IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 7, 2021, accepted May 14, 2021, date of publication May 20, 2021, date of current version June 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3082160

Host-Based Intrusion Detection Model Using Siamese Network

DAEKYEONG PARK[1], SANGSOO KIM[2], HYUKJIN KWON[3], DONGIL SHIN[1], AND DONGKYOO SHIN[1]

[1]Department of Computer Engineering and Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea
[2]Agency for Defense Development, Daejeon 305600, South Korea
[3]Center for Military Analysis and Planning, Korea Institute for Defense Analyses (KIDA), Seoul 02455, Republic of Korea

Corresponding author: Dongkyoo Shin (shindk@sejong.ac.kr)

This work was supported by the Defense Acquisition Program Administration and the Agency for Defense Development Institute under Grant UD200014ED.

ABSTRACT As cyberattacks become more intelligent, the difficulty increases for traditional intrusion detection systems to detect advanced attacks that deviate from previously stored patterns. To solve this problem, a deep learning-based intrusion detection system model has emerged that analyzes intelligent attack patterns through data learning. However, deep learning models have the disadvantage of having to re-learn each time a new cyberattack method emerges. The time required to learn a large amount of data is not efficient. In this paper, an experiment was conducted using the Leipzig Intrusion Detection Data Set (LID-DS), which is a host-based intrusion detection data set released in 2018. In addition, in order to evaluate and improve the performance of the system, a host-based intrusion detection model consisting of pre-processing, vector-to-image processing, training and testing steps is proposed. In the training and testing steps, a Siamese Convolutional Neural Network (Siamese-CNN) is constructed using the few-shot learning method, which shows excellent performance by learning a small amount of data. Siamese-CNN determines whether the attack type is the same based on the similarity score of each cyberattack sample converted to an image. The accuracy was calculated using the few-shot learning technique. The performance of the Vanilla Convolutional Neural Network (Vanilla-CNN) and Siamese-CNN are compared to confirm the performance of Siamese-CNN. As a result of measuring the accuracy, precision, recall, and F1-score indicators, it was confirmed that the recall of the Siamese-CNN model proposed in this study increased by about 6% compared to the Vanilla-CNN model.

INDEX TERMS Machine learning, LID-DS, few-shot learning, siamese network, HIDS.

I. INTRODUCTION

Currently, as cyberattacks become more intelligent, attackers exploit unknown vulnerabilities and become intelligently diversified. Defending against a variety of advanced attacks is an important issue. One commonly used solution to the problem is an intrusion detection system. Intrusion detection systems can be roughly divided into network-based intrusion detection system (NIDS) and host-based intrusion detection system (HIDS). Unlike network-based intrusion detection systems, host-based intrusion detection systems have the disadvantage of having to observe both inside and outside the system. However, a lot of research is needed because it has the advantage of enabling intrusion detection that cannot be detected with a network-based intrusion detection system. In addition, there are two types of intrusion detection systems: misuse detection and anomaly detection [1]. Misuse detection is a method of ensuring that user and system or program behavior match an attack pattern based on known signatures. Anomaly detection is a method of detecting anomalous behavior based on a normal pattern, unlike the misuse detection method. Misuse detection weakness is its incapacity in detecting new unknown attacks, while anomaly detection is able to detect new unknown attacks. However, anomaly detection weakness is summarized by the difficulty to define various normal usage patterns. The false alarm rate increases because normal patterns that have not been learned are regarded as abnormal behavior [2]. With the recent development of deep learning technology, a lot of research is being conducted in the field of information and communication technology (ICT) and the field of Internet of Things (IoT). Various intelligent services have been created.

The associate editor coordinating the review of this manuscript and approving it for publication was Vicente Alarcon-Aquino.

With such advancement in technology, there are cases where deep learning technology is applied to intrusion detection systems in the security field. Deep learning is a technology that can compensate for the weaknesses in both types of detection systems by learning its own functions through deep neural networks. In other words, machine learning and deep learning can self-learn anomalous behavior and identify normal patterns to reduce false alarms. Currently, various studies on intrusion detection systems are being conducted to detect abnormal behavior using deep learning [3]. The data set used in the experiment is Leipzig Intrusion Detection Data Set (LID-DS), a host-based intrusion detection data set published in 2018. The LID-DS is structured differently from previously published data. It consists of different characteristics, attack methods and scenarios of computer systems that are more modern than previously published data sets [4]. In this paper, research was conducted by converting vector data into images and creating a deep learning-based detection model for anomalous behavior. Deep learning models are not efficient when learning a lot of data because they have to learn every time a new attack is discovered. Therefore, we propose Siamese-CNN using the few-shot learning technique, which has shown excellent performance for learning a small amount of data. The accuracy is calculated using the few-shot learning technique, and the performance is compared with Vanilla-CNN to confirm the performance of Siamese-CNN. Then, we describe whether Vanilla-CNN or Siamese-CNN can best detect the type of attack. Additionally, additional experiments were conducted using NSL-KDD to verify the performance of the proposed model.

II. RELATED WORK

A. INTRUSION DETECTION DATA SET

The KDD99 data set was the first to release standard data for intrusion detection systems at MIT, sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) to evaluate intrusion detection systems. The data types are divided into four attack categories: Denial of Service (DoS), User versus Root (U2R), Remote versus Local attack (R2L), and Probe attack. Much research is being done to evaluate intrusion detection systems using the KDD99 data set [5]. The UNM data set is newer than the KDD99 data set, but the data consists of a series of system calls [6]. In the field of intrusion detection systems, data sets such as KDD99 and UNM are outdated and therefore do not contain the capabilities of the computer systems currently in use. To address this, in 2013, the Australian Defense Forces Academy (ADFA) published an ADFA dataset to evaluate host-based intrusion detection systems. The ADFA data set consists of a series of system calls for normal and attack data [5], [6].

