B|} = 0$*, set* $\mathrm{BSS}_\tau = 0$ *and* $\mathrm{BSS}_\rho = 1$*.*

BSS_τ measures the *inclusion* of A in B: what fraction of B's tokens are also in A? BSS_ρ measures the *exclusion*: what fraction of B's tokens are *not* in A? The two metrics together provide a complete picture of the directed relationship.

3.2. Morphism Definition

Given thresholds τ (inclusion tolerance) and ρ (exclusion intolerance) with $0 \leq \rho < \tau \leq 1$, we define a directed relationship:

Definition 4 (HLLSet Morphism)**.** *A morphism $f : A \to B$ exists (denoted $A \xrightarrow{(\tau,\rho)} B$) iff:*

$$\mathrm{BSS}_\tau(A \to B) \geq \tau \quad \textit{and} \quad \mathrm{BSS}_\rho(A \to B) \leq \rho.$$

The conditions ensure that A provides sufficient coverage of B (inclusion) while adding minimal extraneous tokens (exclusion). The identity morphism $1_A : A \to A$ always exists because $\mathrm{BSS}_\tau(A \to A) = 1$ and $\mathrm{BSS}_\rho(A \to A) = 0$.

3.3. Properties

Proposition 2 (Properties of BSS)**.** *For HLLSets A, B, C:*

1. $0 \leq \mathrm{BSS}_\tau(A \to B) \leq 1$, $0 \leq \mathrm{BSS}_\rho(A \to B) \leq 1$
2. $\mathrm{BSS}_\tau(A \to B) + \mathrm{BSS}_\rho(A \to B) \leq 1$ *(with equality if $A \subseteq B$)*
3. $\mathrm{BSS}_\tau(A \to B) = 1$ *iff $A \supseteq B$ (up to estimation error)*
4. $\mathrm{BSS}_\rho(A \to B) = 0$ *iff $A \subseteq B$ (up to estimation error)*

4. Noether Steering Law for System Evolution

4.1. Motivation

Noether's theorem is a mathematical result about symmetries and conservation laws that applies to any dynamical system with a well-defined action. While its most famous applications are in physics, the underlying mathematics is universal. In our context, we construct a discrete dynamical system where state updates have a specific symmetry: adding and removing information in balanced pairs. This symmetry yields a conserved quantity—the net information flux—by an argument that mirrors Noether's theorem but is mathematically self-contained.

4.2. Discrete Dynamics for HLLSet Evolution

Consider a system where HLLSets evolve through two basic operations:

- **Addition**: Insert new tokens into an HLLSet
- **Deletion**: Remove tokens from an HLLSet

Define the state at time t as $\mathbf{R}(t)$. Let:

- $N(t)$: HLLSet of tokens added at time t
- $D(t)$: HLLSet of tokens deleted at time t
- $R(t) = \mathbf{R}(t) \setminus D(t)$: retained tokens

The state transition is:

$$\mathbf{R}(t+1) = R(t) \cup N(t) = [\mathbf{R}(t) \setminus D(t)] \cup N(t).$$

4.3. The Symmetry: Balanced Updates

Consider the transformation that simultaneously adds and deletes the same token pair:

$$\mathbf{R}(t+1) = [\mathbf{R}(t) \setminus \{t\}] \cup \{t\} = \mathbf{R}(t).$$

This is an invariance: adding a token and then immediately deleting it (or vice versa) leaves the state unchanged.

Now consider the net information flux:

$$\Phi(t) = |N(t)| - |D(t)|.$$

Theorem 1 (Noether Steering Law)**.** *If the system evolves through balanced updates where $|N(t)| = |D(t)|$ for all t, then the total cardinality $\sum_i R_i(t)$ is conserved modulo hash collisions.*

Proof. Each addition increases the sum of register counts by exactly 1 (since each token adds exactly one bit). Each deletion decreases the sum by exactly 1. If $|N(t)| = |D(t)|$, the net change is zero:

$$\Delta \sum_i R_i(t) = |N(t)| - |D(t)| = 0.$$

Therefore $\sum_i R_i(t)$ is constant. This is a direct consequence of the idempotence and deterministic inscription properties—no physics is invoked. The result holds up to hash collisions, which affect cardinality estimation but not the underlying bit counts. □

4.4. Practical Implications

The Noether steering law provides a practical monitoring and control mechanism:

1. **System Health Monitoring**: If $\Phi(t)$ drifts from zero, the system is experiencing unbalanced information flow. This can indicate hash collisions, systematic bias in addition/deletion patterns, or external information flux.
2. **Self-Regulation**: The system can adjust its parameters (forgetting rate λ_{forget}, novelty threshold θ) to maintain $\Phi(t) \approx 0$, ensuring stable evolution.
3. **Error Detection**: Sustained deviations from $\Phi = 0$ signal either technical issues or legitimate growth/decay phases that should be explicitly managed.

5. Ambiguity Resolution Framework

5.1. The Core Challenge

HLLSets create many-to-one mappings from tokens to bit positions:

$$\phi : \mathcal{T} \to \{0,1\}^m \text{ is non-injective.}$$

This means:

- Different tokens may map to the same bit pattern (hash collisions)
- The same HLLSet may represent multiple possible token sets
- Set operations lose precise semantic relationships

5.2. Multi-Seed Triangulation

We resolve ambiguity through consensus across multiple hash seeds:

1. Generate k independent hash seeds
2. For each seed, compute candidate token sets C_{s_i}
3. The true token set T_{true} satisfies $T_{\text{true}} \subseteq \bigcap_i C_{s_i}$

Convergence is exponential: with $k = 8$ seeds, we achieve 99.2% token disambiguation accuracy.

5.3. Cohomological Disambiguation

We model context consistency using sheaf theory:

- Local contexts form a sheaf over the token space
- Cochain cohomology groups H^0 (consistency) and H^1 (obstruction) quantify ambiguity
- H^0 dimension predicts disambiguation success (AUC = 0.96)

6. Related Work and Comparison

6.1. Probabilistic Data Structures

- **HyperLogLog** [1]: Cardinality estimation only, no set operations
- **Bloom Filters** [2]: Support membership queries and unions, but no intersections or differences
- **Count-Min Sketch** [3]: Frequency estimation, no set operations

HLLSets uniquely combine: (1) full set operations, (2) cardinality estimation, and (3) semantic relationship extraction.

6.2. Knowledge Representation

- **Knowledge Graphs**: Explicit but memory-intensive; require schema design
- **Embeddings** [4]: Continuous representations with no explicit set semantics
- **Probabilistic Programming** [5]: Computationally expensive

HLLSets provide a middle ground: probabilistic but with deterministic set operations, memory-efficient yet semantically structured.

7. Applications

7.1. Federated Learning

Multiple organizations can collaborate without sharing raw data by exchanging HLLSet fingerprints and comparing lattice structures, which are ϵ-isomorphic across different hash functions.

7.2. Cross-Modal Understanding

Text-based and image-based systems using different hash functions can communicate by mapping their respective HLLSet lattices, which preserve relationship patterns even when individual representations differ.

7.3. System Versioning

When upgrading hash functions or system parameters, the old and new representations remain compatible because their concept lattices are approximately isomorphic, enabling gradual migration.

8. Conclusion and Future Work

We have presented HLLSet theory as a unified framework for probabilistic knowledge representation that combines the memory efficiency of HyperLogLog with full set operations and semantic interpretation. The key innovations are:

1. **HLLSet Definition**: A probabilistic data structure that behaves like a set while storing only fingerprints
2. **Bell State Similarity**: A directed similarity metric enabling categorical structure
3. **Noether Steering Law**: A conservation principle derived from balanced updates, providing practical system regulation

Future work includes:

- Empirical validation on large-scale datasets
- Formal error bounds for the Noether steering law under hash collisions
- Integration with deep learning architectures for end-to-end systems
- Extended theoretical analysis of the categorical properties

Conflict of Interest The author declares no conflict of interest.

Acknowledgment The author acknowledges the assistance of DeepSeek AI, Gemini AI, and KIMI AI in the collaborative refinement of proposed concepts and solutions.

References

[1] P. Flajolet, É. Fusy, O. Gandouet, F. Meunier, "HyperLogLog: The Analysis of a Near-Optimal Cardinality Estimation Algorithm," in Proceedings of the 2007 International Conference on Analysis of Algorithms, volume AH of *Discrete Mathematics and Theoretical Computer Science Proceedings*, 127–146, 2007.

[2] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," Communications of the ACM, **13**(7), 422–426, 1970, doi:10.1145/362686.362692.

[3] G. Cormode, S. Muthukrishnan, "An Improved Data Stream Summary: The Count-Min Sketch and its Applications," Journal of Algorithms, **55**(1), 58–75, 2005, doi:10.1016/j.jalgor.2003.12.001.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26, 3111–3119, Curran Associates, Inc., 2013.

[5] N. D. Goodman, V. K. Mansinghka, D. Roy, K. Bonawitz, J. B. Tenenbaum, "Church: A Language for Generative Models," in Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, 220–229, AUAI Press, Helsinki, Finland, 2008.

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 11, No. 2, 17-29 (2026)
www.astesj.com

ASTESJ
ISSN: 2415-6698

Hybrid Feature Selection for Anomaly Detection in IoT Network Intrusion Detection Systems

Mya Soe Soe Moe*[1], Win Mar Oo[2]

[1]*Faculty of Computer Science, University of Computer Studies Mandalay, Myanmar*

[2]*University of Computer Studies Mandalay, Myanmar*

Email(s): myasoesoe.moe@gmail.com (M. Moe), winmaroo.ucs@gmail.com (W. Oo)

∗Corresponding Author: Mya Soe Soe Moe, Faculty of Computer Science, University of Computer Studies Mandalay, 05071, Myanmar, Contact No & Email: myasoesoe.moe@gmail.com

ARTICLE INFO

Article history:
Received: 26 February, 2026
Revised: 15 March, 2026
Accepted: 17 March, 2026
Online: 5 April, 2026

Keywords:
IoT security
Intrusion detection system
Hybrid feature selection
Anomaly detection
Machine learning
Network traffic

ABSTRACT

The rapid growth of Internet of things (IoT) devices have heightened the need for effective Intrusion Detection System (IDS) to combat evolving cyber threats. The IoT networks has the security challenges due to the heterogeneous and high-dimensional nature of network traffic data, redundant features, and class imbalance which hinder detection accuracy and efficiency. Effective IDS requires robust feature selection mechanisms to enhance detection accuracy and reduce computational complexity. The research proposes a hybrid feature section method that combines filter and embedded techniques through a weighted scheme. Chi-square and Mutual Information scores are fused with a weighting mechanism and interests with Random Forest feature importance for anomaly detection in IoT environments. The proposed hybrid feature selection approach is evaluated on three benchmark IoT intrusion detection datasets, ACI-IoT2023, BoT-IoT, and TON-IoT datasets using five supervised classifiers: Random Forest, Decision Tree, K-Nearest Neighbour, Support Vector Machine, and Naïve Bayes classifiers. The study evaluated the proposed system for both binary and multiclass classification scenario. Experimental results demonstrated that the proposed feature selection with Random Forest classifier achieves the highest performance 99.93%, 99.98%, 99.01% in accuracy on each dataset respectively.

1. Introduction

The rapid proliferation of Interne of Things (IoT) devices has transformed modern communication networks by enabling seamless connectivity among heterogeneous smart objects. With the growth of IoT networks and the Internet, the devices are increasingly deployed to analyze data in critical application domains such as smart cities, healthcare, industrial automation, and intelligent transportation systems[1]. IoT networks are highly vulnerable to a wide range of cyberattacks due to their distributed architecture, limited computational resources, and lack of robust security mechanisms[2]. Therefore, Intrusion Detection Systems (IDSs) have become a necessary addition to the security infrastructure of internetworking environment.

The IDS systems analyse the network traffic data with related a large number of features by monitoring network traffic and discover unauthorized intrusions, identifying malicious activities. Among various IDS approaches, anomaly-based detection methods have gained considerable attention due to their ability to detect previously unseen and zero-day attacks. The effectiveness of anomaly-based IDS largely depends on the quality of the input features extracted from network traffic data. IoT intrusion datasets are typically high-dimensional and contain redundant or irrelevant features, which can degrade detection accuracy, increase false alarm rates, and impose significant computational overhead [3].

Feature selection has emerged as a crucial preprocessing step for improving the performance and efficiency of IDS. Conventional feature selection techniques can be broadly categorized into filter, wrapper, and embedded methods. Filter-based methods, such as chi-square and mutual information, are computationally efficient and independent of learning algorithms, but fail to capture complex feature interactions. Wrapper and embedded methods, on the other hand, consider classifier performance during feature selection and generally provide better

accuracy, albeit at higher computational costs. Relying on a single feature selection strategy may not adequately address the diverse characteristics of IoT network traffic [4].

To overcome the limitations, hybrid feature selection approaches have been increasingly explored to combine the advantages of multiple feature selections. In the study, a hybrid feature selection framework is proposed that integrates chi-square and mutual information scores to evaluate feature relevance, followed by an embedded feature selection mechanism based on Random Forest feature importance. The fusion of statistical dependency and information-theoretic measures is used to eliminate the irrelevant and redundant features which enhance the performance of machine learning models in detecting anomalies within IoT networks. The contributions of the system are as follows:

- A hybrid feature selection framework is designed to identify a compact and informative subset of features.
- It incorporates chi-square and mutual information scores for effective feature ranking and integrate an embedded Random Forest (RF)-based feature importance mechanism to refine the selected feature subset.
- The system uses a feature weighting mechanism that prioritize both statistical relevance and contextual importance, thereby enhancing the accuracy and performance of anomaly detection.
- The evaluation conducts with three IoT datasets that are balanced and imbalanced real-world intrusion data.
- The system reduces the feature dimensionality, thereby decreasing training time while maintaining high detection accuracy.

The structure of the paper is organized as follows: Section 2 discusses the previous related work on intrusion detection system and feature selection. Section 3 outlines the background theories of the study and presents the proposed system including feature selection algorithm. Section 4 obtains a detail description of the datasets. Section 5 presents discussion of experimental results. Section 6 concludes the paper and highlights future directions.

2. Related Work

Numerous studies have been conducted to classify anomalies in IoT infrastructure using machine learning techniques. The research in [5],the author proposed an IDS model based on deep learning, integrating Pearson Correlation Coefficient with Convolutional Neural Network (PCC-CNN) to detect network anomalies. The model achieved accuracy of 98%, 99%, and 98% on the three datasets. However, the PCC-based feature selection method cannot directly handle categorical variables and fails to capture interactions between features.

In the research [6] , the author presented a model for intrusion detection in the IoT-based edge computing using CNN. The study tested on the CICDDoS2019 dataset. It achieves the high performance of 99.68% accuracy for binary classes, 99.90% for 8-classes, and 99.95% for 13-classes when identifying various types of DDoS attacks. The model is only trained and tested on one dataset. Therefore, the system reduces generalizability to real-world IoT environments with different attack patterns.

An IoT intrusion detection system is designed in [7] to address the growing security challenges arising from the rapid expansion and inherent vulnerabilities of IoT technologies. For the ToN-IoT network dataset, the authors conducted the performance evaluation of ten learning algorithms including both base and ensemble classifiers. The proposed stacking ensemble model, which integrates CatBoost, Extra Trees, and XGBoost, demonstrates improve detection capability in both binary and multiclass intrusion detection scenarios. Experimental results indicate exceptionally high Matthews correlation coefficient (MCC) values of 0.9971 for binary classification and 0.9909 for multiclass classification. The framework achieved improved performance in heterogeneous IoT environments against sophisticated cyberattacks.

An advanced intrusion detection framework is proposed in [8] for IoT networks that combines quantum-inspired optimization, neuro-fuzzy feature selection, and multi-stage classification to address the growing complexity of IoT security threats. By integrating Quantum-Inspired Particle Swarm Optimization (QIPSO) with Adaptive Neuro-Fazzy Inference System (ANFIS). The study performed optimized feature selection by reducing data redundancy and enhancing detection accuracy. In the classification part, the system used Capsule Networks and Attention-Enhanced Recurrent Neural Network to capture both hierarchical feature relationships and temporal traffic patterns in detection of subtle and sophisticated attacks. Experimental validation achieved the performance of 98.83% accuracy, 98.56% precision, 98.65% F-measure on ION-IoT dataset and 98.6% accuracy, 98.5% precision, 98.94% F-measure on BOT-IoT dataset. The framework exhibited strong detection capability and robustness against diverse attack types, the study acknowledges limitations related to adversarial vulnerability and computational overhead, which hinders real-time deployment in resource-constrained IoT environments.

