User behavior analysis for insider attack detection using a combination of memory prediction model and recursive feature elimination algorithm

Yaya Sudarya Triana[1], Mohd Azam Osman[2], Deris Stiawan[3], Rahmat Budiarto[4]
[1]Department of Information System, Faculty of Computer Science, Universitas Mercu Buana, Jakarta, Indonesia
[2]School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia
[3]Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya, Indralaya, Indonesia
[4]Department of Computer Science, College of Computing and Information, Al-Baha University, Albaha, Saudi Arabia

Article Info

Article history:

Received Dec 31, 2023
Revised Mar 20, 2024
Accepted Dec 2, 2024

Keywords:

Entity behavior analysis
Insider attack
Memory prediction model
Real time detection
Recursive feature elimination

ABSTRACT

Existing defense tools against the insider attacks are rare, not in real time fashion and suffer from low detection accuracy as the attacks become more sophisticated. Thus, a detection tool with online learning ability and better accuracy is required urgently. This study proposes an insider attack detection model by leveraging entity behavior analysis technique based on a memory prediction model combined with the recursive feature elimination (RFE) feature selection algorithm. The memory-prediction model provides ability to perform online learning, while the RFE algorithm is deployed to reduce data dimensionality. Dataset for the experiment was created from a real network with 150 active users, and mixed with attacks data from publicly available dataset. The dataset is simulated on a testbed network environment consisting of a server configured to run 4 virtual servers and other two computers as traffic generator and detection tool. The experimental results show 94.01% of detection accuracy, 95.64% of precision, 99.28% of sensitivity, and 96.08% of F1-score. The proposed model is able to perform on-the-fly learning to address evolving nature of the attacks. Combining memory prediction models with the RFE for user behavior analysis is a promising approach, and achieving high accuracy is definitely a positive outcome.

Corresponding Author:

Yaya Sudarya Triana
Department of Information System, Faculty of Computer Science, Universitas Mercu Buana
Jl. Meruya Selatan No. 1 Kembangan, Jakarta Barat 11650, Indonesia
Email: yaya.sudarya@mercubuana.ac.id

1. INTRODUCTION

Network security is getting smarter all the time when it comes to finding threats in real-time. The trick is finding those threats quickly without slowing down the whole system. We need systems that keep learning as they go, adjusting even the tiniest changes. The biggest challenge is making sure the system only detects real threats, and does not get confused by normal activity. Since how people use networks changes, security systems need to change with them to fight off new attacks. Adapting on the fly and re-using knowledge across different areas is the key to staying secure.

Memory prediction is gaining steam, but it is still early days compared to network security. As a result, the “best way” to do it is still being figured out. In contrast, user behavior analysis (UBA), which also considers devices, applications, and even network traffic, is already a vital tool in modern cybersecurity. UBA uses different Supervised and Unsupervised Learning algorithms to understand normal user behavior and spot any red flags. It also analyzes how users interact with systems, like their typing patterns, mouse movements, and logins, to catch potential unauthorized access. Despite real-time attack/anomaly detection in network security has made big strides, there are still some important challenges researchers are working on. By tackling these challenges, researchers can create even more effective and reliable real-time attack/anomaly detection systems that make networks more secure against the ever-changing world of cyber threats. These challenges include: i) Balancing how accurate detections are with how efficiently the system runs; ii) Making sure the system can adapt to new threats and changes in how people use the network; and iii) Understanding how the system makes its decisions, especially when using artificial intelligence (AI) based models.

Fending off attacks from outside the institutional/organizational network is relatively easier to do than fending off cybercrime attacks from within the network. External attacks can be prevented by using firewalls, anti-virus and special software for intruder/malware detection. However, defense tools against the insider attacks are rare and suffer from low detection accuracy as the attacks become more sophisticated. Many big cyber-attacks involving insiders have happened, including millions of Yahoo email accounts, illegal downloading of digital movie files from Sony Film Company and ransomware in hospitals in the UK.

There are many attack detection/prevention systems available in the market, apart from being expensive; these systems still have several weaknesses, such as: low detection accuracy, too many false alarms, and the inability to carry out real-time learning for new variants of attacks. The ability of the system to carry out real-time learning is needed to deal with rapidly evolving attacks/viruses/malware. Detecting an attack involving the insiders is more difficult because the defense system used may think that the attack is the normal activity of an entity within the system/network. Besides, the attacks may be able to learn to act as legitimate users.

Referring to the background above, this study attempts to build an intelligent system for detecting insider attacks using entity behavior analysis. Instead of following the traditional way of detecting attacks/anomalies using rule-based or knowledge-based systems, this study prefers to use entity behavior analysis by utilizing human memory modeling to predict entity behavior based on the entity's traffic data [1]. In this case, the system is firstly trained to build a profile of entities in the network, and then examines the normality of that behavior. In addition, to achieve fast detection, the proposed detection system considers the recursive feature elimination (RFE) for reducing the unrelated features of the traffic that degrading the detection accuracy level. Experiments are carried out in a testbed environment with several entities and the system will predict whether a particular entity is carrying out illegal activities, which then ends with making a decision to determine whether the entity is normal or an attack/anomaly.

Zhang *et al.* [2] employed optimization theory to examine which users connected with the strongest short- and long-term effects for their respective target users based on the mobile social environment of users. The goal of integrating these user behavior samples into a target user sample database is to create a sampling process that will greatly increase the accuracy of user behavior predictions. Next, two optimization models are developed based on the degree of interaction and similarity, respectively, to choose the best associated users for examining the two primary components of target user behavior; Furthermore, an adaptive updating strategy based on fuzzy theory is proposed to describe the importance of two factors in real time and quantitative manner. Next, Apriori theory is introduced to predict the user's next service behavior accurately; in particular, the Apriori sample database update mechanism was built to effectively integrate the optimal sample of correlated users. Finally, extensive simulation results show that the proposed algorithm outperforms several related algorithms in terms of accuracy, predictability and operating efficiency. More researches related to the analysis of entity behavior in the field of cybersecurity can be seen in [3]–[6].