B. INTRUSION DETECTION RELATED RESEARCH

Intrusion detection systems are systems that detect and block abnormal behavior by comparing normal and abnormal cyberattack patterns [7], [8]. In Laskov *et al.* [9], various machine learning algorithms such as Decision Tree, K-Nearest Neighbor (K-NN), Multi-Layer Perceptron (MLP), K-means, and Support Vector Machine (SVM) were applied to intrusion detection. The algorithm was compared using a Receiver Operator Characteristic (ROC) curve. Kim and Kim [10] conducted a study to solve the high false alarm rate problem using machine learning algorithms, such as SVM and K-NN, in intrusion detection systems. Kim *et al.* [11] proposed an LSTM-based system called the language modeling method to design an abnormal behavior-based host intrusion detection system. The author used a new ensemble method to solve the high false alarm rate problem that often occurs in the existing method. Ravipati and Abualkibash [12] experimented with eight machine learning algorithms using the KDD99 data set, which is most similar to the function of the LID-DS, showing performance evaluation and false positive rate figures. Khan *et al.* [13] pointed out the drawbacks of using machine learning algorithms to propose new intrusion detection models. A method that combines the CNN-based network intrusion detection model and the Soft Max algorithm was proposed and evaluated using the KDD data set. The experimental results showed that it is a more efficient model for intrusion detection than the SVM and DBN (Deep Belief Network) algorithms. Due to recent advances in deep learning technology, numerous studies are underway to detect binary and various attack categories based on CNN [14], [15]. According to this research direction, various deep learning architectures have been investigated in recent intrusion detection literature [16]. The limitation of text-based malware image analysis is that it cannot easily analyze specific malware. Malware can be packaged in a variety of packaging methods. Therefore, a solution is needed to classify malicious codes by analyzing images [17]. Recently, applications of deep learning models have been used to detect malicious codes. In deep learning training using images, malicious code binaries are converted into grayscale image representations, and deep learning models are used to learn complex features [18], [19]. Upadhyay and Pantiukhin [20] used 36 randomly selected columns from 41 columns of the KDD data set. After that, the CNN model was trained by transforming the data set into images 6 × 6 in size and then storing the remaining functions in other variables. That experiment's results showed that the intrusion detection error of the proposed model was less than 2%, and it was more efficient to convert the data into images and analyze them. Yajamanam *et al.* [21] analyzed image features called gist descriptors to classify obfuscated malware, and compared these techniques to deep learning techniques to assess their robustness.

C. SIAMESE NETWORKS

Siamese Networks are networks that process two different input data using the same type of network. The two networks share weights and generate feature vectors for the input image. Images of the same class are taught to represent closer

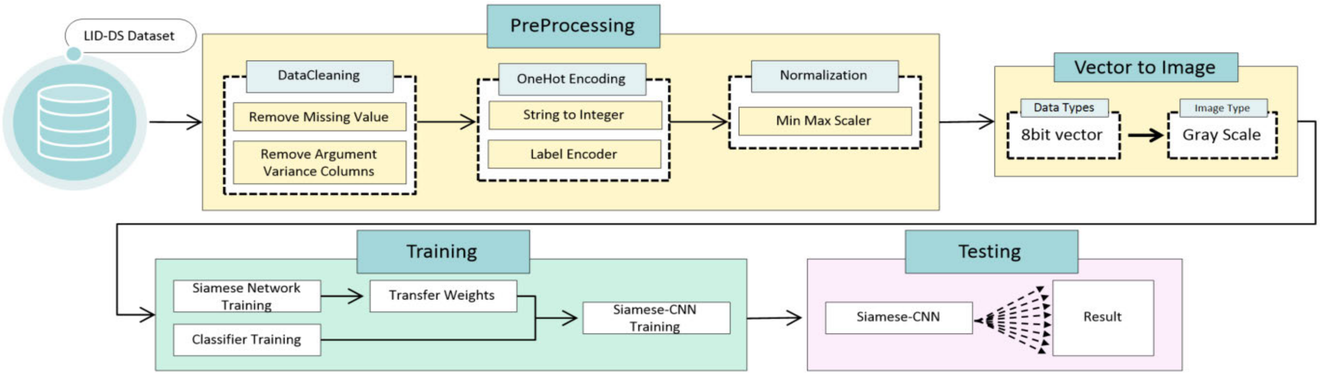
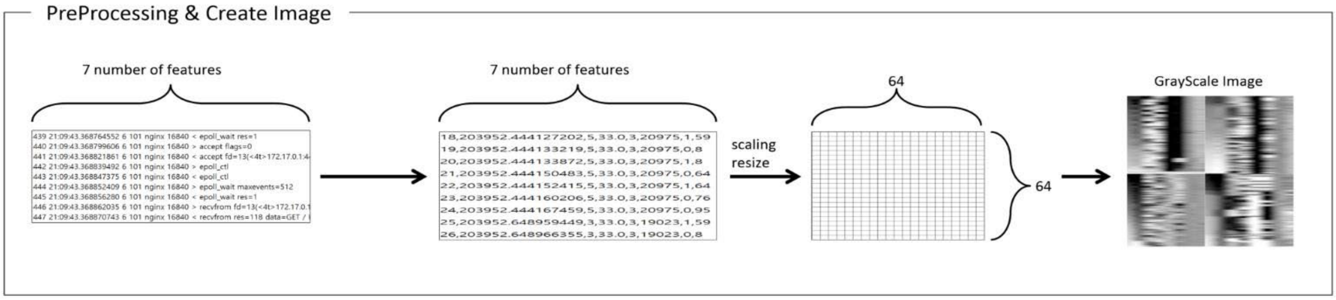


FIGURE 1. **Proposed host-based intrusion detection model structure.**



```
439 21:09:43.368764552 6 101 nginx 16840 < epoll_wait res=1
440 21:09:43.368799606 6 101 nginx 16840 > accept flags=0
441 21:09:43.368821861 6 101 nginx 16840 < accept fd=13(<4t>172.17.0.1:4
442 21:09:43.368839492 6 101 nginx 16840 > epoll_ctl
443 21:09:43.368847375 6 101 nginx 16840 < epoll_ctl
444 21:09:43.368852409 6 101 nginx 16840 > epoll_wait maxevents=512
445 21:09:43.368856280 6 101 nginx 16840 < epoll_wait res=1
446 21:09:43.368862035 6 101 nginx 16840 > recvfrom fd=13(<4t>172.17.0.1
447 21:09:43.368870743 6 101 nginx 16840 < recvfrom res=118 data=GET /
```

```
18,203952.444127202,5,33.0,3,20975,1,59
19,203952.444133219,5,33.0,3,20975,0,8
20,203952.444133872,5,33.0,3,20975,1,8
21,203952.444150483,5,33.0,3,20975,0,64
22,203952.444152415,5,33.0,3,20975,1,64
23,203952.444160206,5,33.0,3,20975,0,76
24,203952.444167459,5,33.0,3,20975,0,95
25,203952.648959449,3,33.0,3,19023,1,59
26,203952.648966355,3,33.0,3,19023,0,8
```

