



# Cyberattack detection model using deep learning in a network log system with data visualization

Jung-Chun Liu<sup>1</sup> · Chao-Tung Yang<sup>1,2,3</sup> · Yu-Wei Chan<sup>4</sup> · Endah Kristiani<sup>5,6</sup> · Wei-Je Jiang<sup>1</sup>

Accepted: 25 February 2021 / Published online: 16 March 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Network log data is significant for network administrators, since it contains information on every event that occurs in a network, including system errors, alerts, and packets sending statuses. Effectively analyzing large volumes of diverse log data brings opportunities to identify issues before they become problems and to prevent future cyberattacks; however, processing of the diverse NetFlow data poses challenges such as volume, velocity, and veracity of log data. In this study, by means of Elasticsearch, Logstash, and Kibana, i.e., the ELK Stack, we construct an analysis and management system for network log data, which provides functions to filter, analyze, and display network log data for further applications and creates data visualization on a Web browser. In addition, an advanced cyberattack detection model is facilitated using deep neural network (DNN), recurrent neural networks (RNN), and long short-term memory (LSTM) approaches. By knowing cyberattack behaviors and cross-validating with the log analysis system, one can learn from this model the characteristics of a variety of cyberattacks. Finally, we also implement Grafana to perform metrics monitoring.

**Keywords** Information security · ELK stack · DDoS · Cyberattack · Deep learning

## 1 Introduction

With the rapid growth and diversity of network applications, incidents of cyberattacks are becoming more diverse, and the adverse outcomes caused by information security compromises are gaining more attention. As a result, the public pays much attention to cyber security issues, especially the administrators who want to obtain timely, accurate, and trustworthy network information so that they can plan effective countermeasures

---

✉ Chao-Tung Yang  
ctyang@thu.edu.tw

Extended author information available on the last page of the article

[12]. The purpose of this study is to provide a management and analysis system to monitor and analyze log data. In addition, continuous data visualization is provided, so that users can instantly monitor and track network traffic at any time. Recently, many research works proposed to apply deep learning methods to learn and train for cyberattack detection. Through continuously training, the attack detection system can accurately identify types of attacks and detect the predicted attacks. To verify the credibility of results [29], analysis and comparison were made.

Usually, cyberattacks are perceivable by high computer network traffic. Networks of large organizations such as a university are the most common targets of cyberattacks, so an effective network log management system that can be easily used by network administrators is eagerly called for. After processing the network log data, and according to predetermined requirements, charts were sorted and displayed by the log analysis system on the management interface [23, 28]. The source data were collected by the university computer center. One solution was to regularly upload the original log data to the database so that the network log management system could maintain it with ease [32]. The results and abnormalities produced by the log analysis system could also be fed back to the computer center for relevant responses. In [31], a deep learning model was used to classify the abnormal behaviors of networks to cross-check with the log system to attain more accurate prediction [13].

In this study, a network log management and analysis system using the ELK Stack is built. The system stores and analyzes log data and visually displays the analysis outcomes on dashboards. In addition, we propose to train and build a cyberattack detection model with deep learning methods. By using emulated cyberattack data, the model can be trained. After the model reaches certain degrees of accuracy, the attack data can be categorized according to the characteristics of cyberattacks. The main contributions of this study are summarized as follows:

- To utilize the ELK Stack to build a computer network log data and management system, which can process and analyze network log data, and present log information to users in a more understandable way by data visualization. In addition, cyberattacks are filtered out and visualized.
- To build a classification model using deep learning methods to detect and classify cyberattack events. The model can effectively help network managers evaluate and track network flows, detect suspicious network sources or anomalous behaviors through log data, and maintain good network security levels.

## 2 Background review and related works

In this section, the key technologies used in this study, including the ELK Stack, deep learning methods, and Grafana, are described; also related works are reviewed.

## 2.1 ELK stack

The ELK Stack [11, 18] consists of three open-source software systems, namely Elasticsearch, Logstash, and Kibana. Through log data processing, users are able to directly search the log files and get the required information. However, for systems with large-scale logs, this approach is very inefficient, since it lacks a centralized log management system to collect and aggregate the logs. The common solution is to set up a centralized log management system to collect, manage, and access logs coming from all sources.

Generally, a large-scale data system is composed of a distributed deployment architecture, in which different service modules are deployed on different servers. When a problem arises, in most situations it needs to locate specific service areas according to the key information in the problem [10]. Hence, by constructing a centralized log system, the locating inefficiency can be easily alleviated.

For large-scale data systems, the ELK Stack provides a complete set of solutions, which can be used in conjunction with each other effectively to meet the specific needs of numerous occasions. Hence, it has been adopted in mainstream log systems at present.

## 2.2 DNN

Artificial intelligence (AI) is not a new concept, and even deep neural network (DNN) is also an old one. In the past, AI as well as DNN were limited by technological development. Although there were occasional signs of revival, they never lasted for long. Since 2005, AI has gradually received attention, but not much improvement was made. However, it saw a sudden rise in 2012 and has become the focus of research in recent years [24].

DNN is a branch of machine learning, which mainly uses supervised or unsupervised learning as the means to train machine in order to improve efficiency and accuracy. The difference between DNN and recursive neural network (RNN) or convolutional neural network (CNN) is that DNN refers to the fully connected neuron structure, and does not contain convolution units or temporal associations. However, DNN would have some problems in practice. For example, the upper and lower neurons of a fully connected DNN can form a connection with each other, which easily causes overfitting and results in regional optimality [33].

## 2.3 RNN

RNN [6, 27] has a feature that the output of each layer in a multi-layer neural network is directly appended to the self-loop of the input. By this architecture, the input before the input of the layer can be memorized. When the input data is a

continuous sequence, the input memory before the input can be incorporated into the thinking mode of the next input.

That is to say, the current output is affected not only by the input of the previous layer, but also by the output of the same layer (i.e., the previous one), similar to the statistical time series [17].

## 2.4 LSTM

Long short-term memory (LSTM) is a modified RNN, mainly used to solve the problem of gradient disappearance and gradient explosion in the process of constant time series. In simple terms, LSTM can perform better in long-term sequence training than a normal RNN because LSTM solves the above-mentioned RNN problem by adopting an improved memory management architecture [22].

