Artificial Neural Networks Based Detection of Cyber Threats or Event Profiles

S. SURYA PRAKASH[1], G.V.RAMMOHAN[2]

#1. M.Tech Scholar in Department of Artificial Intelligence & Data Science,
#2. Assistant Professor, Department Of Computer Science and Engineering, Kakinada Institute Of Engineering & Technology, AP, India.

ABSTRACT *One of the major challenges in cybersecurity is the provision of an automated and effective cyber-threats detection technique. In this paper, we present an AI technique for cyber-threats detection, based on artificial neural networks. The proposed technique converts multitude of collected security events to individual event profiles and use a deep learning-based detection method for enhanced cyber-threat detection. For this work, we developed an AI-SIEM system based on a combination of event profiling for data preprocessing and different artificial neural network methods, including FCNN, CNN, and LSTM. The system focuses on discriminating between true positive and false positive alerts, thus helping security analysts to rapidly respond to cyber threats. All experiments in this study are performed by authors using two benchmark datasets (NSLKDD and CICIDS2017) and two datasets collected in the real world. To evaluate the performance comparison with existing methods, we conducted experiments using the five conventional machine learning methods (SVM, k-NN, RF, NB, and DT). Consequently, the experimental results of this study ensure that our proposed methods are capable of being employed as learning-based models for network intrusion detection, and show that although it is employed in the real world, the performance outperforms the conventional machine- learning methods.*

I. INTRODUCTION

With the emergence of artificial intelligence (AI) techniques, learning-based approaches for detecting cyber attacks, have become further improved, and they have achieved significant results in many studies. However, owing to constantly evolving cyber attacks, it is still highly challenging to protect IT systems against threats and malicious behaviors in net- works. Because of various network intrusions and malicious activities, effective defenses and security considerations were given high priority for finding reliable solutions [1]–[4].

Traditionally, there are two primary systems for detecting cyber-threats and network intrusions. An intrusion prevention system (IPS) is installed in the enterprise network, and can examine the network protocols and flows with signature- based methods primarily. It generates appropriate intrusion alerts, called the security events, and reports the generating alerts to another system, such as SIEM. The security information and event management (SIEM) has been focusing on collecting and managing the alerts of IPSs. The SIEM is the most common and dependable solution among various security operations solutions to analyze the collected security events and logs [5]. Moreover, security analysts make an effort to investigate suspicious alerts by policies and threshold, and to discover malicious behavior by analyzing correlations among events, using knowledge related to attacks.

Nevertheless, it is still difficult to recognize and detect intrusions against intelligent network attacks owing to their high false alerts and the huge amount of security data [6], [7]. Hence, the most recent studies in the field of intrusion detection have given increased focus to machine learning and artificial intelligence techniques for detecting attacks. Advancement in AI fields can facilitate the investigation of network intrusions by security analysts inautomated manner. These learning-based approaches require to learn the attack model from historical threat data and use the trained models to detect intrusions for unknown cyber threats [8], [9].

AI-SIEM system which is able to discriminate between true alerts and false alerts based on deep learning techniques. Our proposed system can help security analysts rapidly to respond cyber threats, dispersed across a large amount of security events.

For this, the proposed the AI-SIEM system particularly includes an event pattern extraction method by aggregating together events with a concurrency feature and correlating between event sets in collected data. Our event profiles have the potential to provide concise input data for various deep neural networks. Moreover, it enables the analyst to handle all the data promptly and efficiently by comparison with long- term history data. The main contributions of our work can be summarized as follows:

Our proposed system aims at converting a large amount of security events to individual event profiles for processing very large scale data. We developed a generalizable security event analysis method by learning normal and threat patterns from a large amount of collected data, considering the frequency of their occurrence. In this study, we specially propose the method to characterize the data sets using the base points in data preprocessing step. This method can significantly reduce the dimensionality space, which is often the main challenge associated with traditional data mining techniques in log analysis.