FIGURE 2. **The detailed dataset preprocessing steps and create image steps.**

to the vector space, while images of different classes are taught to represent farther from the vector space. In other words, by calculating the distance between the feature vectors created using the distance function, it is determined whether the two images are of the same class. The distance function uses a general similarity function, such as Euclidean distance or cosine distance. Hsiao *et al.* [22] used the Siamese Network to rank similarities between samples. In addition, accuracy was calculated through an N-way one-shot operation. As a result, the Siamese Network was shown to be more efficient than typical deep learning models. Moustakidis and Karlsson [23] proposed a Vec2im method to convert feature vectors into images and a pipeline for extracting new features. In addition, in order to reduce the input data dimension into one dimension, a Siamese convolutional neural network was used and applied to the NSL-KDD intrusion detection data set. Taigman *et al.* [24] trained the Siamese Network using standard cross entropy losses and error backpropagation. Their model predicts the L1 distance similarity between each sample and predicts whether the faces of each sample are the same.

D. FEW-SHOT LEARNING

Few-shot learning is a learning method for meta-learning using a data set with sufficient data and classifying a data set with little data included in each class [25]. Few-shot Learning is largely divided into meta-Learning method and metric-Learning method. The main way to achieve excellent performance is the metric learning method. Metric learning is a distance-based learning method. According to some distance metrics, metric learning is a way of learning so that samples of the same class are close and samples of different classes are far apart [26]. This is a method to predict the class of samples with the highest similarity when providing query data by learning the similarity or distance between images [27].

In this paper, we learn the feature vectors for two images from the Siamese Network and compare the distances between the vectors. In addition, we propose a model to detect whether the attacks are the same by comparing the similarity scores for each cyberattack method.

III. INTRODUCING THE DATA SETS AND IMPLEMENTING MODEL

A host-based intrusion detection model consisting of preprocessing, vector to image processing, training, and testing steps is proposed to evaluate and improve the performance of the system using the LID-DS, as shown in Fig. 1. The structure proposed in this paper is shown in Fig. 1. It consists of LID-DS, preprocessing, image generation, Siamese Network, Siamese-CNN, and N-way K-Shot Learning. Each part is described separately in this section. Section III.A describes the LID-DS. Section III.B describes the NSL-KDD. Section III.C describes the preprocessing part and describes the data normalization process according to the data format. Section III.D describes the process of converting 1D vector data into 3D image data. Section III.E describes the structure of two convolutional neural networks that have the same form as the Siamese Network part. Section III.F is the N-way K-Shot Learning part. After explaining the descriptions and features of N and K, the structure of the Siamese-CNN proposed in this paper is examined through Section III.G. Section III.H is the Train Test Split part and describes the ratio of the Train Test data used in the experiment.

A. LID-DS

The LID-DS used in this paper was published at Leipzig University in 2018 for anomaly detection studies on host-based intrusion detection systems. The LID-DS has the latest computer system characteristics, cyberattack methods, and cyberattack scenarios comparted to previously published data sets. Fig. 3 shows the data generation process using the LID-DS cyberattack scenario.

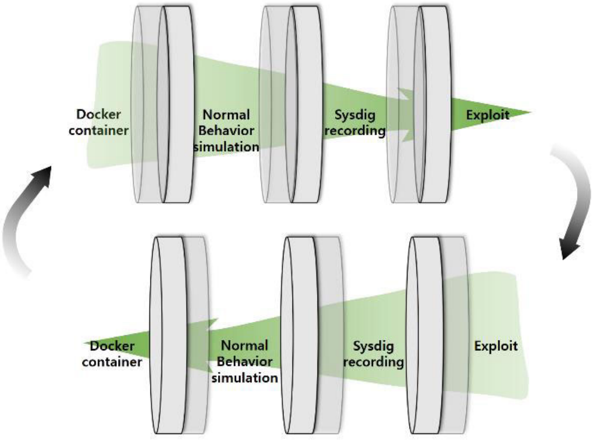


FIGURE 3. Attack simulation procedure of LID-DS data set.

Each cyberattack scenario is a cyberattack that uses the discovered vulnerability. LID-DS runs inside Docker 10 container virtualization software to record system call traces, define the initial state of the target of the attack, and return to the initial state after each cyberattack. The method is as follows. For the record, we first start a Docker container that hosts the attack target using the LID-DS framework. Then, according to the scenario, the initialization task is executed and the simulation of normal operation begins. After that, it waits for a short time before Sysdig is activated in order not to record the starting effect of the target software. In case of recording an cyberattack action, the cyberattack starts after a certain period of time. After the recording is executed for the desired time, the recording is stopped by the control script. It also stops and removes the normal working and simulation of used Docker containers. Table 1 is a table comparing the existing intrusion detection data set and the LID-DS. The features in Table 1 are features of LID-DS and show that there are features that are not included in the existing intrusion detection data. Also, as can be seen from Table 1, previously published data sets are either too old to reflect current system characteristics or consist of a series of system calls, making them unsuitable for use in intrusion detection system research.

TABLE 1. Feature comparison LID-DS with other datasets.

Feature	LID-DS	ADFA-LD	UNM	KDD99
Arguments	O	X	X	O
Returnvalues	O	X	X	O
Timestamps	O	X	X	O
Process ID	O	X	O	O
Data buffers	O	O	X	X
Meta data	O	X	X	O
Thread	O	X	O	O

LID-DS is the first HIDS data set containing system calls and timestamps, thread IDs, process names, arguments, return values, and data buffer excerpts. Many of the features included cannot be extracted from previous data sets [4].

B. NSL-KDD

The NSL-KDD data set is a data set proposed by improving the KDD CUP 99 data set generated through the DARPA Intrusion Detection Assessment Program in 1999. It was created by modeling the US Air Force network and simulating 38 network intrusion detection attacks [28]. Recently, many researchers have demonstrated excellence in experimental evaluation using NSL-KDD as a standard benchmark data set [29], [30], [31]. The NSL-KDD data set has an advantage in that the training data set and the test data set are configured separately, and the number of records is a reasonable number. The NSL-KDD dataset consists of 43 features including labels and 39 attack types. However, the peculiar thing is that the NSL-KDD data set contains only 24 attack types in the training set. It is a data set that trains only 24 attack types and evaluates the performance of untrained attacks that can be detected through the test set. Therefore, in this paper, we performed an experiment of the proposed model using LID-DS and NSL-KDD data set, which are composed of current system characteristics and various cyberattack methods.