## 2.5 Grafana

Grafana, an open-source visualization and analytics tool, is used on top of a variety of databases, but most widely used together with Prometheus, Graphite, and Elasticsearch. Essentially, it is an upgrade of Graphite-web; it provides more flexible dashboard functionality, more options for editing, and no extra tracking overheads due to different data sources. Compared to other monitoring software, Grafana allows users to create a variety of charts and also has simpler installation settings [2, 4].

Once a machine learning model was built using DNN or CNN, it can be converted into an application programming interface (API). The API deployed in a Web server will convert data in a standard exchange format, such as JSON or XML. In Grafana, JSON plugin will inference the value passed from the API. In this way, log data can be monitored in real time to decide whether to categorize the log as being under cyberattacks or not, and alerts can be sent to network administrators.

## 2.6 Related works

In recent years, cybersecurity incidents [3, 5] have been reported and highly regarded, such as loopholes in OpenSSL Heartbleed, cyberattack on JP Morgan Chase, distributed denial of service (DDoS) attacks threatening GitHub. The various cyberattack incidents [9, 20] demonstrate that the importance of information security should not be overlooked. According to Global Risks 2020 published by World Economic Forum, cyberattacks on critical infrastructure were rated the fifth top risk in 2020.

In normal circumstances, the network log data [25] is recorded whenever a computer network is used. Network log data [26] is essential to Web administrators, since it provides information such as system errors, warnings, and alerts. The purpose of this study is to provide a network log data management system, which can

perform visualization analysis for different types of users. The proposed system uses the ELK Stack technology to filters, screens, and analyzes network log data based on different user demands. And finally it applies data visualization [15, 19] on a Web browser. The services of the implemented system mainly consist of Elastic-search, Logstash and Kibana [1] software, which provides a comprehensive network log management and visual analysis service by combining distributed search and analysis services, data collection, data filtering processing, and visualization of data processing results.

Kozik [8] developed a combination of NetFlows with an extreme learning machines (ELM) classifier, in which a reliable tool for a network incidents detection using a Map-Reduce programming model was implemented. Kiran and Chhabra [7] investigated the real-time classification of network flow based on unsupervised and semi-supervised machine learning methods. Their results indicated that the proposed algorithm reaches 90% accuracy in classifying elephants and mice clusters.

A survey paper related to network attack detection was presented by Navarro et al. [16], in which a survey of publications using multi-step attack detection methods was conducted, and 181 publications covering 119 methods were reviewed. Mahmoud et al. [14] surveyed the literature of cyber physical systems (CPS) security. They focused on three main cyberattacks: denial of service (DoS), deception, and replay attacks. Some available attack models, defense approaches, and monitoring methods were also surveyed and discussed.

Our study provides the network monitoring system with pre-trained models based on DNN, RNN, and LSTM algorithms. Our experimental results, especially the performance comparisons of the classification accuracies of DNN, RNN, and LSTM, can provide useful suggestions for network administrators.

### 3 System design and implementation

This section first introduces implementation of the network log data analysis system using the ELK Stack, then shows visualization of analysis results, and finally discusses the three deep learning models for cyberattack detection. The network log data collected in this study is from the University Computer Center of Tunghai University. There are more than 8 million pieces of data processed per day. According to the actual amount of data collected during the school period, the size of a single piece of data is about 2 to 3 GB. At present, it has been accumulated to 6 TB, and the relevant equipment level will be upgraded according to the hardware demand in the future.

#### 3.1 System architecture

In this section, we first introduce deployment of the entire ELK Stack system and then use the log data of the computer center to import and write the configuration file so that the corresponding log data field can be read by the log analysis system to perform visual analysis. Then, we discuss how to use the ELK Stack to build a

network log system that will perform a variety of visual analysis of network usage on the campus, and use deep learning models to detect attacks and to assist the reliability of the network log system for information security.

The main environment of the network log system consists of a set of the ELK Stack system on the server and related assist kits, combined with the network resources of the university computer center, and a variety of visual analysis on the Web browser to provide administrators and users with a clear view of the campus network usage information [30].

To implement the system, we installed Anaconda3 on Windows 10 and used Jupyter Notebook as the development environment of Python, preprocessed the log data, then imported deep learning models to train and learn, and then to detect the attack behaviors of other network log data. Different types of deep learning models were implemented for performance comparisons, and then, we selected the best deep learning model according to accuracies of classification of the log data.

Finally, Grafana was used to monitor the performance of Elasticsearch. After the log system is deployed, as the volume of data accumulates continuously, the system needs to be monitored all the time. Grafana can instantly monitor the data traffic and current performance metrics of the system. Figure 1 shows the detailed system architecture of this study.

### 3.2 NetFlow log system

First, we used a shell script to download files from the server, which collects NetFlow log data of local computers. Then, we used the ELK Stack for preliminary analysis: Logstash will continue to collect and filter log data, and do file format conversion; Elasticsearch stores data sent by Logstash. Finally, we used Kibana to visualize the log data on the Web site.

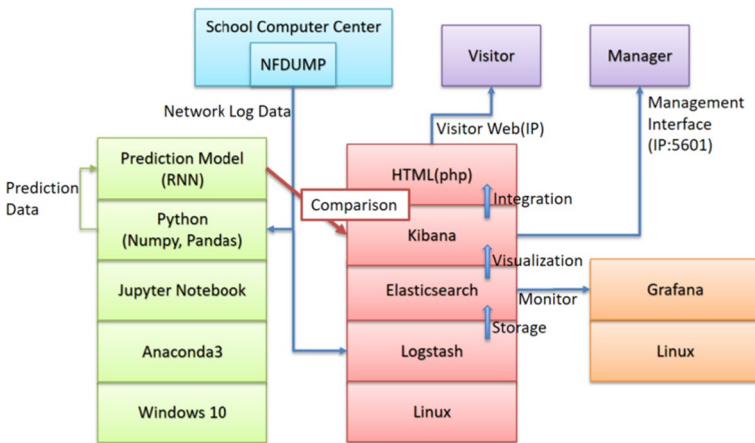


Fig. 1 System architecture

### 3.2.1 Visualization analysis of network usage

Network log data is continually generated. To provide administrators with simple and understandable information as quickly as possible, the solution is data visualization. On the university campus, each building has its own network domain. In order to facilitate the management of the network usage of each building, it is necessary to filter out the network domain of each building separately. This study adopts the commonly used flow analysis method to visualize the network flow of each building. There are two important elements in the network flow, that is, source and destination. Through analysis of IP usage profiles between source and destination, the relationship charts of IPs, ports, and protocols are displayed, making it easier for administrators to view relevant information. Also, geographical information can be combined to create visual heat maps to show the frequency of network usages from local regions and around the world simultaneously.

