ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 4, No. 4, 256-264 (2019)
www.astesj.com
Special Issue on Advancement in Engineering and Computer Science

ASTES Journal
ISSN: 2415-6698

Event Monitoring using Distributed Pattern Recognition Approach on Integrated IoT-Blockchain Network

Anang Hudaya Muhamad Amin*,[1], Nazrul Muhaimin Ahmad[2], Subarmaniam Kannan[2]

[1]*Computer Information Sciences Division, Higher Colleges of Technology, United Arab Emirates*

[2]*Centre of Intelligent Cloud Computing (CICC), Multimedia University, Malaysia*

ARTICLEINFO

Article history:
Received: 12 June, 2019
Accepted: 17 July, 2019
Online: 08 August, 2019

Keywords:
Event Monitoring
Internet-of-Things (IoT)
Blockchain Technology

ABSTRACT

With the advancement in the field of Internet-of-Things (IoT), event monitoring applications have rapidly evolved from a simple event data acquisition towards predictive event analytics involving multi-sensory data aggregation in a distributed environment. Existing event monitoring schemes are mainly relying on inefficient centralized processing mechanism, which may lead to the common single-point of failure for the entire system. In addition, there is no proper method for verifying the event data generated by the monitoring system. In this paper, we present a distributed event monitoring scheme using a Hierarchical Graph Neuron (HGN) distributed pattern recognition algorithm. HGN is a single-cycle learning graph-based recognition scheme that is modelled for in-network deployment. In this work, event data retrieved from multi-sensory IoT devices within a distributed event monitoring network is converted into pattern. To address the event data verification problem, we integrate our proposed scheme with blockchain technology. Combining this IoT event monitoring capabilities with blockchain-based data storage and verification could leads towards a scalable event detection and monitoring model for large-scale network. The results obtained from our simulation shows that the proposed scheme offers high event detection accuracy and capable of minimizing the event storage requirements on blockchain network.

1 Introduction

Since a decade ago, Internet-of-Things (IoT) has been extensively explored as a common platform for event monitoring. In industrial manufacturing applications, IoT plays an important role in monitoring and controlling processes. The monitoring systems designed for such applications require seamless interoperability between heterogeneous distributed IoT sensor devices and processing nodes, as well as an ability to analyze different types of sensor data.

The importance of such systematic interoperability may be conceptualized through an industrial scenario, wherein the plant maintenance is automatically scheduled by coordinating and synchronizing different sub-systems including distributed engineering, control, and maintenance sub-systems. Once a maintenance event is detected, the system can directly generate suitable downstream process controls, computed according to the applications requirements as to generate a correct response within the entire system.

A conventional way of performing event detection and monitoring usually involves a set of interconnected sensor devices that are linked to a centralized processing node (sink). These sensor devices made up the IoT infrastructure, enabling data acquisition process. With this method of implementation, common issues are related to communication and processing latency. Edge computing [1] is a form of computing model that enables data acquisition and processing to be performed within the close proximity of IoT devices.

Event monitoring involving critical applications such as environmental and disaster monitoring usually requires spatio-temporal data analysis for detecting unusual events or incidents. Machine learning and neural network algorithms are commonly being applied for such use cases. Nevertheless, these approaches involve complex computational procedures to be performed on resource-abundant processing nodes. In the context of large-scale and highly distributed IoT network, these approaches may not be suitable due to the resource-constrained specification of most IoT devices. Furthermore, massive data transmission may occur between the

*Anang Hudaya Muhamad Amin, Higher Colleges of Technology, P.O.Box 15825, Dubai, United Arab Emirates, +9712-2064772 & aamin@hct.ac.ae

https://dx.doi.org/10.25046/aj040432

sensor nodes and the processing node, due to large volume of data to be processed continuously.

A number of approaches have been carried out to enable complex machine learning and neural network algorithms to perform in a fully-distributed manner. However, most of the existing schemes suffer from the tightly-coupled processing mechanism in detecting and classifying event/non-event. Distributed algorithms that implement a graph-based approach show a promising way towards achieving purely-distributed processing mechanism.

Graph Neuron (GN) [2] is a distributed pattern recognition algorithm that follows the principles of graph theory with (vertex, edge) representation. GN algorithm has been implemented in several different case studies, including event recognition in wireless sensor networks (WSNs). GN performs a single-cycle in-network processing in its learning mechanism. As such, this improves its scalability against other graph-based pattern recognition schemes.

This paper is an extension of work originally presented in ICIAS 2018 [3]. In this paper, we extend our proposed idea on integrating a distributed pattern recognition algorithm and blockchain technology for event monitoring in IoT network. GN-based algorithm known as Hierarchical Graph Neuron (HGN) [4] is chosen as the distributed pattern recognition algorithm to be incorporated in the proposed IoT-blockchain infrastructure, enabling event detection and monitoring to be performed in a distributed manner, close to the event data acquisition point. Integrating blockchain network and IoT infrastructure allows the event signal or data from IoT devices to be stored and verified by blockchain nodes in the form of transaction using the distributed ledger mechanism. This approach provides a way to collect and preserve significant series of events that were captured and monitored by the IoT devices. Within this infrastructure, distributed pattern recognition could be implemented by identifying patterns of multiple sensor activation on IoT devices, and thus allowing complete transactions of event data to be recorded within the blockchain network.