C. PREPROCESSING

In the LID-DS, the argument function and missing values were deleted for all data, and the colon (:) was removed for the event_time function, as in the preprocessing part of Fig. 1. LID-DS does not contain duplicate data and event_direction and event_type are converted to numbers using LabelEncoder. The Process category consists of a total of 16 processes. The number of processes is different for each attack method. Therefore, the processes used in each cyberattack method have been consolidated into one process. As a result, labels consisting of a total of 10 processes were used by attaching labels to each process using LabelEncoder. MinMaxScaler proceeded to the normalization step using values from 0 to 255.

The data format of the NSL-KDD data set is divided into three categories: nominal, numeric, and binary. The nominal data are categorical text data that cannot be used to train deep learning models. Therefore, all were converted to integer type and converted to one-hot vector. For numeric data, min-max normalization was performed. The NSL-KDD data set consists of 39 attack types. The number of classification targets in a deep learning model is inversely proportional to the classification accuracy. Therefore, as shown in Table 2, it was classified into four attack classes (DoS, Probe, R2L, U2R) [32].

TABLE 2. Attack types merged into 4 classes.

Merged Attack Type	Attack Type
DoS	Back, Land, Neptune, Pod, Smurf, Teardrop, Mailbomb, Apache2, Processtable, UdpStorm
Probe	Ipsweep, Nmap, Portsweep, Satan, Mscan, Saint
R2L	Ftp_write, Guess-passwd, Imap, Multihop, Phf, Spy, Warezclient, Warezmaster, Sendmail, Named, Snmpgetattack, Snmpguess, Xlock, Xsnoop, Worm
U2R	Buffer_overflow, Loadmodule, Perl, Rootkit, Httptunnel, Ps, Sqlattack, Xterm

In addition, the NSL-KDD data set has a data imbalance problem because the number of data in the R2L and U2R classes is smaller than the number of data in the Normal, DoS, and Probe classes. Data imbalance is a problem caused by the difference in the number of samples with each class in the data set. Classes with a large number of samples learn many types well, while classes with a small number of samples do not learn many types. To solve the data imbalance problem, we used RandomUndersampler, one of the Undersampling techniques, for the top three classes that make up a large portion of the data set. Table 3 shows the number of samples of the changed NSL-KDD data set.

TABLE 3. Number of NSL-KDD data set samples changed.

	Original Data	Undersmapled Data
Normal	67343	1000
DoS	45927	1000
Probe	11656	1000
R2L	995	995
U2R	52	52
Total	1259735	4047

D. CREATE IMAGE

Each sample has a value between 0 and 255. In this paper, as in the Vector to Image part of Fig. 1, the samples were converted into 8-bit vectors, and grayscale-type image data was created. Fig. 2 is a picture showing pre-processing data and the process created as an image. A grayscale image has a structure made up of one type of color that is converted into an M x N x 1 pixel array. M and N represent the number of columns and rows, respectively. The converted sample is a 64 × 64 pixel image.

Table 4 shows the number of images converted from 1D vector data of LID-DS to 3D images. The converted image data is an image of malicious data.

E. SIAMESE NETWORK

The Siamese Network used in this paper is composed of two convolutional neural networks that have the same shape as the Siamese Network in Fig. 4. The two input images generate each feature vector through the convolution layer. The distance between the two generated feature vectors is calculated by the Euclidean distance method, and whether the two images are of the same class is determined through the similarity score. Table 5 shows the structure of the Siamese Network proposed in this paper. LeakyReLU was used as the activation function of the layers, except for the last layer.

TABLE 4. Number of images per cyberattack method.

Attack Type	Number of Images
Bruteforce	825
CVE-2012-2122	1019
CVE-2014-0160	786
CVE-2017-7529	1157
CVE-2018-3760	1022
CVE-2019-5418	1073
EPS-CWE-434	1060
PHP-CWE-434	1112
SQL Injection	1078
Zip Slip	1062
Total	10194

TABLE 5. The siamese network configuration used in the experiment.

Parameter	Value
Layer	64-128-128-256
MaxPooling	2, 2
Dropout	0.25
Activation	LeakyReLU, Sigmoid
Optimizer / Learning Rate	Adam / 0.0004
Loss	Binary_crossentropy

F. N-WAY K-SHOT LEARNING

To check if the Siamese Network model was trained properly, we used the N-way K-shot learning method, one of the few-shot learning methods. N-way K-shot learning consists of a data set with support data used for training and query data used for testing. N is the number of categories, and K is the number of supporting data for each category. In N-way K-shot learning, the smaller the N value, the more accurate prediction is possible; the larger the N value, the lower the accuracy. Typically, in an experiment, N is set to 2-10 or less, and K is set to 1 or 5.

G. SIAMESE-CNN AND VANILLA-CNN

In this paper, a convolutional neural network (CNN) is used as an intrusion detection classification model. As shown in the Siamese-CNN part of Table 6 and Fig. 4, the structure of the proposed convolutional neural network can be confirmed, and learning is performed using the weights of the Siamese Network, respectively. Relu was used as the activation function of the layer, except for the last layer. The input_shape is of size 64 × 64 and has 1 color channel, so the tuple value is (64, 64, 1). To ignore minor changes, only the main values were extracted through Maxpooling2D, and a small output value was created and used. In addition, since Conv2D and Maxpooling mainly deal with 2D, to transfer it to the fully connected layer it must be transferred in 1D. Therefore, the layer was converted to 1D using the Flatten function.

