An Effective Anomaly Detection Approach based on Hybrid Unsupervised Learning Technologies in NIDS

Kangseok Kim[1*]
[1] Dept. of Cyber Security, College of Computing and Informatics at Ajou University
16499, Suwon, Korea
[e-mail: kangskim@ajou.ac.kr]
[*]Corresponding author: Kangseok Kim

Received November 2, 2023; revised November 27, 2023; accepted December 12, 2023; published February 29, 2024

Abstract

Internet users are exposed to sophisticated cyberattacks that intrusion detection systems have difficulty detecting. Therefore, research is increasing on intrusion detection methods that use artificial intelligence technology for detecting novel cyberattacks. Unsupervised learning-based methods are being researched that learn only from normal data and detect abnormal behaviors by finding patterns. This study developed an anomaly-detection method based on unsupervised machines and deep learning for a network intrusion detection system (NIDS). We present a hybrid anomaly detection approach based on unsupervised learning techniques using the autoencoder (AE), Isolation Forest (IF), and Local Outlier Factor (LOF) algorithms. An oversampling approach that increased the detection rate was also examined. A hybrid approach that combined deep learning algorithms and traditional machine learning algorithms was highly effective in setting the thresholds for anomalies without subjective human judgment. It achieved precision and recall rates respectively of 88.2% and 92.8% when combining two AEs, IF, and LOF while using an oversampling approach to learn more unknown normal data improved the detection accuracy. This approach achieved precision and recall rates respectively of 88.2% and 94.6%, further improving the detection accuracy compared with the hybrid method. Therefore, in NIDS the proposed approach provides high reliability for detecting cyberattacks.

Keywords: Anomaly Detection, Data Augmentation, Hybrid Approach, NIDS, Unsupervised Learning Technologies

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT: Ministry of Science and ICT) (No. NRF- 2019R1F1A1059036).

http://doi.org/10.3837/tiis.2024.02.012 **ISSN : 1976-7277**

1. Introduction

Since vast amounts of information and data now flow through the Internet, users are exposed to a variety of new and sophisticated cyberattacks. Therefore, the development of intrusion detection technology to detect new cyberattacks is becoming more important. Intrusion detection system (IDS) can be classified according to intrusion detection types and techniques [1, 2, 3]. From the point of view of intrusion detection types, IDSs can be network-based (NIDS) or host-based (HIDS) [1, 3, 4]. While NIDS detects intrusions based on external network traffic, HIDS focuses on detecting intrusions on specific host systems internally rather than externally to the network [5, 6, 7]. From the point of view of intrusion detection techniques, IDS can be further classified into rule-based detection technique and statistics-based anomaly detection technique [8]. Both detect intrusions using information extracted from network or host-related data [9, 10, 11]. Misuse detection techniques used in rule-based IDS are effective at detecting known attacks (or predefined patterns). However, it is vulnerable to intrusion by unknown attacks. Anomaly detection is the identification of abnormal behaviors based on deviations from typical normal behaviors [12]. Therefore, anomaly detection technology is needed to detect abnormal behaviors from normal behaviors based on data such as network traffic and system calls [6, 8, 10, 13].

Cyberattacks are becoming increasingly complex and sophisticated, making them difficult to detect [14]. The ability to detect new types of cyberattacks and benign behaviors is an important metric for evaluating the performance of intrusion detection systems. Therefore, research is increasing into intrusion detection methods that can detect new cyberattacks by applying artificial intelligence technology. However, the accuracy of intrusion-detection models based on machine deep learning may vary depending on whether abnormal samples in the training data are used during learning. various studies have been conducted on machine learning-based anomaly detection for IDSs. They are typically based on supervised and unsupervised machine learning. In general, the supervised learning-based anomaly detection method performs well but requires sufficient labeled data for learning and consumes considerable resources to distinguish between normal and abnormal data. In addition, because the amount of abnormal data is small compared to the amount of normal data, it is difficult to use in a real environment. Therefore, with the recent development of deep learning technology, research is actively being conducted on unsupervised learning-based algorithms that learn only from normal data and detect abnormal behaviors by finding patterns in the learned data [15]. To define normal behavioral patterns in an unsupervised learning approach, IDS technologies use information extracted from the target data, such as the features or distribution of network traffic in a NIDS.

Therefore, this study developed an anomaly-detection method based on an unsupervised machine and deep learning in a NIDS. The main objectives were:

(1) This paper presents an anomaly detection method based on unsupervised learning using autoencoder (AE) [16], Isolation Forest (IF) [17], and Local Outlier Factor (LOF) [18] algorithms. This method required three steps. The first was to extract latent vectors that preserve useful information from the features or distributions of the normal network flow data. Unsupervised learning-based autoencoders were used to extract latent vectors and define normal behavior patterns from the latent vectors. In this step, two autoencoder models were developed: Conv1D-DAE (one-dimensional convolutional neural network-based denoising autoencoder) and an SAE (Sparse Autoencoder). In the second step, the

LOF and IF algorithms were trained by inputting the extracted latent vectors. Subsequently, the LOF and IF identified the normal and anomalies, respectively, and separated them into two datasets. The separated data were fed into an IF or LOF. Outliers were identified as either attack or normal outliers. To evaluate the performance of the proposed hybrid-based anomaly detection method, the latent vectors encoded in the AEs for the test data were input into the trained LOF and IF. The data used in this experiment were obtained from the NSL-KDD dataset [19].