The outline of this paper is as follows: Section 2 provides an overview on the current approaches towards distributed event monitoring. Section 3 focuses on the Graph Neuron (GN) based approach for event detection in a distributed environment. The model for integrated IoT-Blockchain platform for event monitoring is presented in Section 4. The proposed IoT-blockchain scheme with distributed pattern recognition for event monitoring is presented in Section 5. This section also includes description of the simulation works that have been performed. Finally, Section 6 concludes the paper.

2 Distributed Event Monitoring

A typical setup for distributed event monitoring network is shown in Figure 1. In this model, each IoT device act as an observer, which observe a stream of events. These observations would then be used for analysis involving either a simple function as collecting the number of observations or other complex computations such as identifying anomalies in the input distributions. Such computations usually being carried out by the coordinator node, who is overseeing the entire observations within the network.

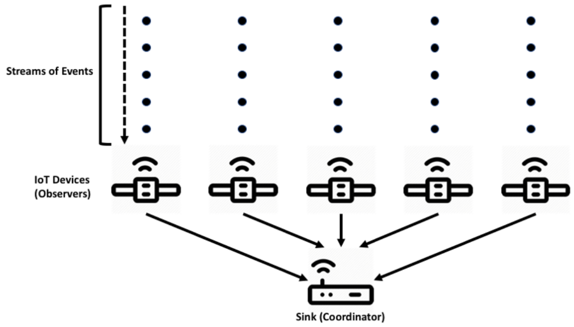


Figure 1: Distributed event monitoring setup with series of IoT devices as observers and processing node as coordinator.

There are a number of related works that demonstrate the distributed event monitoring activities carried out in different kinds of applications. These include the work by Wu et al. [5] and Ahmad Jan et al. [6].

In the survey conducted by Cormode [7], there are two approaches in deriving useful computations for the observations obtained from the sensor/IoT network. The first approach involves all the observers to simply send all the observations to a single and centralized coordinator. A drawback of such approach is such that it burdens the communication network between the coordinator and the observers. A second approach implements a periodic polling, in which at a particular time interval, the coordinator polls each observer for information on the observations and collates this information to get a snapshot of the current status since the last poll. A limitation of this approach is that it does not fit well, when considering event analysis using a non-linear computational model as in the continuous event monitoring applications.

The current centralized event monitoring systems that implement such coordinator-observer model can be seen in the works recorded by Ahmad Jan et al. [6] usually suffer from several limitations. Firstly, centralized event data processing can lead towards single-point of failure whereby failure of centralized node basically fails the entire event monitoring activity. Secondly, magnitude of data transmission to a central repository would be significantly increased as a result of massive number of IoT devices connected. This phenomena creates a connection gridlock from IoT devices to the centralized server.

Apart from the coordinator-observer model, other scheme such as presented by Basirat and Khan [8] utilizes the in-network processing capabilities that enable each IoT device to perform computation on observations within the body of the network, without relying on a centralized data processing node. With the in-network processing model, each IoT node is responsible for collecting the observations, as well as performing data aggregation. Analysis is carried out within the collaborative effort by each node, in exchanging critical information obtained from the observations.

Event monitoring with pattern recognition approach is widely applied in different application fields such as health informatics, manufacturing automation, and structural monitoring. Distributed pattern recognition algorithms such as Graph Neuron (GN) [9] and Vector Symbolic Architecture (VSA) [10] offer better schemes towards improved event monitoring capability that runs on a distributed processing environment.

The proposed Graph Neuron (GN) based approach for event monitoring utilizes the in-network processing model that enables event identification to be conducted within the body of the network, rather than complex analysis and computations being performed at the processing node (coordinator). Discussion on how GN implements such processing model will be discussed in the next section.

3 GN-Based Approach for Event Monitoring

Graph Neuron (GN) is a single-cycle learning distributed pattern recognition algorithm that utilizes the in-network processing model [11]. The main characteristic of any distributed pattern recognition is such that it requires recognition of patterns to be implemented in a decentralized manner. Graph Neuron with similar feature, has been designed for fully-decentralized event detection scheme.

GN algorithm utilizes the graph-like pattern representation with each element of the pattern is represented with a *(value, position)* format. A variation of GN algorithm with hierarchical structure, known as Hierarchical Graph Neuron (HGN) was later developed by Nasution and Khan [4] that eliminates the crosstalk issue in GN implementation. In this paper, we present our proposed approach of utilizing HGN algorithm for distributed event monitoring on multi-sensor IoT devices.