- Our event profiling method for applying

artificial intelligence techniques, unlike typical sequencebased pat- tern approaches, provides featured input data to employ various deep-learning techniques. Hence, because our technique is able to facilitate improved classification for true alerts when compared with conventional machine- learning methods, it can remarkably reduce the number of alerts practically provided to the analysts.

- For the applicability, we evaluate our system with real IPS security events from a real security operations center (SOC) and validate its effectiveness through performance metrics, such as the accuracy, true positive rate (TPR), false positive rate (FPR) and the Fmeasure. Moreover, to evaluate the performance comparison with existing methods, we conducted experiments using the five conventional machine-learning methods (SVM, k-NN, RF, NB and DT). And we also perform an evaluation by applying our method to two benchmark datasets (i.e., NSLKDD, CICIDS2017), which are most commonly used in the field of network intrusion detection research.

In this study, to decompose a large amount of collecting events into individual event occurrence profiles, we apply the TF-IDF mechanism. We also generate the event profiles by computing the similarity value among each TF-IDF event sets and appointed base points. The generated event profiles are fed into the input-layer of the FCNN, CNN, and LSTM models, which are executed in AI-SIEM. Consequently, using two well-known benchmark datasets and two real datasets collected from operating IPS, we aim to show the applicabil ity of our system for defending IT systems against the cyber threats.

For evaluation, we are aware of the limitation of NSLKDD and CICIDS 2017 datasets, but they remain widely used benchmarks for comparing machine-learning methodologies. Hence, we also conduct a performance comparison with existing methods using the real datasets and additional two benchmark datasets. Above all, machine-learning approaches obtained a good performance using benchmark datasets ,also need to achieve satisfactory performance for the real data. The remainder of this paper is structured as follows. In Section II, we introduce the background information for the proposed system. Section III provides existing works on learning-based intrusion or attack detection. In Section IV, we describe the overview for our proposed system and data labeling. In section V, we specify the methodology used in this study in more detail. Section VI provides the implementation of the FCNN, CNN, and LSTM models for this study. Section VII introduces datasets for experiments. Section VIII presents the detailed evaluation results of experiments and comparison with other methods. Finally, the conclusion and future work discussed in Section IX

II. RELATED WORK

In this section, we discuss previous studies for deep learning- based intrusion detection and real security event analysis research. In recent years, many studies in cybersecurity focus on AI-based intrusion detection, and different AI and machine learningbased techniques have been proposed to improve the ability of cyber threat detection [1]–[3], [15], [38], [39]. Although these studies have achieved significant result using AI and machine learningbased techniques, they are still limited to specific test datasets such as NSLKDD. Other research studies however, have used security events and logs collected from the real world [8], [10], [40]–[42]. These studies are closer to our study for addressing the above-mentioned challenges. Especially, Du *et al.* [39], Liao and Vemuri [40], and Zhang *et al.* [42] have used the TF-IDF mechanism like our method.

A. DEEP LEARNING-BASED INTRUSION DETECTION Naseer *et al.* [1] proposed, implemented and trained intrusion detection models using different deep neural network architectures including CNNs, Auto encoders, and RNNs. These models were trained on the NSLKDD training dataset and evaluated on both test datasets provided by NSLKDD. DCNN and LSTM models showed a performance of 85% and 89% accuracy, respectively, on test dataset.

Zhang *et al.* [2] divided methods for network intrusion detection into two types: direct methods using single algorithm and combination method by combination of several methods. The author proposed a new detection model based on a directed acyclic graph (DAG) and a belief rule base (BRB). The results showed that compared with conventional detection models, the DAG-BRB combination model had a higher detection rate using KDD 99 dataset.

Wang *et al.* [3] proposed a hierarchical spatial and temporal features-based intrusion detection system (HAST-IDS) that automatically learns network traffic features. The main idea is that the spatial features of network traffic are first learned using deep CNNs and then learns the temporal features are learned LSTM networks. The experiments were conducted by DARPA and ISCX datasets.