(2) This paper also presents an oversampling approach that increases the detection rate, which should be considered when evaluating the performance of an intrusion detection system. Machine-learning-based IDS must be periodically updated with newly acquired data patterns to improve detection performance. However, this method is expensive. Therefore, in this study, we evaluated how learning on oversampled data for a normal class affects detection performance. Performance was evaluated using the detection and false alarm rates with augmented samples.

The remainder of this paper is organized as follows. Section 2 briefly provides a selection of current studies on unsupervised learning-based anomaly detection methods in NIDS. Section 3 describes the hybrid-based anomaly detection and oversampling approaches based on the unsupervised machine and deep learning technologies used in this study. Section 4 presents the experimental results and analysis. Finally, Section 5 summarizes the main conclusions and directions for future research.

2. Related Work

Various machine learning techniques have been widely used in anomaly detection-based NIDS [21, 22, 23, 24]. Additionally, with the development of deep learning, various studies on supervised deep learning-based anomaly detection methods are being conducted. Kwon et al. [25] presented an overview of RBM-based DBN, DNN, and RNN in network anomaly detection. Performance results on the NSL-KDD dataset showed an accuracy of 84.2%. Xiao et al. [26] studied Convolution Neural Network (CNN)-based anomaly detection through dimensionality reduction algorithms (PCA and Autoencoder) using KDD-CUP99 dataset in supervised learning method. Zhang et al. [27] proposed a supervised deep learning-based anomaly detection method that fuses flow features learned from two branching (parallel) convolutional neural networks. The study focuses on improving the detection results of multi-class imbalanced abnormal flows. Zhang et al. [28] proposed a multilayer approach consisting of a CNN model (GoogLeNetNP) and fine layers based on gcForest (caXGBoost). GoogLeNetNP focuses on identifying abnormal and normal classes. caXGBoost then classifies the abnormal classes into subclasses to detect various attacks. Zhang et al. [29] proposed a compact multilayer perceptron (MLP)-based intrusion detection technique that uses a denoising autoencoder for feature selection in UNSW-NB dataset. YIN et al. [30] proposed RNN-IDS (recurrent neural networks based IDS) in both binary and multiclass classification using NSL-KDD dataset.

Recently, various researches on anomaly detection methods based on unsupervised machine and deep learning, which is an artificial intelligence implementation technology, are being conducted in NIDS. Existing unsupervised machine learning-based anomaly detection algorithms include LOF, IF, and OC-SVM etc. Local Outlier Factor (LOF) [19] is an unsupervised learning-based anomaly detection algorithm that calculates the local density deviation of the neighbors of a given data point, and Isolation Forest (IF) [18] identifies

anomalies by isolating them from data based on a decision tree algorithm. OC-SVM [31] is an unsupervised learning method that uses support vectors to create a hyperplane that divides normal data from outlier data and distinguishes normal from outliers based on the hyperplane. In the paper [32], a deep learning-based intrusion detection method using autoencoder and Isolation Forest in fog environment was proposed. This approach aims to differentiate normal packets from attacks in real time.

In addition, anomaly detection algorithms using unsupervised deep learning, such as autoencoder and deep SVDD (Deep Support Vector Data Description) [12], which are effective in handling non-linear data, have recently begun to be developed. An autoencoder is a type of unsupervised learning network used to learn efficient coding (latent vectors) through encoding and decoding functions. The encoding function (encoder) compresses the input to produce a coding vector, and the decoding function (decoder) uses only this coding vector to reconstruct the input. A method of detecting anomalies using a threshold according to the reconstruction error of the autoencoder has also been used [33], but there is a disadvantage that human subjective judgment must be included when setting the threshold. To overcome these shortcomings, an anomaly detection method using latent vectors generated from the encoder of an autoencoder was proposed [34]. Mishra et al. [4] proposed a method to handle the network intrusion detection problem by combining unsupervised and supervised methods. The unsupervised approach trains two autoencoders, each trained separately in the normal flow and the attack flow. The supervised approach trains a CNN classifier on the inputs and reconstructed outputs of the two autoencoders. Elsayed et al. [35] investigated unsupervised learning algorithms such as K-Means, Self-Organizing Maps, Deep Autoencoder Gaussian Mixture Model, and Adversarially Learned Anomaly Detection on two benchmark datasets for network-based anomaly detection, and also described the importance of integrating deep learning algorithms with traditional algorithms.