3.1 Hierarchical Graph Neuron (HGN)

Hierarchical Graph Neuron (HGN) algorithm extends the functionality of Graph Neuron (GN) algorithm by increasing the pattern matching accuracy for highly-distorted patterns. HGN has been proven to eliminate the crosstalk issue found in GN implementation. The neuron firing in GN relies on the comparative function between stored and input elements on each neuron, which is called bias entry. These entries composed of *(value, position)* pairs that represent pattern elements. The approach is completely different from conventional neural network schemes whereby firing of neuron is based upon weight adjustment and calibration. Figure 2 shows a typical HGN network composition.

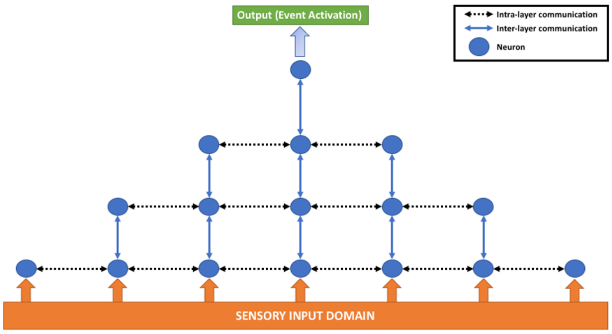

Output (Event Activation)
Intra-layer communication
Inter-layer communication
Neuron
SENSORY INPUT DOMAIN

Figure 2: HGN network composition in a pyramid-like formation. Each neuron at the base layer corresponds to each input element within the sensory domain.

As shown in the figure, neurons composition in HGN network are arranged in a pyramid-like structure with base layer neurons act as input neurons to the network.

3.2 HGN Learning Mechanism

The neurons composition in each HGN network, n_{hgn} is represented by the following equation:

$$n_{hgn} = \left(\frac{p+1}{2}\right)^2 \tag{1}$$

Where p represents the number of input elements within the sensory input domain.

Following the graph-based structure, each neuron in HGN network composition forms a vertex that contains pattern element information (value or identification (ID)) while the adjacent inter-neuron communication is represented by the edge of a graph.

The HGN learning mechanism involves finding matched indices between input and stored patterns. In performing this comparison, each neuron compares its input pattern with the inputs obtained from its adjacent neurons. Adjacency information for each neuron is represented using the *(left, right)* formation. This adjacency information is known as bias entry and each neuron maintains an array of such entries. Each neurons bias array only stores the unique adjacency information derived from the input patterns. The pattern storage process involving N bias array sizes is represented in the following equation:

$$E^B = (\langle x, y \rangle; x \in N, y \in N) \tag{2}$$

Where E^B, comprises the sets of two-element ordered pair respectively, while x and y are the inputs within each entry. E^B can also be formed as one- or three-element ordered pair, depending upon the neuron location within the HGN hierarchical structure.

In retrieving matched bias entry, a linear search method is used on the adjacency information obtained for a given input pattern within each neuron through the bias array composition. The bias array entry is unique. Thus, the following equation shows a function to estimate the required number of comparisons for each input entry C_i, given n_{bEnt} number of bias entries in the array and r number of occurrences for each entry:

$$C_i = \frac{n_{bEnt} + 1}{r + 1} \tag{3}$$

In principle, HGN implements a single-cycle learning for recognizing patterns, through implementation of a graph-matching scheme on its pattern representation. Within HGN network composition, each neuron performs a forward propagation of index values obtained from the matching process as shown in Figure 3, from the base layer neurons towards the top neuron. The top layer neuron will decide on the event/non-event status based on the entire pattern recall search on the bias array composition. The procedures only involved a one-pass cycle for each input pattern, without any iterations involving value alteration as to obtain a recognition output.

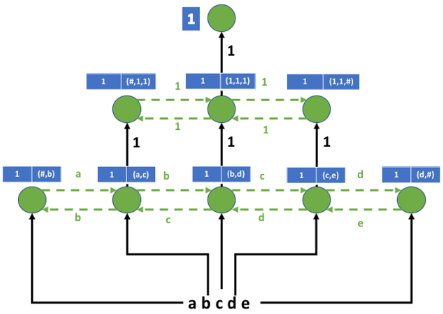


Figure 3: Neuron firing mechanism within the HGN network composition. Neuron activation is achieved through value comparison between adjacent neurons.

3.3 HGN for Event Monitoring

With distributed and lightweight features of HGN, an event detection scheme for IoT network can be carried out at the IoT device level. It could act as a front-end middleware that could be deployed within each IoT nodes in the network, forming a network of event detectors. Hence, the processing scheme minimizes the processing load at the sink and provides near real-time detection capability. Preliminary work on distributed HGN integration for distributed networks such as wireless sensor network (WSN) has been conducted by Muhamad Amin and Khan [12].