Vinaya kumar *et al.* [15] developed a hybrid intrusion detection system which has the capability to analyze the net- work and host-level activities. It employed distributed deep learning model with DNN for processing and analyzing very large scale data in real-time. The DNN model was selected by comprehensively evaluating their performance in comparison to classical machine learning classifiers on various bench- mark IDS datasets such as NSLKDD and UNSW-NB15.

Khan *et al.* [38] propose a novel two-stage deep learning model, based on a stacked auto-encoder with a soft-max classifier, for efficient network intrusion detection. The authors conducted several experiments on two public datasets: the benchmark KDD99 and UNSWNB15 datasets. This study achieved results, up to 99.9% for the KDD99 dataset and 89.1% for the UNSW-NB15 dataset.

Du et al. [39] proposed a new algorithm based on the kNN classifier method using TF-IDF for modeling program behavior in intrusion detection regarding system calls. In [29], with the k-NN classifier, the frequencies of system calls are used to describe the program behavior. For this, text categorization techniques, such as TF-IDF, are adopted to transform each system call data to a vector and measure the similarity between two program system call activities. Authors report that the TF-IDF-based kNN classifier appears to be well applicable to the domain of intrusion detection in the field of malware detection.

B. REAL SECURITY EVENT ANALYSIS Shen *et al.* [8] developed the system for predicting security events through deep learning, which is called Tiresias. Authors presented a system that leverages RNNs to predict future events on a machine, based on previous observations. It tested on a dataset of 3.4 billion security events collected from a commercial IPS, and showed that its approach is effective in predicting the next event that will occur on a machine with a precision of up to 0.93. In addition, the system also accomplished a high precision for a complex situation and maintained stable results.

Veeramachaneni *et al.* [10] developed end-to-end machine learning techniques that predict cyber attacks significantly better than existing systems by continuously incorporating input from human experts. The analyst directly labeled data with a ranked metric over several months, and these labeled data were provided to the supervised learning module to predict whether an attack would occur. This study showed that the technique, using six anomaly detection methods, can detect 85 percent attacks, which is roughly three times better than previous benchmarks, while also reducing the number of false positives by a factor of 5. The system was tested on

3.6 billion pieces of data known as ‘‘log lines,’’ which were generated by millions of users over a period of three months. Specially, the hybrid approaches of auto-encoders have been recently proposed for anomaly detection.

Liao and Vemuri [40] proposed DeepLog, a deep neural network model employing LSTM to train a system’s log patterns (e.g., log key patterns and corresponding parameter value patterns) from normal execution. This work uses the term frequency inverse

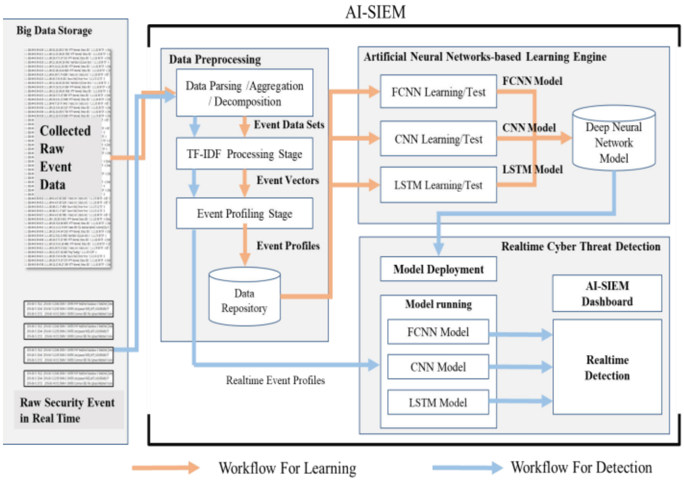

FIGURE. The workflow and architecture for the developed AI-based SIEM system.

engines. It also utilizes the processing capability of the several graphical processing unit (GPU) cores for faster and parallel analysis.