Despite advances in deep learning, class imbalance, which means imbalance between classes, is still a task that needs to be addressed [36]. There is usually a problem of class imbalance between the different types of attacks in an intrusion detection dataset. In supervised learning-based anomaly detection methods, there are generally techniques such as Random Oversampling, SMOTE (Synthetic Minority Over-sampling Technique) [37], and GAN (Generative Adversarial Network) [38] to solve the class imbalance problem that lowers the detection rate for minority classes. Various studies have been conducted to solve the class imbalance problem using various datasets such as NSL-KDD, CICIDS2017, and CSE-CIC-IDS2018 [39, 40, 41, 42, 43, 44]. Experimental results using oversampling methods in supervised learning-based binary and multiclass classification generally show better performance in terms of accuracy, recall, and precision.

In an unsupervised learning-based NIDS for anomaly detection, oversampling may be unnecessary. However, in this study, we used an oversampling approach to reduce false-negative rates by increasing the sample size of normal data and learning more normal behavioral patterns. NSL-KDD datasets were used to test the proposed approach for binary classification using a Variational Autoencoder (VAE)-based oversampling processing model.

3. Methodology

This section describes a series of hybrid approaches conducted to detect anomalies through unsupervised learning in NIDS. Section 3.1 explains the experimental dataset and data preprocessing used in this study. Section 3.2 describes the use of autoencoder to preserve the useful information of the experimental data. The latent vector extracted from autoencoder is

converted into an embedding vector, which is the input data for the next step. We also describe a hybrid anomaly detection method based on unsupervised deep learning / machine learning using Autoencoders, Isolation Forest, and Local Outlier Factor. Section 3.3 describes an oversampling approach to augment data in normal training set. Section 3.4 describes performance metrics for evaluating the anomaly detection method performed in this study. **Fig. 1** shows a schematic representation of the hybrid-based anomaly detection method proposed in this study.

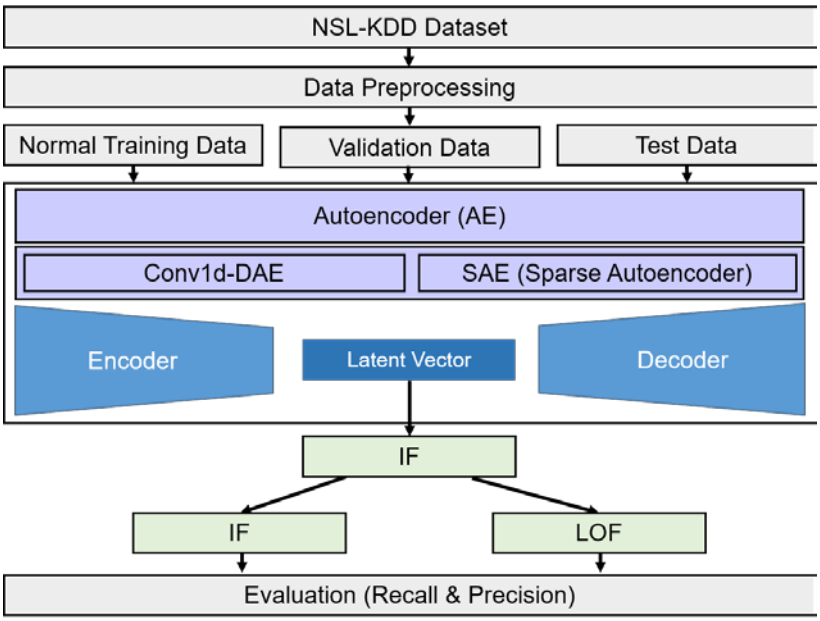

Fig. 1. Overall workflow of the unsupervised learning-based anomaly detection method performed in this study

3.1 Dataset description and preprocessing

As computers and network systems have evolved, new attack vectors and vulnerabilities have emerged. Therefore, datasets reflecting current attack vectors have been proposed, examples being the KDD-CUP'99 datasets. However, analysis of this dataset revealed a problem with many duplicate records. The analysis revealed that 78% and 75% of duplicate records were found in the training and test sets, respectively [19]. Redundancy in datasets can lead to bias in the training models. The NSL-KDD dataset had duplicate records removed to mitigate this problem. Although it imperfectly represents real networks and has limitations in that the amount of data per attack type is imbalanced, it remains widely used in network anomaly detection research.

The experiments in this study were conducted using NSL-KDD. The NSL-KDD dataset comprises 43 characteristics and labels that classify the difficulty and type of attacks (including normal attacks). In this study, difficulty was not considered, and the target value was the attack-type category. For data preprocessing, the categorical data such as protocol type, service, and flag were one-hot encoded, and each characteristic value of the dataset was normalized using a scaler (MinMaxScaler). The class types were divided into two classes: normal and abnormal (including DoS, Probe, R2L, and U2R attack classes). In this study, the

samples were classified into two classes for binary classification. **Table 1** lists the number of samples used in the unsupervised-learning experiments.