In integrating HGN within event monitoring, we have considered mapping each HGN processing cluster into each IoT node, with the assumption that each node is having more than one sensor inputs. Our proposed scheme is composed of a collection of IoT nodes and a sink. We consider a deployment of IoT network in two dimensional plane with M sensors, represented by a set $M = (m_1, m_2, ..., m_n)$, where m_i is the ith sensor. The placement for each of these sensors is uniformly located in a grid-like area, $A = (x \times y)$, where x represents the x-axis coordinate of the grid area and y represents the y-axis coordinate of the grid area. Each IoT node is assigned to a specific area of the grid as shown in Figure 4. The location of each sensor node is represented by the coordinates of its grid area (x_i, y_i).

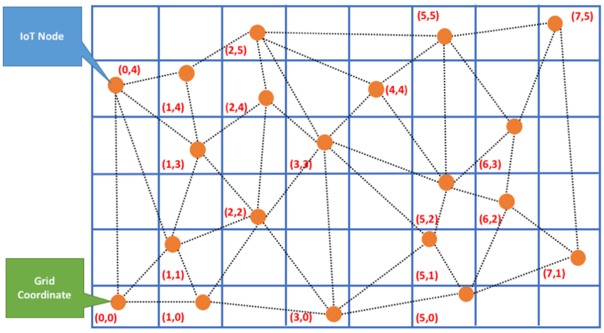


Figure 4: Cartesian grid layout used to identify location of IoT nodes in a two-dimensional space.

For communication model, HGN uses a single-hop mechanism for data transmission from IoT node to the sink. The use of *autosend* approach has been selected following the work by Saha and Bajcsy [13], to minimize error due to the loss of packets during data transmission. HGN event processing scheme involves minimal transmission of event data from IoT nodes to the sink, due to the ability for the front-end processing.

4 Integrating Blockchain in Distributed Event Monitoring

In this section, we present an overview of blockchain and its integration with the distributed event monitoring framework on IoT network.

4.1 Blockchain Technology

Blockchain provides a capability for verification of credentials and transactions using a distributed ledger mechanism. Such capability is of interest, especially when considering its integration with event monitoring system in IoT network.

Implementation of blockchain is not strictly limited to cryptocurrencies. In fact, there are many differences between cryptocurrency implementation and blockchain for industrial applications [14], including immutability and provenance. Blockchain has also been a point of interest in the field of shared economy applications (e.g. Uber, AirBnB) [15]. Nevertheless, past implementations of blockchain come with several limitations, including slow block generation period as mentioned in the Ethereum White Paper [16] and lack of loop implementation (Turing incomplete) [17]. Blockchain enables a decentralized mechanism for verification that allow participating nodes in the network to validate unique transactions through a distributed ledger (block). Figure 5 shows the decentralization approach in blockchain, in comparison with existing centralized approach.

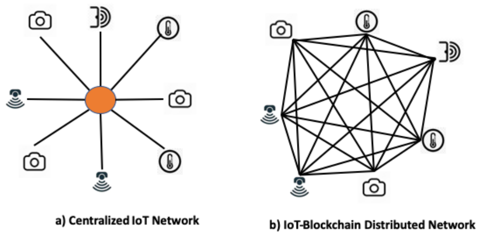

a) Centralized IoT Network
b) IoT-Blockchain Distributed Network

Figure 5: A comparison between conventional IoT network with IoT-Blockchain distributed network.

One of the important requirements for IoT integration on blockchain network is such that it requires fast validation and verification to be performed on blockchain network. Existing blockchain network such as Ethereum [18] has made this possible. Ethereum offers a smart contract deployment, as shown in Figure 6. It captures details of transactions including the value and its corresponding state. Developers can write a program that runs on top of Ethereum. With Ethereum as a blockchain platform, we can configure IoT devices to connect to this main network. This infrastructure would allow authentication of IoT devices through public key infrastructure. IoT devices can utilize Ethereum platform for updating their action and behavior.

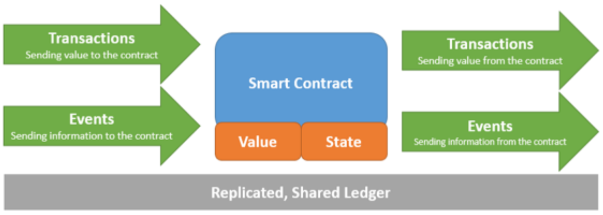

Transactions
Sending value to the contract
Events
Sending information to the contract
Smart Contract
Value
State
Transactions
Sending value from the contract
Events
Sending information from the contract
Replicated, Shared Ledger

Figure 6: Smart contract deployment on Ethereum platform that captures details about transactions and events.

4.2 Device Synchronization

One of the important issues to be addressed in IoT network implementation within the IoT-Blockchain environment is device synchronization. As the number of devices increased, the need for proper synchronization mechanism is important. A work presented by Huh et al [19], demonstrated the synchronization of IoT devices on Ethereum blockchain network can be obtained using smart contract feature, which specify certain action and behavior of IoT devices. The smart contract can be used to place a code to indicate the detection of event by the IoT device.