Figure 2 presents the workflow and architecture for the developed artificial intelligent (AI)-based SIEM system. The AI-SIEM system comprises three main phases: The data pre- processing, artificial neural networks-based learning engine, and real-time threat detection phase. The first preprocessing phase in the system, termed event profiling, aims at providing concise inputs for various deep neural networks by transforming raw data. In the data preprocessing phase, data aggregation with parsing, data normalization stage using TF-IDF mechanism, and event profiling stage are consecutively performed in the AISIEM system. Each stage generates event data sets, event vectors, and event profiles, respectively, and the output is utilized in next each stage, as shown in Figure 2. This phase not only precedes the data learning stage but also precedes the conversion of raw security events to the deep-learning engine's input data when the system operates on detecting network intrusions in real time. The second AI-based learning engine employs three artificial neural networks for modeling. For the data learning stage, the preprocessed data are fed into the three artificial neural networks, and each ANN performs learning to find the most accurate model. Finally, in real-time threat detection, each ANN model mechanically classifies each security raw event using the trained model, and the dashboard shows the only recognized true alerts to security analysts for reducing false ones.

Each stage for data preprocessing is detailed in Section V, and second ANNs for data learning phase are described in Section VI.

*A. **DATA LABELING FOR LEARNING***

In this subsection, we discuss the data labeling of security events for supervised learning. As mentioned above, to employ the supervised learning method, a labeled data is essential. For this, analysts should be able to label several months of data heuristically. In other words, analysts need to label the raw events as ''Normal'' or as ''Threat,'' based on whether it belongs to a type of attack by analyzing correlations among raw security events. However, owing to a rapidly growing number of security events and unknown cyber threats, the labeling of numerous data is time- consuming and costly. In addition, it is difficult to acquire the labeled security event dataset based on the action of SOC security experts in the real world. By investigating occurred cyber attacks, most of detected attacks can be categorized as system hacking, denial of service, network attacks, scanning attacks, and suspicious authentication activities. These attack types are determined by the SOC security analysts based on correlation among attack duration time, the number of attacker's IP, and importance of victim system.

In our study, to provide an available dataset for supervised learning, we had to carry out dataset labeling according to utilizing recorded information in the threat detection report list (e.g., attack start time, attack end time, and attacker's ip address information). The threat detection reports are made by the SOC analysts during raw data collecting periods. The labeling operation is automatically performed by the data labeling module in our system. First, the system extracts timestamps and network information from the threat detection report, for each recorded threat detection result. Next, the data labeling tool in the system, investigates correlation of extracted threat information on raw security event, with each threat using the big data platform. The security events that are correlated with IP address and time of each threat are labeled as ''THREAT (Attack name),'' and others are labeled as ''NORMAL.'' The labeled result of our collected datasets is explained in Section VII.

III. METHODOLOGY

In this section, we describe an event profiling method for preprocessing. The method is composed of data aggregation and decomposition, TF-IDF normalization, and generating event profile. we first present an event set extraction method for the data preprocessing. Then, the event vectorization using TF-IDF for event profiles is described in detail. Finally, we present the event profiling method for inputs into three deep learning models. The proposed method was basically motivated by the observation that raw event data can be profiled by concurrent event sets. By combining each pro- posed method sequentially, the preprocessing for AI engine is operated as shown in Figure 2:

DATA AGGRERATION AND DECOMPOSITION

The sequence may be changed in IPS by system processes, resources, and network,; therefore we adopt the concurrency- based method that depends on cooccurrence information, which is not as tight as the sequence, but allows the calibration of the gap of changeable sequence.

After performing repetitive testing, we adopted a multi- layer perceptron (MLP) model with eleven layers comprising one input layer, nine hidden layers, and an output layer. In particular, we built a suitable architecture that has one input layer and, nine hidden layers that had 1650, 1850, 2048, 1792, 1536, 1280, 1024, 768, and 512 nodes, respectively. We composed the activation functions using the leaky rectified linear unit (leaky ReLU) scheme as the activation function, instead of Sigmoid. The softmax function with a cross entropy cost function at the output layer, produces the final outputs, as shown in Figure 4. The softmax layer, which is composed of a cross-entropy cost function at the output layer, produces the final multiple outputs.