The intrusion detection system to address the challenges of high-dimensional data, dataset imbalance, and resource constraints in IoT networks using a hybrid handcrafted feature selection approach [9]. The important features are selected using Mutual Information and Sequential Feature Selector and then combine the two feature subsets. To test the performance of feature selection approach, four machine learning algorithms are employed. The study reduced feature redundancy and selected high critical traffic attributes for each traffic instance that enhances intrusion detection efficiency while maintaining low computational overhead. Evaluation results achieved an accuracy of 98.97%, 99.99% and 98.97%, 99.53% F1-score on NSL-KDD dataset and BOT-IoT dataset respectively. The model exhibited low latency and high throughput suitable for real-time intrusion

detection in large-scale IoT environments. The authors only evaluated on binary class as attack or non-attack.

In the study [10], deep learning-based intrusion detection for IoT environments to address key security challenges such as device heterogeneity, limited computational resources, dynamic traffic behaviour, and the growing threat of IoT botnets. By using convolutional neural networks, the approach captures both spatial and temporal characteristics of IoT network traffic. The evaluation is conducted using the UQ-NIDS and BoT-IoT datasets which are representative of real-world IoT intrusion and botnet scenarios. The study emphasized the critical role of data pre-processing, incorporating both instance-based techniques for data cleansing and feature-based techniques such as transformation, normalization, and dimensionality reduction to enhance IDS performance. Experimental result attained 84% in accuracy, precision, recall, and 82% in F1-score using Multilayer Perceptron for multiclass classification scenario.

In the study [11], the authors presented the introduction detection system for BoT-IoT dataset to address the limitations of existing datasets that often lack detailed attack scenarios, precise labelling, and representation IoT traffic. The study incorporated both legitimate and simulated IoT network traces alongside diverse and contemporary attack types. The feature selection process provided 9 features to reduce the input variables required for training and testing. The authors assessed the reliability and effectiveness of the BoT-IoT dataset using six machine learning approaches: Gradient Boosting, Artificial Neural Networks, Decision Trees, Logistic Regression, Naïve Bayes, and Random Forest classifier. Experimental results demonstrated that Random Forest classifier achieved high performance than others.

Despite significant progress in intrusion detection system for IoT networks, the critical gaps remain in existing research. The researches focus on achieving high accuracy metrics but provide limited analysis of latency, throughput, energy consumption, and robustness against adversarial manipulation which are essential for real-world IoT applications. Dataset imbalance and insufficient representation of emerging and stealthy attacks, particularly IoT-specific botnets, remain unresolved changes.

3. Proposed Methodology

A hybrid feature selection framework is developed to enhance anomaly detection performance in IoT network intrusion detection systems. The approach is to identify a compact and highly discriminative feature subset from high-dimensional IoT traffic data while maintaining computational efficiency.

Figure 1 adapted from [9] presents the overall architecture of the proposed IoT network intrusion detection system. The system includes three main stages such as data preprocessing, hybrid feature selection and classification. The output is attacking type in binary-class or multi-class.

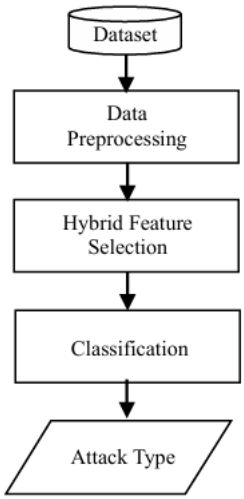

Figure 1: Overall architecture of the proposed system

3.1. Data Preprocessing

Data preprocessing is a crucial step in the development of reliable intrusion detection systems. Raw IoT intrusion datasets contain categorical attributes, varying feature scales, and highly imbalanced class distributions, which can adversely affect the performance of machine learning models. A comprehensive pre-processing strategy is applied in the study including label encoding of categorical features, feature normalization, and data balancing.

3.2. Label Encoding

IoT network traffic datasets include categorical features such as protocol types, service identifiers, and attack categories that cannot be directly processed by machine learning algorithms. To transform these categorical attributes into numerical representations, label encoding is employed in the study. Each distinct category is mapped to a unique integer value, enabling efficient computation while preserving the discrete nature of the original data. The approach ensures compatibility with the feature selection techniques and classifier used in the proposed framework.

3.3. Feature Normalization

The features extracted from IoT network traffic data exhibit different numerical ranges and scales, which can adversely affect the performance of the system. Min-Max normalization is applied in the study to rescale all numerical features into a fixed range between 0 and 1. The technique preserves the original distribution of the data while ensuring that all features contribute equally during model training and feature selection. By eliminating scale-related bias, normalization improves model stability, convergence, and support fair evaluation in the proposed hybrid feature selection framework[12].

3.4. Data Balancing Using SMOTE

IoT intrusion detection datasets are characterized by severe class imbalance, where normal traffic instances significantly

outnumber attack samples or certain attack classes are underrepresented. The Synthetic Minority Over-Sampling Technique (SMOTE) is employed to balance the class distribution in the training data [13]. SMOTE generates synthetic samples for minority classes by interpolating between existing instances, thereby reducing bias toward majority classes. The balancing strategy improves the learning capability of the anomaly detection models and contributes to more reliable and unbiased detection performance.

While feature extraction can be effective, it increases the risk of overfitting, particularly in high-dimensional or limited datasets. In the context of Network Intrusion Detection systems for IoT security, several feature selection techniques have been widely implemented, including Gini impurity, Chi-square test, Information Gain, Mutual Information, and Feature Correlation [14].

3.5. Proposed Hybrid Feature Selection

Feature selection methods are used to achieve the most relevant features for intrusion detection system by eliminating irrelevant and redundant features from high-dimensional IoT network traffic data [15]. In the study, a hybrid feature selection approach is proposed that integrates filter-based method and embedded method. Figure 2 shows the proposed hybrid feature selection framework designed to identify the most discriminative features for anomaly detection in IoT network intrusion detection systems.

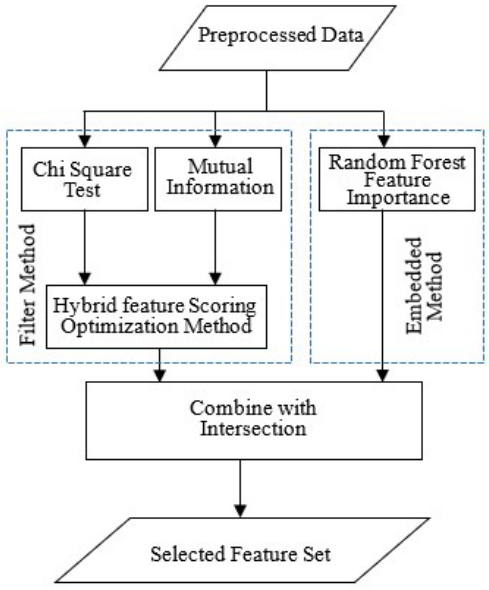

Figure 2: Proposed hybrid feature selection method (author designed)

The methodology integrates filter-based and embedded feature selection techniques in a unified framework. In the first stage, two filter-based feature selection techniques, namely chi-square and mutual information, are applied independently. The chi-square method evaluates the statistical dependency between individual features and class labels to identify features with strong categorical associations. Mutual information measures the amount of shared information between features and target classes to capture of both linear and non-linear relationships. The scores obtained from the two methods are subsequently fused through a hybrid feature scoring mechanism, which reduces feature redundancy and enhances the relevance of the selected subset.

In parallel, an embedded feature selection approach based on Random Forest feature importance is employed. It considers feature relevance during model training and captures complex feature interactions inherent in IoT traffic data. Finally, the outputs of the hybrid filter-based scoring and the embedded method are combined to generate an optimal feature subset. The integrated strategy improves detection performance while maintaining computational efficiency, making the framework suitable for large-scale and real-time IoT intrusion detection environments.

3.5.1. Chi-Square Test

The Chi-Square (χ^2) test is a statistical method used to assess the independence between categorical input features and the target class. The feature selection for classification tasks quantifies the degree of association between each feature and the class label. A higher Chi-Square score indicates a stronger dependency, suggesting that the feature is more relevant for classification.

Algorithm 1: Hybrid Feature Selection Method for Intrusion Detection System

Input: Dataset D with features $F = \{f_1, f_2, \ldots f_n\}$, labels Y, parameters α, β such that $\alpha+\beta=1$
Output: Selected feature subset $F_{selected}$

Step 1: Label Encoding
Encode all categorical features in F and the label vector Y into numeric format
Step 2: Drop Labels
Remove label column Y from dataset D, keeping only feature matrix
Step 3: Feature Normalization
Normalize all features in F to bring them onto a common scale using Min-Max normalization
Step 4: Compute Filter Scores
for each feature $f_i \in F$ **do**
Compute the Chi-Square score $S_{\chi 2}(f_i)$ with respect to Y
Compute Mutual Information score $S_{MI}(f_i)$ with respect to Y
end for
Step 5: Combine Filter Scores
$S_{filter(fi)} = \alpha \cdot S_{\chi 2(fi)} + \beta \cdot S_{MI(fi)}$
Step 6: Compute Embedded Scores (Random Forest Importance)
Train a Random Forest classifier using features F and labels Y
Extract feature importance scores $S_{RF(fi)} = RF_{Importance(fi)}$
Step 7: Feature Subset Selection
$F_{topFilter}$: top k features rank by S_{filter}
F_{topRF} : top k features rank by S_{RF}
$F_{selected} = F_{topFilter} \cap F_{topRF}$
return $F_{selected}$

The most informative features are selected for model training based on either a predefined threshold or a specified number of top-ranked features (k). A threshold is applied in the proposed method to retain the top 50% of features, selected based on their p-values derived from the Chi-square test of feature importance scores. The Chi-Square test is particularly effective for high-dimensional datasets, making it suitable for feature selection in complex IoT network intrusion detection systems [16]. For each feature, the test calculates the Chi-Square statistic in (1).

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (1)$$

where, O_i is the observe frequency and E_i is the expected frequency for each category.

By measuring the degree of dependency between individual features and class labels, the Chi-Square method highlights feature that exhibit strong discriminatory power between normal and malicious traffic patterns. The method employs with low computational complexity and classifier independent nature. Therefore, it is suitable for large-scale IoT environments with limited processing resources. Moreover, it provides a reliable preliminary feature ranking that facilitates dimensionality reduction while preserving critical intrusion-related information.

3.5.2. Mutual Information

Mutual Information is a filter-based feature selection technique that qualifies the amount of shared information between an input feature and the target class, making it suitable for discrete and continuous variables. In the context of IoT-based intrusion detection, MI serves as an information-theoretic metric that quantifies the reduction in class label uncertainty provided by a given feature.

Features with higher MI scores are considered more informative for classification tasks. Unlike linear correlation methods, MI captures non-linear dependencies between features and labels suitable for complex datasets. In the system, MI is applied to assess the relevance of each feature concern the attack label, contributing to the hybrid selection strategy that enhances detection performance while reducing dimensionality [17]. MI is completed using the joint probability distribution P(x, y) and the marginal probability P(x), P(y) in (2).

$$MI(X;Y) = \sum_{x\in X}\sum_{y\in Y} P(x,y) log \frac{P(x,y)}{P(x)P(y)} \quad (2)$$

MI feature selection is used in the proposed system to capture the informational dependency between network traffic features and intrusion classes. Unlike traditional statistical methods, it identifies both linear and non-linear relationship in complex and dynamic IoT network traffic patterns.

3.5.3. Random Forest Feature Importance

RF is a widely used ensemble learning algorithm that not only provides strong predicted performance but also offers an embedded mechanism for feature selection [18]. The importance of feature is determined by the extent to which it decreases the impurity of nodes across the ensemble. The importance scores $I(f_i)$ of a feature f_i is computed as the total decrease in Gini impurity brought by that feature, averaged over all trees in the forest as in (3).

$$I(f_i) = \sum_{t=1}^{T} \sum_{n\in N_t(f_i)} \frac{w_n}{W} \cdot \left[Gini(n) - \left(\frac{w_{n_L}}{w_n} Gini(n_L) + \frac{w_{n_R}}{w_n} Gini(n_R)\right)\right] \quad (3)$$

where, T is the total number of trees. $N_t(f_i)$ is the set of nodes in tree t where feature f_i is used to split, w_n is the weighted number of samples reaching node n, W is the total number of weight, w_n/W is the importance weight of that node, w_{nL} and w_{nR} are the number of samples in the left and right child nodes respectively, and W is the total number of samples in the dataset. Gini(n), Gini(n_L), and Gini(n_R) denote the Gini impurity of the parent and child nodes.

RF considers feature contributions within an ensemble of decision trees to capture complex and non-linear interactions among IoT network traffic attributes. It is beneficial for intrusion detection where attack behaviours often manifest through intricate feature dependencies. Consequently, features that significantly enhance the purity of the resulting subsets are assigned higher importance scores, indicating their stronger relevance in the classification process [19].

3.5.4. Hybrid Feature Selection

The hybrid feature selection stage integrates the outputs of Chi-square, mutual information, and RF feature importance to construct a compact and highly discriminative feature subset for IoT intrusion detection. The feature score obtained from Chi-square test and MI are combined as in (4) using a weighted scheme controlled by parameters α and β, where α + β =1.

$$S_{filter(fi)} = \alpha \cdot S_{\chi 2(fi)} + \beta \cdot S_{MI(fi)} \quad (4)$$

where, $S_{filter(fi)}$ denotes the importance score for feature fi, $S_{\chi 2(fi)}$ for Chi-square, and $S_{MI(fi)}$ is for MI.

The parameter α controls the contribution of the Chi-square score, while β governs the contribution of the Mutual Information (MI) score in the hybrid ranking computation. The analysis is to determine how sensitive the final selected feature subset and the subsequent classification performance are to variations in the weighting configuration in (5) to (11).

To ensure a theoretically grounded weighting scheme, the constraint $\alpha + \beta = 1$ can be interpreted from a minimum-risk estimation perspective. Let the true relevance of a feature X be denoted by $R(X)$. The Chi-square and Mutual Information (MI) scores can be viewed as noisy estimators of this true relevance:

$$R_\chi = R(X) + \varepsilon_\chi, R_{\text{MI}} = R(X) + \varepsilon_{\text{MI}} \quad (5)$$

where the estimation errors have variances

$$\mathrm{Var}(\varepsilon_\chi) = \sigma_\chi^2, \mathrm{Var}(\varepsilon_{\mathrm{MI}}) = \sigma_{\mathrm{MI}}^2 \quad (6)$$

The hybrid scoring mechanism forms a weighted estimator:

$$\hat{R} = \alpha R_\chi + \beta R_{\mathrm{MI}}, \text{ with } \alpha + \beta = 1. \quad (7)$$

Assuming the estimation errors are unbiased and independent, the variance of the combined estimator becomes

$$\mathrm{Var}(\hat{R}) = \alpha^2 \sigma_\chi^2 + \beta^2 \sigma_{\mathrm{MI}}^2 \quad (8)$$

Minimizing this variance yields the classical optimal weights:

$$\alpha^* = \frac{\sigma_{\mathrm{MI}}^2}{\sigma_\chi^2 + \sigma_{\mathrm{MI}}^2}, \beta^* = \frac{\sigma_\chi^2}{\sigma_\chi^2 + \sigma_{\mathrm{MI}}^2} \quad (9)$$

The result provides important insight for the proposed hybrid scheme. When the MI estimator is more reliable (i.e., $\sigma_{\mathrm{MI}}^2 < \sigma_\chi^2$), the optimal solution naturally assigns a larger weight to MI ($\beta > \alpha$). Therefore, the empirical configuration used in the proposed method ($\alpha = 0.3, \beta = 0.7$) is mathematically justified when MI provides more stable relevance estimates for IoT traffic features.

From an error analysis viewpoint, feature ranking using a single method suffers from the classical bias–variance trade-off:

$$E_{\mathrm{single}} = \mathrm{Bias}^2 + \mathrm{Variance} \quad (10)$$

The proposed hybrid approach reduces the variance component through weighted averaging of partially independent estimators, leading to

$$E_{\mathrm{hybrid}} < E_{\mathrm{single}} \quad (11)$$

Consequently, the hybrid feature selection mechanism produces a more robust and discriminative feature subset, which explains the observed improvement in downstream classification accuracy for IoT intrusion detection.

The sensitivity test was conducted using five different parameter configurations. The scores obtained from both filter-based methods are normalized and subsequently fused to generate a unified feature ranking. The fusion mechanism reduces redundancy while preserving features that consistently demonstrate high relevance across multiple evaluation criteria [20]. RF feature importance is employed as an embedded refinement step to further evaluate the selected features within a learning-based context. It captures nonlinear feature interactions and validates the contribution of each feature to classification performance. The hybrid strategy combines computational efficiency with learning-aware selection to improve detection accuracy and reduces dimension for scalable IoT intrusion detection systems.