In this paper, we demonstrate the use of blockchain network for the purpose of event data monitoring and verification. Blockchain offers immutability to the event data being captured from IoT network. Within the proposed model, sensor data are converted into activation patterns as shown in Figure 7.

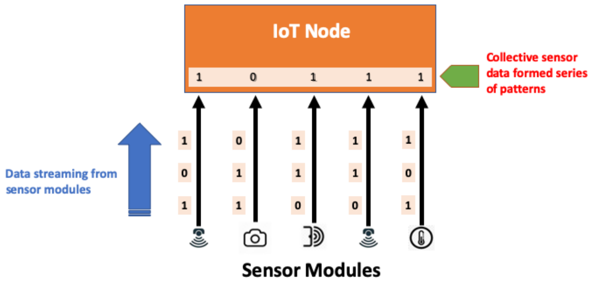

IoT Node
1 0 1 1 1
Collective sensor data formed series of patterns
Data streaming from sensor modules
1 0 1
0 1 1
1 1 0
1 1 0
1 0 1
Sensor Modules

Figure 7: Inputs from sensor modules are treated as a pattern, to be processed by the IoT node.

From the temporal perspective, this sensor data patterns vary across time. Our aim is to capture unique event data to be stored and verified on blockchain network. HGN distributed pattern recognition scheme is used to detect unique event, prior to storage and verification on blockchain network. In this work, similar data patterns would be treated as non-event and will be discarded. The hypothesis for this work is such that the block generation could be minimized accordingly.

5 Proposed Infrastructure

In this section, we present our proposed IoT-blockchain with distributed pattern recognition framework for event detection and monitoring. In addition, this section will also include details on the simulation works and corresponding results involving forest fire detection using HGN distributed pattern recognition scheme.

5.1 System Model

The proposed event monitoring infrastructure incorporates a distributed pattern recognition on IoT- blockchain network. Hierarchical Graph Neuron (HGN) has been selected as the distributed pattern recognition algorithm to be used in our model. In this implementation, assumption is made as such that each IoT node is capable of performing event data collection. Each IoT node is linked to a blockchain-edge (BC-Edge) node that hold a distributed ledger, as part of the blockchain network. Figure 8 shows the proposed infrastructure for HGN deployment in IoT-blockchain network.

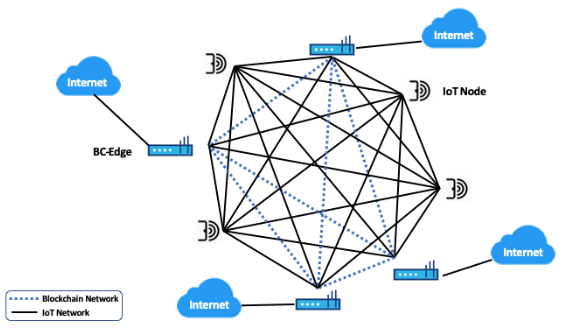


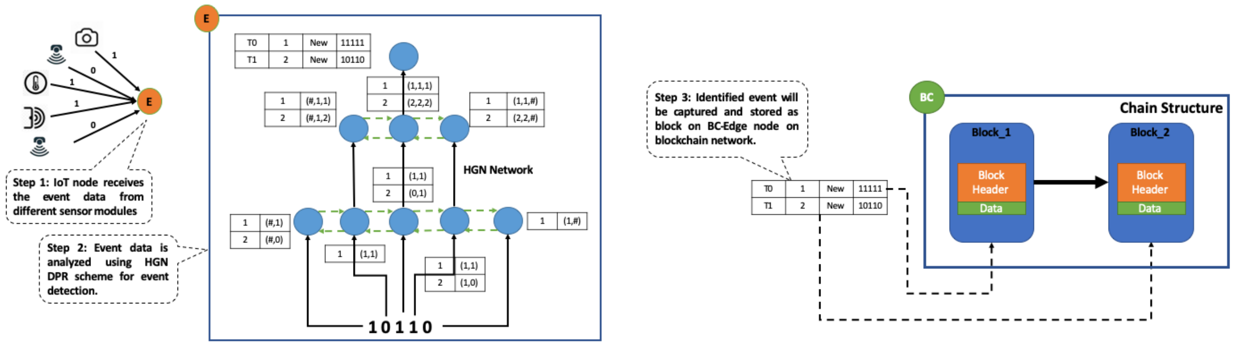
Figure 8: IoT-Blockchain network infrastructure for distributed event monitoring.

Note that the proposed scheme considered the use of multi-sensory IoT devices. Event monitoring is basically carried out at the device level, in which each device is capable of performing event detection using the HGN distributed pattern recognition algorithm. Figure 9 shows the detail event detection and verification procedures on IoT-blockchain network.

The network configuration used in this work is based upon the premise that all the nodes acquired communication capability with low latency in a full-mesh network structure.