### 3.2.2 Visualization analysis of attack detection

To analyze and detect cyberattacks, we used the characteristics of attack behaviors and applied filtering operations to extract the anomalous data for visual inspections. Features like CodeRed and Worm are relatively fixed and can be easily scrutinized. The characteristics of DDoS are relatively unfixed, so a feature range is set for data processing and analysis [21].

## 3.3 Deep learning models

In this section, we investigate the deep learning models used for training and demonstrate the accuracy of cyberattack classification of the log data. This study experiments and tests the cyberattack behaviors on the university computer network with a large amount of log data. To emulate the network log data with cyberattacks data, the source data uses attacks such as the CodeRed, Nimda, and Worm with other data to perform the training process.

In the experiment, Keras, a powerful Python library for deep learning, was used to train and test the deep learning models. The used Python version is version 3.6. Then, Anaconda3's Jupyter Notebooks suite was used to write Python codes. And finally, we compared accuracies of attack classification using three deep learning models.

### 3.3.1 Network log data preprocessing

First, the log data was preprocessed to convert it into a format that can be used for the deep learning models to learn. Then, we extracted cyberattack behaviors of the data, classified the data according to its characteristics of behavior, and prepared a suitable amount of data to have a reliable training process and enhance classification

accuracy. The volumes of the used network log dataset samples were 76374 records for training and 37618 records for testing.

For the multiple attacks classification, in order to make training more effective, we tried to use uniform amounts of various cyberattacks. Our training set was collected from log data recorded at different times, so the model did not learn the same data collected at the same time interval. The results of the three used deep learning models will be described in detail as follows.

### 3.3.2 DNN model training and classification

The used DNN model adopts the supervised learning method for training. We found that if the preset parameters have not been properly adjusted at first, the overfitting situation is easily encountered, and the test result would be unexpected. However, by adjusting learning rates, numbers of neurons, and optimizers, the overfitting situation is gradually lessened, and the result of training sets can be improved. Finally, the accuracy of the validation set can reach 99.98%.

In the aspect of the test set, four sets of test data were extracted from different dates and times, and the accuracies of attack classifications were found to be 98.88%, 99.97%, 99.47%, and 99.91%, respectively. To conclude, the average classification accuracy could reach 99% or more, demonstrating that the DNN model has high accuracy.

### 3.3.3 RNN model training and classification

The used RNN model is also trained by the supervised learning method. In order to compare performances of the three neural network models, the training set of each model uses the same data set. In the process of training, we found that the RNN model is obviously better than the DNN model, since the RNN model achieves high accuracy more easily than the DNN model. In the RNN model, the optimizer was also used in the same way as DNN. We found that almost no overfitting occurred in the training process.

In the aspect of the test set, four sets of test data were extracted from different dates and times, and the accuracies of attack classifications were found to be 100.0%, 99.9%, 100.0%, and 99.98%, respectively. To conclude, the average classification accuracy is close to 100.0%.

### 3.3.4 LSTM model training and classification

For the used LSTM model, the training process is also carried out using the supervised learning method. The training result of it is found to be very similar to that of the RNN model. The reason may be that the long-term sequence is not obvious in the training set, and the test results of the LSTM and RNN models are very close to 100.0%, making it difficult to distinguish the difference of accuracies between them.

The test set was composed of the same four sets of test data used in the other two models, and the accuracies of attack classifications were found to be 100.0%, 99.98%, 100.0%, and 99.98%, respectively, which are similar to that in the RNN model.



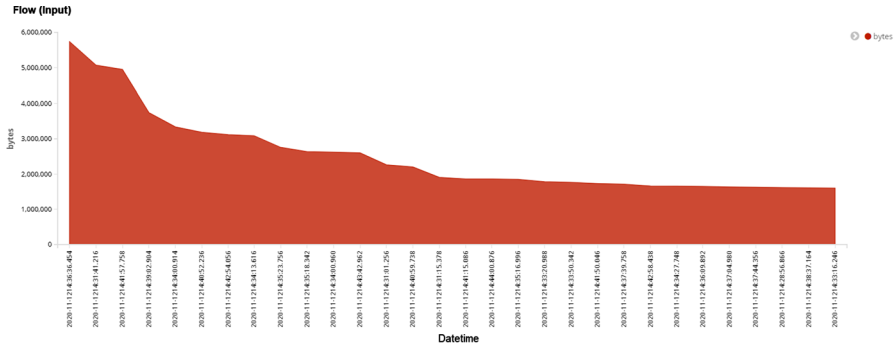


Fig. 2 Count of packet

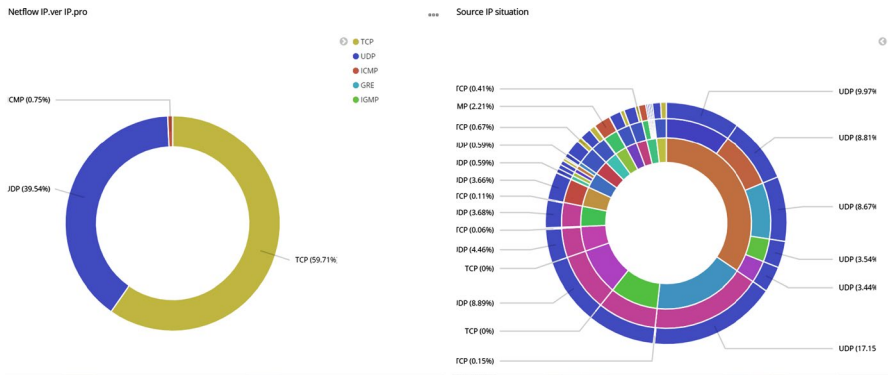


Fig. 3 Source IP details

## 4 Experimental results

### 4.1 Visualization of the network log system

Charts used in visual presentations of log data analysis are designed to deal with diverse types of usages, such as that of the student dormitory, which reveals the most frequent anomalies of network traffic. There are many causes of the anomalous traffic: computers infected with viruses, hacker intrusions, and the use of plug-in software, which can be clearly observed over the log data.