Table 1. Number of normal and abnormal samples generated

Number of Samples	Training Dataset	Validation Dataset	Test Dataset
Normal	53,874	13,469	9,710
Abnormal	0	11,726	12,833

3.2 Proposed hybrid intrusion detection approach based on unsupervised learning methods

This section describes a hybrid intrusion detection approach based on unsupervised deep and machine learning methods using autoencoders, isolation forests, and local outlier factors in the NIDS. A hybrid approach was used to reduce false positive rates and provide higher detection rates. The proposed approach comprises three steps for anomaly detection by integrating machine learning and deep learning algorithms. First, a one-dimensional convolutional neural network-based denoising autoencoder (Conv1d-DAE) extracts useful data representations (latent vectors) from normal data and defines normal behavior patterns from the latent vectors. The Conv1d-DAE was trained to learn normal patterns and extract latent representations of normal data. Second, another sparse autoencoder (SAE) is trained to learn normal patterns and extract latent representations from normal data. Third, the LOF and IF algorithms are trained by inputting the latent vectors extracted from the SAE. In the second step, the LOF and IF algorithms were trained by inputting the latent vectors extracted from Conv1d-DAE. The test data are fed into the trained Conv1d-DAE to obtain a latent representation. The latent vectors encoded in the Conv1d-AE for the test data were input into the LOF and IF trained by the Conv1d-DAE, as shown in **Fig. 2**. Subsequently, LOF and IF identified normal data and anomalies, respectively, and separated them into two datasets. That is, one was for normal data, and the other was for attack data. In the third step, the separated data were input into another IF or LOF, as shown in **Fig. 2**. The outliers were then identified as attack outliers in IF (left-hand side) and as attack outliers in LOF (right-hand side). Generally, IDSs must periodically be updated with newly acquired data and oversampling methods, such as VAE, were used to augment normal training data to increase detection rates by learning more normal behavioral patterns.

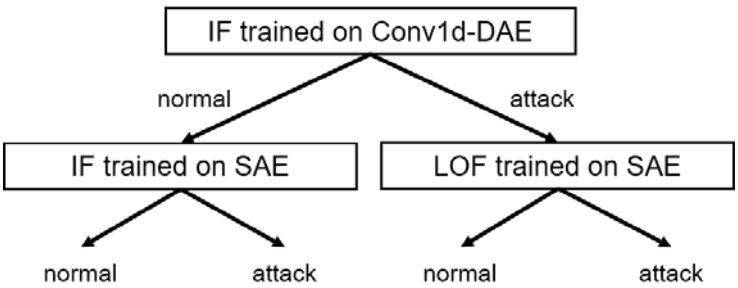

Fig. 2. Proposed hybrid intrusion detection approach based on unsupervised learning algorithms

3.2.1 Conv1D based denoising autoencoder (Conv1D-DAE) and Sparse autoencoder (SAE)

Conv1D is a one-dimensional (1D) convolutional neural network suitable for detecting one short pattern per kernel by sliding several kernels over input data in a one-dimensional convolutional layer. Conv1D-DAE is a Conv1D-based denoising autoencoder for preserving useful information by discarding unimportant details from the input data. Conv1D-DAE then extracts a latent representation of the input data from the compressed feature vector. The encoder consisted of three 1D convolutional layers and a linear transformation layer for latent vector extraction. A dropout layer was used as the encoder input. The decoder also consisted of three 1D convolutional layers and a linear transformation layer for input reconstruction. A SELU (Scaled Exponential Linear Unit) activation function was used for all 1D convolutional layers of the encoder and decoder. The autoencoder was configured to generate reconstruction data similar to the input data. The architectural details of the Conv1d-DAE are summarized in **Table 2**.

Table 2. Architectural details of Conv1d-DAE

Layer	Number of Kernel	Kernel Size	Stride	Activation Function	Padding
Conv1D_1	16	7	1	Selu	Yes
MaxPooling1D_1	-	-	-	None	-
Conv1D_2	32	5	1	Selu	Yes
MaxPooling1D_2	-	-	-	None	-
Conv1D_3	64	3	1	Selu	Yes
Dense_1	-	-	-	Selu	-
Dense (coding layer)	-	-	-	-	-
Conv1DTranspose_1	64	3	1	Selu	No
Conv1DTranspose_2	32	5	1	Selu	Yes
Conv1DTranspose_3	16	7	1	Selu	Yes
Dense_2	-	-	-	-	-

The SAE is an autoencoder for expressing useful features by a combination of a small number of activated nodes by reducing the number of nodes activated in the coding layer by adding an appropriate term to the cost function. The SAE was trained to learn normal patterns and extract latent representations of normal data. The encoder consists of three dense layers and an activity regularization layer ($\ell 1$ regularization to the coding layer's activations). The decoder consists of three dense layers. The architectural details of the SAE are summarized in **Table 3**.