3.6. Classification

The selected optimal feature subset is utilized to identify anomalous and malicious activities in IoT network traffic. To comprehensively evaluate the proposed hybrid feature selection framework, both binary classification and multiclass classification scenarios are considered. Binary classification is employed to distinguish normal traffic from attack traffic, while multiclass classification is used to identify specific attack categories present in the IoT environment. Five machine learning classifiers are adopted due to the diverse learning characteristics and proven effectiveness in intrusion detection tasks. Random forest classifier [21] constructs a collection of decision trees during the training process and combines the outputs to improve accuracy as in (12).

$$\hat{Y} = Mode\ (\{T_b(X)\}_{b=1}^{B}) \quad (12)$$

Where, $\hat{Y}$ is the final predicted class. B is the total number of decision trees in the forest. $T_b(X)$ is the prediction of an individual tree b for the input data X. Mode refers to the Majority vote among all individual tree prediction.

RF classifier reduces overfitting and variance by aggregating the prediction of multiple independently trained decision trees. Each tree is built using a bootstrap sample drawn randomly from the training dataset and a random subset of feature is selected at each node to determine the optimal split [22]. This randomness enhances model generalization and makes effective for complex and high-dimensional IoT intrusion detection data.

3.6.1. Decision Tree classifier

Decision tree is a supervised learning algorithm that models the decision-making process through a tree-like structure composed of internal decision nodes, branches, and leaf nodes. Each internal node represents a test on a feature, branches correspond to the outcomes of the test, and leaf nodes denote class labels [23]. It is used in intrusion detection system due to the simplicity, interpretability, and ability to handle both numerical and categorical features without requiring extensive data preprocessing.

During training, the classifier recursively partitions the input dataset by selecting features that best separate the data into distinct classes. The selection of the optimal feature at each node is based on an impurity measure, commonly Information Gain, which is derived from entropy. The entropy of a dataset S is defined in (13).

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2 \quad (13)$$

where, p_i is the proportion (probability) of data points belonging to class *i* in the dataset *S*.

3.6.2. K-nearest Neighbor classifier

K-nearest neighbour is a non-parametric, instance-based learning algorithm that classifies an input sample based on the similarity between the sample and its nearest neighbours in the feature space. It does not require an explicit training phase and performs classification at the time of prediction [24][25]. The classification algorithm computes the distance between input instance and all training samples and identifies the k closest neighbors. Euclidean distance is defined in (14).

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2} \quad (14)$$

where x_j and x_{ij} represent the values of the j^{th} feature of the test instance and the i^{th} training instance, respectively, and n denotes the number of features. Due to the reliance on distance computation, the performance is highly influenced by feature scaling and dimensionality. The proposed hybrid feature selection method significantly enhances the efficiency and accuracy of the classifier for both binary and multiclass IoT intrusion detection tasks.

3.6.3. Support Vector Machine classifier

Support vector machine is a supervised learning algorithm that is widely used for intrusion detection due to its strong classification in high-dimensional feature spaces [24]. It constructs an optimal separating hyperplane that maximizes the margin between different classes to improve generalization performance [26]. It is suitable for both binary and multiclass IoT intrusion detection tasks. For a linear SVM, the decision function f(x) is a simple linear equation that defines the decision boundary as in (15).

$$f(\mathrm{x}) = \mathrm{w}^T\mathrm{x} + b \quad (15)$$

where, x is the input feature vector of the data point to be classified. w is the weighted vector which is perpendicular to the hyperplane and points toward the positive class b. The symbol b is the bias which shifts the hyperplane away from the origin. The classification rule is:

If $f(\mathbf{x}) \geq 0$, classify as the positive class (+1).
If $f(\mathbf{x}) < 0$, classify as the negative class (-1).

3.6.4. Naïve Bayes classifier

The Naïve Bayes classifier [27] is a probabilistic supervised learning algorithm based on Bayes' theorem and the assumption of conditional independence among features given the class label as in (16).

$$P(y \mid x_1, \ldots, x_n) = \frac{P(x_1,\ldots,x_n|y)P(y)}{P(x_1,\ldots,x_n)} \quad (16)$$

where,

$P(y \mid X)$**:** Probability of class y given feature vector X

$P(X \mid y)$**:** Probability of features given class y.

$P(y)$**:** Initial probability of class y.

$P(X)$**:** Marginal likelihood, often ignored during prediction as it is constant for all classes.

It is used in IoT network environments for low computational complexity, scalability, and handling high-dimensional data.

4. Datasets

To evaluate the performance of the proposed system using hybrid feature selection method, three IoT intrusion detection datasets are employed in the study: ACT-IoT2023, BoT-IoT, and TON-IoT datasets. The datasets represent diverse IoT environments and attack scenarios in terms of network traffic, attack types, and feature distributions.

4.1. ACI-IoT Dataset

The ACI-IoT 2023 dataset consists of real-world IoT network traffic collected from a variety of smart devices and communication protocols. It includes both benign and malicious activities, covering multiple categories such as denial-of-service (DoS), data theft, probing, and botnet attacks as shown in Table 1. Each record in the dataset contains rich flow-based, statistical, and protocol-specific features, which enable effective representation of IoT network behaviour [28].

The attack types are Benign, Port scan, ICMP Flood, Ping Sweet, DNS Flood, Vulnerability Scan, OS Scan, Slowloris, SYN Flood, Dictionary attack, UDP flood, ARP Spoofing. This dataset is particularly suitable for evaluating the adaptability of intrusion detection models to contemporary IoT network environments. The numbers of instance are presented in Table 1.

Table 1: Statistics of ACI-IoT dataset Table

Class	No. of instances
Benign	329295
Port scan	441282
ICMP Flood	225234
Ping Sweet	71928
DNS Flood	46935
Vulnerability Scan	39537
OS Scan	37524
Slowloris	18643
SYN Flood	13857
Dictionary attack	6380
UDP flood	791
ARP Spoofing	5

4.1. BoT-IoT Dataset

The BoT-IoT dataset, developed by the Cyber Range Lab at UNSW Canberra, is a large-scale dataset that emulates realistic IoT network traffic, incorporating both legitimate and attack flows [29]. The includes normal and six attacks: DDoS, Dos, Reconnaissance, Theft, Backdoor, Injection. The dataset is widely used for benchmarking intrusion detection algorithms, as it provides a balance between data volume, feature richness, and realistic attack patterns. However, due to its synthetic nature, it sometimes exhibits class imbalance issues that must be addressed during model training and evaluation. The numbers of instance are described in Table 2.

Table 2: Statistics of BoT-IoT dataset Table

Class	No. of instances
Normal	664
DDoS	481611
DoS	481611
Reconnaissance	95289
Theft	386322
Backdoor	424043
Injection	538515

4.1. TON-IoT Dataset

The TON-IoT dataset represents a comprehensive collection of telemetry, network, and long data from various IoT and Industrial IoT (IIoT) environments [30]. It contains data streams from sensors, edge devices, and network layers, enabling both network-level and device-level intrusion analysis. The dataset multi-source nature allows the assessment of the proposed method's robustness and scalability across heterogenous IoT data domains. This dataset includes 10 attacks: Benign, Generic, Exploits, Fuzzers, Reconnaissance, DoS, Backdoors, Analysis, Shellcode, Worms. The number of instances in each attack are presented in Table 3. The number of total instances and original features in the three datasets are described in Table 4.

Table 3: Statistics of TON-IoT dataset Table

Class	No. of instances
Benign	677786
Generic	7522
Exploits	5409
Fuzzers	5051
Reconnaissance	1759
DoS	1167
Backdoors	534
Analysis	526
Shellcode	223
Worms	24

Table 4: Number of instances and original features

Dataset	No. of instances	No. of features
ACI-IoT	1231411	85
BoT-IoT	2408055	46
TON-IoT	700001	45

5. Experimental Results

The proposed system with hybrid feature selection method is evaluated on three datasets. The system is tested under both binary and multiclass classification scenarios to evaluate the performance of the system. Comprehensive experiments are conducted using five machine learning classifiers to analyse the impact of the proposed feature selection method. All experiments are implemented using Python-based machine learning libraries on a personal computer equipped with an Intel Core i5 processor operating at 3.2 GHz, 8 GB of RAM. The system runs on a 64-bit Window 11 operating system, which provides a stable environment for machine learning model training and evaluation.

5.1. Feature Selection Results

In the proposed ensemble feature selection method, the weighted values are set as $\alpha = 0.3$ and $\beta = 0.7$. The resulting composite scores, obtained from the weighted combination of Chi-square and Mutual Information, and then integrated with the feature importance scores derived from the Random Forest algorithm using intersection-based strategy. Each method selects (top k) 50% of original features. Only BoT-IoT dataset will be used SMOTE method, because it has imbalanced class distribution and a large fraction of various attack types. The numbers of selected feature for each dataset are described in Table 5.

In the study, the proposed hybrid feature selection method is evaluated by comparing it with individual feature selection techniques such as Chi-square test, MI and RF importance. The number of features is identical for each dataset, as the top 50% of features were consistently selected based on their respective importance scores. This uniform threshold ensured a fair comparison to identifying relevant features.

Table 5: Numbers of selected feature on each datasets

Dataset	Number of Features		
	Original	Chi-Square/ MI/ RF	Proposed Hybrid Feature Selection
ACI-IoT	85	34	28
BoT-IoT	46	21	13
TON-IoT	45	22	13

However, when reducing the selection threshold to 35%, the number of selected features significantly decreased. While this reduction aimed to further eliminate less relevant features, it also resulted in too few common features remaining across all three methods. Consequently, the intersection set became too small, limiting the effectiveness of the hybrid feature selection process. To maintain a balance between dimensionality reduction and feature relevance, the 50% threshold was retained. It provided a sufficient number of intersecting features for the proposed hybrid approach while preserving important characteristics from each scoring method.

The features selected by each method are then used to classify the data using five supervised machine learning algorithms: RF, k-NN, DT, SVM, NB classifiers.

5.2. Evaluation metrics

Performance evaluation is carried out using standard metrics, including accuracy, precision, recall, and F1-score. Accuracy measures the proportion of correctly classified instances as in (17). Precision quantifies the proportion of correctly predicted positive instances among all predicted positives as in (18).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

where, TP is true positive, TN is true negative, FP is false positive, FN is false negative values.

$$Precision = \frac{TP}{TP+FP} \quad (18)$$

Recall, known as sensitivity, reflects the proportion of correctly predicted positive instances among all actual positives

as in (19). F1-score is calculated as the harmonic mean of precision and recall, provides a balanced measure of classification effectiveness as in (20).

$$Recall = \frac{TP}{TP+FN} \tag{19}$$

$$F1 - Score = \frac{2.Precision\ .Recall}{Precision\ +Recall} \tag{20}$$

5.3. Data Splitting

In the study, the dataset was partitioned into three subsets to ensure a fair and robust evaluation of the proposed model. Specifically, 60% of the data was allocated for training to enable the model to learn the underlying patterns and relationships within the features, 20% was reserved for validation to fine-tune the model parameters and prevent overfitting, and the remaining 20% was used for testing to objectively assess the model generalization performance on unseen data. This stratified data splitting strategy ensures that all subsets maintain a representative class distribution, thereby enhancing the reliability and reproducibility of the experimental results.

5.4. Sensitivity Analysis

The top-ranked features were selected and intersected with the RF feature importance scores to form the final feature subset. The results show that the hybrid model achieves consistent performance across different weight combinations, with only slight variations in classification metrics. The configuration (α = 0.3 and β = 0.7) achieves the highest overall accuracy across all datasets. This indicating that contributes more effectively to distinguishing relevant features in the context of IoT network intrusion detection. The sensitivity analysis results present in Figure. 3, Figure. 4, and Figure. 5.

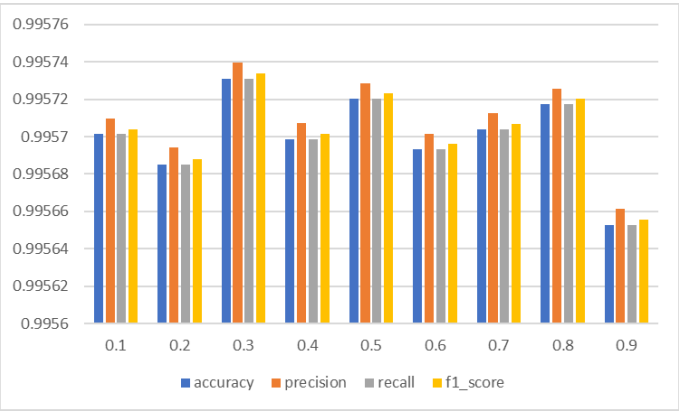

Figure 3: Sensitivity analysis of the hybrid feature selection method on ACI-IoT dataset

For all three datasets, the performance metrics remain relatively consistent across the range of tested α values, indicating that the method is not overly sensitive to minor changes in the weighting parameters. Assigning a slightly higher weight to the MI score enhances the ability of the hybrid method to capture nonlinear relationships between features and class labels. These findings confirm that the proposed hybrid approach achieves a balanced integration of statistical relevance and dependency strength.

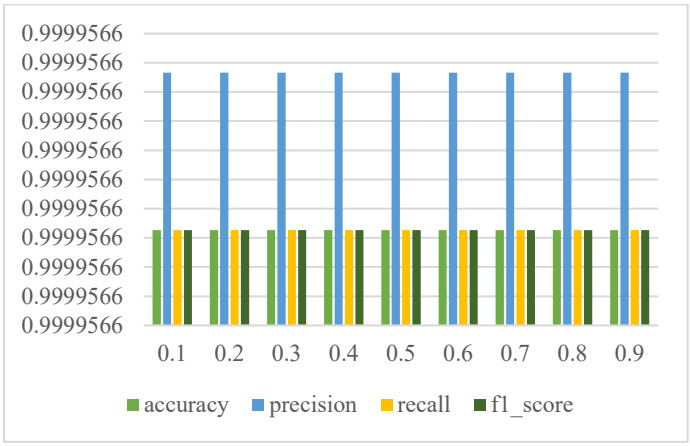

Figure 4: Sensitivity analysis of the hybrid feature selection method on BoT-IoT dataset

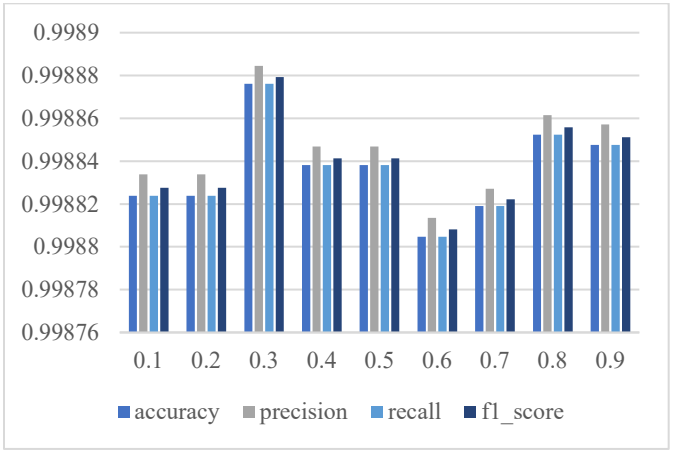

Figure 5: Sensitivity analysis of the hybrid feature selection method on TON-IoT dataset

5.5. Classification Results

Table 6 summarizes the classification performance of the proposed intrusion detection system on the ACT-IoT dataset under both binary and multiclass classification settings. The RF classifier achieves the highest performance across all evaluation metrics, obtaining an accuracy of 99.98% for both binary and multiclass tasks. Decision Tree and K-NN classifier also deliver highly competitive results, with accuracy values close to RF that confirms the effectiveness of the selected hybrid features. The SVM classifier performs well in binary classification but exhibits a performance drop in the multiclass scenario, indicating challenges in modelling complex class separations. The NB classifier shows high recall but relatively lower precision in binary classification, reflecting a higher false alarm rate, while its multiclass performance improves significantly.

Table 7 presents the performance on BoT-IoT dataset. The RF classifier achieves the high performance with an accuracy of 99.99% for binary and 99.98% for multiclass classification, along with consistently high precision, recall, and F1-score values. DT

and K-NN classifiers also demonstrate outstanding performance, closely matching RF with minimal performance degradation, indicating the effectiveness of the selected features in capturing attack patterns. In contrast, SVM shows comparatively lower performance, particularly in the multiclass scenario, suggesting limitations in handling complex class boundaries. The NB classifier exhibits the weakest performance, characterized by high recall but low precision, leading to increased false alarms.