To train our FCNN, the preprocessed data were fed to the FCNN, and training was performed by tuning the parameter configuration to over 1000 epochs with a learning rate of

0.001. The implemented FCNN diagram is shown in Figure layers, and an output layer with one fully connected layer. Each of the front three convolutional layers in CNN was followed by max pooling layers for subsampling. We placed the dropout layer at the front of each convolutional layer except for the last.

The input layer in the implemented CNN is dynamically shaped. Because the CNN is generally specialized for 2D or 3D pixel data of the processing image, we need to transform each pre-processed event profile row into a2D array. Hence, we transform each element of the input data vector into an N ×N 2D array form, where empty positions in the 2D array are replaced with zero. Each input layer can then be variously shaped by the size of defined features for learning based on CNN. The implemented architecture for CNN is described in Figure 5, and the depicted CNN can be used to learn the data where the features ranging from number of features is 169-196.

B. LSTM MODEL

An LSTM has a special recurrent architecture designed to advance the storage ability.

Figure 5 presents the constructed architecture of the recur- rent neural network in our deep learning model. An input layer's vector sequence x = {$x_{t\,L\,1}$, $x_{t\,L}$, . . . x_t 1, x_t with length L is passed wi}th weighted connections to a layer of multiple recurrently connected hidden layers to compute first the hidden
layer's vector sequences h = $h_{t-L}+_1$, h_{t-L} , . . . h_{t-1} , h_t , and then the output vector sequence y = {$y_{t-L}+_1$, y_{t-L} , . . . y_{t-1} , y_t .} In common LSTM, each output vector y_t is used to parameterize the probability distribution $Pr(x_t +_1 | y_t)$ of the next inputs x_{t+1} [42], [44].

Given the temporal dependencies between the event profiles, in this work, we employ LSTM to model the temporal correlations of event profiles. An RNN is a connectivity

A. CNN MODEL

CNNs are neural network architectures especially designed to deal with spatial data. For CNN, the data of input layer consists of 2D or 3D array such as the pixel value of the image information. The core layers of CNN are convolutional layers (Conv) and max pooling layers. A Conv layer receives input as a unit and convolves it using filters to produce an ongoing data to transfer into next layers.

In a Conv layer, the filters read overall inputted data by the slicing and extract the key features. In addition, convolution is performed by calculating the scalar product between the input chunk and each filter. The features that are extracted by each filter are aggregated to a new feature set, which is called the feature map. Because the convolutional layer consists of a group of filters, it produces a feature map for each filter, and the data of feature maps are aggregated together to generate data for output [8], [22].

A. CNN MODEL

CNNs are neural network architectures especially designed to deal with spatial data. For CNN, the data of input layer consists of 2D or 3D array such as the pixel value of the image information. The core layers of CNN are convolutional layers (Conv) and max pooling layers. A Conv layer receives input as a unit and convolves it using filters to produce an ongoing data to transfer into next layers.

In a Conv layer, the filters read overall inputted data by the slicing and extract the key features. In addition, convolution is performed by calculating the scalar product between the input chunk and each filter. The features that are extracted by each filter are aggregated to a new feature set, which is called the feature map. Because the convolutional layer consists of a group of filters, it produces a feature map for each filter, and the data of feature maps are aggregated together to generate data for output [8], [22].

The designed and implemented CNN was comprised an input layer, four convolutional layers, three max pooling using a recurrence formula of the form h_t $f_\theta(h=_t\ _1, x_t)_-$, where f , an activation function and θ , a parameter, are use dat each timestamp to process. To avoid the vanishing gradient problems with RNNs, gradient clipping and gating concepts are introduced [33].

An LSTM is an upgraded network of RNN. Unlike classical RNNs, LSTM tries to address the problem of long-term dependencies by introducing a purpose-built memory cell to store information of previous time steps [42].