In the following, the meanings or causes of various data visualizations will be described, and relevant explanations and actions will be deliberated.

Figure 2 extracts the count of packets. Since several attack patterns are correlated with this factor, this graph is separately and distinctly plotted for display.

Figure 3 shows the source IP usage in pie charts, which depicts ratios of the source IPs with used network protocols such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Control Message Protocol

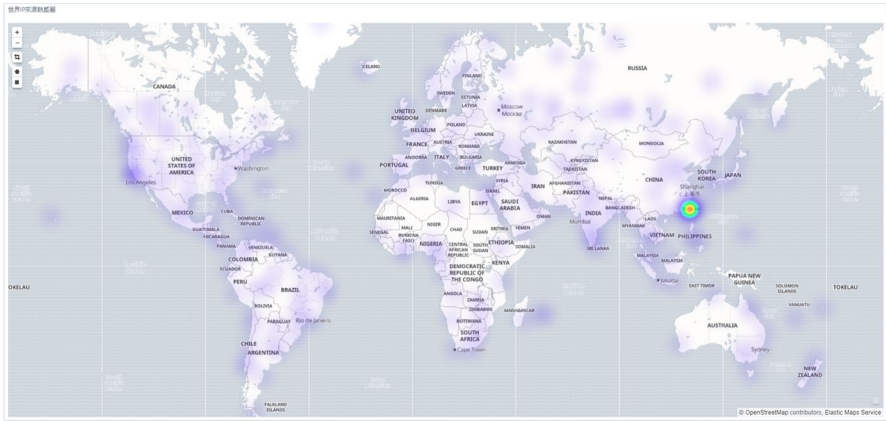


Fig. 4 World IP source heat map

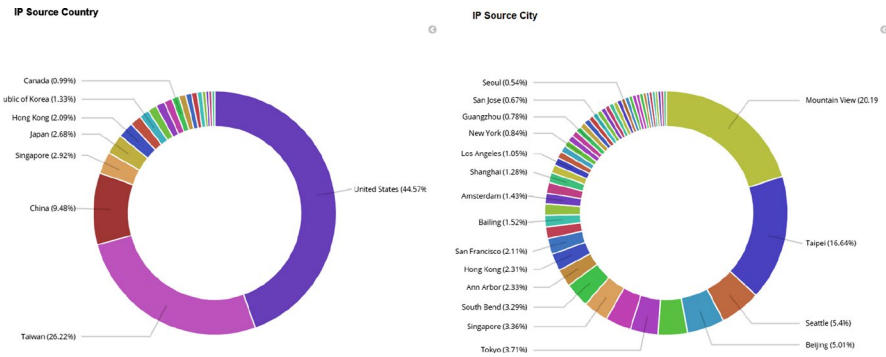


Fig. 5 Geographic information of source IP

(ICMP), and Internet Group Management Protocol (IGMP). In particular, the IP with the highest usage ratio can be selected to regulate the network usage.

Figure 4 shows the source of connections around the world with a zoomable heat map. Figure 5 shows the ratios of detailed geographic information in pie charts of the source IPs around the world from various countries and cities, with clear visual information available for users.

Figures 6 and 7 show related patterns of several cyberattack behaviors, such as that of CodeRed, Nimda, ICMP Flood, and DDoS. By classifying the suspicious source of attacks, and analyzing the time of the network traffic to decide the time of attack, the attack behavior can be queried. Therefore, the source of cyberattacks can be quickly identified.

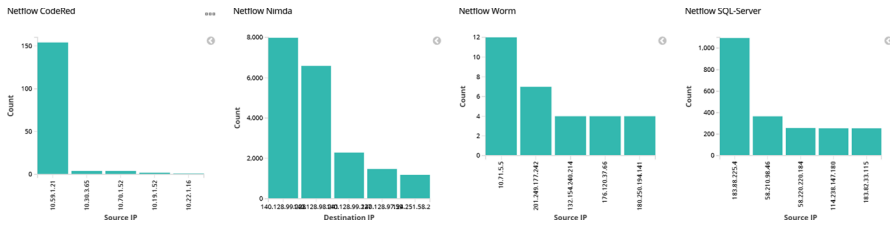


Fig. 6 Cyberattack detection1

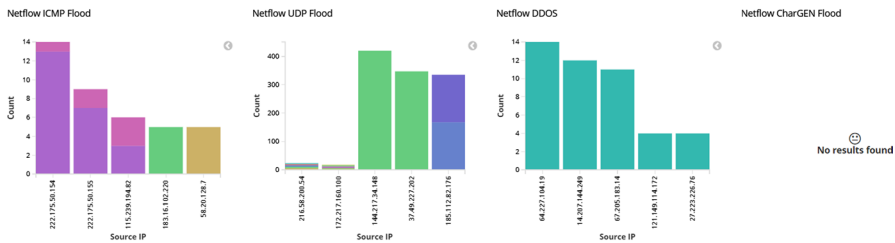


Fig. 7 Cyberattack detection2

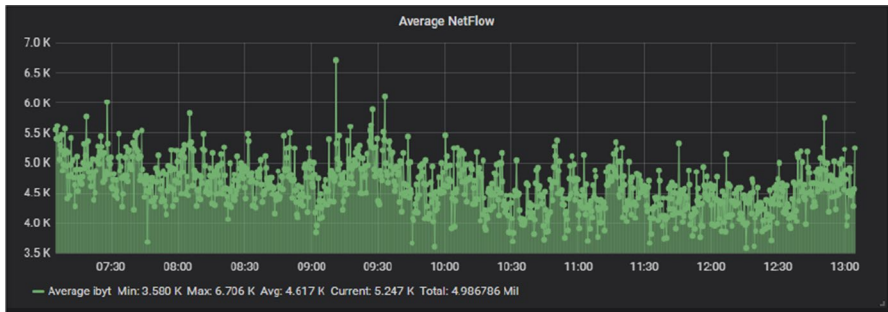


Fig. 8 Average NetFlow

## 4.2 System monitoring

For system monitoring purpose, Grafana is used to link Elasticsearch in the log system, so that the system can immediately send log information to Grafana for data analysis and processing. As shown in Figs. 8, 9, and 10, the average NetFlow, packets counts, and NetFlow performance metrics can be continuously displayed, making it easier for network administrators to monitor the network usage.