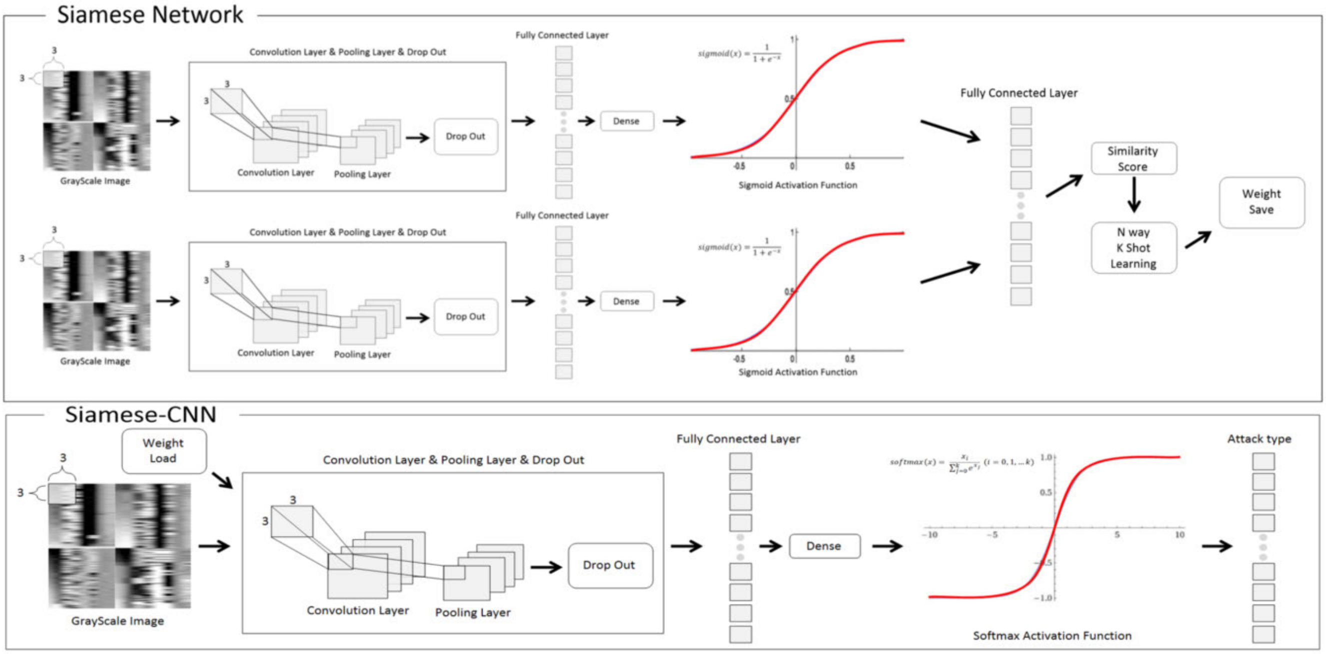


FIGURE 4. The structure of siamese convolutional neural networks and convolutional neural networks.

TABLE 6. The neural network configuration used in the experiment.

Parameter	Value
Layer	64-128-128-256
MaxPooling	2, 2
Dropout	0.25
Activation	relu, softmax
Optimizer / Learning Rate	Adam / 0.0004
Loss	categorical_crossentropy

The structure of the proposed Vanilla-CNN convolutional neural network is shown in Table 6. It is the same except that it uses the weights of the Siamese Network during the Siamese-CNN training process in Fig. 4.

H. TRAIN TEST SPLIT

The data set ratio used in the experiment for 1D vector data is as follows. The training data and test data were divided by the commonly used 8:2 ratio using the train_test_split module to conduct experiments. After training the training data, we evaluated the model using the test data and discovered an overfitting phenomenon. Overfitting means the model overtrains the training data and fails to make correct predictions. To avoid overfitting, the model was evaluated by cross-validation, including data divided by different proportions. As a result, the data set divided by a ratio of 7:3 had the highest model performance. Therefore, in this paper, the training data and the test data were divided into 7:3 ratios and used. The data set proportions used in the 3D image data experiment are as follows. The image data generated in Part III.D was randomly extracted and divided at a ratio of 7:3. Table 7 shows the number of LID-DS data used in the experiment.

TABLE 7. Number of LID-DS data used in the experiment.

	Train	Test
Bruteforce	577	248
CVE-2012-2122	714	305
CVE-2014-0160	550	236
CVE-2017-7529	810	347
CVE-2018-3760	715	307
CVE-2019-5418	751	322
EPS-CWE-434	742	318
PHP-CWE-434	778	334
SQL Injection	755	323
Zip Slip	743	319
Total	7135	3059

The proportions of the NSL-KDD data set used for further experiments are as follows. In Part III.C, the undersampled data was experimented using the train_test_split module by dividing it by the commonly used 7:3 ratio. Table 8 shows the number of NSL-KDD data used in the experiment.

TABLE 8. Number of NSL-KDD data used in the experiment.

	Train	Test
Normal	700	300
DoS	700	300
Probe	700	300
R2L	695	300
U2R	37	15
Total	2832	1215

IV. EVALUATION INDICATORS AND EXPERIMENTAL RESULTS

A. EVALUATION INDICATORS

To evaluate the performance of the trained model, Precision, Recall, F1 Score, and False Positive Rate(FPR) were used, and the equations for performance evaluation and accuracy are as follows.

"Equation (1) is an example of the Precision formula."

$$Precision = \frac{TP}{FP + TP} \tag{1}$$

"Equation (2) is an example of the Recall formula."

$$Recall = \frac{TP}{FN + TP} \tag{2}$$

"Equation (3) is an example of the F1 Score formula."

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

"Equation (4) is an example of the FPR formula."

$$FPR = \frac{FP}{FP + TN} \tag{4}$$

"Equation (5) is an example of the Accuracy formula."

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \tag{5}$$

The properties of equations (1), (2), (3), (4) and (5) are the same as Table 9.

TABLE 9. Confusion matrix.

		Real correct answer	
		True	False
Classification Result	True	True Positive(TP)	False Positive(FP)
	False	False Negative(FN)	True Negative(TN)

Precision is the ratio between the True Positives and all the Positives. The recall is the measure of our model correctly identifying True Positives. F1-Score is the harmonic average of precision and recall. This means that Precision and Recall are useful when measuring the performance of a model, but there is no way to describe how effective the model is, so we use a method called F1-Score to determine whether the model is effective. FPR is the rate at which the model predicts true for false data. Accuracy is calculated as in Equation (4), because, unlike Precision and Recall, the example predicting False as False is also calculated as a correct case.

B. EXPERIMENTAL RESULTS

The experiment was conducted using the Siamese Network created in part III.E using the LID-DS that converted the image to the size of 64 × 64. Fig. 5 is a picture that visualizes the created Siamese Network learning process. To check the performance of the trained Siamese Network model, we used the N-way One-shot learning proposed in Part III.F. N conducted N-way tests for 1, 2, 3, and 4. As the data used for performance evaluation, part III.H of the test data were used. For the test, step_epoch was performed 2000 times for the N value test, and the average accuracy was calculated by performing the process 2000 times. The results are shown in Table 10. To classify each cyberattack method in the LID-DS, an experiment was performed by loading the weight of the stored Siamese Network into the Siamese-CNN, as shown in Fig. 4.