Within this model, instead of propagating the state without multiplicative updates at each step, it is stored in memory cell C_t , which receives additive updates, merged with a method for removing irrelevant inputs from the memory cell of previous time steps [45]. Following the notation in Shin *et al.* [45], Zaremba *et al.* [46] the computation of LSTM unit at time step *t* is formally represented as follows:

In our study, we constructed LSTM with 1–8 multi-layers and N hidden layers; an example of the architecture is shown in Figure 6. It must be noted that if there is one multi-layer, the neural network is an RNN. The RNN cell and LSTM cell can be easily substituted for each other because they both support in TensorFlow. To construct a suitable LSTM network with the optimal number of multi-layers and hidden layer, we used several dynamic configurations until the best performance was obtained. Consequently, we observed that the optimal number of multi-layers is 2–4 and optimal the number of hidden layers is 256pattern that c₋o512mpu. Although the multites on a sequence of- vlayersectors are*x*1, *x*deeper,2· ,· this accuracy is not considerably advanced. However, a longer training period is required. Moreover, because our proposed AI-SIEM system can model the LSTM through dynamic configuration, the optimal LSTM network

V. DATASETS

This section describes the datasets. The four datasets used for testing, are NSLKDD, CICIDS 2017, and the two real datasets collected in the SOC.

A. NSLKDD

The NSLKDD dataset is the new revised version of the KDDCUP99. Tavallaee *et al*. [47] had discovered a number of duplicated records in the original KDDCUP99 dataset, which had an impact on the performance of model training and evaluation on the dataset. NSLKDD is a refined version of the dataset to address discovered statistical problems.

VI EXPERIMENTS AND RESULTS

In this section, we report the experimental results are per- formed with the two benchmark datasets and our two collected real datasets. We start by describing test environment with test bed. We then present the metric for experiment. Continually, we present the SVD and conventional machine- learning methods for various comparison of evaluation the

performance. we discuss the experimental results in sub- section E, and finally we present the implemented system by attacks classified, where TP (True Positive) is the number of attack data that is correctly classified as an attack, and FP (False Positive) is the number of normal data that is incorrectly classified as an attack. TN (True Negative) the number of normal data that is correctly classified as normal, and FN (False Negative) is the number of attack data that is incorrectly classified as normal. The definitions for accuracy, TPR, FPR, and Fmeasure are presented below:

The t-SNE is not only commonly utilized for vector data visualization but also considered as embedding tools to visualize high-dimensional data. The t-SNE is able to visualize high-dimensional data into two-dimensional maps by learning two-dimensional embedding vectors that preserves neighbor structures among high-dimensional data. The N data rows in dataset are randomly selected, which are visualized by performing analysis in t-SNE [3], [49]. Figure 7 and Figure 8 represent the maps that are visualized by t-SNE for CICIDS 2017 and ESX-2, respectively. The t-SNE plots in the figure show that the normal and attack data points located nearby in the same space, which makes it very hard to classify them into either normal or attack. Although the t- SNE plots of normal and attack data are clustered, it clearly finds out that those are not linearly separated. In general, it is known that deep learning is then effective at dealing with highdimensional data with non-linearity [50], which is one of the reasons we employ deep learning approaches to detect cyber threats.

In addition, as shown in Figure 7 and Figure 8, the data distribution visually seen by t-SNE regarding our dataset means that the dataset is not to be easily categorization in comparison with the benchmark datasets.

After minor data filtering, we constructed the dataset using collected data for performance evaluations as described in the previous section. In general, the format of security event of IPS/IDS is different between devices or vendors, but majority of events always contain timestamp, source ip address, destination ip address, port information, protocol, flow information, and rule names. When these security events are stored in conventional SIEM, they are stored in a standardized format with minor additions such as data tagging and data enrichment. Because the collected ESX-1, ESX-2 is a set of several types of IPS / IDS data stored through this process, it can be considered that it is sufficiently applicable to other SIEM and SOC.