In the proposed design, an IoT node N will establish its identity, and broadcast it to the BC-Edge nodes, B. For a given specific time interval T, the IoT nodes will communicate the event data (result from event activation using HGN scheme). This will then be processed by the BC-Edge node as event data for T. The analysis produces an output in the form of combined pattern index, event status, and event pattern as shown in the examples in Table 1. The event status contains information on the event, whether it is classified as new or recalled. Using this approach, only newly-recorded event will be stored and verified in the blockchain network.

As shown in Table 1, at T_2, event detected by IoT node *X02* is recalled and will be discarded. Similarly, at T_3, event


Step 1: IoT node receives the event data from different sensor modules

Step 2: Event data is analyzed using HGN DPR scheme for event detection.

Step 3: Identified event will be captured and stored as block on BC-Edge node on blockchain network.

(a) Event data monitoring and detection.

(b) Unique event storage and verification on blockchain network.

Figure 9: Event monitoring with HGN distributed pattern recognition on integrated IoT-blockchain network.

Table 1: Outputs recorded by BC-Edge node, retrieved from the IoT nodes. Note that only unique event pattern will be stored in the blockchain network

Timestamp	IoT Node ID	Event Pattern	Index	Status
T_1	X01	10101	1	new
T_1	X02	11111	1	new
T_2	X01	10111	2	new
T_2	X02	11111	1	recall
T_3	X01	10111	2	recall
T_3	X02	11110	2	new

recorded by *X01* is recalled. Only unique event pattern will be captured and stored in the distributed ledger.

5.2 Event Monitoring Simulation

To demonstrate our proposed distributed event monitoring scheme, we designed a simulation work on event data analysis using the HGN algorithm. The analysis was conducted on the data related to forest fire, which contains 517 records as reported by Cortez and Morais [20]. Each event pattern detected was stored and verified using the SHA-256 secure hash algorithm on blockchain network.

We extracted five important features that contribute towards possibility of fire ignition, namely the ISI index, temperature, relative humidity (RH), wind and rain. For the purpose of simulation, the dataset is transformed into a binary representation using an interactive binning method. The range of values for each bin was specified according to the potential event (binary value 1) and non-event (binary value 0) as shown in the Table 2.

Table 2: Data values in different range used to identify each parameter as event.

Parameter	Data Range 1	Data Range 2
ISI	$x \geq 42.075$	$x \leq 14.025$
Temperature	$x \geq 23.325$	$x \leq 7.775$
Relative Humidity (RH)	$x \geq 63.75$	$x \leq 21.25$
Wind	$x \geq 6.75$	$x \leq 2.25$
Rain	$x \geq 4.8$	$x \leq 1.6$

Our assumption is such that the event data is continuous and recorded in a timely manner from T_0 till T_{516}, respectively.

5.3 Results and Discussion

In this section, we present the results of our simulation on event pattern recognition using the Hierarchical Graph Neuron (HGN) approach. The proposed scheme capable of detecting 19 unique events from the total of 517 event records. Consequently, the number of data blocks generated is reduced by 97%. This was made possible by identifying unique event patterns from the dataset. Figure 10 shows the event count distribution based on the 19 unique events detected using the proposed HGN distributed pattern recognition scheme.

Table 3 shows samples of event data for each corresponding unique event detected using our proposed HGN recognition scheme.

Apart from implementing the event data classification using HGN approach, we extended our analysis to examine the accuracy of our proposed recognition scheme on the simulated dataset. Based on the observation made, there were 40 event data that were misclassified as false positive (duplication of indices). As a result, the recognition accuracy for HGN in this simulation is at the rate of 92.3% with error rate of 7.7%. Further details on the misclassification can be seen in [21].

The simulation work that has been carried out demonstrate a high recall accuracy for HGN distributed pattern recognition implementation. The error rate for the event classification obtained is considerably low, as it only indicates value around 7.7%.In addition, the results prove that the number of unique events detected is significantly reduced. This is critical for ensuring efficient data block generation