Meanwhile, Sharipuddin *et al.* [7] built an intrusion detection system (IDS) and succeeded in improving detection accuracy and precision by using recursive feature elimination (RFE) algorithm as feature extraction. Experiments on the feature extraction dataset from an internet of things (IoT) testbed network have been carried out to investigate the effect of the extraction process on attack detection and the results show perfect of 100% detection accuracy. More research works on attack/anomaly detection using RFE algorithm are presented in [8]–[11] A new framework for expressing the function of the human brain's neocortex was proposed by Hawkins and Ahmad [12]; Hawkins *et al.* [13]; Eichenbaum [1]; Triana *et al.* [14]; and Liu and Lam [15]. Furthermore, other research has revealed that grid cells—which resemble neurons—may potentially exist in the neocortex [12], [16]. Applications for memory prediction frameworks can be found in a variety of domains, including real-time networking [17], [18], online education [19], [20], object identification [21], [22], and medicine [23]. Mohamed [24] carried out an extremely thorough review of the literature on AI methods for attack and anomaly detection. In order to detect intrusion in heterogeneous networks, Sharipuddin *et al.* [25] suggest combining a deep learning strategy with an RFE-based feature extraction method. The experimental results on created dataset from a testbed network show that the accuracy of the proposed method reaches accuracy level above 99%. Apruzzese *et al.* [26] analyze machine learning techniques applied to the detection of intrusion, malware, and spam. The authors reveal the detection accuracy of 98.88%.

Optimal ensemble IDS generation is presented by Stiawan *et al.* [27]. information gain, gain ratio, symmetrical uncertainty, Relief-F, One-R, and Chi-Square are the six feature selection techniques that are employed. The following classification techniques were applied: naïve Bayesian, Bayesian network, decision tree: J48, and self-organizing map (SOM). Subsequently, the optimal features from each feature selection methodology are coupled with each classifier method to create ensemble IDSs. On the ITD-UTM dataset, experimental results demonstrate five optimal ensemble IDSs: symmetrical uncertainty+Bayesian network; chi-square+Bayesian network, Chi Square+SOM, information gain and naïve Bayes; and One-R+Bayesian network, with respective ten, four and seven best selected features achieve 81.0316%, 85.2593%, and 80.8625% of accuracy, respectively. Furthermore, the best F-measure values, *i.e.*, 0.853 and 0.830, respectively, are achieved by ensemble IDSs utilizing symmetrical uncertainty+Bayesian network and one-R+J48 with the ten and six best selected features, respectively. Additionally, the long short term memory (LSTM) model, which is furnished with the RFE feature selection method, was employed by Stiawan *et al.* [28]. Meanwhile, Kurniabudi *et al*. [29] combined PSO-search and random forest to improve attack detection performance on the internet of things. Concurrently, a great deal of research has been done on the subject of online/real-time attack/anomaly detection. Some of these studies are those by Rivera *et al.* [30], Kandhro *et al.* [31], Mohammed and Bouchachia [32], and Tyagi and Kumar [33].

Budiarto *et al.* [34] proposed a memory model by applying a memory prediction framework, called "simplified single cell assembled sequential hierarchical memory (s.SCASHM)". Then this model is used as a tool to predict entity behavior and detect attacks involving insider attacks/anomalies. The experimental results show that the proposed memory model successfully predicts the traffic behavior of entities with varying degrees of accuracy from 72% to 83% and is capable of learning on-the-fly, when new patterns of attacks come. The research in this paper adopts the model proposed by Budiarto *et al.* [34] and combined it with the RFE method as feature selection. Thus, the proposed model is called memory prediction model with recursive feature elimination (MPM-RFE). This research work contributes towards the development of intelligent real-time anomaly/insiders' attack detection.

The rest of the paper is structured as follows: Section 2 discusses the method used in this research. Section 3 presents the research results and discussion. In closing, section 4 concludes the entire research.

2. METHOD

The research method is divided into three stages. Stage 1 is the dataset creation for experiments. Stage 2 is the feature extraction using FRE algorithm. Stage 3 is the development of an engine to predict the user behavior traffic based on a memory prediction model. Figure 1 shows the workflow of the proposed model and is explained in detail in the following sections.

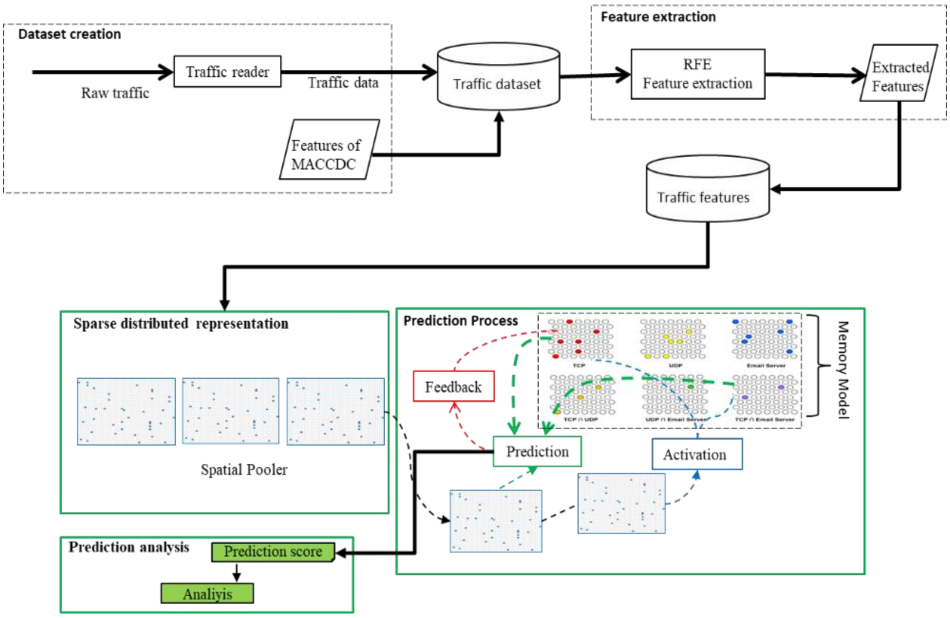


Figure 1. Overall architecture of the proposed model (adopted from [34])

Hierarchical temporal memory provides a framework that models several computational principles that occur in the neocortex (a part of the human brain). The spatial pooler models how neurons learn feedforward and form representations of input data efficiently (fast and accurately). This spatial pooler converts binary input patterns (0, 1) into sparse distributed representations (SDRs) using a combination of Hebbian competitive learning rules. This SDR data is used to model the memory for certain types of traffic (email, HTTP, and applications). For the prediction process: input data in SDR form is used to predict, activate memory and also provide feedback for changes to the memory model if there are significant changes in existing patterns in the memory model. After that, the prediction score is calculated, analyzed and a pattern is determined from the input data.