TABLE 10. Siamese network performance with n-way one-shot learning.

N-way	1	2	3	4
Accuracy	100%	98%	94%	92%

Table 11, Table 12 shows the performance comparison results of the Siamese-CNN created using the weights of Siamese Network and Vanilla-CNN not using weights. In addition, Naive Bayes, Decision Tree, Logistic Regression, and MLP algorithm performance were compared using vector data. The reason for conducting experiments using vector data is to compare the performance of experiments conducted using image data. Naive Bayes, Decision Tree, Logistic Regression, and MLP algorithms used algorithms provided by Scikit-learn.

TABLE 11. Performance evaluation of learning algorithms.

	Precision	Recall	F1-Score
Vanilla-CNN	87%	86%	87%
Siamese-CNN	89%	90%	90%
Naïve Bayes	81%	82%	81%
Decision Tree	81%	80%	78%
Logistic Regression	72%	75%	72%
MLP	69%	68%	65%

TABLE 12. Accuracy and FPR of learning algorithms.

	Accuracy	FPR
Vanilla-CNN	88%	1.3%
Siamese-CNN	91%	1.1%
Naïve Bayes	82%	1.9%
Decision Tree	79%	2.4%
Logistic Regression	75%	3.0%
MLP	68%	3.9%

Table 11 shows that Siamese-CNN has outperformed Vanilla-CNN by 3% in f1-score and 4% in recall. Table 12 shows that Siamese-CNN has outperformed Vanilla-CNN by 3% in accuracy and 0.2% in FPR. That experiment's results showed that the FPR of the proposed Siamese-CNN model is 1.2%, and it was more efficient to convert the data into images and analyze them. In order to confirm that each cyberattack method was classified correctly, the confusion matrix of the Siamese-CNN was checked, and the result is shown in Fig. 6. Also, the cyberattack methods and names classified in Fig. 6 are shown in Table 13. Fig. 6 shows confirmation that (0, 2), (4, 5), and (7, 8) cyberattack methods were not properly classified. Excluding Bruteforce and CVE-2014-0160, cyberattack methods (CVE-2018-3760, CVE-2019-5418) are cyberattack methods for information leakage, with CVE-2019-5418 complementing the vulnerability of CVE-2018-3760.

TABLE 13. LID-DS data cyberattack method.

Num	Cyberattack type
0	Bruteforce
1	CVE-2012-2122
2	CVE-2014-0160
3	CVE-2017-7529
4	CVE-2018-3760
5	CVE-2019-5418
6	EPS-CWE-434
7	PHP-CWE-434
8	SQL Injection
9	ZIP Slip

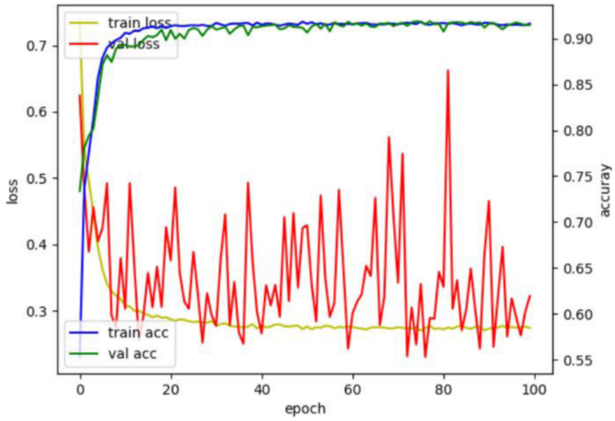


FIGURE 5. Visualize the learning process of siamese network.

(PHP-CWE-434, SQL Injection) The cyberattack method is a vulnerability identified by the Open Web Application Security Project (OWASP), and PHP-CWE-434 and SQL Injection are classified as attacks in which an attacker can send hostile data to the interpreter.

Therefore, the (CVE-2018-3760, CVE-2019-5418) and (PHP-CWE-434, SQL Injection) cyberattack methods are the same type of cyberattack method, so they can be viewed as one attack method. Ten cyberattack methods were converted into 8 cyberattack methods.

Table 14 is a table converted from 10 cyberattack methods to 8 cyberattack methods. We used the transformed eight cyberattacks to double-check the performance of the Siamese-CNN. Fig. 7 visualizes the learning process of Siamese-CNN using 8 types of cyberattacks.

The result of confirming the performance of the learned Siamese-CNN is as shown in Table 15, Table 16. The confusion matrix for the performance is shown in Fig. 8. As shown in the above result, after converting to 8 cyberattack methods, the experiment was conducted. The results confirmed that the accuracy was improved by about 2% and recalled by about 2% and FPR by about 0.1% compared to Siamese-CNN, which was classified into 10 cyberattack methods.

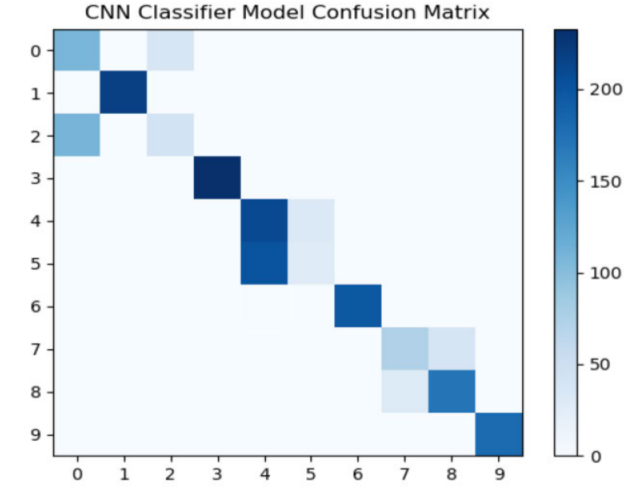


FIGURE 6. Confusion matrix for CNN model classification performance.

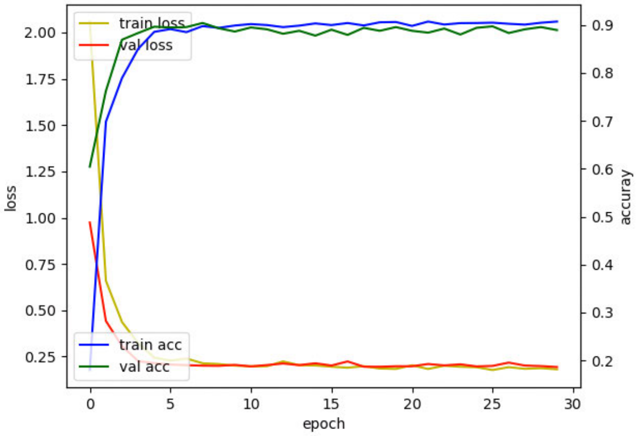


FIGURE 7. Visualize the learning process of classifier model.