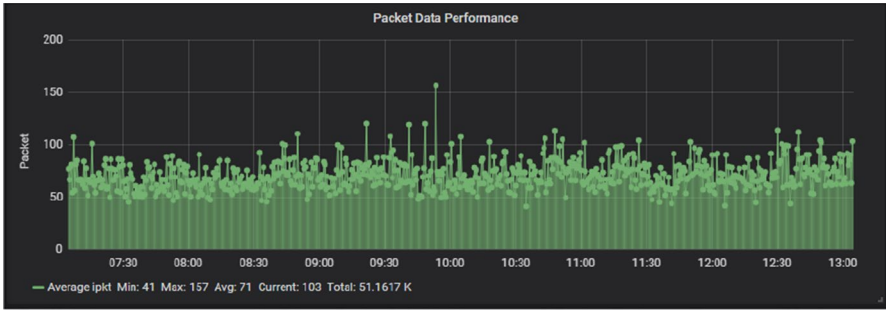


Fig. 9 Packet data performance

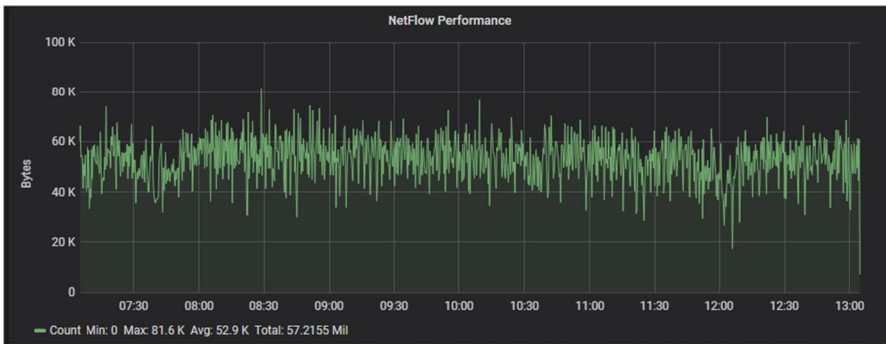


Fig. 10 NetFlow performance

```

Train on 76374 samples, validate on 37618 samples
Epoch 1/40
76374/76374 [=====] - 2s 20us/step - loss: 0.7148 - acc: 0.9528 - val_loss: 0.1583 - val_acc: 0.9874
Epoch 2/40
76374/76374 [=====] - 1s 15us/step - loss: 0.1527 - acc: 0.9891 - val_loss: 0.0486 - val_acc: 0.9967
Epoch 3/40
76374/76374 [=====] - 1s 15us/step - loss: 0.0699 - acc: 0.9950 - val_loss: 0.0407 - val_acc: 0.9975
Epoch 4/40
76374/76374 [=====] - 1s 15us/step - loss: 0.0467 - acc: 0.9963 - val_loss: 0.0383 - val_acc: 0.9975
Epoch 5/40
76374/76374 [=====] - 1s 15us/step - loss: 0.0339 - acc: 0.9965 - val_loss: 0.0134 - val_acc: 0.9976
Epoch 6/40
76374/76374 [=====] - 1s 15us/step - loss: 0.0203 - acc: 0.9971 - val_loss: 0.0078 - val_acc: 0.9981
Epoch 7/40
76374/76374 [=====] - 1s 15us/step - loss: 0.0189 - acc: 0.9974 - val_loss: 0.0073 - val_acc: 0.9981
Epoch 8/40
76374/76374 [=====] - 1s 15us/step - loss: 0.0149 - acc: 0.9977 - val_loss: 0.0058 - val_acc: 0.9988
Epoch 9/40
76374/76374 [=====] - 1s 15us/step - loss: 0.0137 - acc: 0.9979 - val_loss: 0.0062 - val_acc: 0.9988
    
```

Fig. 11 Training and validation loss of DNN model

### 4.3 Training models

This section shows results of the deep learning models used for training and tests in the study. The training and validation loss, and the neural layer hierarchy of the three used models are shown in the following figures.

Figure 11 shows the training and validation loss of the DNN model.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 64)	512
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 128)	8320
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 256)	33024
dropout_3 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 128)	32896
dropout_4 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 3)	387
Total params: 75,139		
Trainable params: 75,139		
Non-trainable params: 0		

Fig. 12 Neural layer hierarchy of DNN model

Figure 12 shows the neural layer hierarchy of the DNN model.

Figure 13 shows the training and validation loss of the RNN model.

Figure 14 shows the neural layer hierarchy of the RNN model.

Figure 15 shows the training and validation loss of the LSTM model.

Figure 16 shows the neural layer hierarchy of the LSTM model.

#### 4.4 Performance comparison of models

In view of the classification of the network attacks, all the three deep learning models have high accuracies in validation. Figures 17 and 18 show the training loss and validation loss, and training accuracy and validation accuracy of the DNN model, respectively.

```

Train on 76374 samples, validate on 37618 samples
Epoch 1/40
76374/76374 [=====] - 1s 18us/step - loss: 0.1223 - acc: 0.9592 - val_loss: 0.0046 - val_acc: 0.9983
Epoch 2/40
76374/76374 [=====] - 1s 10us/step - loss: 0.0026 - acc: 0.9991 - val_loss: 0.0020 - val_acc: 0.9989
Epoch 3/40
76374/76374 [=====] - 1s 10us/step - loss: 0.0011 - acc: 0.9995 - val_loss: 7.1630e-04 - val_acc: 0.9997
Epoch 4/40
76374/76374 [=====] - 1s 10us/step - loss: 7.9759e-04 - acc: 0.9998 - val_loss: 3.2164e-04 - val_acc: 0.9999
Epoch 5/40
76374/76374 [=====] - 1s 11us/step - loss: 6.0564e-04 - acc: 0.9998 - val_loss: 2.8309e-04 - val_acc: 0.9999
Epoch 6/40
76374/76374 [=====] - 1s 11us/step - loss: 6.0700e-04 - acc: 0.9998 - val_loss: 2.4609e-04 - val_acc: 0.9999
Epoch 7/40
76374/76374 [=====] - 1s 11us/step - loss: 4.9710e-04 - acc: 0.9999 - val_loss: 8.3594e-05 - val_acc: 0.9999