For real environments when we conduct the test, we implemented a sensor emulator that can substitute for a real IPS system. It uses the syslog protocol to send to the AI-SIEM system, by reading security event dataset and synthetically generating syslog packets

Our proposed EP-ANN in AI-SIEM was implemented using TensorFlow [51]. The hardware used to evaluate the performance of the EP-ANN methods are clusters of server with Intel Xeon with 2.5 GHz (32 CPU cores) and 128GB memory. Two Nvidia Tesla P100 GPUs are used as the accelerator.

A. METRICS AND EXPERIMENTAL SETUP

1) FOUR meTRICS

To evaluate the performance, four metrics are adopted: accu- racy, TPR, FPR, and F-measure, which are all commonly used for learning-based methods in the field of intrusion detection. TPR is used to evaluate the system's performance with respect to its threat detection. FPR is used to evaluate misclassifications of normal data. F-measure is the harmonic mean of the precision and TPR(recall), where Precision = TP / (TP+FP) is the percentage of true attacks among all

In order to evaluate the quality of detection performance, we show a receiver operating characteristic (ROC) curve and measure an area under curve (AUC) value as significant comparison metrics.

ROC curve is a plot of FPR against TPR of binary classifiers. FPR corresponds to the proportion of normal data points incorrectly predicted as attack to all normal data points. TPR, also called sensitivity or recall, corresponds to the proportion of attack data points that are correctly predicted attack to all attack data points. ROC curve shows a trade-off between sensitivity and FPR. The closer the ROC curve is to the top- left border, the better the quality of predictions by the prediction model and vice versa [1]. Additionally, AUC is defined as area under the ROC curve, which is a measure of how well a binary classifier can perform predictions of labels.

B. COMPARISON WITH SVD

As singular value decomposition (SVD) is the one of the most commonly used methods for dimensionality reduction in machine learning, we compare the performance of our method with SVD.

SVD is the method to diagonalize a matrix as in eigenvalue decomposition. Note that eigenvalue decomposition by eigenvalues and eigenvectors is applicable only to square matrices, and is also a diagonalization method applicable only to some square matrices [52]. Whereas, SVD is useful because the

technique is applicable to all m× n matrices whether they are square matrices or not. SVD for an m × n matrix in real space is defined as follows:

$$A = U \times \Sigma \times V^T \qquad (17)$$

where *U* is an *m-by-m* orthonormal matrix, *V* is an nby-n orthonormal matrix and *Σ* is an *m-by-n* diagonal matrix. Here, an orthogonal matrix is a matrix in which the result of

VII CONCLUSION

In this paper, we have proposed the AI-SIEM system using event profiles and artificial neural networks. The novelty of our work lies in condensing very large-scale data into event profiles and using the deep learning-based detection methods for enhanced cyber-threat detection ability. The AI-SIEM system enables the security analysts to deal with significant security alerts promptly and efficiently by comparing long- term security data. By reducing false positive alerts, it can also help the security analysts to rapidly respond to cyber threats dispersed across a large number of security events.

For the evaluation of performance, we performed a performance comparison using two benchmark datasets (NSLKDD, CICIDS2017) and two datasets collected in the real world. First, based on the comparison experiment with other methods, using widely known benchmark datasets, we showed that our mechanisms can be applied as one of the learning-based models for network intrusion detection. Second, through the evaluation using two real datasets, we presented promising results that our technology also outperformed conventional machine learning methods in terms of accurate classifications. In the future, to address the evolving problem of cyber attacks, we will focus on enhancing earlier threat predictions through the multiple deep learning approach to discovering the long-term patterns in history data. In addition, to improve the precision of labeled dataset for supervised-learning and construct good learning datasets, many SOC analysts will make efforts directly to record labels of raw security events one by one over several months.

REFERENCES

[1]. S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, ''Enhanced network anomaly detection based on deep neural networks,'' IEEE Access, vol. 6, pp. 48231–48246, 2018.