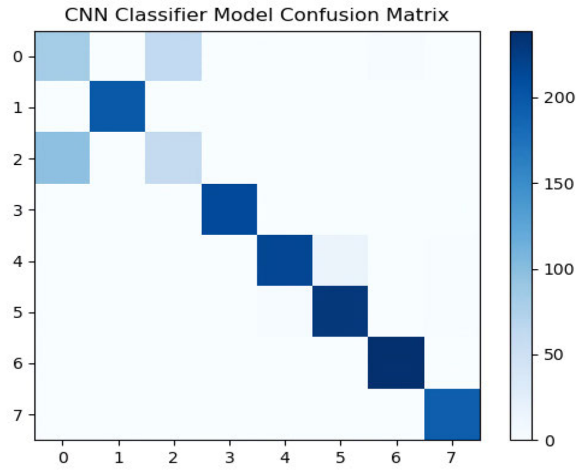


FIGURE 8. Confusion matrix for model performance classifying 8 cyberattack methods.

However, experiments on LID-DS were first conducted in this paper. Therefore, to confirm the performance of the proposed model, an additional experiment was performed using

TABLE 14. Modified LID-DS data cyberattack method.

Num	Cyberattack type	Num	Changed cyberattack type
0	Bruteforce	0	Bruteforce
1	CVE-2012-2122	1	CVE-2012-2122
2	CVE-2014-0160	2	CVE-2014-0160
3	CVE-2017-7529	3	CVE-2017-7529
4	CVE-2018-3760	4	XSS
5	CVE-2019-5418	5	EPS-CWE-434
6	EPS-CWE-434	6	SCRIPT
7	PHP-CWE-434	7	ZIP Slip
8	SQL Injection		
9	ZIP Slip		

TABLE 15. Performance evaluation of siamese-CNN.

	Precision	Recall	F1-Score
Siamese-CNN	93%	92%	91%

TABLE 16. Accuracy and FPR of siamese-CNN.

	Accuracy	FPR
Siamese-CNN	93%	1.0%

TABLE 17. Performance evaluation of learning algorithms.

	Precision	Recall	F1-Score	Accuracy
Siamese-CNN	81%	83%	82%	81%
Random Forest	80%	77%	73%	76%
Decision Tree	76%	74%	72%	74%
Naïve Bayes	75%	72%	71%	72%

the NSL-KDD data set, which has similar characteristics to LID-DS. The ratio of the data used in the experiment is shown in Table 8 of Part III.H. Table 17 compares the performance of the Siamese-CNN proposed in this paper and the Random Forest, Decision Tree, and Naïve Bayes algorithms provided by Scikit-learn.

As can be seen in Table 17, the Siamese-CNN model proposed in this white paper performs better than the model provided by Scikit-learn. Although the NSL-KDD data set is a very unbalanced data set, the proposed Siamese-CNN is judged to outperform the Scikit-learn model as it extracts feature vectors and classifies them by comparing similarity scores.

V. CONCLUSION AND FURTHER RESEARCH

The LID-DS presented in this paper comprises up-to-date system security vulnerabilities and can be used to evaluate various types of HIDS without losing basic thread information. In order to check the similarity among the data, 1D vector data was converted into 3D image data and reconstructed. And to accurately identify and classify image data, we proposed a new hybrid deep learning model. The existing deep learning-based intrusion detection system model has the disadvantage of having to retrain every time a new attack is discovered. To solve this problem, a Siamese Network was created to test the performance of the model using the few-shot learning technique, which shows excellent performance when learning a small amount of data. After that, to classify each cyberattack method, the performance evaluation of Siamese-CNN and Vanilla-CNN using the weight of the Siamese Network proposed in this paper was conducted. In addition, additional experiments were conducted to compare the performance experimented with image data and the performance experimented with one-dimensional vector data. Naive Bayes, Decision Tree, Logistic Regression, and MLP algorithm performance were compared using vector data. As a result, image analysis showed better performance than vector data analysis. Also, the results confirmed that the accuracy of the Siamese-CNN is about 3% higher than that of the typical Vanilla-CNN, and Recall is about 4% higher. In order to confirm that each cyberattack method was properly classified, we checked the confusion matrix and found that 4 types of cyberattack methods can be transformed into 2 types of cyberattack methods. Therefore, the performance was re-evaluated by reducing 10 cyberattack methods to 8 cyberattack methods. As a result, accuracy increased by about 2% compared to Siamese-CNN, which classified 10 cyberattack methods, Precision by 4%, Recall by 2%, and F1-Score by 1%. In addition, since there are no experiments on LID-DS, additional experiments were conducted using the NSL-KDD data set to verify the performance of the proposed model. As a result, it was confirmed that the proposed Siamese-CNN model outperforms the model provided by Scikit-learn. In summary, the advantage of the proposed method is that it has lower training costs compared to traditional deep learning-based methods. In addition, the proposed deep learning-based model can be quickly trained in real time to cope with new attack data in the future. In future work, we will conduct a study to detect intrusion against various cyberattacks by using the LID-DS converted into images. Further, by optimizing the hyper parameter value of the proposed model, we will conduct research to further increase the accuracy of intrusion detection for new cyberattacks and internal cyberattacks. Finally, the experiment can be expanded for the recently created intrusion detection data set.

REFERENCES

[1] Y. G. Choi and S. S. Park, "Reinforcement mining method for anomaly detection and misuse detection using post-processing and training method," in *Proc. Korean Inf. Sci. Soc. Conf.*, 2006, pp. 238–240.

[2] S. O. Choi and W. N. Kim, "Control system intrusion detection system technology research trend," *Rev. KIISC*, vol. 24, no. 5, pp. 7–14, 2014.

[3] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: A review," 2020, *arXiv:2007.02500*. [Online]. Available: http://arxiv.org/abs/2007.02500

[4] M. M. Röhling, M. Grimmer, D. Kreußel, J. Hoffmann, and B. Franczyk, "Standardized container virtualization approach for collecting host intrusion detection data," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2019, pp. 459–463.