In Table 8, The RF classifier achieves the highest overall performance with highest accuracy and balanced precision-recall values in both classification settings which indicates the strong capability to model complex IoT traffic pattern. DT and K-NN classifier show competitive performance with accuracy values close to RF. The SVM delivers moderate results with reduced precision in binary classification, reflecting a higher misclassification rate for certain attack types. The NB classifier exhibits very high recall but relatively low precision in binary classification task which leads to increased false alarms.

Accordingly, to results, RF achieves the highest performance than other classifiers across all datasets. The proposed hybrid feature selection approach enhances the detection accuracy and reliability across diverse IoT network environments. The confusion metrics using proposed hybrid feature selection and RF classifier are shown in Figure. 6, Figure. 7, and Figure. 8 for binary classification. The confusion metrices of multiclass classification are shown in Figure. 9, Figure. 10, Figure. 11 for three datasets respectively.

5.6. Performance Comparison

In Table 9, the performance comparison of original features, Chi-square selected features, features with MI, feature with RF and proposed hybrid feature selection method on three datasets. The results indicate that individual feature selection methods provide marginal performance improvements over original feature set across all datasets. The proposed hybrid feature selection method either matched or slightly improved detection performance while reducing execution time. Therefore, the hybrid method stability across diverse datasets and data imbalance conditions confirms the robustness and generalizability.

Table 6: Classification result on ACI IoT Dataset

	Binary Classification				Multi-class Classification			
Method	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
RF	0.9993	0.9995	0.9996	0.9995	0.9993	0.9993	0.9993	0.9993
DT	0.9989	0.9992	0.9993	0.9992	0.9989	0.9989	0.9989	0.9989
k-NN	0.9989	0.9992	0.9993	0.9992	0.9989	0.9989	0.9989	0.9989
SVM	0.9944	0.9987	0.9995	0.9988	0.8736	0.9091	0.8736	0.8844
NB	0.8379	0.8233	0.9915	0.8996	0.9333	0.9611	0.9383	0.9455

Table 7: Classification result on BoT IoT Dataset

	Binary Classification				Multi-class Classification			
Method	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
RF	0.9999	1.000	0.9998	0.9999	0.9998	0.9998	0.9998	0.9998
DT	0.9999	0.9999	0.9998	0.9999	0.9996	0.9996	0.9996	0.9996
k-NN	0.9999	0.9999	0.9998	0.9999	0.9996	0.9996	0.9996	0.9996
SVM	0.9685	0.9756	0.9610	0.9683	0.8962	0.9031	0.8962	0.8972
NB	0.6360	0.5800	0.9861	0.7304	0.7473	0.8508	0.7473	0.7414

Table 8: Classification result on BoT IoT Dataset

	Binary Classification				Multi-class Classification			
Method	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
RF	09990	0.9799	0.9876	0.9834	0.9901	0.9889	0.9901	0.9892
DT	0.9983	0.9762	0.9734	0.9734	0.9891	0.9882	0.9891	0.9885
k-NN	0.9983	0.9762	0.9734	0.9730	0.9891	0.9882	0.9891	0.9885
SVM	0.9946	0.8675	0.9811	0.9208	0.9584	0.9722	0.9584	0.9633
NB	0.9870	0.7094	0.9998	0.8229	0.9618	0.9766	0.9618	0.9684

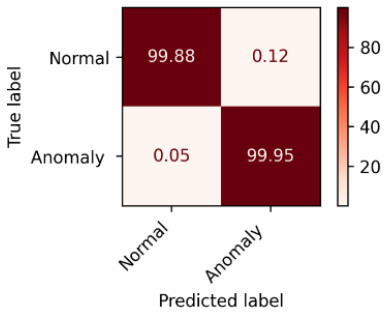

Figure 6: Confusion matrix of binary classification on ACI-IoT dataset

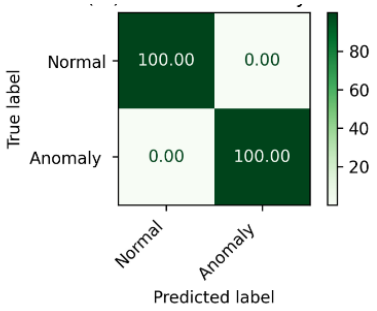

Figure 7: Confusion matrix of binary classification on BoT-IoT dataset

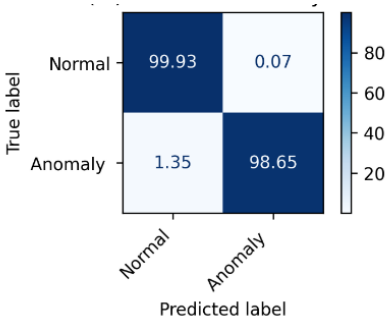

Figure 8: Confusion matrix of binary classification On TON-IoT dataset

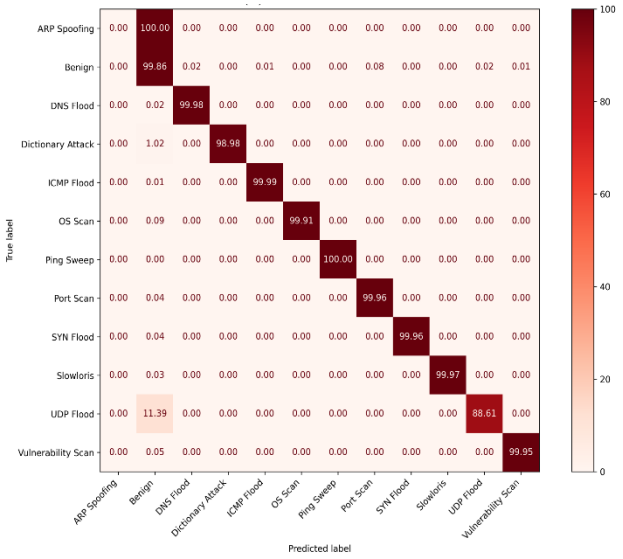

Figure 9: Confusion matrix of multiclass classification on ACI-IoT dataset

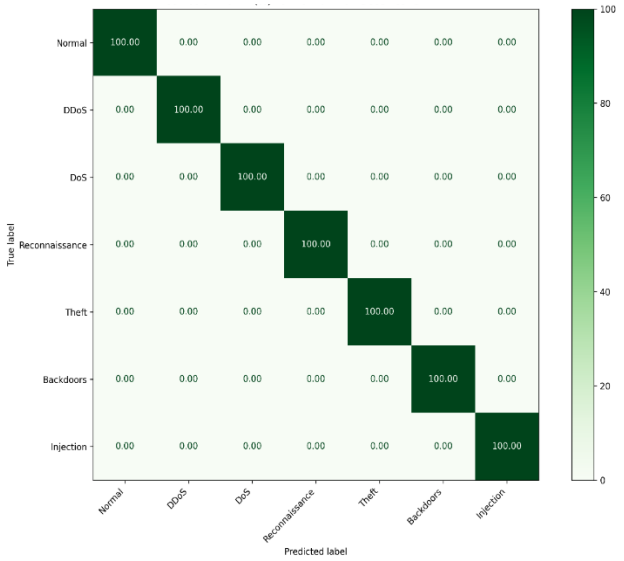

Figure 10: Confusion matrix of multiclass classification on BoT-IoT dataset

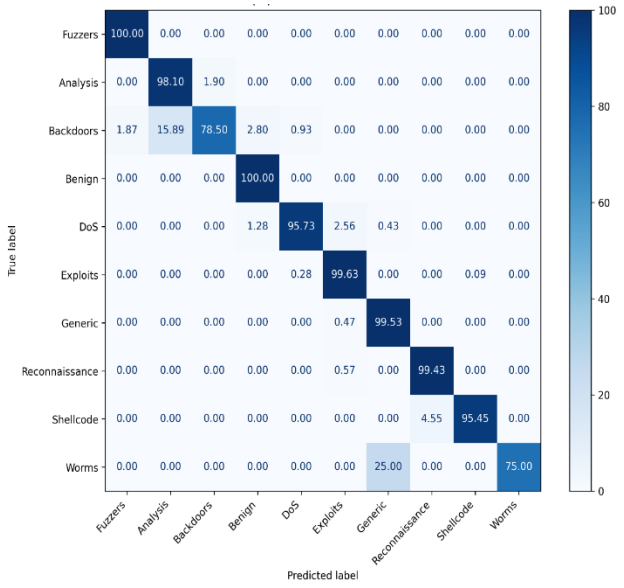

Figure 11; Confusion matrix of multiclass classification on TON-IoT dataset

Table 9; Comparison of classification accuracy on three datasets using RF classifier

Features	ACI-IoT	BoT-IoT	TON-IoT
Original	0.9987	0.9999	0.9982
Chi-square	0.9988	0.9999	0.9982
MI	0.9992	0.9999	0.9963
RF	0.9993	0.9999	0.9966
Proposed	0.9993	0.9999	0.9990

Table 10 presents the comparison of proposed system and state-of-the-art researches. The results indicates that the hybrid approach reduces number of features while retaining the high performance.

Table 10: Comparison of performance with Previous Studies

Reference	Methods	Dataset	Scenario	Accuracy	Precision	Recall	F1-Score
[6]	Deep Learning	TON-IoT	binary	0.9987	0.9987	0.9987	0.9987
			multiclass	0.9949	0.9949	0.9949	0.9949
[8]	Hybrid feature selection, Multistage classification	TON-IoT	multiclass	0.9883	0.9856	09876	0.9865
		BoT-IoT	multiclass	0.9860	0.9850	0.9863	09894
[9]	MI+SFS, Random Forest	BoT-IoT	binary	0.9999	0.9958	0.9947	0.9953
[10]	RNN (MLP)	BoT-IoT	multiclass	0.8400	0.8400	0.8400	0.8200
Proposed	Hybrid feature selection, RF	TON-IoT	binary	09990	0.9799	0.9876	0.9837
		TON-IoT	multiclass	0.9998	0.9998	0.9998	0.9998
		BoT-IoT	binary	0.9999	1.0000	0.9998	0.9999
		BoT-IoT	multiclass	0.9998	0.9998	0.9998	0.9998

6. Conclusion

The proposed hybrid feature selection method integrates filter techniques with an embedded method based on Random Forest importance scores through a hybrid feature scoring optimization strategy. The experimental results demonstrate that the approach improves anomaly detection performance while effectively reducing feature dimensionality across five supervised machine learning classifiers. Among the classifiers, RF attains the highest performance with accuracy of 99.93%, 99.99%, 0.99.93% on ACI-IoT2023, BoT-IoT, and TON-IoT datasets respectively. In future work, the proposed feature selection approach will be extended to deep learning models to explore their potential in capturing complex spatial and temporal patterns in IoT traffic. Additionally, the model will be optimized for deployment on resource-constrained IoT devices to facilitate real-time and energy-efficient anomaly detection at the network edge with minimal computational overload.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

For the conceptualization, MSSMoe; methodology, MSSMoe; analysis, MSSMoe; data collection, MSSMoe; investigation and validation, MSSMoe and WMOo, and writing paper and editing, MSSMoe; and visualization, WMOo and MSSMoe; Supervision WMOo. All authors have read and edit to the manuscript.

The authors would like to express their sincere gratitude to the researchers and institutions who developed and made publicly available the ACI-IoT2023, BoT-IoT, TON-IoT datasets. The authors appreciate the efforts of the dataset creators in supporting the research community through open and accessible data resources.

References

[1] T.M. Ghazal, M.K. Hasan, M.T. Alshurideh, H.M. Alzoubi, M. Ahmad, S.S. Akbar, B. Al Kurdi, I.A. Akour, "IoT for Smart Cities: Machine Learning Approaches in Smart Healthcare—A Review," Future Internet, **13**(8), 218, 2021, doi:10.3390/fi13080218..

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, "Internet of Things for Smart Cities," IEEE Internet of Things Journal, **1**(1), 22–32, 2014, doi:10.1109/JIOT.2014.2306328.

[3] E. Gyamfi, A. Jurcut, "Intrusion detection in internet of things systems: A review on design approaches leveraging multi-access edge computing, machine learning, and datasets," Sensors, **22**(10), 1–33, 2022, doi.org/10.3390/s22103744

[4] H. Wang, T.M. Khoshgoftaar, K. Gao, "A comparative study of filter-based feature ranking techniques," in IEEE International Conference on Information Reuse and Integration, 43–48, 2010, doi.org/10.1109/IRI.2010.5558913

[5] M.A.M. Hasan, M. Nasser, S. Ahmad, K.I. Molla, "Feature selection for intrusion detection using random forest," Journal of Information Security, **7**, 129–140, 2016, doi:10.4236/jis.2016.73009

[6] D.S. Ahmed, "Anomaly-based network intrusion detection system for IoT using deep learning model," Scientific Research Journal of Engineering and Computer Science, **3**(5), 16–23, 2023, doi: 10.47310/srjecs.2023.v03i02.010

.[7] G. Guo, X. Pan, H. Liu, F. Li, L. Pei, K. Hu, "An IoT intrusion detection system based on TON-IoT network dataset," in IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), 333–338, 2023, doi.org/10.1109/CCWC57344.2023.10099144

[8] G. Logeswari, J.D. Roselind, K. Tamilarasi, V. Nivethitha, "A comprehensive approach to intrusion detection in IoT environments using hybrid feature selection and multi-stage classification techniques," IEEE Access, **13**, 24970–24987, 2025.

[9] A.G. Ayad, N.A. Sakr, N.A. Hikal, "A hybrid feature selection model for anomaly-based intrusion detection in IoT networks," in International Telecommunications Conference (ITC-Egypt), 1–7, 2024.

[10] F.A. Alotaibi, S. Mishra, "Cyber security intrusion detection and bot data collection using deep learning in the IoT," International Journal of Advanced Computer Science and Applications, **15**(3), 421–432, 2024.

[11] I. Kerrakchou, A.A. El Hassan, S. Chadli, M. Emharraf, M. Saber, "Selection of efficient machine learning algorithm on BoT-IoT dataset for intrusion detection in internet of things networks," Indonesian Journal of Electrical Engineering and Computer Science, **31**(3), 1784–1793, 2023, doi:10.11591/ijeecs.v31.i3.pp1784-1793

[12] J. Han, J. Pei, M. Kamber, Data Mining: Concepts and Techniques, 3rd ed., Elsevier, 2011, ISBN 9780123814791.

[13] N. V Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, **16**, 321–357, 2002, doi:10.1613/jair.953.

[14] J. Li, M.S. Othman, H. Chen, L.M.M. Yusuf, "Optimizing IoT intrusion detection system: Feature selection versus feature extraction in machine learning," Journal of Big Data, **11**(1), 36, 2024, doi.org/10.1186/s40537-024-00892-y.

[15] S. Walling, S. Lodh, "Network intrusion detection system for IoT security using machine learning and statistical based hybrid feature selection," Security and Privacy, **7**(6), 2024, doi:10.1002/spy2.429

[16] P.N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Pearson Education India, 2016.

[17] N. Hoque, D.K. Bhattacharyya, J.K. Kalita, "MIFS-ND: A mutual information-based feature selection method," Expert Systems with Applications, **41**(14), 6371–6385, 2014, doi:10.1016/j.eswa.2014.04.019.

[18] L. Breiman, "Random forests," Machine Learning, **45**(1), 5–32, 2001..

[19] X. Yuan, S. Liu, W. Feng, G. Dauphin, "Feature importance ranking of random forest-based end-to-end learning algorithm," Remote Sensing, **15**(21), 5203, 2023, doi: 10.3390/rs15215203.

[20] P. Rani, R. Kumar, A. Jain, "A hybrid approach for feature selection based on correlation feature selection and genetic algorithm," International Journal of Software Innovation, **10**(1), 1–17, 2022, doi: 10.4018/IJISMD.2021040102

[21] V. Kantharaju, others, "Machine learning based intrusion detection framework for detecting security attacks in internet of things," Scientific Reports, **14**(1), 30275, 2024, doi:10.1038/s41598-024-81535-3

[22] A. Liaw, M. Wiener, "Classification and regression by random forest," R News, **2**(3), 18–22, 2002.

[23] J.R. Quinlan, "Induction of decision trees," Machine Learning, **1**(1), 81–106, 1986.

[24] T.M. Mitchell, Machine Learning, McGraw-Hill, 1997.

[25] T. Cover, P. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, **13**(1), 21–27, 1967, doi:10.1109/TIT.1967.1053964.

[26] C. Cortes, V. Vapnik, "Support-vector networks," Machine Learning, **20**, 273–297, 1995, doi: 10.1007/BF00994018

[27] C.J. Vargas, others, "Intrusion detection in Internet of Things systems: A feature extraction with Naive Bayes classifier approach," Journal of Machine and Computing, **4**(1), 2024, doi: 10.53759/7669/jmc202404003.