Table 3. Architectural details of SAE

Layer	Number of Nodes	Activation Function
Dense_1	50	Selu
Dense_2	100	Selu
Dense_3	300	Sigmoid
Activity Regularization	-	None
Dense_4 (coding layer)	-	Selu
Dense_5	300	Selu
Dense_6	100	Selu
Dense_7	50	Selu
Dense_8 (reconstruction layer)	115	Sigmoid

3.2.2 IF (Isolation Forest)

Isolation Forest is an unsupervised anomaly detection algorithm based on decision trees that randomly selects features and then separates the samples by assigning higher anomaly scores to those requiring fewer splits. The outlier score of the sample was calculated as the average outlier score of trees in the forest. A sample's normality is measured, given a tree, by the depth of the leaf containing it, which is equal to the number of splits required to separate it. Thus, a forest of random trees collectively produces shorter path lengths for anomalies. Because IF has low computational and memory costs compared with distance- or density-based unsupervised anomaly-detection algorithms [14], the proposed hybrid approach used the IF algorithm to detect anomalies in the test data. The Python package scikit-learn was used for the IF algorithm [44].

3.2.3 LOF (Local Outlier Factor)

This is an unsupervised anomaly-detection method that computes the local deviation of a given sample's density with respect to its neighbors. The anomaly score depends on how isolated an object is relative to its surrounding neighborhood. Locality is given by the k-nearest neighbors, whose distances are used to estimate the local density. By comparing the local densities of samples, outliers have lower densities than their neighbors. In this study, we used the Python package from scikit-learn [45].

3.3 Data oversampling

This was studied using a VAE for data oversampling to augment the normal samples in the training dataset. The VAE is a type of unsupervised learning model used for dimensionality reduction, visualization, and feature extraction [46]. As a generative model, it can also be used for data oversampling. For learning the algorithm can, therefore, produce new data that are similar to the training (or input) dataset. Thus, a data augmentation approach using a VAE network was applied to the preprocessed normal training data. The encoder in the VAE network produces a mean coding μ and a standard deviation σ. The actual coding is then sampled randomly from a Gaussian distribution with the mean μ and the standard deviation σ. The network decoder then decodes the sampled coding. For oversampling, data similar to the normal input data were generated by sampling random coding from a Gaussian distribution using a VAE decoder trained on normal training data. The encoder consists of three dense layers and a sampling layer for coding-vector extraction. The decoder also consists of three dense layers and a linear transformation layer for input reconstruction. A SELU activation function was used for all the layers of the encoder and decoder. The architectural details of the VAE are summarized in **Table 4**.

Table 4. Architectural details of VAE

Layer	Number of Nodes	Activation Function
Dense_1	100	Selu
Dense_2	50	Selu
Dense_3	30	Selu
Sampling layer (μ, σ)	-	None
Dense_4	30	Selu
Dense_5	50	Selu
Dense_6	100	Selu

3.4 Performance metrics

This section briefly describes the performance metrics used to evaluate the proposed anomaly-detection method used in this study. For evaluation, the True Positive Rate (TPR) and false alarm rate (FPR) were defined as

True Positive Rate (TPR) or Recall: TP / (TP +FN) = the proportion of data samples correctly identified as belonging to a positive class.
False Positive Rate (FPR) or Precision: TP / (TP +FP) = number of true positives divided by the total number of positive predictions, where P represents real positive (anomalous) cases in the data, N represents real negative (normal) cases in the data, TP represents true positive, FP represents false positive, TN represents true negative, and FN represents false negative.
Precision refers to the accuracy of the positive predictions. The recall (or true positive rate) refers to the proportion of positive instances that a classifier correctly detects. The F1-score is the harmonic average of the precision and recall for combining them into a single metric.

4. Experimental results and analysis

This section presents the experimental results of the proposed hybrid anomaly-detection approach using AEs, LOFs, and IFs, and further describes an oversampling approach to increase detection rates. Section 4.1 describes the anomaly detection results using IF and LOF with Conv1d-DAE for different latent vector dimensions. Section 4.2 describes the experimental results of the proposed hybrid anomaly detection approach. Section 4.3 describes the experimental results of an oversampling approach to improve the detection performance by learning more normal patterns. Section 4.4 describes the model performance according to the learning time and detection time of the proposed approach. The experimental conditions are listed in **Table 5**. The hyperparameters of the LOF and IF algorithms used in these experiments are listed in **Table 6**.