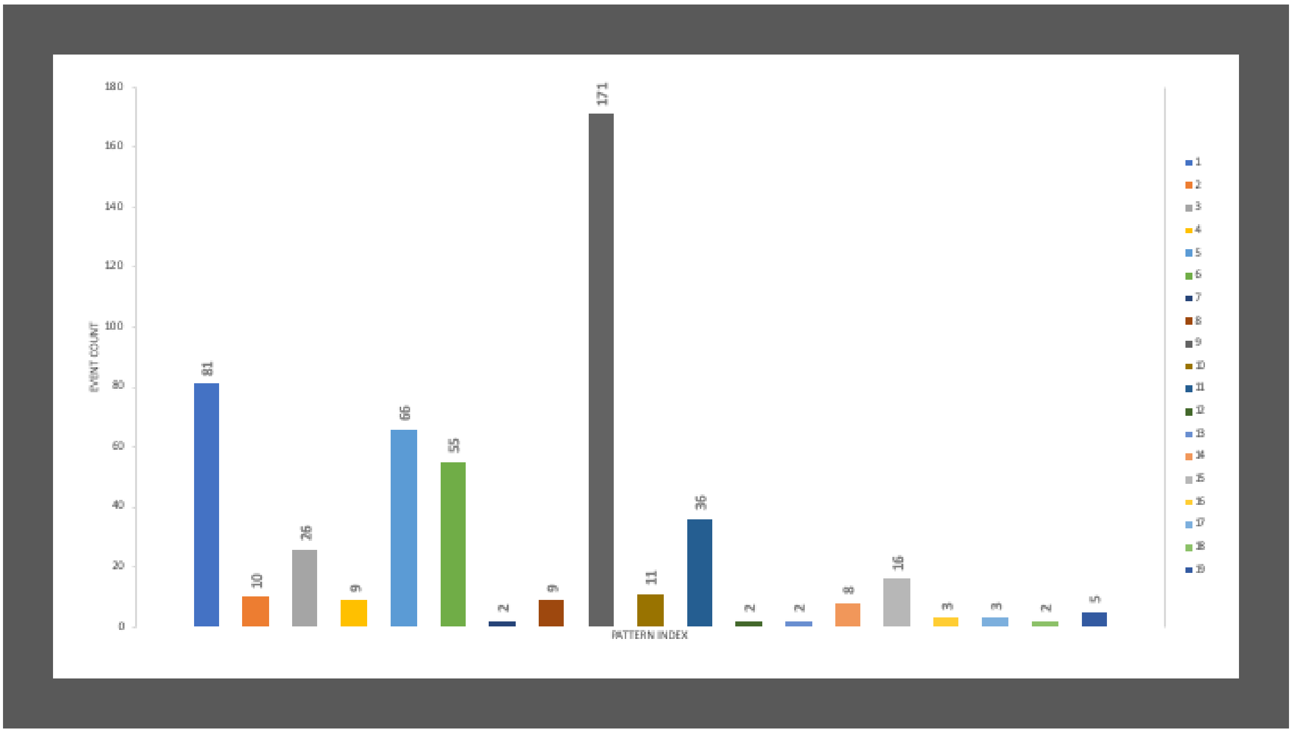


Figure 10: Unique event patterns detected using HGN distributed pattern recognition algorithm from the total of 517 event data recorded.

Table 3: Samples of forest fire event data that has been classified as unique event using HGN distributed pattern recognition algorithm.

Event ID	ISI Index	Temperature (°C)	Rel. Humidity (RH) (%)	Wind (km per hour)	Rain (mm per m^2)
1	5.1	8.2	51	6.7	0
2	7	21.3	42	2.2	0
3	5.8	23.4	22	2.7	0
4	6.2	12.9	74	4.9	0
5	17	20.1	40	4	0
6	7	21.6	33	2.2	0
7	8.9	18.4	42	6.7	0
8	15.9	25.9	24	4	0
9	10.7	10.3	74	2.2	0
10	7.8	17.4	24	5.4	0
11	3.8	15.2	51	2.7	0
12	8.1	29.6	27	2.7	0
13	6.7	18.4	25	3.1	0
14	14.3	19.1	53	2.7	0
15	14.1	33.1	25	4	0
16	10.8	26.4	35	2.7	0
17	14	30.8	30	4.9	0
18	17.7	26.4	34	3.6	0
19	14.3	27.3	63	4.9	6.4

within the blockchain network.

5.4 Future Work

Based on the simulation work that has been carried out, it was clearly indicated that the HGN distributed pattern recognition scheme is able to provide accurate classification of unique events. Our future work is basically to fully integrate the event classification module with blockchain network for event data storage and verification. The use of open-source hyperledger fabric platform [22] under the Linux Foundation Projects will be considered.

We intend to expand our research into effective scheme for data storage on blockchain. As the amount of data arises from events monitored on IoT network, the need for efficient data storage mechanism is important. At a preliminary stage, we propose a two-type storage mechanism, comprising event digest and event data. Event digest will be permanently stored in the blockchain network in the form of recorded blocks (on-chain), while the event data will be stored in conventional databases (off-chain). In implementing this scheme, it is expected that the storage requirement for event data could be further reduced.

In addition to the storage optimization, we would also aim to study on different consensus mechanisms for data verification on blockchain network. These include the proof-of-work (PoW) and proof-of-stake (PoS). The consensus mechanism enables peers and users within the network to verify the authenticity of the data that is stored on the blockchain network. Results of the study would assist us in implementing more efficient IoT-blockchain infrastructure for event monitoring on ioT network.

6 Conclusions

In this paper, we presented our on-going work on distributed event monitoring using a distributed pattern recognition algorithm on integrated IoT-Blockchain network. The proposed scheme implements a HGN recognition algorithm that performs a distributed event detection. Accuracy of HGN pattern recognition may improves the efficiency of distributed ledger storage mechanism in blockchain network. This enables unique event data to be stored and verified in a distributed environment. The simulation work being carried out shows that HGN algorithm exerts high recall accuracy with one-cycle learning. Furthermore, the algorithm is highly-suitable for in-network deployment.