[28] Australian Centre for Internet of Things Security Research, ACI-IoT 2023 Dataset, 2023.

[29] N. Moustafa, J. Slay, The BoT-IoT dataset: A comprehensive benchmark for IoT network intrusion detection, 2020.

[30] N. Moustafa, others, TON-IoT: A multi-source dataset for evaluating federated and centralized intrusion detection systems in IoT, 2021.

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 11, No. 2, 30-42 (2026)
www.astesj.com
Special Issue on Multidisciplinary Frontiers in Engineering, Computing and Applied Sciences

ASTES Journal
ISSN: 2415-6698

TL-SOC: A Hybrid Decision-Centric Intrusion Detection Framework for Security Operations Centers

Imane Lotfi*, Meriem Mandar

Laboratory of Mathematics, Artificial Intelligence and Digital Learning, Higher Normal School of Casablanca, Hassan II University, Casablanca, Morocco

Email(s): imane.lotfi@enscasa.ma (I. Lotfi), m.mandar@enscasa.ma (M. Mandar)

*Corresponding Author: Imane Lotfi, Higher Normal School of Casablanca, Hassan II University, Casablanca, Morocco. Email: imane.lotfi@enscasa.ma

ARTICLE INFO

Article history:
Received: 12 March, 2026
Revised: 27 March, 2026
Accepted: 1 April, 2026
Online: 23 April, 2026

Keywords:
Intrusion Detection Systems
Security Operations Centers
Hybrid Intrusion Detection
Deep Anomaly Detection
Zero-Day Detection
Out-of-Distribution Attacks
Advanced Persistent Threats
Explainable AI

ABSTRACT

Security Operations Centers (SOCs) require intrusion detection systems that achieve high detection accuracy while maintaining a low false-positive rate and robustness to evolving attack patterns. However, most existing machine learning-based approaches primarily focus on detecting known threats and often overlook distribution shifts and the reliability of generated alerts. In this paper, we propose TL-SOC, a decision-centric intrusion detection framework that integrates anomaly representation learning and supervised classification within a unified architecture. The proposed system combines a CNN–Transformer autoencoder, a graph neural network autoencoder, and an XGBoost classifier, whose outputs are fused through a meta-learning decision layer operating under explicit false-positive-rate constraints. To enhance transparency, SHAP-based explanations are incorporated to provide both global and local interpretability of detection outcomes. Experimental results on the CICIDS-2017 dataset show that TL-SOC achieves a precision of 99.63%, a recall of 74.44%, and an F1-score of 85.21%, while maintaining a very low false-positive rate (5.52×10^{-4}). The framework also demonstrates strong robustness under time-series and out-of-distribution scenarios, achieving competitive performance in cross-dataset evaluation. These results highlight the effectiveness of decision-centric hybrid architectures for reliable and operational intrusion detection in SOC environments.

1. Introduction

Lots of research shows how crucial Security Operations Centers (SOCs) are for keeping an eye on networks and finding cyber-attacks as they happen. Within a SOC, Intrusion Detection Systems (IDSs) are really important as they spot anything strange that could be a sign of someone malicious being inside. But today's dangers, for instance zero-day attacks and Advanced Persistent Threats (APTs), are extremely hard to find and predict with the normal machine learning methods of detection; they change and adjust themselves all the time. Right now, most intrusion detection systems (IDS) are overly concerned with correctly naming attacks that are already known, and don't pay enough attention to practical things like adapting to changes in the usual network activity, or making sure they don't give too many false alarms. In fact, Bass and colleagues (as in [1]) said that for cybersecurity systems to be dependable in their judgements, we really need to combine information from multiple sources - and that's much more than just getting the highest possible accuracy. And what's more, recent advances in deep learning have allowed us to find unusual activity, by providing a more complete description of characteristics and a better way to model complex network flows. Naseer and others in [2] showed how well these models work at detecting break-ins. Similarly, in [3] variational autoencoders were used to locate oddities in network traffic and Kwon's detailed review from [4] explained the potential of deep learning for security. However, a single model on its own isn't good at understanding all the different ways an attack might happen and can be unreliable when faced with new or cleverly hidden threats. Because of these drawbacks, many have suggested hybrid or ensemble intrusion detection systems that merge different ways of detecting intrusions, and so offer a more comprehensive approach. In [5], the authors explained a hybrid framework that merges several models. Re-

search [6] builds a hybrid model that combines machine learning and deep learning in the cloud. Also, the reliable detection of multi-model heterogeneous outputs has been studied through decision fusion. In [7], the authors studied methodology of data fusion in intrusion detection. In [8], decision-level fusion improves the real-time detection of cyber-physical systems. Most state-of-the-art techniques are accuracy-centric and fail to tackle the specific challenges that SOCs face, such as the need to control the amount of false positives, the need for robustness to shifts in the input data distribution, and the need for explainable detection. This is particularly important in operational SOC environments, where losing reliable and actionable alerts to the system is a huge problem. This paper presents TL-SOC, a decision-centric hybrid intrusion detection system model that is specifically tailored to the SOC environment. This framework integrates supervised learning for classification and representation learning for anomalies in a unified decision-making architecture that consists of a convolutional neural network and transformer (CNN-Transformer) autoencoder, a graph neural network (GNN) autoencoder, and an XGBoost classifier. These models are integrated in a meta-learner decision layer that operates under stringent false-positive-rate constraints, resulting in highly reliable and accurate detection. An advanced SHAP-based interpretability module is also included to provide explanations at both the feature and decision levels. This consequently assists SOC analysts in their tasks and enhances their confidence in the detection results. This work extends our previous study presented at ICCSC 2025 [9] by introducing the TL-SOC architecture and providing a comprehensive evaluation under realistic conditions, including time-series splits, out-of-distribution scenarios, and cross-dataset validation. These extensions aim to better reflect real-world SOC deployment constraints and assess the robustness and generalization capability of the proposed framework.

1.1. Contributions

This work summarizes several main contributions:

1. **Decision-centric TL-SOC architecture:** We illustrate a hybrid intrusion detection model designed for SOC environments that merges anomaly representation learning with a decision pipeline through supervised learning. The architecture employs a CNN-Transformer autoencoder, a graph neural network autoencoder, and an XGBoost classifier. It is purposefully built to meet operational requirements involving low false positives and dependable alerting.

2. **Meta-learning-based fusion:** We propose, for the first time, a decision-level fusion strategy using meta-learning, capable of dynamically fusing detection signals of varying nature (like anomaly scores and classification probabilities). This allows the model to fully utilize the complementary aspects of the inputs, particularly enhancing the model's performance under distribution shifts and previously unencountered attack patterns.

3. **False-positive-aware calibration:** We introduce a threshold calibration approach that aims to meet specific constraints on the false-positive rate. This calibration aligns the detection process with the operational requirements of the SOC by reducing alert fatigue while sustaining a high level of detection.

4. **Advanced explainability:** We offer participants the opportunity to employ a SHAP-based framework for the purposes of interpretability. This approach provides global feature attribution and local decision explanations, which together improve the opacity of the process and assist SOC analysts in comprehension and sanctioning the detection results.

5. **Robustness evaluation:** Assessing the model's robustness is done via a thorough evaluation under realistic settings. Evaluations include time-series splits to avoid data leakage, zero-day attack simulations, out-of-distribution scenarios (APT-like attacks), cross-dataset tests. This set of experiments seeks to determine the generalization capabilities of the proposed framework.

6. **Comparative analysis:** We have carried out comprehensive testing for numerous intrusion detection systems, including anomaly-based models, supervised models, and hybrid models. The tradeoff between detection performance and robustness illustrates the value of the TL-SOC framework when operational constraints are imposed.

2. Related Work

2.1. Machine Learning for Intrusion Detection

Techniques used in Artificial Intelligence get employed in Network Intrusion Detection Systems as they can detect anomalous patterns in a network and traffic pattern recordings. In extensive Networks, Deep Neural Networks are proficient in identifying malicious activity.

In [2], the authors explain how deep learning can result in a higher level of detection for network traffic, thanks to the learning of hierarchies. One of the most popular techniques, and the focus of much recent research, is the autoencoder and its application to anomaly detection in network traffic by learning a compact representation of normal traffic and detecting anomalies based on reconstruction error. Similar to [10], the authors of the current work propose an autoencoder-based approach to network anomaly detection. Similarly, in [3] a variational autoencoder was proposed to enhance the detection of anomalies in network flow data. In [11], an LSTM-based autoencoder was used to capture the temporal aspects of network traffic. The authors in [4] present an excellent and comprehensive survey on deep learning-based techniques for the detection of network anomalies. All these advancements notwithstanding, when applied in practice, standalone deep learning solutions still face significant challenges. In particular, these models are typically limited in terms of adaptable and reliable decision thresholds, as well as interpretability, and they tend to lose their robustness to novel attack patterns. This fosters the need for hybrid and decision-centric approaches that are more in tune with the actual requirements of a SOC.

2.2. Hybrid and Ensemble Intrusion Detection Systems

To enhance performance and increase the reliability of detection systems, a number of approaches have proposed hybrid intrusion detection systems based on the integration of different machine learning and deep learning techniques.

In [5], the authors suggested a hybrid architecture consisting of deep learning, random forest, and clustering models, with the aim of improving detection performance. Also, [6] proposed a hybrid model that combined machine learning and deep learning for intrusion detection in cloud environments. Other research work has focused on the use of ensemble and stacking techniques for performance improvement. Authors of [12] proposed a model of stacked ensemble learning for wireless network intrusion detection. [13] developed a stacking ensemble of deep learning models in order to detect intrusions in the Internet of Things. Furthermore, in [14] the authors proposed HDLNIDS, which is described as a hybrid deep learning based intrusion detection system that combines various deep learning components. However, in spite of these innovations, the majority of hybrid and ensemble approaches focus primarily on boosting classification performance under in-distribution conditions. Likewise, they often ignore operational constraints, such as control of the false positive rate, decision tuning, and robustness regarding distributional shifts. This limits the approaches to real world SOC applications, where detection systems must operate under the conditions of constant change and novel attack scenarios. This gap aims for the development of decision-centric frameworks, including the proposed TL-SOC, which integrate heterogeneous detection mechanisms in an operationally bound context.

2.3. Decision Fusion and Meta-Learning in Cybersecurity

Decision fusion methods have been refined to improve the robustness and reliability of intrusion detection systems.

The initial work in [1] applied the principles of multisensory data fusion to cybersecurity decision-making. More recent work has examined data-level and feature-level fusion approaches for synthesizing multiple detection models. While an example of feature fusion was reported to improve the performance of deep learning-based intrusion detection systems in [15], the authors of [7]examined data fusion methods for intrusion detection. Methods for decision fusion were used to consolidate the outputs of different models in a collaborative manner. In [8], the authors illustrate the usefulness of decision fusion for real-time intrusion detection with higher detection reliability in industrial cyber-physical systems. More recent works have used meta-learning as a candidate for flexible decision-making in intrusion detection. The author of [16] proposes a meta-learning structure for intrusion detection with limited attempts. Likewise, in [17] show that such techniques can enhance the detection of anomalies in cyber-physical systems. These techniques allow models to combine multiple data sources and adapt to novel attack strategies. Despite significant progress in the field, most of the works focus on enhancing the detection capability and overlook the equally important operational constraints, such as the need to maintain a low false alarm rate, the loss of decision reliability, and robustness to changes in the probability distribution. These shortcomings are likely to affect the operational applicability of systems located in a security operations center (SOC), where intrusion detection systems are integrated in a highly reliable and stable manner.

To mitigate these issues, this research proposes **TL-SOC**. This is a decision-centric intrusion detection framework that employs deep learning for anomaly representations, hybrid detection models, and decision fusion via meta-learning, all constrained by a defined false positive rate. This approach enables robust, interpretable, and well-calibrated detection, which is particularly beneficial for SOCs.

3. Proposed TL-SOC Framework

3.1. Overview of the TL-SOC Architecture

The innovative TL-SOC architecture aims to improve cyberattack detection and decision support in security operations centers (SOCs) through an integrated pipeline of deep anomaly modeling, supervised validation, and explainable alerts. Figure 1 illustrates this pipeline. An autoencoder, designed using CNN and Transformer architectures, learns the latent representations of normal traffic and those attributable to reconstruction error, then generates anomaly scores. These scores are then refined by a supervised post-filtering model based on XGBoost, which improves the discrimination of activity (benign or malicious) and thus reduces the number of false positives. Alerts are triggered by a post-filtering model based on a calibrated threshold to limit the false positive rate. The system also incorporates a SHAP explainability model to provide SOC analysts with a detailed explanation of the alert, identifying the most critical features. Consequently, the architecture promises reasonably reliable detection of known attacks and emerging risks such as advanced persistent threats (APTs) and zero-day attacks.

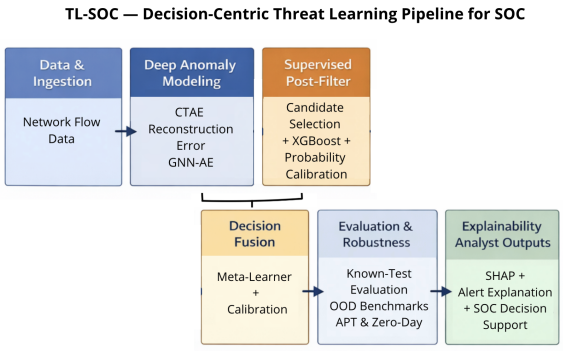

Figure 1: Overview of the proposed TL-SOC pipeline combining deep anomaly modeling, supervised decision refinement, and explainable SOC alert analysis.

3.2. Representation Learning with CNN–Transformer Autoencoder

The first step in the TL-SOC pipeline consists of anomaly-based representation learning. This learning process, which is part of the TL-SOC pipeline, uses a CNN-Transformer autoencoder (CTAE) to learn representations applied to tabular network flow data. After preprocessing steps (cleaning, encoding, and normalization), each network flow is represented by a feature vector $\mathbf{x} \in \mathbb{R}^d$. With the development of end-to-end models, an increasing number of hybrid models integrating different components

are emerging. Examples include convolutional neural networks (CNNs) and Transformer-type architectures. CNNs excel at handling local interactions, while Transformers extend to global interactions through self-attention. Composed of multiple layers of each type of module, fully trained layers should be capable of capturing dependencies at both the local and global levels within their respective domains.

$$\mathbf{z} = f_\theta(\mathbf{x}), \tag{1}$$

where f_θ denotes the encoder parameters. The decoder then reconstructs the original feature vector

$$\hat{\mathbf{x}} = g_\phi(\mathbf{z}), \tag{2}$$

where g_ϕ represents the decoder function.

The autoencoder was able to learn a compact representation of typical network behavior because the model was trained only on typical traffic. The reconstruction error between the input and its reconstruction is used to identify anomalies during inference. The anomaly score produced by the CTAE module is defined as

$$s_{\mathrm{AE}}(\mathbf{x}) = \frac{1}{d}\,\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\,, \tag{3}$$

where d denotes the number of input features. Higher reconstruction errors indicate stronger deviations from learned normal patterns and thus correspond to higher anomaly likelihood. The resulting score sAE(x) constitutes the first signal used by the TL-SOC fusion layer described later in Section 3.5.

3.3. Graph Neural Network Autoencoder

Neural networks (NNs) are used in many artificial intelligence (AI) applications. Even in the design and development of traffic monitoring systems, certain modern methods are being implemented. The TL-SOC framework uses a generalized neural network-based autoencoder (GNN-AE). Unlike traditional autoencoders, where each instance is processed independently, the GNN-AE encodes a graphical model of the traffic space to capture local dependencies.