Table 5. Experimental environment

OS	Ubuntu 18.04.6 LTS
CPU	Intel(R) Xeon(R) Gold 5120
GPU	NVIDIA RTX A5000
RAM	264GB
Python	3.10.9
Scikit-Learn	1.2.1
Keras	2.11.0

Table 6. Hyperparameters of IF and LOF algorithms used in the experiment

	Number of Neighbors	**Contamination**	**Number of Estimators**	**Max of samples**
LOF	10 ~ 20	0.1 ~ 0.4	None	None
IF	None	0.1 ~ 0.4	50 ~ 100	10 ~ 20

4.1 Anomaly detection results using IF and LOF, respectively, with Conv1d-DAE for different latent vector dimensions

This section describes the anomaly detection results using IF and LOF for latent vectors drawn from the training samples using Conv1d-DAE. The Conv1d-DAE was trained with samples consisting entirely of normal data. We investigated the impact of vector dimensions on the detection performance using latent vector (embedding vectors for the next steps) dimensions of 15, 25, 30, and 50 generated from the encoder of the trained Conv1d-DAE. IF and LOF were trained using the generated embedding vectors. To evaluate the anomaly detection performance, the NSL-KDDTest+ dataset, consisting of normal and abnormal data (normal:9,710, attack:12,833), was used. A latent vector for the test data was extracted using the encoder of the learned Conv1d-DAE. The detection performance was evaluated using the extracted latent vector as the input to the learned IF and LOF. The experimental results are shown in **Table 7** and **Fig. 3**. Each of the best performances is shown in bold and underscored. These results show that the anomaly detection performance can vary according to the latent vector dimensions. The experimental results show that 25-dimensional embedding vectors perform best on average. Based on the obtained experimental results, the vector dimension was used as the dimension for latent vector extraction in the following experiments. Additionally, when comparing IF and LOF, the experimental results showed that IF had higher precision and recall scores than LOF. In the following experiments, these observations were considered in the hybrid anomaly-detection approach.

Table 7. Performance of anomaly detection results using LOF and IF for latent vectors extracted from Conv1d-DAE

Number of test samples	Vector dim.	LOF			IF		
		Precision	Recall	F-1 score	Precision	Recall	F-1 score
NSL-KDDTest+ datasets - normal: 9,710 - attack: 12,833	15	0.82	0.918	0.866	0.838	0.967	0.898
	25	**0.822**	**0.933**	**0.874**	**0.848**	**0.969**	**0.904**
	30	0.816	0.903	0.857	0.839	0.963	0.897
	50	0.817	0.918	0.864	0.843	0.96	0.897

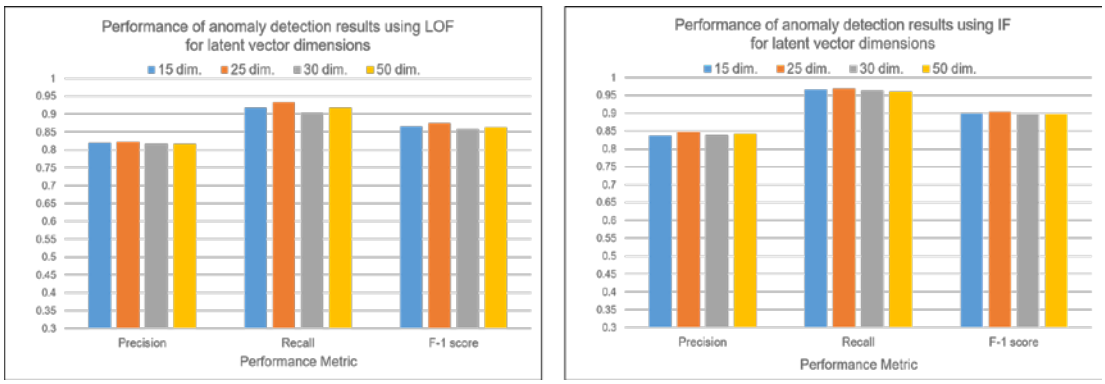

Fig. 3. Performance of anomaly detection results using LOF and IF for latent vectors extracted from Conv1d-DAE

4.2 Experimental results of the proposed hybrid approach

This section describes the results of the proposed hybrid-based anomaly-detection method that combines AE with IF and LOF and considers the latent vector dimensions obtained from previous experiments. The learned autoencoders (Conv1d-DAE and SAE) generated 25-dimensional latent vectors from normal training samples. The IF and LOF are then trained using latent vectors. The latent vectors encoded for the test data were input into the trained IF and LOF, as shown in **Fig. 2**. Subsequently, the IF and LOF identified the normal and anomalies, respectively, and separated them into two datasets. Thereafter, the separated data were input to each IF or LOF. The outliers were then identified as attack outliers in IF (left-hand side) and as attack outliers in LOF (right-hand side). The hybrid architecture was composed as shown in **Fig. 2**. The experimental results are shown in **Table 8** and **Fig. 4**. The best performance is shown in bold with an underscore. The experimental results show that the hybrid approach can improve anomaly detection performance by reducing the false alarm rate. Through experiments, we confirmed that the hybrid method of combining IF (left-hand side) and LOF (right-hand side) had higher precision and recall than combining IF (left-hand side) and IF (left side) alone, even though IF had higher precision and recall scores than LOF, as shown in previous experiments. This is likely because the right-hand side LOF process samples were identified as outliers of the trained IF on the Conv1d-DAE. Therefore, LOF, which computes distances to estimate the local variation for a given sample's neighbors, appears to perform better than IF, which separates samples based on decision trees.