2.1. Dataset creation

Data for the experiment was taken from the network in the building of the Faculty of Computing and Information, Albaha University, Saudi Arabia, for 2 weeks (1 – 14 April 2023). The rational of selecting this network as the source of dataset, is the network in the building is a flat network connected through switches that makes the traffic capturing process become easy. Thus, the captured traffic reflects a real-world network where users access internet applications such as: web browsing, chatting, email, ftp, streaming video, and games. It was recorded that there were more than 150 users accessing the network during this period. In order for the client application to be able to collect all traffic, we used a network tap to sniff the network traffic before it being sent from and to the particular segment. In this experiment, the machine where the client application runs are configured to collect network traffic in promiscuous mode and stored into a dataset in $.pcap$ format. The attack traffic is taken from the Mid-Atlantic Collegiate Cyber Defense Competition (MACCDC) dataset [35]. Therefore, the created dataset consists of normal and attacks traffics. Because it is not possible to inject attack/anomaly traffic into the production network, the data that has been obtained is simulated again in a testbed network environment consisting of 3 computers connected to a switch. Figure 2 illustrates the testbed network for the experiments. A server is configured to run 4 virtual servers. PC-2 computer is used to inject traffic packets of the captured dataset into the network. The detection module is installed on PC-1, which is connected to the mirrored port on the switch so that the detector is able to see all traffic within the network segment being monitored. The specifications of PC-1 and PC-2 are as follows: Intel Core i7 CPU, 8 GB RAM and 500 GB hard drive. For server, a computer with Intel Core i7 CPU, 16 GB RAM and 1 TB hard disk is used. All computers run the Windows 10 operating system. The modules for the memory prediction model is implemented in the Java programming language, while the prediction module using deep learning, *i.e.* the LSTM is implemented in the Python programming language and the Scikit-learn library.

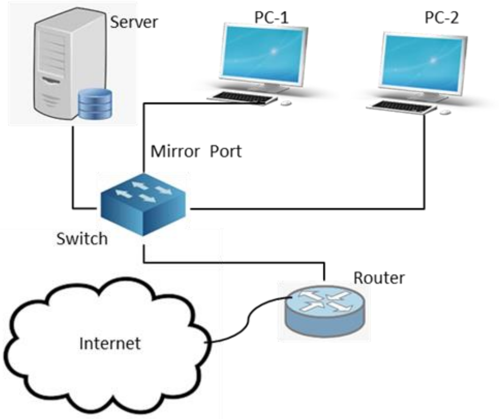


Figure 2. Testbed network topology

2.2. Features extraction using RFE

The next step is extracting significant features of the traffic. The aim is to reduce the dimensionality of the dataset, which will reduce the detection time while preserve a good accuracy level. This study uses the RFE algorithm as a feature extraction/selection method. The RFE transforms the original data linearly into a new coordinate system by maximizing the variance value [36]. It examines the dataset containing observations that have inter-correlated quantitative dependent variables. The fields of traffic packet header such as: source, and destination IP addresses, source and destination ports, protocols, time stamp and attack. are considered as the input for the RFE. The RFE then extracts the significance information from the datasets with the aim is to signify the dataset as a set of new orthogonal variables, *i.e.* principal components and represents the pattern similarity of the observations graphically. The nine extracted features with the highest produced by the RFE implementation are: *ip.ttl, ip.hdr_len, ip.len, tcp.flags, window.size, tcp.hdr_len, wpan.src_pan, wpan.dst16,* and *wpan.c_md*. The RFE algorithm for performing the traffic feature extraction is shown in algorithm 1 [37].

Algorithm 1. The recursive feature elimination

```
Input: Traffic Data
Output: FEATURE (a set of selected attributes)
Import modul decomposition from sklearn
data ← load_dataset
def main()
      Y ← read(data)
      RFE = decomposition.RFE(n_components=8), RFE.fit(Y))
      Y=RFE.transform(Y)
     FEATURE←Y
End
```

2.3. Prediction process

This paper adopts the method introduced by Budiarto *et al.* [34] for predicting attack/anomaly traffic. The method uses a human neocortex inspired hybrid gene-controlled machine intelligence approach. The method aims to use the human neocortex memory as a model to generate human intelligence for recognition and to apply a neurogenetics method for complex identification and prediction.

2.3.1. Sparse distributed representation (SDR)

Memories that share some common features tend to cluster together in human memory, even if they have no connection. The SDR is a way of expressing human memory with mathematics, and it uses a space with many dimensions to capture the huge amount of memory that resembles the human brain’s network of neurons. A key feature of such spaces with many dimensions is that two vectors picked at random are very far apart, which means they have no relation. The SDR saves a lot of data in a small space, using some special places called hard locations. These places are spread out evenly in a bigger space that is not real. Each piece of data is saved by using some of these places, and taken back by combining them. However, this may not work very well, and the quality of the data may change depending on how full the space is.

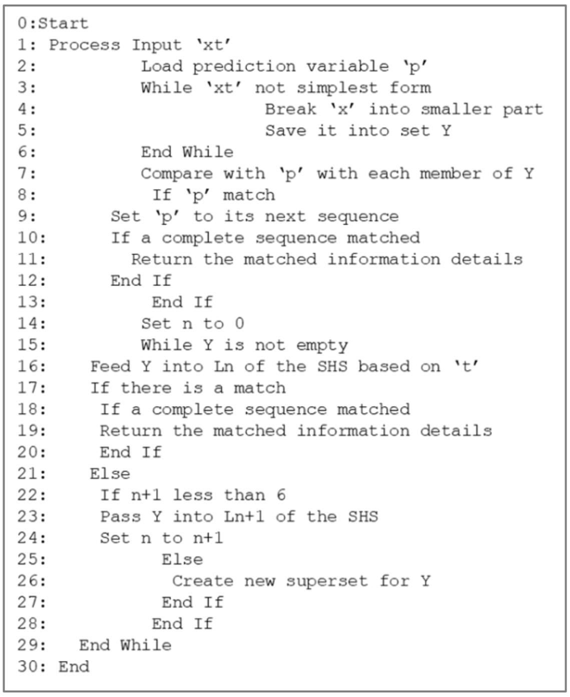
The traffic data obtained from the features extraction phase is converted into a series of individual network packets, by representing each byte of the traffic data in an atomic form as SDR. This basic form is in the form of a vector consisting of a sequence of 2048 bits. For example, consider the value of *hdr_len* feature is 800 bytes. This value is represented as 7 atomic sequences of SDR. These bit vectors are inputted into the memory prediction model module for analysis. In the prediction process, this value is compared with the thresholds inside the memory model and will be decided whether it is a new feature or not. If so, then the memory model is updated.