```

Fig. 13 Training and validation loss of RNN model

Layer (type)	Output Shape	Param #
simple_rnn_1 (SimpleRNN)	(None, None, 16)	384
dense_1 (Dense)	(None, None, 32)	544
simple_rnn_2 (SimpleRNN)	(None, None, 16)	784
simple_rnn_3 (SimpleRNN)	(None, 16)	528
dense_2 (Dense)	(None, 3)	51
Total params: 2,291		
Trainable params: 2,291		
Non-trainable params: 0		

Fig. 14 Neural layer hierarchy of RNN model

```

Train on 76374 samples, validate on 37618 samples
Epoch 1/40
76374/76374 [=====] - 3s 44us/step - loss: 0.1599 - acc: 0.9690 - val_loss: 0.0082 - val_acc: 0.9975
Epoch 2/40
76374/76374 [=====] - 1s 17us/step - loss: 0.0042 - acc: 0.9984 - val_loss: 0.0033 - val_acc: 0.9984
Epoch 3/40
76374/76374 [=====] - 1s 17us/step - loss: 0.0016 - acc: 0.9996 - val_loss: 0.0011 - val_acc: 0.9997
Epoch 4/40
76374/76374 [=====] - 1s 18us/step - loss: 9.2654e-04 - acc: 0.9997 - val_loss: 9.0691e-04 - val_acc: 0.9997
Epoch 5/40
76374/76374 [=====] - 1s 18us/step - loss: 5.7890e-04 - acc: 0.9998 - val_loss: 9.2351e-04 - val_acc: 0.9994
Epoch 6/40
76374/76374 [=====] - 1s 18us/step - loss: 3.8739e-04 - acc: 0.9999 - val_loss: 1.4893e-04 - val_acc: 1.0000
Epoch 7/40
76374/76374 [=====] - 1s 19us/step - loss: 3.2720e-04 - acc: 0.9999 - val_loss: 1.4190e-04 - val_acc: 1.0000
Epoch 8/40

```

Fig. 15 Training and validation loss of LSTM model

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, None, 16)	1536
dense_1 (Dense)	(None, None, 32)	544
lstm_2 (LSTM)	(None, None, 16)	3136
lstm_3 (LSTM)	(None, 16)	2112
dense_2 (Dense)	(None, 3)	51
Total params: 7,379		
Trainable params: 7,379		
Non-trainable params: 0		

Fig. 16 Neural layer hierarchy of LSTM model

Figures 19 and 20, respectively, show the training loss and validation loss, and training accuracy and validation accuracy of the RNN model, which has better performance than that of the DNN model.

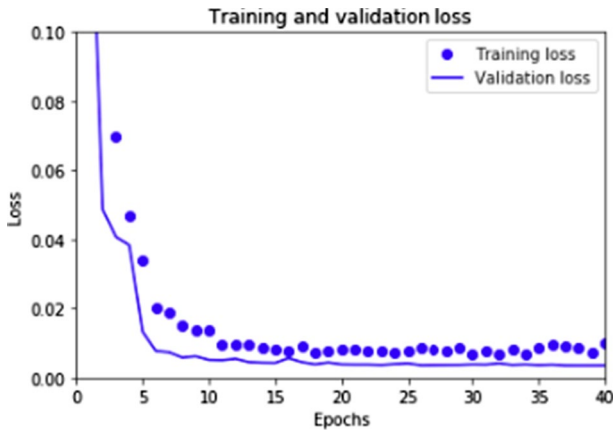


Fig. 17 DNN model's training and validation loss

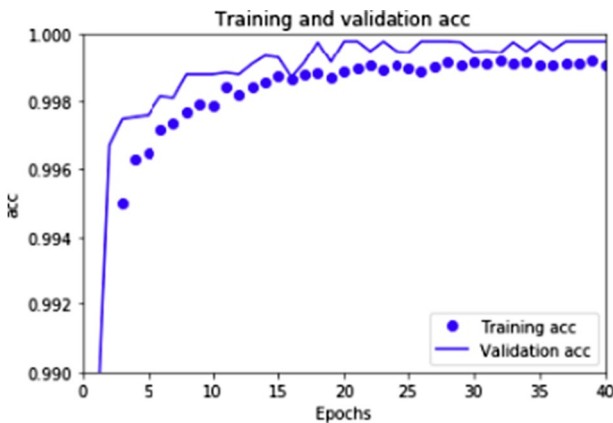


Fig. 18 DNN model's training and validation accuracy

The training loss and validation loss, and training accuracy and validation accuracy of the LSTM model are, respectively, shown in Figs. 21 and 22, which are similar to that of the RNN model.

To conclude, the comparison of classification accuracies of DNN, RNN, and LSTM models is shown in Fig. 23, in which the LSTM model is observed to have the best classification accuracies among the three used models.

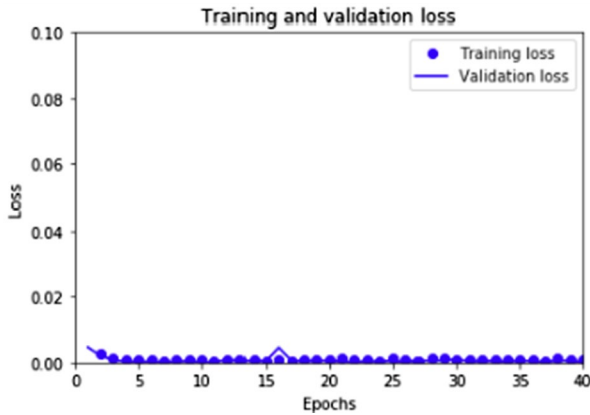


Fig. 19 RNN model's training and validation loss

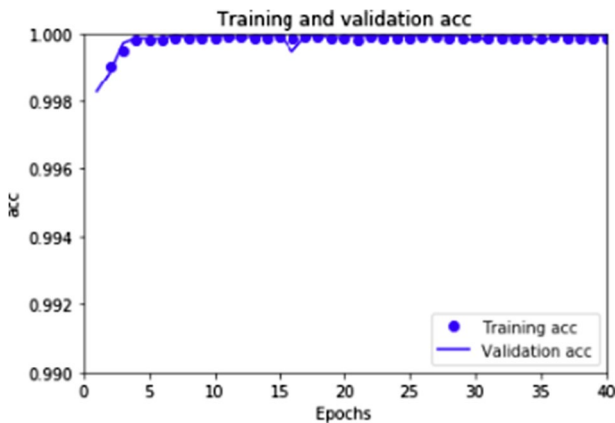


Fig. 20 RNN model's training and validation accuracy

## 5 Conclusions and future work

This study designs and implements a network log management and analysis system based on the ELK Stack. By the use of the proposed system, administrators can easily comprehend the general computer network usages in all the service areas. As a result, they can swiftly respond and make timely adjustments according to the visual analysis of network usages in each area. The used RNN and LSTM models for network cyberattack classification can achieve more than 99.0% in accuracy and are useful for identifying attack types. The log data and the cyberattack behaviors are closely correlated, so that network administrators can immediately acquire accurate information of network usage through the implemented network log management and analysis system.