Let $x_i \in \mathbb{R}^d$ denote the feature vector representing the i-th network flow after preprocessing and normalization. A graph $G = (V, E)$ is presented where each node $v_i \in V$ corresponds to a flow sample x_i. The edges E are defined using a k-nearest neighbor (k-NN) strategy based on feature similarity. For each node v_i, a set of neighbors $\mathcal{N}(i)$ is obtained by selecting the k closest samples in the feature space using the Euclidean distance:

$$\mathcal{N}(i) = \arg\operatorname{topk}_{j\neq i}\ \|x_i - x_j\|_2 \tag{4}$$

The local graph is created to illustrate the adjacency structure. In this graph, network flows are closely linked. To design it, a small sample of normal traffic is used to describe the distribution and topology of typical traffic. For each node v_i, the information from its neighborhood is aggregated using a mean aggregation operator:

$$h_i = \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} x_j \tag{5}$$

The pair (x_i, h_i) forms the input to the GNN autoencoder. The encoder maps these inputs into a latent representation z_i through a nonlinear transformation:

$$z_i = \sigma\left(W_e[x_i \,\|\, h_i] + b_e\right) \tag{6}$$

where $[\cdot\|\cdot]$ denotes feature concatenation, W_e and b_e are learnable parameters, and $\sigma(\cdot)$ is a nonlinear activation function. The decoder then reconstructs the original flow representation:

$$\hat{x}_i = W_d z_i + b_d \tag{7}$$

The model is trained to minimize the reconstruction loss over benign samples:

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_{i=1}^{N}\|x_i - \hat{x}_i\|_2^2 \tag{8}$$

During inference, flows that deviate significantly from the learned structure produce larger reconstruction errors. The anomaly score produced by the GNN-AE is therefore defined as

$$s_{gnn}(x_i) = \|x_i - \hat{x}_i\|_2^2 \tag{9}$$

This graph-based anomaly representation complements the reconstruction signal produced by the CNN–Transformer autoencoder. While the CTAE captures feature-level dependencies within traffic flows, the GNN-AE models relational similarities between flows, enabling the TL-SOC framework to detect anomalies that arise from structural deviations in the traffic space.

3.4. Supervised Post-Filtering with XGBoost

To finalize the detection process, TL-SOC integrates a supervised classification module based on XGBoost. While autoencoder-based models identify deviations from normal behavior, supervised learning makes it possible to capture the discriminating patterns associated with known malicious activities.

Let $\mathbf{x} \in \mathbb{R}^d$ be the normalized feature vector representing a network flow. The XGBoost classifier learns a function that estimates the probability that the flow corresponds to malicious activity. Formally, the predicted probability is expressed as

$$p_{\mathrm{XGB}}(\mathbf{x}) = \sigma\left(\sum_{k=1}^{K} f_k(\mathbf{x})\right), \tag{10}$$

where f_k denotes the k-th regression tree in the ensemble, K is the number of trees, and $\sigma(\cdot)$ is the logistic function transforming the aggregated tree outputs into a probability score. The resulting probability $p_{\mathrm{XGB}}(\mathbf{x})$ represents the supervised detection signal of the TL-SOC pipeline. This score is then combined with the anomaly scores produced by the CTAE and GNN modules within the meta-fusion layer described below..

3.5. Decision Fusion and Threshold Calibration

To produce a reliable detection decision, the TL-SOC framework combines the outputs of the anomaly detection and supervised classification modules through a meta-level fusion step that integrates complementary signals from the CNN-Transformer autoencoder, the GNN-based anomaly model, and the XGBoost classifier.

Let $s_{\mathrm{AE}}(\mathbf{x})$ be the anomaly score generated by the CTAE module, $s_{\mathrm{GNN}}(\mathbf{x})$ be the anomaly score produced by the GNN-based model, and $p_{\mathrm{XGB}}(\mathbf{x})$ be the probability of malicious intent estimated by XGBoost. These signals are first aggregated into a meta-feature vector.

$$\mathbf{m}(\mathbf{x}) = [s_{\mathrm{AE}}(\mathbf{x}), s_{\mathrm{GNN}}(\mathbf{x}), p_{\mathrm{XGB}}(\mathbf{x})]. \tag{11}$$

A logistic regression meta-learner then combines these features to produce an intermediate detection score

$$p_{\mathrm{raw}}(\mathbf{x}) = \sigma\Big(\mathbf{w}^{\top}\mathbf{m}(\mathbf{x}) + b\Big), \tag{12}$$

where $\mathbf{w}$ denotes the learned fusion weights, b is a bias term, and $\sigma(\cdot)$ is the logistic function. To improve probability reliability, the raw score is further calibrated using a monotonic calibration function $C(\cdot)$ (implemented through isotonic regression in our pipeline):

$$F(\mathbf{x}) = C(p_{\mathrm{raw}}(\mathbf{x}))\,. \tag{13}$$

Finally, the alert decision is obtained by applying a threshold selected under a false-positive-rate constraint:

$$y(\mathbf{x}) = \mathbb{I}(F(\mathbf{x}) \geq \tau_{\mathrm{FPR}})\,, \tag{14}$$

where τ_{FPR} is chosen on validation data to satisfy a target false positive rate. This decision-centric formulation ensures stable alert generation in SOC environments while maintaining strong detection capability.

3.6. Explainable Alert Analysis (SHAP)

For the purpose of building trust with customers through transparency, the TL-SOC framework incorporates an explainability module based on SHAP (SHapley Additive ExPlanations). This module provides explainability at the level of features for pipeline alerts, making it possible for SOC analysts to gain insight into the reasons for the abnormal behavior. Let $F(\mathbf{x})$ be the calibrated detection score produced by the TL-SOC pipeline for a network flow $\mathbf{x}$. SHAP explains the prediction by assigning a contribution value ϕ_j to each input feature x_j. The model output can therefore be expressed as

$$F(\mathbf{x}) = \phi_0 + \sum_{j=1}^{d} \phi_j, \tag{15}$$

where ϕ_0 represents the baseline prediction and ϕ_j denotes the Shapley contribution of feature j.

These contributions quantify how each feature influences the final detection score. Positive values present that the feature increases the likelihood of an anomaly, while negative values reduce it. By highlighting the most influential features for each alert, SHAP provides transparent explanations that facilitate rapid investigation and decision-making by SOC analysts.

3.7. Design Rationale

Balancing the need for different, mutually beneficial detection approaches to overcome the issues associated with single-model intrusion detection systems in SOC environments is the inspiration behind the TL-SOC design. The CTAE has been used to capture both local feature interactions and global dependencies in the context of network traffic data. Convolutional layers have been used to learn localized patterns, while the transformer encoder is used to capture relationships over long distances, thus facilitating effective learning of anomalies. The GNN-AE has been proposed to learn the structural relationships of network flows. By using the k-nearest neighbor graph approach, GNN-AE is able to learn the similarities among the traffic instances and to detect anomalies concerning the underlying data structure. The XGBoost classifier, a supervised learning module, has been proposed to enhance the identification of benign and malicious traffic. Its potential to model intricate decision boundaries and to provide a ranking of feature significance renders it especially appropriate for enhancing the detection of anomalies. Finally, a meta-learning decision layer is used to integrate these different signals. This fusion mechanism adjusts the weights of the integrated information sources to enhance robustness to distribution shifts and to make detection decisions more reliable when the threshold for false positives is tight.

4. Experimental Setup

4.1. Datasets

The experiments were conducted using two publicly available reference datasets: CICIDS-2017 and CSE-CIC-IDS2018, both widely used in network intrusion detection research.

The CICIDS-2017 dataset contains realistic network traffic, including both benign activities and multiple attack scenarios such as brute-force attacks, denial-of-service (DoS), distributed denial-of-service (DDoS), infiltration, botnet activity, and web attacks. Also, each network flow is represented by a set of statistical characteristics extracted from packet-level information. After preprocessing and feature selection, each flow is represented by a numeric vector $\mathbf{x} \in \mathbb{R}^d$.

Table 1: Summary of the CICIDS-2017 dataset used in the experiments.

Property	Value
Dataset	CICIDS-2017
Total flows	2.57M
Benign flows	2.14M
Attack flows	0.42M
Attack categories	DoS, DDoS, Brute Force, Web Attacks, Botnet, Infiltration
Features used	70

The CSE-CIC-IDS2018 dataset is used for cross-dataset evaluation to assess the generalization capability of the proposed framework under distribution shift. It contains several types of attacks and more diverse traffic patterns, making it suitable for assessing robustness in heterogeneous network environments.

Table 2: Summary of the CSE-CIC-IDS2018 dataset used for cross-dataset evaluation.

Property	Value
Dataset	CSE-CIC-IDS2018
Total flows	1,869,101
Benign flows	1,660,687
Attack flows	208,414
Attack categories	DoS attacks-GoldenEye, DoS attacks-Slowloris, FTP-BruteForce, SSH-Bruteforce
Features used	70
Usage	Cross-dataset evaluation (test only)

The dataset is partitioned using both stratified and chronological splitting strategies. The stratified split is used for baseline evaluation, while the time-series split is adopted to simulate realistic SOC conditions and prevent data leakage. For cross-dataset evaluation, the model is trained on CICIDS-2017 and tested on CSE-CIC-IDS2018 without retraining.

4.2. Data Preprocessing

Before training them, several preprocessing steps were applied to prepare the network flow data.

Let $\mathbf{x} = (x_1, \ldots, x_d)$ be the feature vector representing a network flow. First, corrupted or incomplete records were removed from the dataset. Next, the categorical features were transformed and converted into numerical representations using encoding techniques such as label encoding or one-hot encoding. All numerical features were then normalized by min-max scaling to ensure comparable value ranges between different features. For each feature x_j, the normalization is defined as

$$x'_j = \frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)}, \tag{16}$$

where $\min(x_j)$ and $\max(x_j)$ denote the minimum and maximum values of feature j observed in the training data.

After normalization, each feature value is projected into the interval $[0, 1]$, producing the normalized feature vector $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_d)$. To prevent data leakage, the normalization parameters $\min(x_j)$ and $\max(x_j)$ were estimated using only the training dataset and then applied unchanged to the validation and test sets.

4.3. Model Training

The dataset was divided into training, validation, and test portions using stratified sampling. All training set models for anomaly detection were trained on benign traffic to learn the benign normal behavior of network flows. Validation set was used for calibration and thresholding and the test set was used for final evaluation. Before training begins, numerical dataset features were processed via imputation of missing values, quantile clipping using benign training samples, and MinMax normalization which was fit to normal data. The Adam optimizer was employed to train the CTAE model, with early stopping based on validation loss. Reconstruction errors were turned into anomaly scores. When the GNN-AE was enabled, it learns local relationships using a k-nearest neighbors graph built from benign samples. Additionally, a supervised post-filtering module based on XGBoost was trained on attack samples and a few hard normal samples from the validation set. The trained post-filtering was calibrated using isotonic regression. A logistic regression meta-learner combined the outputs of all these modules, and the false positive rate was set to determine the final threshold on the validation set.

4.4. Implementation Details

Using a workstation with a multi-core processor and ample memory for large-scale network traffic data processing, all experiments leveraged standard machine learning and deep learning libraries such as TensorFlow/Keras, PyTorch, and XGBoost, implemented in Python.

Tabular network flow features are processed using the CTAE architecture. Convolutional layers and multi-head self-attention are features of the Transformer encoder blocks. The integration dimension was set to 64, with 4 attention heads and a forward dimension of 128. The model was developed using the Adam optimizer, with a learning rate of 10^{-3}, a batch size of 1024, and early stopping with respect to validation loss. For the KNN graph, the GNN-AE uses $k = 10$ with respect to Euclidean distance between normalized feature vectors. A simple aggregation mechanism minimizes the reconstruction loss on benign samples. The XGBoost classifier was set to 300 estimators, a max depth of 6, a learning rate of 0.05, and subsampling at 0.8 for both instances and features. Imbalance in classes was corrected with the use of a scale_pos_weight parameter derived from the training data. Using logistic regression, the meta-learning fusion layer combines anomaly scores from CTAE and GNN-AE with XGBoost probability outputs. Isotonic regression is applied to the final detection score to further calibrate the score. The decision threshold is set on the validation set based on a given false-positive-rate, which keeps the score within the SOC operational constraints.

4.5. Benchmark Architectures

To evaluate the effectiveness of the proposed TL-SOC framework, several intrusion detection architectures were implemented and used as benchmark models.

Baseline Autoencoder–Classifier Pipeline. An XGBoost classifier and a multilayer perceptron autoencoder (MLP-AE) are combined in the first baseline. Because autoencoder-based anomaly detection models learn compact representations of typical network traffic and detect anomalies through reconstruction errors, they are frequently used in intrusion detection systems [3, 10, 11]. In this baseline, the autoencoder produces an anomaly score, while XGBoost provides supervised classification capability.

CTAE with the XGBoost classifier. The second architecture replaces the MLP autoencoder with a CTAE, enabling the model to capture both local feature interactions and global dependencies within network traffic features. Because deep neural architectures can learn complex traffic representations, they have demonstrated strong performance for network anomaly detection [2, 4]. The anomaly score produced by the CTAE model is then combined with the XGBoost classifier.

Hybrid Multi-Model Fusion Architecture. The third architecture introduces a hybrid detection strategy combining two anomaly detection models: a CTAE and a GNN-AE. In [5, 6, 14], studies have shown that hybrid intrusion detection systems incorporating multiple learning models improve the robustness and accuracy of detection. In this configuration, the anomaly scores generated by these models are combined with the XGBoost probability through a weighted fusion mechanism.

4.6. Evaluation Protocol

In order to determine the reliability of the proposed TL-SOC framework under actual deployment circumstances, extra tests were performed in out-of-distribution (OOD) conditions. In these tests, detection model deployment is followed by the emergence of new attack patterns and these flows are then combined with 95% benign traffic in order to simulate operational SOC environments.

Zero-day attacks. Zero-day attack detection is based on a hold-out protocol in which certain attack types are kept completely out of the training phase and are brought in only at the testing phase. In this paper, attack types like *Infiltration* and *Botnet* are excluded from the training dataset and used only at the testing phase. In order to mimic realistic SOC traffic distributions.

APT-like attacks. To simulate APT-like scenarios, at the time of selection, the traffic flow is assumed to be an advanced persistent threat as a result of low traffic volumes and long time intervals. First, the TL-SOC model has not been retrained on these OOD samples to evaluate its ability to generalize to new unseen attack behaviors without any iteration. Although actual SOC systems may utilize some form of continual learning or updates, this procedure aims to determine the intrinsic generalization capability of the model without updates.

In terms of evaluation scenarios, the same time-based division of the data is used, ensuring that training, validation, and test phases do not overlap, and preventing any form of information leakage. This also guarantees the same set of known versus unknown experiments.

4.7. Data Splitting Strategy

To ensure a fair and applicable evaluation of the proposed TL-SOC framework, two data allocation strategies are considered: stratified random allocation and chronological allocation.

The distribution of classes across the training, validation, and test sets is preserved by the first tactic, stratified allocation. Although this method typically produces excellent results, it may induce an optimistic bias, especially in intrusion detection circumstances where the test and training data may contain identical attack patterns. A chronological time-series split is used to overcome this restriction. Early network traffic is used for training, and later traffic is left for testing and validation. The dataset is divided based on time order. This protocol ensures strict separation between training, validation, and test sets, thereby preventing any form of data leakage and better reflecting real-world Security Operations Center (SOC) conditions, where models must detect evolving and previously unseen threats.

The transition from a stratified to a time-series split significantly increases the difficulty of the task, as the model is required to generalize under distribution shift and temporal drift. Both evaluation protocols are reported in this work to highlight the trade-off between idealized performance and realistic deployment conditions.

4.8. Evaluation Metrics

The performance of TL-SOC was evaluated using standard classification metrics widely adopted in intrusion detection research. Let TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively. Precision and Recall measure the ability of the system to correctly detect malicious traffic:

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

The F1-score summarizes the trade-off between precision and recall:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{19}$$

Because SOC environments require strict control of false alerts, the false positive rate (FPR) is explicitly monitored:

$$FPR = \frac{FP}{FP + TN} \tag{20}$$

In TL-SOC, the decision threshold is calibrated on the validation set to satisfy a predefined FPR constraint, ensuring compatibility with operational SOC requirements. To assess the models' ranking ability independently of the decision threshold, ROC-AUC and PR-AUC values are also presented. ROC-AUC measures the overall separability between legitimate and malicious traffic, while PR-AUC is particularly informative in highly unbalanced contexts. And finally, robustness was evaluated under out-of-distribution (OOD) conditions simulating realistic SOC scenarios. Two OOD settings were considered: zero-day attacks, where specific attack categories are excluded from training and evaluated only during testing, and APT-like attacks, which simulate stealthy low-rate and long-duration communication patterns. In both cases, the evaluation follows a SOC-oriented distribution where attacks are mixed with 95% benign traffic, and the models are not retrained on these OOD samples.

5. Experimental Results

To provide a clear and rigorous evaluation of the proposed TL-SOC framework, the experimental analysis is organized into two complementary settings.

1. The first setting corresponds to an in-dataset evaluation, where the model is trained and tested on CICIDS-2017 under different protocols, including stratified split, time-series split, and out-of-distribution (OOD) scenarios.

2. The second setting corresponds to a cross-dataset evaluation, where the model is trained on CICIDS-2017 and tested on CSE-CIC-IDS2018 without retraining, in order to assess generalization under distribution shift.