2.3.2. Memory prediction model

In human neocortex memory model, among the brain’s neural circuits, the cerebellum’s cortex is the most similar to the sparse distributed memory. An associative memory keeps a world model that connects sensory input to action. The memory receives the world’s events as a series of large patterns. These patterns represent sensor data, internal-state variables, and commands to the actuators. The memory’s capacity to store and retrieve these series under similar situations enables its use for prediction [38]. Thus, a human neocortex inspired hybrid gene-controlled machine intelligence approach is considered. The hybrid approach aims to model human neocortex memory to produce human intelligence for recognition and to implement a neurogenetics approach for complex identification and prediction. The cortex of the cerebellum is the neuronal circuit in the brain that most closely mimics sparse distributed memory. An associative memory keeps track of a world model that connects perception to behavior. The world's events are presented to the memory as a series of expansive patterns. These patterns represent commands to the actuators, internal state variables, and sensor data. The memory can be used for prediction since it has the capacity to store these sequences and retrieve them in situations similar to the past.

Inspired by the biological concept of cell assembly, a single neuronal cell model is introduced to form artificial cell assembly that stores the data. Compared to neurons, which are typically thought of in terms of artificial cell assembly models or neural networks, the nature of a single neuronal cell model is essentially different. If the value that the neural network stores is altered, the synapse—which does not keep the data—must be strengthened or trained again [38]. Within the memory model, the single neural cell model sequentially assembles cells without the need for complicated training and learning computations. The purpose of the connection between cells is to determine which cell, in a specific sequenced constructed cell, may activate next when another cell does. The cell is a data container by itself. Because no weights need to be retrained, this greatly reduces the amount of training time needed when the data needs to be updated.

In addition, a platform called sequential hierarchical superset is introduced for materializing the replication of the actual human neocortex memory. The idea is to mimic the 6 layer hierarchical structure of human neocortex by having a hierarchical superset implementation [38]. Inside the platform, an assembled single neuronal cell model is placed as a set and a sequence of these sets form a superset in a hierarchy, starting from the lowest layer 6, which contains set of cells with a specific and detailed data, until the highest layer 1 which contains cells with data that can be considered as object in its abstracted form. The object in abstracted form is to realize one of the key characteristics of human neocortex as explained in the memory-prediction framework, which is the neocortex stores data that is invariant representation [38]. The output of the memory model module is then used for the error prediction process and the probability that the event is included in the attack/anomaly category. The pseudocode of the memory model module is shown in Figure 3.



```
Start
 Process Input 'xt'
          Load prediction variable 'p'
          While 'xt' not simplest form
                      Break 'x' into smaller part
                      Save it into set Y
          End While
          Compare with 'p' with each member of Y
           If 'p' match
       Set 'p' to its next sequence
       If a complete sequence matched
         Return the matched information details
       End If
           End If
          Set n to 0
          While Y is not empty
     Feed Y into Ln of the SHS based on 't'
     If there is a match
      If a complete sequence matched
      Return the matched information details
      End If
     Else
      If n+1 less than 6
      Pass Y into Ln+1 of the SHS
      Set n to n+1
            Else
             Create new superset for Y
            End If
           End If
    End While
  End
```

Figure 3. Pseudocode of memory model construction

2.4. Experiment scenario

After the testbed network is prepared, the simulation of traffic data that has been obtained during the 2 weeks of observation is run with the following scenario.

- Traffic data simulation design and plan, $i.e.$ times to inject and determine type of attack/anomaly. (When an attack/anomaly occurs? and what attack/anomaly occurs?).
- Define twelve users/nodes (four server nodes and eight most active users) based on the traffic volume.
- Create required normal traffic packages (from the captured dataset) as well as attack/anomaly packages (from MACCDC dataset) to be injected into the network.
- Start injecting traffic data into the network and at the same time logging traffic via port mirroring.
- Labeling simulated anomalies and specific application traffic manually is necessary for experiments to validate results.
- Save the obtained traffic to a file in $.pcap$ format as a data set for entity/user behavior analysis experiments.

The experiments involved inputting raw traffic data for ongoing learning and detection purposes. In the LSTM learning trials, the initial 7 days of traffic data serve as the training dataset, while the final 7 days serve as the testing dataset. The final result is determined by averaging the scores obtained.

2.5. Performance evaluation

In evaluating the experimental results, the authors use several metrics to measure the performance of the proposed model. These performance metrics include: Accuracy, Precision, Sensitivity, F1-score, and Specificity using the formula in (1) – (5) [39].

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{1}$$

$$Precision = TP/(TP + FP) \tag{2}$$

$$Sensitivity = TP/(TP + FN) \tag{3}$$

$$Specificity = TN/(TN + FP\ ) \tag{4}$$

$$F1\ Score = (2 * (Sensitivity * Precision))/(Sensitivity + Precision) \tag{5}$$

Where,
$TP =$ True Positive, namely positive data that is detected correctly.
$TN =$ True Negative, namely negative data that is correctly detected.
$FP =$ False Positive, namely negative data but detected as positive data.
$FN =$ False Negative, namely positive data but detected as negative data.
TP, TN, FP, FN are referred to as the components of the confusion matrix and their values are obtained from observations during the experiment.

3. RESULTS AND DISCUSSION

3.1. Created dataset and feature extraction results

Table 1 shows the recapitulation of traffic data that was successfully recorded during the experiment, and then this traffic data was extracted and selected using the RFE method. The created dataset is dominated by traffic from users browsing web (78.91%), including attack traffic of Edonkey (5.75%), and Unknown traffic (2.9%). The results of feature extraction are shown in Table 2. Streaming and Games traffics have high number of features, because the two applications have many types of traffic. Bittorent traffic has the lowest number of features among the traffic classes since it is very specific application that has specific features. Edonkey malware/attack traffic consists of 12 features.