HGN recognition approach also minimizes the needs for storing large amount of continuous event data on blockchain network. The detection mechanism in the proposed scheme also consider the entire state of the IoT network, rather than examining individual sensor input values.

For our future development, we intend to look into several aspects of IoT-blockchain integration, including the efficiency and effectiveness of the proposed scheme, through the scalability factor of the system in implementing multi-heterogeneous IoT-blockchain network. Apart from this, different consensus mechanisms in blockchain technology will be reviewed to analyze their effectiveness in the proposed IoT-blockchain scheme.

References

[1] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[2] Mohamed Baqer, Asad I Khan, and Zubair A Baig. Implementing a graph neuron array for pattern recognition within unstructured wireless sensor networks. In *International Conference on Embedded and Ubiquitous Computing*, pages 208–217. Springer, 2005.

[3] Anang Hudaya, Muhamad Amin, Nazrul Muhaimin Ahmad, and Subarmaniam Kannan. Integrating distributed pattern recognition technique for event monitoring within the iot-blockchain network. In *2018 International Conference on Intelligent and Advanced System (ICIAS)*, pages 1–6. IEEE, 2018.

[4] Benny B Nasution and Asad I Khan. A hierarchical graph neuron scheme for real-time pattern recognition. *IEEE Transactions on Neural Networks*, 19(2):212–229, 2008.

[5] Kangheng Wu, Xiaokang Xiong, Bert W. Leung, Jihyoun Park, and Zhibin Lei. Event processing of monitoring data of large hi-tech manufacturing equipment: Debs grand challenge. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 387–392, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1315-5. doi: 10.1145/2335484.2335535. URL http://doi.acm.org/10.1145/2335484.2335535.

[6] Mian Ahmad Jan, Priyadarsi Nanda, Xiangjian He, and Ren Ping Liu. A sybil attack detection scheme for a forest wildfire monitoring application. *Future Generation Computer Systems*, 80:613 – 626, 2018. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2016.05.034. URL http://www.sciencedirect.com/science/article/pii/S0167739X16301522.

[7] Graham Cormode. Continuous distributed monitoring: a short survey. In *Proceedings of the first international workshop on algorithms and models for distributed event processing*, pages 1–10. ACM, 2011.

[8] Amir H Basirat and Asad I Khan. Graph neuron and hierarchical graph neuron, novel approaches toward real time pattern recognition in wireless sensor networks. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pages 404–409. ACM, 2009.

[9] Asad I Khan and Patrik Mihailescu. Parallel pattern recognition computations within a wireless sensor network. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 1, pages 777–780. IEEE, 2004.

[10] Simon D Levy and Ross Gayler. Vector symbolic architectures: A new building material for artificial general intelligence. In *Proceedings of the 2008 Conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference*, pages 414–418. IOS Press, 2008.

[11] Ram Kumar, Vlasios Tsiatsis, and Mani B Srivastava. Computation hierarchy for in-network processing. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 68–77. ACM, 2003.

[12] A. H. M. Amin and A. I. Khan. Parallel pattern recognition using a single-cycle learning approach within wireless sensor networks. In *2008 Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies*, pages 305–308, Dec 2008. doi: 10.1109/PDCAT.2008.47.

[13] Sunayana Saha, Peter Bajcsy, et al. System design issues in single-hop wireless sensor networks. *Proc. of 2nd IASTED ICCIIT 2003*, 2003.

[14] Gideon Greenspan. Four genuine blockchain use cases. *MultiChain [blog]*, 10, 2016.

[15] Steve Huckle, Rituparna Bhattacharya, Martin White, and Natalia Beloff. Internet of things, blockchain and shared economy applications. *Procedia computer science*, 98:461–466, 2016.

[16] Vitalik Buterin et al. Ethereum white paper. *GitHub repository*, pages 22–23, 2013.

[17] Vitalik Buterin. Toward a 12-second block time. *Ethereum Blog*, 2014.

[18] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

[19] Seyoung Huh, Sangrae Cho, and Soohyung Kim. Managing iot devices using blockchain platform. In *2017 19th international conference on advanced communication technology (ICACT)*, pages 464–467. IEEE, 2017.

[20] Paulo Cortez and Aníbal de Jesus Raimundo Morais. A data mining approach to predict forest fires using meteorological data. 2007.

[21] Anang Hudaya Muhamad Amin, Sujni Paul, Fred N Kiwanuka, Imtiaz Ahmad Akhtar, et al. Improving event monitoring in iot network using an integrated blockchain-distributed pattern recognition scheme. In *International Congress on Blockchain and Applications*, pages 134–144. Springer, 2019.

[22] Hyperledger fabric - hyperledger. `https://www.hyperledger.org/projects/fabric`. Accessed: 2019-07-12.