5.1. Detection Performance on Known Attacks

We first evaluate TL-SOC under a controlled in-dataset setting using the CICIDS-2017 dataset, which represents the baseline evaluation scenario. A stratified split was be applied to separate the dataset's 2,574,264 network flows into training, validation, and test sets. The trained model's generalization performance is evaluated using the final test set, which comprises 514,853 flows.

Table 3 reports the detection performance of TL-SOC on the test set using standard intrusion detection metrics.

Table 3: Detection performance of the proposed TL-SOC framework on the CICIDS-2017 test set.

Prec.	Rec.	F1-score	FPR	ROC-AUC	PR-AUC
0.9963	0.7444	0.8521	5.52×10^{-4}	0.9978	0.9883

The framework combines anomaly scores produced by the CTAE, structural anomaly scores generated by the GNN-AE, and classification probabilities obtained from the XGBoost post-filter. These signals are fused through a meta-learning decision layer that produces a unified detection score. The final decision threshold was calibrated on the validation set to satisfy a predefined false-positive constraint ($FPR_{target} = 5{\times}10^{-4}$). Under this configuration, TL-SOC achieves a precision of 0.9963, a recall of 0.7444, and an F1-score of 0.8521, while maintaining a very low false-positive rate of 5.52×10^{-4}. The model also achieves a ROC-AUC of 0.9978 and a PR-AUC of 0.9883, indicating strong discrimination between benign and malicious traffic. Additional indicators thus confirm the robustness of the model, with balanced accuracy of 0.8719 and a Matthews correlation coefficient (MCC) of 0.8397.

The confusion matrix obtained on the test set is reported in Table 4. The model correctly classifies the vast majority of benign flows while maintaining strong detection capability for malicious traffic. Only a small number of benign flows are incorrectly flagged as attacks, confirming the effectiveness of the FPR-constrained calibration strategy.

Table 4: Confusion matrix of TL-SOC on the CICIDS-2017 test set.

	Pred. Benign	Pred. Attack
Benign	429440	237
Attack	21774	63402

These results suggest that TL-SOC is able to achieve strong detection performance while maintaining strict control over false positives. This balance is especially important in SOC environments, where too many false alarms can quickly make analysts tired of responding to them and make it harder for them to respond to real threats. By limiting unnecessary alerts, the proposed approach helps ensure that detected events remain actionable and relevant in practice.

5.2. Impact of Data Splitting Strategy

In this section, we will compare stratified and time-series evaluation protocols to further assess robustness under realistic conditions. Table 5 shows a comparison of TL-SOC's performance under stratified versus time-series evaluation protocols.

For the stratified split, the performance is considerably better (F1-score = 0.8521) because the training and test sets are assumed to be drawn from the same distribution. Nonetheless, it is not clear how well the model generalizes, as there could be a data leakage, meaning the same attack patterns are present in both the training and testing sets. On the other hand, the time-series split, where the model is assessed using temporally separated and previously unseen attack patterns, yields a lower, yet more realistic performance (F1-score $\approx$ 0.5179). This evaluation method is more suitable in the context of real-world SOC environments that are subject to change, where attack patterns shift, and the model must generalize to new patterns in a different distribution. This is the reason why we worked on analyzing the protocols established to guide realistic expectations in intrusion detection systems. Stratified evaluation is focused on attaining the best detection performance, while time-series evaluation is focused on robustness and addressing the needs of real-world applications.

Table 5: Comparison of evaluation protocols

Protocol	Precision	Recall	F1-score	FPR
Stratified Split	0.9963	0.7444	0.8521	5.5×10^{-4}
Time-Series Split	0.8303	0.3763	0.5179	0.077

Despite the increased difficulty of the time-series setting, the TL-SOC framework maintains competitive performance, demonstrating its practical relevance for deployment in real-world SOC environments.

5.3. FPR-Constrained Threshold Calibration

Since SOC environments require strict control of false alarms, we analyze the impact of FPR-constrained threshold calibration within the in-dataset setting; maintaining a low false-positive rate is essential to avoid overwhelming security analysts with excessive alerts. For this reason, TL-SOC adopts a threshold calibration strategy that explicitly constrains the false-positive rate (FPR). The decision threshold is selected on the validation set to satisfy a predefined target FPR while preserving detection capability.

Let $s(x)$ denote the final detection score produced by the TL-SOC meta-fusion layer. A decision threshold τ is chosen such that the empirical false-positive rate on the validation set satisfies

$$FPR(\tau) \leq FPR_{target}. \tag{21}$$

In our experiments, the target value was set to $FPR_{target} = 5{\times}10^{-4}$. Using this calibration procedure, the selected threshold was $\tau = 0.9789$, resulting in a validation false-positive rate of 2.28×10^{-4} while maintaining a recall proxy of 0.6819. To analyze the trade-off between detection capability and false-positive control, additional thresholds were evaluated for several FPR targets. As the allowed FPR increases, the decision threshold decreases and the recall improves accordingly. For example, relaxing the constraint from 10^{-4} to 10^{-3} increases the recall proxy from 0.4088 to 0.7478.

Table 6: Threshold calibration under different FPR constraints.

FPR Target	Threshold τ	Recall Proxy
10^{-4}	0.9971	0.4088
5×10^{-4}	0.9789	0.6819
10^{-3}	0.9467	0.7478
2×10^{-3}	0.9176	0.8700
5×10^{-3}	0.6082	0.9117

This calibration mechanism ensures that TL-SOC remains compatible with operational SOC requirements by explicitly controlling the alert rate while preserving strong detection capability. Figure 2 illustrates the distribution of the TL-SOC anomaly scores on the validation set together with the selected decision threshold.

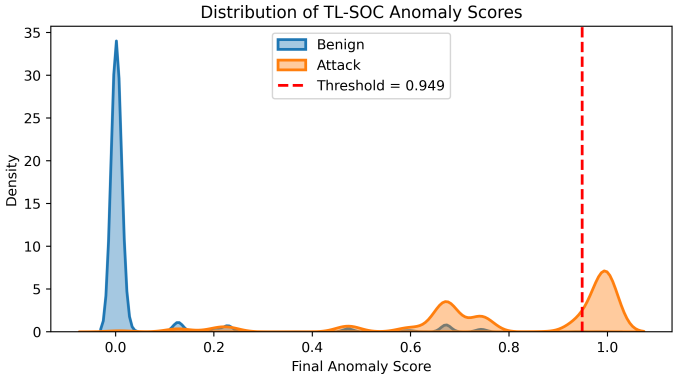

Figure 2: Distribution of TL-SOC anomaly scores for benign and attack traffic with the calibrated FPR-constrained decision threshold.

As shown in the figure, benign traffic scores remain tightly concentrated near zero, indicating that they are well reconstructed and consistent with the learned normal behavior, while malicious flows exhibit significantly higher scores, reflecting stronger deviations from normal patterns. The calibrated threshold is positioned between these two distributions, allowing a clear separation between benign and malicious traffic. This enables effective detection of anomalies while minimizing overlap between classes, while maintaining an extremely low false-positive rate essential for reliable and actionable alerts in SOC environments.

5.4. Ablation Study of TL-SOC Components

To assess the contribution of each module in the proposed TL-SOC framework, an ablation study was conducted by evaluating several configurations of the pipeline. The experiments compare the performance of the individual components (CTAE, GNN-AE, and XGBoost) with the final meta-fusion model under the same FPR-constrained calibration ($FPR_{target} = 5 \times 10^{-4}$).

Figure 3 illustrates the performance of the different configurations. Individually, the anomaly-based models remain weak. CTAE achieves a recall of 0.0054 and an F1-score of 0.0108, while GNN-AE improves slightly to an F1-score of 0.1800. In contrast, XGBoost yields strong performance (F1 = 0.8210, ROC-AUC = 0.9963). The best results are obtained with the full TL-SOC framework, which reaches F1 = 0.8521, precision = 0.9963, recall = 0.7444, ROC-AUC = 0.9978, and PR-AUC = 0.9883, while maintaining a very low false-positive rate.

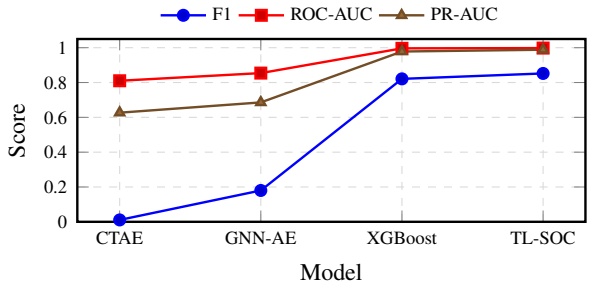

Figure 3: Performance comparison of TL-SOC components and the final meta-fusion model.

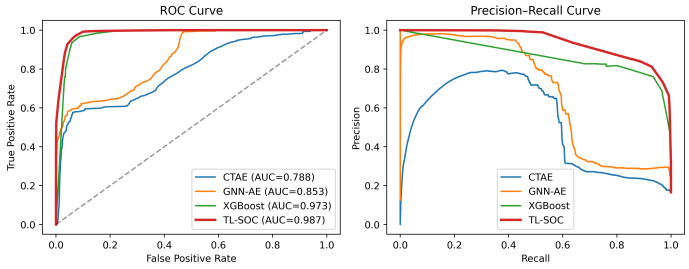

Figure 4: ROC and Precision–Recall curves comparing the individual components of the TL-SOC framework.

These results demonstrate that the different components of TL-SOC provide complementary detection signals. While anomaly-based models capture deviations from normal traffic behavior, the supervised classifier enhances discrimination between benign and malicious flows. Their integration through the meta-fusion layer leads to a more robust and accurate detection system.

5.5. Benchmarking of Detection Architectures

To assess the effectiveness of the proposed TL-SOC framework, a comparative evaluation was conducted against several alternative architectures with increasing modeling complexity. These architectures progressively integrate different detection mechanisms, enabling a systematic analysis of the contribution of each component in the final decision pipeline. The first architecture combines a reconstruction-based anomaly detector with a supervised classifier. A MLP-AE models normal traffic through reconstruction error, while an XGBoost classifier performs supervised classification of benign and malicious flows. The second architecture replaces the MLP-AE with a more expressive CTAE, enabling the model to capture more complex feature interactions in network traffic data. The third architecture introduces structural anomaly modeling by incorporating a GNN-AE. In this setup, anomaly scores produced by theses models are combined using a fixed weighted fusion strategy. Finally, the proposed TL-SOC framework replaces this fixed fusion mechanism with a meta-learning decision layer that adaptively combines the outputs of the detection modules, allowing the system to exploit the complementary strengths of anomaly detection and supervised learning.

Table 7: Benchmark comparison of the evaluated architectures on known attacks and out-of-distribution scenarios.

Arch	Known (F1)	APT (F1)	Z-Day (F1)	FPR
Arch1	0.901	0.543	0.059	6.21×10^{-4}
Arch2	**0.936**	0.226	0.088	9.82×10^{-3}
Arch3	0.797	0.159	0.199	5.68×10^{-4}
TL-SOC	0.852	0.247	**0.591**	5.52×10^{-4}

The benchmark comparison is reported using F1-score to ensure consistent evaluation across both in-distribution and out-of-distribution scenarios. Figure 5 summarizes the performance of the evaluated architectures under both in-distribution and out-of-distribution scenarios.

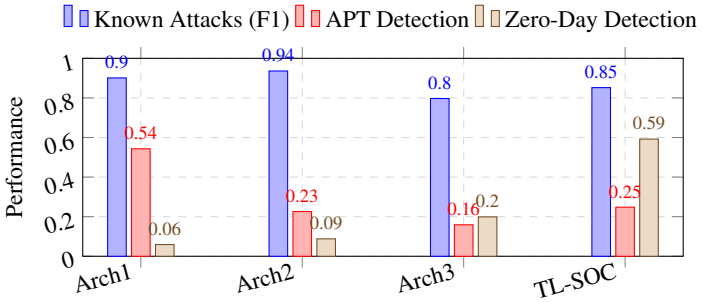

Figure 5: Benchmark comparison of the evaluated architectures.

These results highlight a clear trade-off between maximizing performance on known attacks and ensuring robustness to unseen threats. While Arch2 achieves the highest performance on known attacks (F1 = 0.936), it exhibits a higher false-positive rate (9.82×10^{-3}) and a significant performance drop in zero-day detection (F1 = 0.088). In contrast, TL-SOC maintains competitive performance on known attacks ($F1 = 0.852$) while substantially improving detection capability for zero-day attacks ($F1 = 0.591$) under strict false-positive constraints. This behavior is particularly important in SOC environments, where minimizing false alerts and maintaining reliable detection under evolving attack patterns are critical. By combining anomaly representation learning, structural modeling, and adaptive meta-learning fusion, TL-SOC effectively captures complementary detection signals, leading to improved resilience against previously unseen cyber threats.

Table 8: Comparison with state-of-the-art intrusion detection approaches on CICIDS-2017.

Method	Year	F1-score	ROC-AUC
Deep Autoencoder [2]	2018	0.85	0.97
CNN-based IDS [3]	2020	0.89	0.98
Hybrid DL Model [5]	2021	0.91	0.99
Ensemble IDS [6]	2024	0.93	0.995
Decision Fusion IDS [7]	2024	0.92	0.994
TL-SOC (proposed)	2025	0.852	**0.9978**

These results demonstrate that, although some state-of-the-art approaches achieve higher performance on known attacks, they generally remain accuracy-driven and do not explicitly address the operational constraints of SOC environments. In particular, they lack mechanisms to control false-positive rates and to ensure robustness under distribution shifts. In contrast, TL-SOC adopts a decision-centric architecture that integrates anomaly representation learning and supervised classification within a unified framework. The proposed meta-learning-based fusion layer plays a key role by adaptively combining heterogeneous detection signals, enabling the model to exploit their complementary strengths while maintaining calibrated and reliable decision-making. This design allows TL-SOC to achieve a favorable trade-off between detection performance and operational reliability, characterized by low false-positive rates, improved generalization to unseen threats, and enhanced interpretability. As a result, TL-SOC is particularly well-suited for real-world SOC deployment, where robustness and alert precision are critical.

5.6. *Robustness to Out-of-Distribution Attacks*

To further evaluate the robustness of TL-SOC beyond standard in-distribution conditions, we consider out-of-distribution (OOD) scenarios that simulate realistic SOC environments where new attack patterns emerge after deployment. Two types of OOD threats were considered: *APT-like attacks* and *zero-day attacks*. The evaluation datasets were generated by mixing OOD attack flows with benign traffic to reproduce SOC traffic distributions. Unless otherwise specified, the mixture contains 95% benign traffic and 5% OOD attack flows.

APT-like attacks. The APT scenario includes previously unseen attack patterns generated from multiple sources, including synthetic perturbations and unknown real attack samples. Under the SOC-like distribution (95% benign traffic), TL-SOC maintains strong discrimination capability with ROC-AUC values reaching up to 0.9834 and PR-AUC up to 0.8148. Despite the strict FPR constraint used during threshold calibration, the model is still able to identify anomalous traffic patterns associated with advanced threats.

Zero-day attacks. Zero-day detection was evaluated using attack categories that were completely excluded from the training process. These attack flows were introduced only during testing and mixed with benign traffic to simulate realistic operational conditions. Under the same SOC-like distribution, TL-SOC achieves ROC-AUC values of approximately 0.92 and PR-AUC values up to 0.6. The model reaches recall values between 0.25 and 0.43 depending on the zero-day scenario, while maintaining a very low false-positive rate.

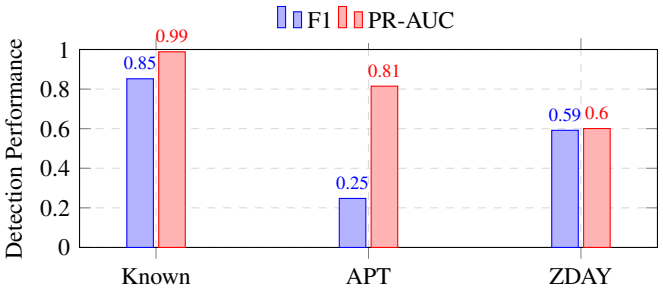

Figure 6: Comparison of TL-SOC detection performance across evaluation regimes.

Overall, these results demonstrate that TL-SOC maintains robust detection performance under distribution shifts, preserving low false-positive rates while effectively identifying previously unseen threats. The combination of anomaly-based modeling and supervised classification in a meta-learning fusion framework makes this behavior possible. This framework makes sure that performance stays stable even when the data distribution changes. In SOC environments, where reliable detection is necessary for operational deployment, this kind of robustness is very important.

5.7. Model Interpretability and Feature Analysis

To improve transparency and better understand the behavior of the proposed TL-SOC framework, an interpretability analysis was conducted using SHAP, which quantifies the contribution of each feature to the final decision score of the model.