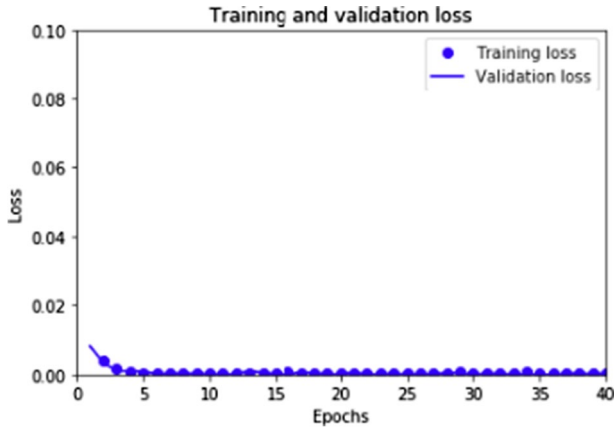


Fig. 21 LSTM model's training and validation loss

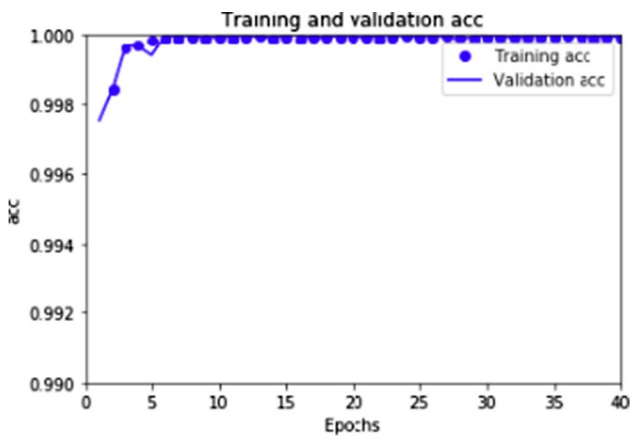


Fig. 22 LSTM model's training and validation accuracy

In the future, we plan to gather more data related to cyberattacks by the ELK Stack log analysis system and employ them for cyberattack classification based on deep learning and visual analysis through Kibana. Network usage can be analyzed to provide appropriate visual analysis of different areas, which will make it easier for network administrators to peruse diverse network log data. By means of deep learning, we hope not only to identify more types of attacks, but also to accurately predict and rapidly defend future cyberattacks.

**Acknowledgements** This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2622-E-029-007-CC3, 109-2625-M-029-001 and 109-2221-E-029-020.

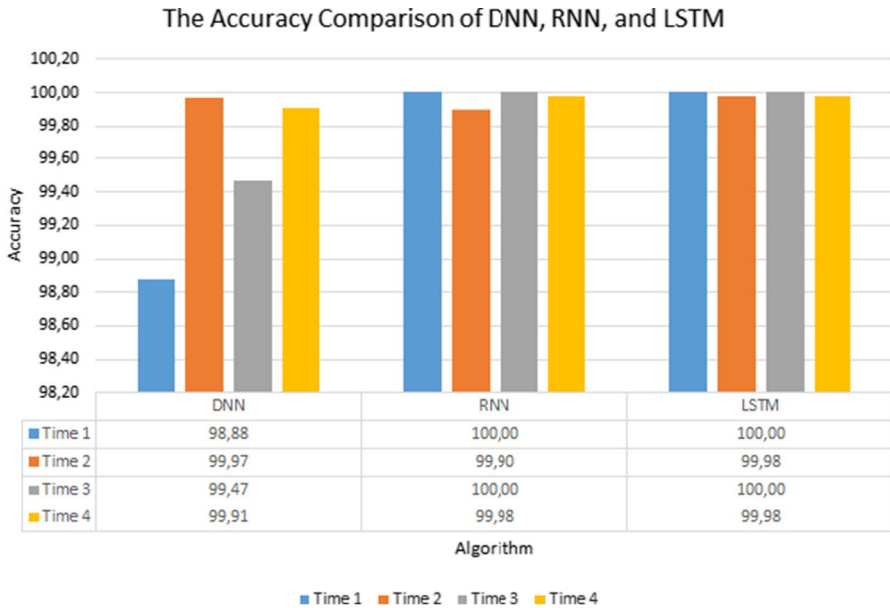



Fig. 23 The accuracy comparison of DNN, RNN, and LSTM

### References

1. Appleyard RH, Adams J (2016) Using the ELK Stack for CASTOR Application Logging at RAL. In: Conference: International Symposium on Grids and Clouds 2015, (p 027). <https://doi.org/10.22323/1.239.0027>
2. Betke E, Kunkel J (2017) Real-Time i/o-Monitoring of Hpc Applications with Siox, Elasticsearch, Grafana and Fuse. In: International Conference on High Performance Computing, Springer, pp 174–186
3. Carela-Español V, Barlet-Ros P, Cabellos-Aparicio A, Solé-Pareta J (2011) Analysis of the impact of sampling on NetFlow traffic classification. *Comput Netw* 55:1083–1099
4. Grafana Labs (2020) The analytics platform for all your metrics. <https://grafana.com/>, online; accessed 20 June 2019
5. Kalech M (2019) Cyberattack detection in SCADA systems using temporal pattern recognition techniques. *Comput Secur* 84:225–238
6. Kim TY, Cho SB (2018) Web traffic anomaly detection using C-LSTM neural networks. *Expert Syst Appl* 106:66–76
7. Kiran M, Chhabra A (2019) Understanding flows in high-speed scientific networks: a netflow data study. *Future Gener Comput Syst* 94:72–79
8. Kozik R (2018) Distributing extreme learning machines with apache spark for netflow-based malware activity detection. *Pattern Recogn Lett* 101:14–20
9. Kozik R, Choraá M, Ficco M, Palmieri F (2018) A scalable distributed machine learning approach for attack detection in edge computing environments. *J Parallel Distrib Comput* 119:18–26
10. Kristiani E, Yang CT, Huang CY, Ko PC, Fathoni H (2020) On construction of sensors, edge, and cloud (ISEC) framework for smart system integration and applications. *IEEE Internet Things J* 8(1):309–319
11. Langi PP, Najib W, Aji TB, et al (2015) An Evaluation of Twitter River and Logstash Performances as Elasticsearch Inputs for Social Media Analysis of Twitter. In: 2015 International Conference on Information & Communication Technology and Systems (ICTS), IEEE, pp 181–186