[2]. B.-C. Zhang, G.-Y. Hu, Z.-J. Zhou, Y.-M. Zhang, P.-L. Qiao, and L.-

[3]. L. Chang, ''Network intrusion detection based on directed acyclic graphand belief rule base,'' Electron. Telecommun. Res. Inst. J., vol. 39, no. 4, pp. 592–604, Aug. 2017.

[4]. W. Wang, Y. Sheng, and J. Wang, ''HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection,'' IEEE Access, vol. 6, pp. 1792–1806, 2018.

[5]. M. K. Hussein, N. Bin Zainal, and A. N. Jaber, ''Data security analysis for DDoS defense of cloud based networks,'' in Proc. IEEE Student Conf. Res. Develop. (SCOReD), Kuala Lumpur, Malaysia, Dec. 2015, pp. 305–310.

[6]. S. S. Sekharan and K. Kandasamy, ‘‘Profiling SIEM tools and correlation engines for security analytics,’’ in Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET), Mar. 2017, pp. 717–721.

[7]. N. Hubballi and V. Suryanarayanan, ‘‘False alarm minimization techniques in signature-based intrusion detection systems: A survey,’’ Comput. Com- mun., vol. 49, p. 1Â17, Aug. 2014.

[8]. Naser, M. A. Majid, M. F. Zolkipli, and S. Anwar, ‘‘Trusting cloud computing for personal files,’’ in Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Busan, South Korea, Oct. 2014, pp. 488–489.

[9]. Y. Shen, E. Mariconti, P. A. Vervier, and G. Stringhini, ‘‘Tiresias: Predict- ing security events through deep learning,’’ in Proc. ACM CCS, Toronto, ON, Canada, Oct. 2018, pp. 592–605.

[10]. K. Soska and N. Christin, ‘‘Automatically detecting vulnerable Websites before they turn malicious,’’ in Proc. USENIX Secur. Symp., San Diego, CA, USA, 2014, pp. 625–640.

[11]. K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li, ‘‘AI^2: Training a big data machine to defend,’’ in Proc. IEEE BigDataSecurity HPSC IDS, New York, NY, USA, Apr. 2016, pp. 49–54.

[12]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, ‘‘A detailed analysis of the KDD cup 99 data set,’’ in Proc. 2nd IEEE Symp. Comput. Intell. Secur. Defense Appl., Jul. 2009, pp. 53–58.

[13]. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ‘‘Toward generating a new intrusion detection dataset and intrusion traffic characterization,’’ in Proc. Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), Jan. 2018, pp. 108–116.

[14]. Song, H. Takakura, and Y. Okabe. (2006). Description of Kyoto University Benchmark Data. [Online]. Available: http://www.

[15]. takakura.com/Kyoto_data/BenchmarkData-Descriptionv5.pdf

[16]. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, ‘‘A deep learning approach to network intrusion detection,’’ IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, Feb. 2018.

[17]. R. Vinayakumar, M. Alazab, K. Soman, P.

[18]. Poornachandran, A. Al-Nemrat, and S. Venkatraman, ‘‘Deep learning approach for intelligent intrusion detection system,’’ IEEE Access, vol. 7, pp. 41525– 41550, 2019.

[19]. W. Hu, W. Hu, and S. Maybank, ‘‘AdaBoost-based algorithm for network intrusion detection,’’ IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 38, no. 2, pp. 577– 583, Apr. 2008.

[20]. T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W.

[21]. Robertson, A. Juels, and E. Kirda, ‘‘Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks,’’ in Proc. 29th Annu. Comput. Secur. Appl. Conf., New York, NY, USA, Dec. 2013, pp. 199–208.

[22]. K.-O. Detken, T. Rix, C. Kleiner, B. Hellmann, and L. Renners, ‘‘SIEM approach for a higher level of it security in enterprise networks,’’ in Proc. IDAACS, Warsaw, Poland, Sep. 2015, pp. 322–327.

[23]. (2016). Security Information and Event Management.

[24]. [Online]. Available: https://en.wikipedia.org/wiki/Security-_information_and _event_management