Table 1. Captured traffic Statistics

Class	Flows		Application protocol
	Traffic Amount	(%)	
WEB	7029000	78.91	Browsing: HTTP, HTTPS
HTTP-STR	153000	1.71	HTTP Streaming
EDONKEY	562500	5.75	eDonkey, eMule obfuscated
BITTORRENT	51300	0.57	Bittorent
CHAT	438300	1.87	MSN, IRC, Yahoo Msn, HTTP Chat, Jabber
EMAIL	533700	4.56	SMTP, HTTP Mail, POP3, POP3s, IMAP, IMAPs
FTP	5400	0.05	FTP-data, FTP control
STREAMING	25200	0.28	Ms. Media Server, Real Player, iTunes, Quick Time
GAMES	4500	0.05	NFS3, HTTP Games, Blizzard Battlenet, Quake II/III, Counter Strike
UNKNOWN	197100	2.19	NBS, Ms-ds, Epmap, Attacks

Table 2. Feature extraction results

Class	Optimized features #	Class	Optimized features #
WEB	9	EMAIL	11
HTTP-STR	9	FTP	8
EDONKEY	12	STREAMING	15
BITTORRENT	7	GAMES	14
CHAT	11	UNKNOWN	13

3.2. Top user profiling

In this experiment, the eight top users were profiled. The profiling results shown in Figure 4 are taken from the data on the average number of packets for 14 days of observation. Four users represent a similar pattern of traffic, while the other four show another similar traffic pattern.

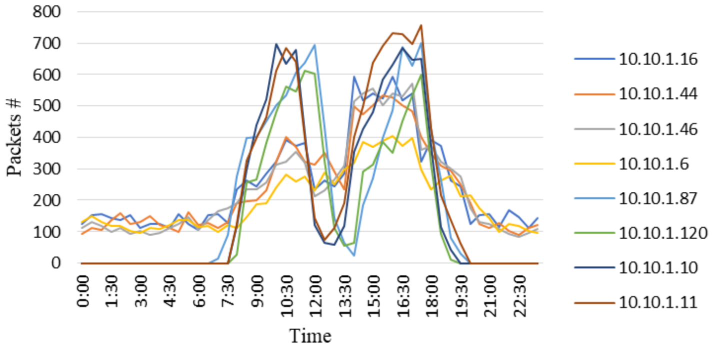


Figure 4. Result of profiling 8 top active users (daily average)

3.3. Comparison with s.SCASHM and LSTM-RFE models

This experiment aims to compare the accuracy in predicting traffic flow between the MPM-RFE model and the s.SCASHM [34] and LSTM-RFE [40] models. For this purpose, the traffic records in the dataset are labeled automatically using Python package. The experimental results shown in Figure 5 exhibit the s.SCASHM and MPM-RFE models are able to maintain consistent average accuracy during periods of fluctuation from 9 am to 5 p.m. per day, while the LSTM-RFE model fails to maintain prediction accuracy because this model does not have the ability to do on-the-fly learning and maybe because the sample used for training is not enough. Nevertheless, in average, the prediction accuracy of MPM-RFE model is better than the s.SCASHM model.

Table 3 shows the results of performance measurement calculations from the detection of attacks/anomalies that occur in the selected applications based on confusion metrics observations. In general, the MPM-RFE model produces better accuracy than the LSTM-RFE model. The accuracy reached 94.01% for the detection of the UNKNOWN class.

The process of forming a memory model for normal and anomalous traffic in Figure 1 was experimentally observed through statistics on the activity of activating cells in memory. Table 4 shows the results of statistical observations of the activation of memory cells that occur during the training of the proposed MPM+RFE model. Measurement on detection and profiling time is shown in Table 5.

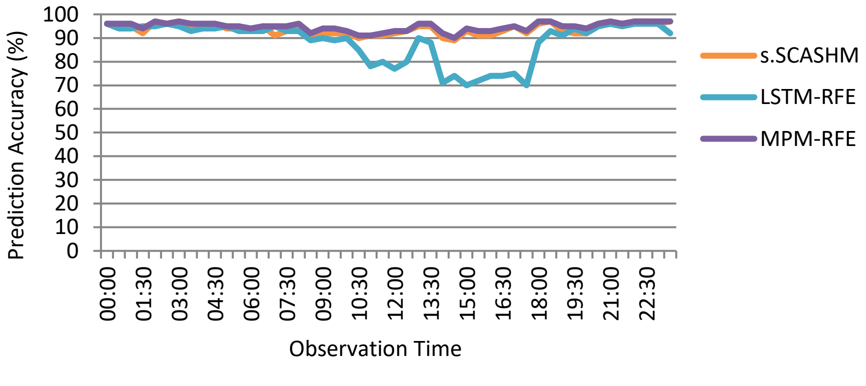


Figure 5. Accuracy comparison: s.SCASHM vs LSTM-RFE vs MPM-RFE

Table 3. Performance metrics results for different applications (LSTM-RFE vs MPM-RFE)

Application	Accuracy (%)		Precision (%)		Sensitivity (%)		F1-score (%)		Specificity	
	MPM-RFE	LSTM-RFE	MPM-RFE	LSTM-RFE	MPM-RFE	LSTM-RFE	MPM-RFE	LSTM-RFE	MPM-RFE	LSTM-RFE
WEB	90.15	79.54	88.55	76.96	98.65	96.32	91.80	84.00	57.93	46.18
CHAT	90.18	72.99	90.33	75.88	98.59	92.43	93.13	82.87	30.00	20.07
EMAIL	92.78	74.10	95.64	75.25	98.77	93.02	95.87	82.11	56.67	28.60
STREAMING	93.66	80.05	95.11	82.01	99.01	95.99	96.08	86.56	55.67	28.56
GAMES	93.22	81.52	94.88	82.17	99.14	96.23	95.65	87.32	59.27	38.90
UNKNOWN	94.01	91.76	93.75	79.98	99.28	96.78	95.45	86.77	60.39	40.79

Table 4. Statistics of memory cell activation during the experiment

	Experiment number				
	1	2	3	4	5
Active cells	86%	51%	80%	85%	49%
Non active cells	12%	37%	14%	9%	41%
Died cells	2%	12%	6%	7%	10%