Table 8. Performance of hybrid-based (IF and LOF combined) anomaly detection results on latent vectors extracted from Conv1d-DAE and SAE (vector dimension: 25)

Number of test samples	**Combining with IF and LOF**			**Combining with IF and IF**		
	Precision	Recall	F-1 score	Precision	Recall	F-1 score
NSL-KDDTest+ datasets - normal: 9,710 - attack: 12,833	**<u>0.882</u>**	**<u>0.928</u>**	**<u>0.904</u>**	0.859	0.925	0.891

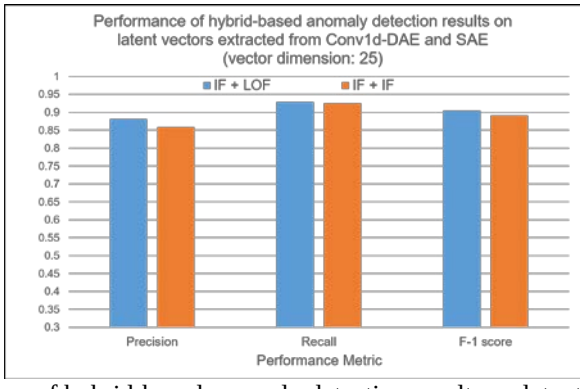

Fig. 4. Performance of hybrid-based anomaly detection results on latent vectors extracted from Conv1d-DAE and SAE (vector dimension: 25)

4.3 Experimental results of an oversampling approach that improves detection performance by learning more normal patterns

This study assumes that learning with a larger number of normal training samples may improve the detection performance compared to learning with fewer training samples because more samples can be used to extract more unknown normal patterns. To validate this assumption, an additional experiment was performed to determine the extent to which the detection performance could be improved if additional normal samples were obtained by applying an oversampling method instead of collecting new types of samples, which is expensive. We present the experimental results regarding measures of precision and recall and compare them with the experimental results in the previous section. The oversampling approach uses the VAE, a generative deep learning model. The data augmentation approach using the VAE was applied only to normal samples with no anomalies in a given training dataset. The VAE networks learn to generate new data similar to the normal training data. The VAE decoder generates new normal samples to increase the diversity of the normal samples. Subsequently, the Conv1d-DAE and SAE were trained using augmented normal samples as well as existing normal samples.

Table 9. Performance of oversampling-based anomaly detection results using augmented data from VAE (vector dimension: 25)

Number of training and oversampling samples	Combining with IF and LOF			Combining with IF and IF		
	Precision	Recall	F-1 score	Precision	Recall	F-1 score
NSL-KDDTrain+ + 1,000 samples	**0.883**	**0.932**	**0.907**	0.841	0.902	0.871
NSL-KDDTrain+ + 3,000 samples	**0.882**	0.946	**0.913**	0.871	**0.950**	0.909
NSL-KDDTrain+ + 5,000 samples	**0.877**	0.925	0.9	0.852	**0.965**	**0.905**
NSL-KDDTrain+ + 7,000 samples	**0.875**	0.922	**0.898**	0.844	**0.923**	0.882
NSL-KDDTrain+ + 10,000 samples	**0.873**	0.947	**0.908**	0.831	**0.964**	0.893

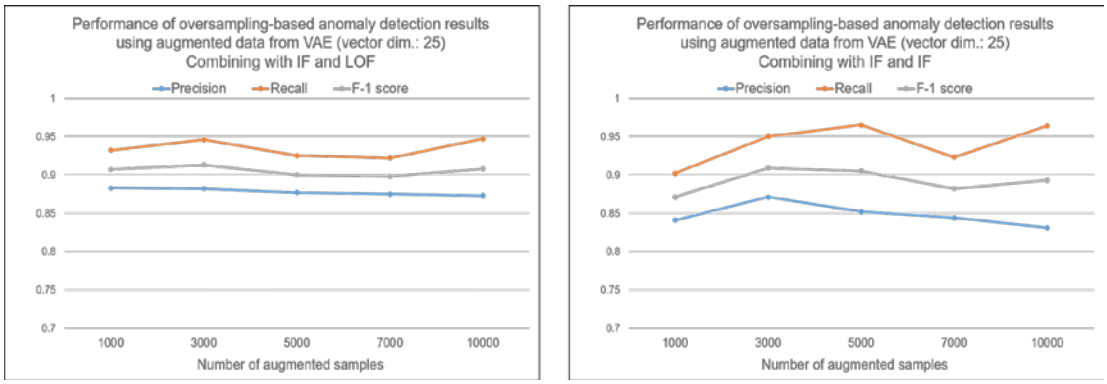

Fig. 5. Performance of oversampling-based anomaly detection results using augmented data from VAE (vector dimension: 25)

We used the previously mentioned NSL-KDDTest+ dataset to evaluate the performance of oversampling-based anomaly detection. Experiments using augmented data showed that the data level-based oversampling approach can provide higher detection rates. Therefore, the proposed approach provides high reliability for cyberattack detection in NIDS. The experimental results are shown in **Table 9** and **Fig. 5**. The best performance is shown in bold with an underscore.