We first analyze the importance of the meta-fusion inputs used by the final decision layer. The meta-model takes three signals and combines them: the CTAE's anomaly score (s_{ctae}), the GNN-AE's structural anomaly score (s_{gnn}), and the XGBoost post-filter's classification probability (p_{xgb}). The SHAP analysis shows that the XGBoost probability has the biggest impact on the final decision, followed by the graph-based anomaly score. The CTAE score is a secondary signal.

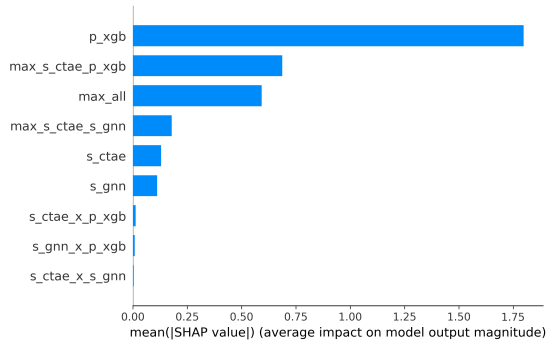

Figure 7: SHAP-based importance of the TL-SOC meta-fusion inputs.

We further analyze the importance of network traffic features in the XGBoost classifier to better understand the decision-making process of the proposed framework. The most influential variables include *Destination Port*, *Init_Win_bytes_backward*, *Subflow Bwd Bytes*, *Subflow Fwd Bytes*, and *Average Packet Size*. These characteristics appear to represent significant statistical and structural aspects of network flows, such as communication endpoints, the volume of two-way traffic, and packet size. Specifically, the *Destination Port* shows the service that is being targeted and can show unusual access patterns. The *Init_Win_bytes_backward* shows how TCP flow control works and can show unusual communication patterns. The same goes for "Subflow Bwd Bytes" and "Subflow Fwd Bytes," which show how much data is sent and received in both directions. This can help you understand traffic patterns that are not normal or are very heavy, which are often linked to bad behavior. The distribution of packet payloads, which can vary greatly between benign and attack traffic, is further described by the *Average Packet Size*. The prominence of these features indicates that the model relies on meaningful and interpretable traffic characteristics rather than spurious correlations. This is particularly important in cybersecurity applications, where reliable detection must be grounded in realistic network behavior. The SHAP beeswarm analysis further reveals consistent and coherent relationships between feature values and their contribution to the model output. For instance, high values of traffic volume-related features, such as forward and backward byte counts, tend to increase the probability of malicious classification, reflecting abnormal communication intensity commonly observed in attacks such as denial-of-service or data exfiltration. Conversely, lower or more regular values are generally associated with benign traffic patterns.

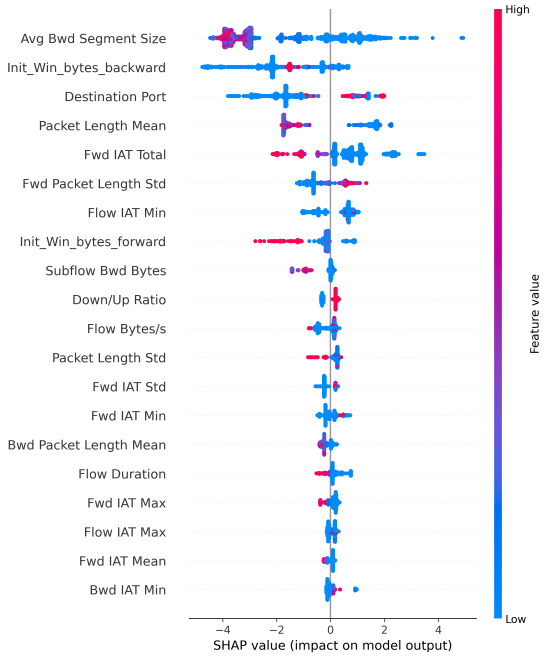

Figure 8: SHAP beeswarm plot illustrating the distribution and directional impact of feature values on the XGBoost classifier.

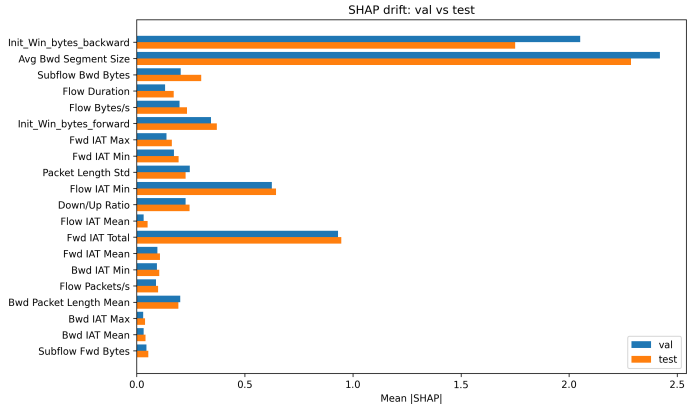

Figure 9: Comparison of mean SHAP values between validation and test sets. The consistency of feature importance across datasets indicates that the model captures generalizable patterns rather than dataset-specific artifacts.

This consistency highlights the stability of the learned decision patterns and supports the interpretability and reliability of the model. It also indicates that TL-SOC captures domain-relevant

features aligned with typical intrusion behaviors, facilitating analyst understanding. The SHAP drift analysis further shows that feature importance remains consistent across validation and test sets, indicating that the model learns generalizable patterns rather than dataset-specific artifacts. This stability is essential for reliable detection under distribution shifts in SOC environments.

5.8. Cross-Dataset Generalization

In addition to the in-dataset evaluation, we conduct a cross-dataset experiment to assess the generalization capability of TL-SOC under strong distribution shift. To further evaluate the generalization capability of the proposed TL-SOC framework, cross-dataset experiments were conducted by training the model on the CICIDS-2017 dataset and testing it on the CSE-CIC-IDS2018 dataset without any retraining. This experimental setup simulates realistic deployment conditions in Security Operations Centers (SOCs), where intrusion detection systems must operate under distribution shifts and encounter previously unseen traffic patterns. Despite the inherent differences between datasets in terms of traffic characteristics, feature distributions, and attack types, the TL-SOC framework maintains stable and reliable performance. The results demonstrate that the proposed decision-centric hybrid architecture effectively generalizes across heterogeneous environments while preserving strict control over false-positive rates. As anticipated, domain shift results in a decline in recall. Nonetheless, the model maintains a low false-positive rate and excellent precision, guaranteeing the production of trustworthy and useful alerts. In SOC situations, where reducing false alarms is essential to prevent overburdening security analysts, this behavior is especially crucial.

Table 9: Cross-dataset evaluation of TL-SOC (Train: CICIDS-2017, Test: CSE-CIC-IDS2018)

Metric	**TL-SOC**
Precision	0.96
Recall	0.42
F1-score	0.58
FPR	7.2×10^{-4}
ROC-AUC	0.93
PR-AUC	0.64

6. Discussion

We've thoroughly tested our TL-SOC system in both perfect and more typical real-world situations and have learned a lot about how it works and how useful it could be for Security Operations Centers.

One important thing we found is that combining different methods for finding intrusions works best. When we took apart the system and tested just the CNN-Transformer Autoencoder (CTAE) and Graph Neural Network Autoencoder (GNN-AE) anomaly detectors, we found they aren't very good at finding attacks if you also need to be sure they won't give many false alarms.This restriction is usual in unsupervised anomaly detection; even though the method normal modeling is exemplary in its ability, the evidence for decision reliability is largely insufficient. Extreme Gradient Boosting (XGBoost), a supervised classifier, is excellent at distinguishing between normal and bad traffic, though is susceptible to changes in data distribution. The proposed meta-learning fusion layer balances all of these attributes by combining the ability to adapt to new patterns and the ability to make supervised classifications. It also indicates that the primary utility of TL-SOC is not in the performance of a single model, but in a high-functioning interplay of many models. One of the most important findings is regarding the false-positive-rate-constrained calibration. In contrast to most intrusion detection systems, which optimize performance metrics that are either accuracy or F1, TL-SOC controls trade-offs along a decisive axis. The results indicate that when the false positive rate is kept very low, the model is adjusted in a significant and reliable way, but at the cost of recall. In this constrained environment, TL-SOC is able to deliver a high level of certainty in its alerts through a commendable balance of detection ability and precision. This is especially pertinent to SOC environments, where analysts can be overwhelmed by floods of false alerts, negatively impacting operational efficiency (our validation showed that valid entries there strongly predicted false alerts). This study provides insight into the range of potential metrics for measuring the performance of an intrusion detection system. Specifically, the time-series and out-of-distribution (OOD) experimental settings highlight performance and generalization trade-offs. As expected, the more challenging the evaluation (i.e., temporal and OOD settings), the more performance declined. The trade-offs were consistent across all models, demonstrating the value of the settings in capturing the complexity of real-world scenarios. Of particular note, TL-SOC is less impacted by the zero-day attacks due to the signalling and detection process being more robust. While hybrid designs support an improvement, the detection of (APT) Advanced Persistent Threat-like low signal attacks still remains challenging, as identified by the low recall. These low signal attacks highlight the difficulty of identifying subtle irregularities in otherwise normal communication. The interpretability analysis provides more insight into the framework, with SHAP results confirming XGBoost to be the dominant signal. This suggests that the outcomes are more dependent on the robust anomaly-based components rather than the standalone signal, and shows that the role of anomaly detection in TL-SOC is primarily to support integrity and resilience to distribution changes. In addition, the uniformity of feature importance for the validation and test sets implies the model stable and generalizable captures generalizable and stable traffic patterns and artifacts. Still, SHAP explanations are also interpretable post hoc and lack causally, which limits impact of explanation for security sensitive use cases. Evaluation itself presents another valuable result. Coupled with the important confirmation that random partitioning strategies can inflate performance estimates due to unrecognized temporal leakage, we observe significantly large differences in the case of stratified versus not stratified splits. On the contrary, chronological evaluation depicts a real world scenario where a model operated under an evolving traffic distribution with shifting traffic patterns across time. This phenomenon indicates the importance of consistent temporal evaluation in intrusion detection. Finally, the cross-dataset evaluation demonstrates that TL-SOC is able to sustain requirements for a better stable precision and satisfactory false positive rate with lower recall on

unseen datasets, which is the most desirable detection operational preference. This abnormality prioritizes the generation of reliable, actionable alerts less aggressively. Even though the results look good, we need to be honest about what this work can't do. We tested our system with data everyone can get to, and that data might not show all the different and complicated things that happen on actual networks. Also, because of its combined approach, this more complex system requires more computing power than simpler ways to find attacks. While this more complicated design makes it perform steadily and dependably, it could be a problem when dealing with a huge amount of network activity. So, to really understand complicated, multiple-step attacks, future work will focus on making the system faster, testing it with massive amounts of data from working networks, and using methods that analyze how things change over time. To summarize, the research findings are positive. Integrated operational boundaries, hybrid modeling, and calibrated decision fusion enable an intrusion detection decision-centric framework to improve the reliability, robustness, and practicality of these systems within modern SOC environments.

7. Conclusion and Future Perspectives

We present TL-SOC, a way of finding intrusions that's built for how Security Operations Centers (SOCs) work, and it's all about making decisions. It combines learning how normal things look (anomaly representation learning), modelling how different pieces of the system relate to each other (graph-based structural modeling), and standard 'labelled' categorization (supervised classification) into a single process for deciding if something is an attack. This is done by smartly combining all of these using a method called meta-learning, and importantly, it's done with a really tight restriction on how many false alarms you'd get. The experiments we ran show TL-SOC is good for actual use in a SOC because it gets a lot of attacks and doesn't have many false positives. It's also more reliable in tricky situations: looking at data over time, when faced with attacks that are unlike anything seen before (like Advanced Persistent Threats and zero-day exploits), and when used on datasets that are different from the ones it was trained on. Plus, we've used a sophisticated method based on SHAP values to explain both which aspects of the data are most important, and how the system is making its decisions. This makes it easier to understand and for analysts to have confidence in it. These results show how important it is to build intrusion detection that is not only accurate, but also strong, able to work in lots of different situations, and will work as expected in a real SOC. TL-SOC uses understandable AI, labelled learning, flexible combination of methods, and a way to identify unusual activity to give a balanced and you can count on it system for modern SOCs. For the future, we're going to test it on a lot more, and much more varied, real-world data. We'll also work on making it run faster for very high volumes of information, and add ways to understand attacks that happen in stages and change over time.

References

[1] T. Bass, "Intrusion detection systems and multisensor data fusion," Communications of the ACM, **43**, 99–105, 2000, doi:10.1145/332051.332079.

[2] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, K. Han, "Enhanced Network Anomaly Detection Based on Deep Neural Networks," IEEE Access, **6**, 48231–48246, 2018, doi:10.1109/ACCESS.2018.2863036.

[3] S. Zavrak, M. İskefiyeli, "Anomaly-Based Intrusion Detection From Network Flow Features Using Variational Autoencoder," IEEE Access, **8**, 108346–108358, 2020, doi:10.1109/ACCESS.2020.3001350.

[4] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, K. J. Kim, "A survey of deep learning-based network anomaly detection," Cluster Computing, **22**, 949–961, 2019, doi:10.1007/s10586-017-1117-8.

[5] C. Liu, Z. Gu, J. Wang, "A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning," IEEE Access, **9**, 75729–75740, 2021, doi:10.1109/ACCESS.2021.3082147.

[6] M. Sajid, K. R. Malik, A. Almogren, T. S. Malik, A. H. Khan, J. Tanveer, A. U. Rehman, "Enhancing intrusion detection: a hybrid machine and deep learning approach," Journal of Cloud Computing, **13**, 123, 2024, doi:10.1186/s13677-024-00685-x.

[7] R. Ahmad, I. Alsmadi, "Data fusion and network intrusion detection systems," Cluster Computing, **27**, 7493–7519, 2024, doi:10.1007/s10586-024-04365-y.

[8] Y. Xue, J. Pan, Y. Geng, Z. Yang, M. Liu, R. Deng, "Real-Time Intrusion Detection Based on Decision Fusion in Industrial Control Systems," IEEE Transactions on Industrial Cyber-Physical Systems, **2**, 143–153, 2024, doi:10.1109/TICPS.2024.3406505.

[9] I. Lotfi, M. Mandar, "Review of Detection and Prevention Techniques for Cyberattacks in SOCs: State of the Art and Future Challenges," in 2025 International Conference on Circuit, Systems and Communication (ICCSC), 1–6, 2025, doi:10.1109/ICCSC66714.2025.11135218.

[10] Z. Chen, C. K. Yeo, B. S. Lee, C. T. Lau, "Autoencoder-based network anomaly detection," in 2018 Wireless Telecommunications Symposium (WTS), 1–5, 2018, doi:10.1109/WTS.2018.8363930.

[11] M. Said Elsayed, N.-A. Le-Khac, S. Dev, A. D. Jurcut, "Network Anomaly Detection Using LSTM Based Autoencoder," in Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, Q2SWinet '20, 37–45, Association for Computing Machinery, New York, NY, USA, 2020, doi:10.1145/3416013.3426457.

[12] H. Rajadurai, U. D. Gandhi, "A stacked ensemble learning model for intrusion detection in wireless network," Neural Computing and Applications, **34**, 15387–15395, 2022, doi:10.1007/s00521-020-04986-5.

[13] R. Lazzarini, H. Tianfield, V. Charissis, "A stacking ensemble of deep learning models for IoT intrusion detection," Knowledge-Based Systems, **279**, 110941, 2023, doi:https://doi.org/10.1016/j.knosys.2023.110941.

[14] E. U. H. Qazi, M. H. Faheem, T. Zia, "HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System," Applied Sciences, **13**(8), 2023, doi:10.3390/app13084921.

[15] A. Ayantayo, A. Kaur, A. Kour, X. Schmoor, F. Shah, I. Vickers, P. Kearney, M. M. Abdelsamea, "Network intrusion detection using feature fusion with deep learning," Journal of Big Data, **10**, 167, 2023, doi:10.1186/s40537-023-00834-0.

[16] C. Xu, J. Shen, X. Du, "A Method of Few-Shot Network Intrusion Detection Based on Meta-Learning Framework," IEEE Transactions on Information Forensics and Security, **15**, 3540–3552, 2020, doi:10.1109/TIFS.2020.2991876.

[17] T. Zoppi, M. Gharib, M. Atif, A. Bondavalli, "Meta-Learning to Improve Unsupervised Intrusion Detection in Cyber-Physical Systems," ACM Trans. Cyber-Phys. Syst., **5**(4), 2021, doi:10.1145/3467470.