[5] O. Yavanoglu and M. Aydos, "A review on cyber security datasets for machine learning algorithms," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 2186–2193.

[6] M. Pendleton and S. Xu, "A dataset generator for next generation system call host intrusion detection systems," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2017, pp. 231–236.

[7] L. N. Tidjon, M. Frappier, and A. Mammar, "Intrusion detection systems: A cross-domain overview," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3639–3681, Jun. 2019.

[8] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "Optimal cluster expansion-based intrusion tolerant system to prevent denial of service attacks," *Appl. Sci.*, vol. 7, no. 11, p. 1186, Nov. 2017.

[9] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: Supervised or unsupervised?" in *Proc. Int. Conf. Image Anal. Process.* Berlin, Germany: Springer, 2005, pp. 50–57.

[10] T.-T.-H. Le, J. Kim, and H. Kim, "An effective intrusion detection classifier using long short-term memory with gradient descent optimization," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Feb. 2017, pp. 1–6.

[11] G. Kim, H. Yi, J. Lee, Y. Paek, and S. Yoon, "LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems," 2016, *arXiv:1611.01726*. [Online]. Available: http://arxiv.org/abs/1611.01726

[12] R. D. Ravipati and M. Abualkibash, "Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets—A review paper," *Int. J. Comput. Sci. Inf. Technol.* vol. 11, 2019.

[13] R. U. Khan, X. Zhang, M. Alazab, and R. Kumar, "An improved convolutional neural network model for intrusion detection in networks," in *Proc. Cybersecur. Cyberforensics Conf. (CCC)*, May 2019, pp. 74–77.

[14] A. K. Verma, P. Kaushik, and G. Shrivastava, "A network intrusion detection approach using variant of convolution neural network," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Jul. 2019, pp. 409–416.

[15] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, no. 6, p. 916, Jun. 2020.

[16] A. Drewek-Ossowicka, M. Pietrołaj, and J. Rumiński, "A survey of neural networks usage for intrusion detection systems," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 497–514, Jan. 2021.

[17] S. Yue, "Imbalanced malware images classification: A CNN based approach," 2017, *arXiv:1708.08042*. [Online]. Available: http://arxiv.org/abs/1708.08042

[18] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-G. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3187–3196, Jul. 2018.

[19] S. Ni, Q. Qian, and R. Zhang, "Malware identification using visualization images and deep learning," *Comput. Secur.*, vol. 77, pp. 871–885, Aug. 2018.

[20] R. Upadhyay and D. Pantiukhin, "Application of convolutional neural network to intrusion type recognition," Ben-Gurion Univ. Negev, Beersheba, Israel, Tech. Rep., 2017.

[21] S. Yajamanam, V. R. S. Selvin, F. Di Troia, and M. Stamp, "Deep learning versus gist descriptors for image-based malware classification," in *Proc. ICISSP*, 2018, pp. 553–561.

[22] S.-C. Hsiao, D.-Y. Kao, Z.-Y. Liu, and R. Tso, "Malware image classification using one-shot learning with siamese networks," *Procedia Comput. Sci.*, vol. 159, pp. 1863–1871, Jan. 2019.

[23] S. Moustakidis and P. Karlsson, "A novel feature extraction methodology using siamese convolutional neural networks for intrusion detection," *Cybersecurity*, vol. 3, no. 1, pp. 1–13, Dec. 2020.

[24] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

[25] S. E. Jang and J. T. Kim, "Few-shot classification of histopathology image using batch hard loss-based siamese networks," Korean Inst. Inf. Sci. Eng., Daejeon, South Korea, Tech. Rep., 2019, pp. 634–636.

[26] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," Tech. Rep., 2016.

[27] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learning Workshop*, vol. 2, 2015, pp. 1–8.

[28] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.

[29] P. Devan and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12499–12514, Aug. 2020.

[30] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[31] 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.

[32] N. Qazi and K. Raza, "Effect of feature selection, SMOTE and under sampling on class imbalance classification," in *Proc. UKSim 14th Int. Conf. Comput. Modelling Simulation*, Mar. 2012, pp. 145–150.

DAEKYEONG PARK received the B.S. degree in computer engineering from Soongsil University, Seoul, South Korea, in 2020. He is currently pursuing the M.S. degree with Sejong University. His research interests include information security, data mining, and machine learning.

SANGSOO KIM received the B.S. degree in electronic engineering and the M.S. degree in computer engineering from Kyungpook National University, Daegu, South Korea, in 1997 and 2003, respectively. Since 2003, he has been a Principal Researcher with the Agency for Defense Development, South Korea. His research interests include cyber security, machine learning, and situational awareness.

HYUKJIN KWON received the B.S., M.S., and Ph.D. degrees in industrial engineering from Sungkyunkwan University, Seoul, South Korea, in 1989, 1991, and 2000, respectively. He served as the Director for Research Planning Division, Korea Institute for Defense Analyses (KIDA), and researched information system assessment and information system development, from 1991 to 2017. He served as the Director General for the Information Planning Bureau, Ministry of National Defense, South Korea, from 2017 to 2020. He is currently a Senior Research Fellow with the Center for Military Analysis Planning, KIDA. He is the author or coauthor of a number of academic articles. His research interests include cyber security and performance for information systems.

DONGIL SHIN received the B.S. degree in computer science from Yonsei University, Seoul, South Korea, in 1988, the M.S. degree in computer science from Washington State University, Pullman, WA, USA, in 1993, and the Ph.D. degree from the University of North Texas, Denton, TX, USA in 1997. He was a Senior Researcher with the System Engineering Research Institute, Daejeon, South Korea, in 1997. Since 1998, he has been with the Department of Computer Engineering, Sejong University, South Korea, where he is currently a Professor. His research interests include information security, bio-signal data processing, data mining, and machine learning.

DONGKYOO SHIN received the B.S. degree in computer science from Seoul National University, South Korea, in 1986, the M.S. degree in computer science from the Illinois Institute of Technology, Chicago, IL, USA, in 1992, and the Ph.D. degree in computer science from Texas A&M University, College Station, TX, USA, in 1997. From 1986 to 1991, he was with the Korea Institute of Defense Analyses, where he developed database application software. From 1997 to 1998, he was a Principal Researcher with the Multimedia Research Institute, Hyundai Electronics Company, South Korea. He is currently a Professor with the Department of Computer Engineering, Sejong University, South Korea. His research interests include machine learning, ubiquitous computing, bio-signal data processing, and information security.