Table 5. Detection processing time vs profiling time

	Node ID															
	10.10.1.16		10.10.1.44		10.10.1.46		10.10.1.6		10.10.1.87		10.10.1.120		10.10.1.10		10.10.1.11	
	Det.	Prof.	Det.	Prof.	Det.	Prof.	Det.	Prof.	Det.	Prof.	Det.	Prof.	Det.	Prof.	Det.	Prof.
s.SCASHM	0.12	0.01	0.14	0.01	0.14	0.01	0.15	0.01	0.27	0.01	0.44	0.03	0.51	0.06	0.49	0.05
LSTM-RFE	0.04	0.14	0.05	0.14	0.05	0.15	0.06	0.16	0.07	0.16	0.09	0.10	0.16	0.17	0.12	0.19
MPM-RFE	0.02	0.01	0.02	0.01	0.02	0.01	0.02	0.02	0.02	0.02	0.04	0.03	0.07	0.06	0.06	0.05

Det.: Detection, Prof.: Profiling

3.4. Discussion

The performance of the proposed model, MPM-RFE is more consistent than the LSTM-RFE model as shown in Figure 5. The consistent performance is achieved because the MPM-RFE model uses streaming data while the LSTM-RFE model uses static data. The inability to adapt to changes in incoming data causes a decrease in accuracy level. On the other hand, MPM-RFE excels in attack/anomaly detection for streaming data, due to its provision of computational neurogenetic traits like adaptability, simplicity, continuous learning, and robust computation for facilitating real-time user behavior analysis, the model stands out. Comprising three primary components, one of which is the memory component., the computational component (prediction process) and the controller component (prediction analysis). The main strength of the proposed MPM+RFE model lies in the integration among these three components. We compare the proposed MPM-RFE with the s.SCASHM in term of detection time and profiling time and the results are shown in Table 5. The MPM-RFE detects the attack/anomaly and profiles the users faster than the s.SCASHM.

The worst accuracy performance of the MPM-RFE model that was observed was 90.15%. The MPM-RFE model achieves faster learning convergence than the LSTM-RFE model because there is no need to update the current active memory cells, as long as a significant change in the data stream does not occur. As can be seen in Table 4, memory cells are formed when new information is obtained from streaming data. While MPM+RFE and LSTM+RFE are promising approaches for predicting future memory accesses, they each have distinct strengths and weaknesses. Table 6 summarizes the analysis of both models.

Table 6. Summary of analysis of MPM+RFE vs LSTM+RFE

	MPM+RFE	LSTM+RFE
Accuracy	Potential for higher accuracy	Potentially lower accuracy
Interpretability	More interpretable	Challenging to interpret (Black box nature)
Flexibility	Able to improve performance in specific scenarios	Less flexibility
Data scarcity	Requires large datasets of memory access patterns for training process	Requires smaller datasets of memory access patterns for training process
Adaptability	Adaptable to different domains	Adaptable to different domains
Limited availability	The models are less commonly used	Well-established and widely used

In general, choosing the best approach depends on the specific requirements and priorities of the application. If maximizing accuracy and interpretability are crucial, and the challenges of data scarcity and limited availability can be addressed, a memory prediction model with feature analysis might be a better choice. However, if established technology, adaptability, and wider resource availability are more important considerations, an LSTM with feature analysis might be a suitable solution. It is important to note that neither approach is definitively “better” as the optimal choice depends on the specific application and its priorities. Combining elements of both approaches or exploring other emerging techniques might also be worth investigating depending on specific needs and research goals.

Combining memory prediction models with the RFE for user behavior analysis sounds like a promising approach, and achieving high accuracy is definitely a positive outcome. However, it is important to consider the ramifications of the findings beyond just accuracy including Fraud detection, i.e.: more accurate user behavior models can help identify unusual activities that might indicate fraudulent transactions or account compromises; User experience improvement, *i.e.* highly accurate user behavior analysis can lead to better personalization and recommendations across various platforms; Enhanced targeting, i.e.: targeted advertising based on accurate user predictions can be more effective.

Building on the success with a highly accurate user behavior analysis model using memory prediction and RFE, some areas are identified that will be valuable for future advancements, include:

a. Context awareness: the future lies in incorporating more data sources beyond traditional user actions, including user demographics, sentiment analysis from text data, which leads to more nuanced predictions.
b. Continuous learning and improvement: develop strategies for continuous learning and improvement. This might involve incorporating new data sources, user feedback loops, or retraining the model periodically to adapt to evolving user behavior patterns.
c. Model generalizability: ensure the model performs well on unseen data (generalizability). Utilize rigorous testing with diverse datasets to maintain accuracy in real-world scenarios.
d. Real-time personalization and recommendation systems: leverage the model for real-time personalization that can significantly enhance user experience across various applications.
e. Ethical considerations and bias mitigation: ethical considerations become paramount when the memory prediction model becomes more powerful. Future success will involve proactively addressing potential biases in the data and training process. Fairness metrics implementation will ensure that the model treats all users equitably.

The proposed MPM-RFE model is not designed for long-term dependent learning of high-order memory sets, because it will require a long processing time, as required by the hierarchical temporal memory model [34]. This fact may be considered as limitation of the proposed approach. Considering the processing time and complexity of the model, the MPM-RFE model has the potential to be adopted as a way to analyze the behavior entity to prevent insiders' attacks in real-time fashion.

4. CONCLUSION

In this paper, the authors have introduced the combined use of RFE as a tool for feature selection and memory prediction model (MPM-RFE) to detect anomalous traffic/cyberattacks involving insiders. The experimental results show that the MPM-RFE model is capable of demonstrating the detection of attacks/anomaly involving insiders with an accuracy rate of up to 94.01% for the detection of unknown traffic classes. In general, MPM-RFE achieves an accuracy of 90.15% to 94.01%. Besides that, MPM-RFE is also able to provide better accuracy than the LSTM-RFE model, because it is able to carry out on-the-fly learning, so that the system can recognize new patterns of attacks/anomalies. Therefore, MPM-RFE can be implemented as a sub-system to support an intelligent and holistic cybersecurity platform, which is being developed at the Networked Computing Lab, Defense Mathematics Study Program, Defense University. This platform is projected to be used for both government and private institutions. The limitations of the MPM-RFE model are the general limitations of the machine learning model, because this model includes an unsupervised, all-time, and continuous learning approach, where the MPM-RFE carries out learning from user data with normal behavior.