12. Lee S, Huh JH (2019) An effective security measures for nuclear power plant using big data analysis approach. *J Supercomput* 75(8):4267–4294
13. Liu H, Lang B, Liu M, Yan H (2019) CNN and RNN based payload classification methods for attack detection. *Knowl-Based Syst* 163:332–341
14. Mahmoud MS, Hamdan MM, Baroudi UA (2019) Modeling and control of cyber-physical systems subject to cyber attacks: a survey of recent advances and challenges. *Neurocomputing* 338:101–115
15. Moh M, Pininti S, Doddapaneni S, Moh T (2016) Detecting Web Attacks Using Multi-Stage Log Analysis. In: 2016 IEEE 6th International Conference on Advanced Computing (IACC), pp 733–738, <https://doi.org/10.1109/IACC.2016.141>
16. Navarro J, Deruyver A, Parrend P (2018) A systematic survey on multi-step attack detection. *Comput Secur* 76:214–249
17. Perry I, Li L, Sweet C, Su SH, Cheng FY, Yang SJ, Okutan A (2018) Differentiating and Predicting Cyberattack Behaviors Using Istm. In: 2018 IEEE Conference on Dependable and Secure Computing (DSC), IEEE, pp 1–8
18. Prakash T, Kakkar M, Patel K (2016) Geo-Identification of Web Users Through Logs Using Elk Stack. In: 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence), IEEE, pp 606–610
19. Rastogi R, Akash S, Shobha G, Poonam G, Pratiba D, Singh A (2016) Design and development of generic web based framework for log analysis. In: 2016 IEEE Region 10 Conference (TENCON), IEEE, pp 232–236
20. Sahingoz OK, Buber E, Demir O, Diri B (2019) Machine learning based phishing detection from URLs. *Expert Syst Appl* 117:345–357
21. Sahoo KS, Panda SK, Sahoo S, Sahoo B, Dash R (2019) Toward secure software-defined networks against distributed denial of service attack. *J Supercomput* 75(8):4829–4874
22. Taylor A, Leblanc S, Japkowicz N (2018) Probing the limits of anomaly detectors for automobiles with a cyberattack framework. *IEEE Intell Syst* 33(2):54–62
23. Tsung CK, Hsieh HY, Yang CT (2019) An implementation of scalable high throughput data platform for logging semiconductor testing results. *IEEE Access* 7:26,497–26,506
24. Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S (2019) Deep learning approach for intelligent intrusion detection system. *IEEE Access* 7:41525–41550
25. Wang CY, Ou CL, Zhang YE, Cho FM, Chen PH, Chang JB, Shieh CK (2018) BotCluster: a session-based P2P botnet clustering system on NetFlow. *Comput Netw* 145:175–189
26. Wu P, Lu Z, Zhou Q, Lei Z, Li X, Qiu M, Hung PC (2019) Bigdata logs analysis based on seq2seq networks for cognitive internet of things. *Future Gener Comput Syst* 90:477–488
27. Xue Q, Chuah MC (2018) New attacks on RNN based healthcare learning system and their detections. *Smart Health* 9–10:144–157
28. Yang CT, Chen ST, Cheng WH, Chan YW, Kristiani E (2019a) A heterogeneous cloud storage platform with uniform data distribution by software-defined storage technologies. *IEEE Access* 7:147,672–147,682
29. Yang CT, Chen ST, Liu JC, Yang YY, Mitra K, Ranjan R (2019b) Implementation of a real-time network traffic monitoring service with network functions virtualization. *Future Gener Comput Syst* 93:687–701
30. Yang CT, Jiang WJ, Kristiani E, Chan YW, Liu JC (2019c) The Implementation of a Network Log System Using Rnn on Cyberattack Detection with Data Visualization. In: International Conference on Frontier Computing, Springer, pp 321–329
31. Yang CT, Kristiani E, Wang YT, Min G, Lai CH, Jiang WJ (2020a) On construction of a network log management system using ELK Stack with Ceph. *J Supercomput* 76:6344–6360
32. Yang CT, Liu JC, Kristiani E, Liu ML, You I, Pau G (2020b) Netflow monitoring and cyberattack detection using deep learning with Ceph. *IEEE Access* 8:7842–7850
33. Yang Y, Zheng K, Wu C, Yang Y (2019d) Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. *Sensors* 19(11):2528

## Authors and Affiliations

Jung-Chun Liu<sup>1</sup> · Chao-Tung Yang<sup>1,2,3</sup>  · Yu-Wei Chan<sup>4</sup> · Endah Kristiani<sup>5,6</sup> · Wei-Je Jiang<sup>1</sup>

Jung-Chun Liu  
jcliu@thu.edu.tw

Yu-Wei Chan  
ywchan@gm.pu.edu.tw

Endah Kristiani  
endahkristi@gmail.com

Wei-Je Jiang  
s22775605@gmail.com

- <sup>1</sup> Department of Computer Science, Tunghai University, Taichung City 407224, Taiwan, R.O.C.
- <sup>2</sup> Research Center for Smart Sustainable Circular Economy, Tunghai University, No. 1727, Sec.4, Taiwan Boulevard, Taichung City 407224, Taiwan, R.O.C.
- <sup>3</sup> Research Center for Nanotechnology, Tunghai University, No. 1727, Sec.4, Taiwan Boulevard, Taichung City 407224, Taiwan, R.O.C.
- <sup>4</sup> College of Computing and Informatics, Providence University, Taichung City 43301, Taiwan, R.O.C.
- <sup>5</sup> Department of Industrial Engineering and Enterprise Information, Tunghai University, Taichung City 407224, Taiwan, R.O.C.
- <sup>6</sup> Department of Informatics, Krida Wacana Christian University, Jakarta 11470, Indonesia