Overall, the state of the art in real-time attack/anomaly detection for network security is promising, with continuous improvements in accuracy, efficiency, and adaptability. However, there are still ongoing challenges like balancing tradeoffs and keeping pace with evolving threats. Research and development efforts are actively exploring new techniques and approaches to enhance the effectiveness and reliability of real-time attack/ anomaly detection in securing networks. While promising advancements are being made, memory prediction frameworks are still under active development. Research is ongoing to overcome limitations in accuracy, generalizability, and interpretability. As the field evolves, we can expect to see these frameworks become increasingly integrated into various computing systems to improve performance and resource utilization. Significant advancements in model architectures, hardware utilization, and transfer learning techniques are among focus area in the near future. The UBA is a rapidly evolving field with continuous advancements in technology and capabilities. By leveraging advanced analytics, behavioral biometrics, and risk-based assessments, UBA empowers organizations to proactively detect and respond to insider threats, compromised accounts, and other sophisticated cyber-attacks. However, ensuring data privacy, addressing false positives, and staying ahead of evolving threats remain ongoing challenges that researchers and security professionals are actively addressing.

REFERENCES

[1] H. Eichenbaum, “Memory systems,” *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 1, no. 4, pp. 478–490, Mar. 2010, doi: 10.1002/wcs.49.
[2] H. Zhang, M. Wang, L. Yang, and H. Zhu, “A novel user behavior analysis and prediction algorithm based on mobile social environment,” *Wireless Networks*, vol. 25, no. 2, pp. 791–803, Oct. 2019, doi: 10.1007/s11276-017-1592-0.
[3] D. Stiawan, A. H. Abdullah, and M. Y. Idris, “Classification of habitual activities in behavior-based network detection,” *Journal of Computing*, vol. 2, no. 8, pp. 1–7, 2010.

[4] Z. Sun, Y. Wang, H. Zhou, J. Jiao, and R. E. Overstreet, "Travel behaviours, user characteristics, and social-economic impacts of shared transportation: a comprehensive review," *International Journal of Logistics Research and Applications*, vol. 24, no. 1, pp. 51–78, Sep. 2021, doi: 10.1080/13675567.2019.1663162.

[5] K. A. P. Perichappan, "Greedy algorithm based deep learning strategy for user behavior prediction and decision making support," *Journal of Computer and Communications*, vol. 06, no. 06, pp. 45–53, 2018, doi: 10.4236/jcc.2018.66004.

[6] K. Deng, L. Xing, L. Zheng, H. Wu, P. Xie, and F. Gao, "A user identification algorithm based on user behavior analysis in social networks," *IEEE Access*, vol. 7, pp. 47114–47123, 2019, doi: 10.1109/ACCESS.2019.2909089.

[7] Sharipuddin *et al.*, "Features extraction on IoT intrusion detection system using principal components analysis (PCA)," in *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Oct. 2020, vol. 2020-Octob, pp. 114–118, doi: 10.23919/EECSI50503.2020.9251292.

[8] Y. Yin *et al.*, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *Journal of Big Data*, vol. 10, no. 1, Feb. 2023, doi: 10.1186/s40537-023-00694-8.

[9] R. Patil, R. Biradar, V. Ravi, P. Biradar, and U. Ghosh, "Network traffic anomaly detection using PCA and BiGAN," *Internet Technology Letters*, vol. 5, no. 1, Sep. 2022, doi: 10.1002/itl2.235.

[10] B. Purnama *et al.*, "Time efficiency on computational performance of PCA, FA and TSVD on ransomware detection," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 102–111, Feb. 2022, doi: 10.52549/ijeei.v10i1.3481.

[11] P. B. Udas, M. E. Karim, and K. S. Roy, "SPIDER: A shallow PCA based network intrusion detection system with enhanced recurrent neural networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 10246–10272, Nov. 2022, doi: 10.1016/j.jksuci.2022.10.019.

[12] J. Hawkins and S. Ahmad, "Why neurons have thousands of synapses, a theory of sequence memory in neocortex," *Frontiers in Neural Circuits*, vol. 10, Mar. 2016, doi: 10.3389/fncir.2016.00023.

[13] J. Hawkins, M. Lewis, M. Klukas, S. Purdy, and S. Ahmad, "A framework for intelligence and cortical function based on grid cells in the neocortex," *Frontiers in Neural Circuits*, vol. 12, Jan. 2019, doi: 10.3389/fncir.2018.00121.

[14] Y. S. Triana, M. A. Osman, A. Pratomo, M. F. Pasha, D. Stiawan, and R. Budiarto, "Neural network models selection scheme for health mobile app development," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 3, p. 1191, Sep. 2023, doi: 10.11591/ijai.v12.i3.pp1191-1203.

[15] T. Liu and K.-M. Lam, "A hybrid egocentric activity anticipation framework via memory-augmented recurrent and one-shot representation forecasting," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 13894–13903, doi: 10.1109/cvpr52688.2022.01353.

[16] H. C. Barron, R. Auksztulewicz, and K. Friston, "Prediction and memory: A predictive coding account," *Progress in Neurobiology*, vol. 192, p. 101821, Sep. 2020, doi: 10.1016/j.pneurobio.2020.101821.

[17] Z. Li *et al.*, "Deep learning-based object detection techniques for remote sensing images: a survey," *Remote Sensing*, vol. 14, no. 10, p. 2385, May 2022, doi: 10.3390/rs14102385.

[18] J. Cai *et al.*, "MeMOT: multi-object tracking with memory," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, doi: 10.1109/cvpr52688.2022.00792.

[19] Y. An, Y. Sun, B. Jin, and X. Wei, "Medication prediction using memory augmented heterogeneous information fusion network," in *2022 8th International Conference on Big Data Computing and Communications (BigCom)*, Aug. 2022, pp. 233–242, doi: 10.1109/bigcom57025.2022.00037.

[20] Y. Cui, S. Ahmad, and J. Hawkins, "The HTM spatial pooler—a neocortical algorithm for online sparse distributed coding," *Frontiers in Computational Neuroscience*, vol. 11, Nov. 2017, doi: 10.3389/fncom.2017.00111.

[21] N. Yan and O. T.-S. Au, "Online learning behavior analysis based on machine learning," *Asian Association of Open Universities Journal*, vol. 14, no. 2, pp. 97–106, Dec. 2019, doi: 10.1108/aaouj-08-2019-0029.

[22] Y. Cui, S. Ahmad, and J. Hawkins, "Continuous online sequence learning with an unsupervised neural network model," *Neural Computation*, vol. 28, no. 11, pp. 2474–2504, Nov. 2016, doi: 10.1162/neco_a_00893.

[23] K. Shaukat *et al.*, "A review of time-series anomaly detection techniques: a step to future perspectives," in *Advances in Information and Communication*, Springer International Publishing, 2021, pp. 865–877, doi: 10.1007/978-3-030-73100-7_60.

[24] N. Mohamed, "Current trends in AI and ML for cybersecurity: A state-of-the-art survey," *Cogent Engineering*, vol. 10, no. 2, Oct. 2023, doi: 10.1080/23311916.2023.2272358.

[25] S. Sharipuddin *et al.*, "Enhanced deep learning intrusion detection in IoT heterogeneous network with feature extraction," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 9, no. 3, Sep. 2021, doi: 10.52549/.v9i3.3134.

[26] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in *2018 10th International Conference on Cyber Conflict (CyCon)*, May 2018, doi: 10.23919/cycon.2018.8405026.

[27] D. Stiawan *et al.*, "An Approach for Optimizing Ensemble Intrusion Detection Systems," *IEEE Access*, vol. 9, pp. 6930–6947, 2021, doi: 10.1109/access.2020.3046246.

[28] D. Stiawan, Susanto, A. Bimantara, M. Y. Idris, and R. Budiarto, "IoT botnet attack detection using deep autoencoder and artificial neural networks," *KSII Transactions on Internet and Information Systems*, vol. 17, no. 5, pp. 1310–1338, May 2023, doi: 10.3837/tiis.2023.05.001.

[29] Kurniabudi *et al.*, "Improvement of attack detection performance on the internet of things with PSO-search and random forest," *Journal of Computational Science*, vol. 64, p. 101833, Oct. 2022, doi: 10.1016/j.jocs.2022.101833.

[30] J. Jose Diaz Rivera, T. Ahmed Khan, W. Akbar, M. Afaq, and W.-C. Song, "An ML based anomaly detection system in real-time data streams," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec. 2021, pp. 1329–1334, doi: 10.1109/csci54926.2021.00270.

[31] I. A. Kandhro *et al.*, "Detection of real-time malicious intrusions and attacks in IoT empowered cybersecurity infrastructures," *IEEE Access*, vol. 11, pp. 9136–9148, 2023, doi: 10.1109/access.2023.3238664.

[32] S. Mohamad and A. Bouchachia, "Deep online hierarchical dynamic unsupervised learning for pattern mining from utility usage data," *Neurocomputing*, vol. 390, pp. 359–373, May 2020, doi: 10.1016/j.neucom.2019.08.093.

[33] H. Tyagi and R. Kumar, "Attack and anomaly detection in IoT networks using supervised machine learning approaches," *Revue d'Intelligence Artificielle*, vol. 35, no. 1, pp. 11–21, Feb. 2021, doi: 10.18280/ria.350102.

[34] R. Budiarto, A. A. Alqarni, M. Y. Alzahrani, M. F. Pasha, M. F. Mohamed Firdhous, and D. Stiawan, "User Behavior Traffic Analysis Using a Simplified Memory-Prediction Framework," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2679–2698, 2022, doi: 10.32604/cmc.2022.019847.

[35] NETRESEC, "Capture files from Mid-Atlantic CCDC," *netresec.com*, 2012. https://www.netresec.com/?page=MACCDC (accessed Dec. 13, 2022).

[36] J. Zhang, "Machine learning with feature selection using principal component analysis for malware detection: a case study," *arxiv preprint arXiv:1902.03639*, 2019.

[37] Z. Wang, F. Jiang, T. Liu, F. Xie, and P. Li, “Attention-based spatial and spectral network with PCA-guided self-supervised feature extraction for change detection in hyperspectral images,” *Remote Sensing*, vol. 13, no. 23, p. 4927, Dec. 2021, doi: 10.3390/rs13234927.

[38] J. Hawkins and S. Blakeslee, *On intelligence*. New York, USA: Henry Holt and Company, 2005.

[39] Y.-K. Gu, B. Xu, H. Huang, and G. Qiu, “A fuzzy performance evaluation model for a gearbox system using hidden Markov model,” *IEEE Access*, vol. 8, pp. 30400–30409, 2020, doi: 10.1109/access.2020.2972810.

[40] D. Stiawan *et al.*, “An improved LSTM-PCA ensemble classifier for SQL injection and XSS attack detection,” *Computer Systems Science and Engineering*, vol. 46, no. 2, pp. 1759–1774, 2023, doi: 10.32604/csse.2023.034047.

BIOGRAPHIES OF AUTHORS

Yaya Sudarya Triana received B.Sc. degree in statistics from University of Padjadjaran, Indonesia in 1988, M.Kom. in Master of Information Technology from University of Indonesia in 2002 and Ph.D. in statistics from Universiti Malaysia Trengganu, Malaysia in 2013, Software Engineer 1991-2003 at an IT Company at Tokyo, Bandung and Jakarta, respectively. Currently, he is an assistant professor at Department of Information System, Universitas Mercu Buana Indonesia. His research interests include fuzzy, data science, artificial intelligent, business intelligent, data analytics, statistics, information system, and machine learning. He can be contacted at email: yaya.sudarya@mercubuana.ac.id.

Mohd Azam Osman is an assistant professor at the School of Computer Sciences, Universiti Sains Malaysia. He obtained bachelor degree and a master degree in computer science from Universiti Sains Malaysia, Pulau Pinang, Malaysia in 1996 and 2000 respectively. His current research focuses on mobile applications, image processing and deep learning to detect and track multi-fish in underwater and human emotional identification. He can be contacted at email: azam@usm.my.

Deris Stiawan received a Ph.D. degree in computer engineering from Universiti Teknologi Malaysia, Malaysia. He is currently a professor at the Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya. His research interests include computer networks, intrusion detection/prevention systems, heterogeneous networks, and intelligent systems. He can be contacted at email: deris@unsri.ac.id.

Rahmat Budiarto received B.Sc. degree in mathematics from Bandung Institute of Technology, Indonesia in 1986, M.Eng. and Dr.Eng. in computer science from Nagoya Institute of Technology, Japan in 1995 and 1998, respectively. Currently, he is a full professor at Department of Computer Science, College of Computer and Information, Albaha University, KSA. His research interests include intelligent systems, brain modeling, IPv6, network security, wireless sensor networks, and MANETs. He can be contacted at email: rahmat@bu.edu.sa.