In this section, we evaluate the proposed approach on the NSL-KDDTrain + and NSL-KDDTest+ datasets for training and testing, respectively. The proposed method achieved a precision and recall respectively of 88.2% and 92.8% in the hybrid form combining AE, IF, and LOF to further improve the precision. Additionally, the over-sampling method achieved a precision and recall respectively of 88.2% and 94.6%, further improving the recall rate compared with the hybrid method. Most of the previous IDS methods using the NSL-KDD dataset focus on supervised learning trained on labeled datasets. On the other hand, the proposed research focuses on unsupervised learning, which involves training algorithms on unlabeled datasets. The proposed study assumed that there was no labeled data in the experimental dataset, despite the presence of labeled data in the dataset. Compared with the methods mentioned in [47], the approach proposed in this study showed better results for cyberattack detection. Therefore, the proposed study provides high reliability for cyberattack detection in NIDS.

4.4 Performance evaluation according to training time and detection time

In this section, we describe the performance of the model according to learning time and detection time to confirm whether the proposed approach is efficient for real-time detection. The average learning time of Conv1d-DAE, SAE, and VAE models and the average detection time using a method combining IF and LOF in the hybrid approach were measured. In addition, the average learning time of the developed models according to the augmented training data was measured. To prevent the model from overfitting the training data and improve generalization performance on the test data, learning and detection performance were measured using the early stopping technique. As shown in **Table 10** and **11**, the proposed method was confirmed to be suitable and effective for real-time detection according to the measured detection time.

Table 10. Performance of the developed models according to learning time

Number of training samples	**Conv1d-DAE**		**SAE**		**VAE**	
	learning time in minutes					
	mean	std	mean	std	mean	std
53874	4.15	0.28	10.54	0.84	7.3	1.04
53874 + 1,000	4.16	0.72	11.52	0.61	n/a	n/a
53874 + 3,000	4.31	0.68	11.50	0.53	n/a	n/a
53874 + 5,000	4.39	0.89	11.58	0.66	n/a	n/a
53874 + 7,000	4.59	0.99	12.06	1.59	n/a	n/a
53874 + 10,000	5.26	0.69	13.34	1.88	n/a	n/a

In general, learning times tend to increase as dataset size increases, but other factors such as model complexity, batch size, and hardware performance may have an impact. For example, the overall time complexity of training a model can be expressed in terms of factors other than dataset size. As the dataset size (n) increases, the learning time complexity is denoted as O(n). More complex models can increase learning time (O(m)), where m represents an additional factor related to model complexity. Additionally, considering batch size and hardware performance, the time complexity of model training is related to O(b) and O(h), where b is the batch size related to memory requirements and h is hardware performance based on GPU or TPU usage. Then, taking these factors into account, the overall time complexity of model training can be expressed as O(n·m·b·h). Therefore, factors beyond dataset size can affect the overall time complexity of model training. Intrusion detection systems must minimize detection delays. Since the model training time varies greatly depending on the complexity of the model, batch Size, and hardware performance, we will consider developing a model that can reduce the training time more reliably in the future.

Table 11. Performance of the proposed hybrid approach according to detection time

	Combining with IF and LOF				**Combining with IF and IF**			
	learning time in seconds		detection time in seconds		learning time in seconds		detection time in seconds	
	mean	std	mean	std	mean	std	mean	std
IF	1.08	0.04	0.36	0.014	1.08	0.04	0.36	0.014
IF (left side)	0.874	0.028	0.087	0.038	0.874	0.028	0.087	0.038
LOF (right side)	1.314	0.167	0.396	0.052	n/a	n/a	n/a	n/a
IF (right side)	n/a	n/a	n/a	n/a	0.88	0.025	0.086	0.038
Total	3.268	0.08	0.843	0.035	2.834	0.031	0.98	0.03

5. Conclusions

This study developed a hybrid anomaly-based NIDS with good detection accuracy and minimal false-positive detection. Owing to the advanced nature of cyberattacks, artificial intelligence (AI) technology was applied to overcome the difficulty of detecting new types of cyberattacks in NIDS. The unsupervised machine-learning and deep-learning technologies AE, IF, and LOF were used. The experimental results show that the proposed hybrid-based anomaly-detection method significantly improves the attack detection rate (reduces the false alarm rate). This study confirms that, depending on the availability of normal data, detection performance can be improved by using AI methods. The normal samples can then be augmented using generative models such as the VAE. The experimental results obtained using augmented data show that the proposed approach increases the detection rate. Therefore, it provides considerable reliability in detecting cyberattacks in a NIDS.

The hybrid approach that integrates unsupervised deep learning algorithms with traditional unsupervised machine learning algorithms was highly effective without requiring subjective human judgment to set thresholds for anomalies. In addition, the oversampling approach to learn more unknown normal data somewhat improved the detection accuracy. In the future, these findings will inform further research and investigation to develop more effective IDS for various intrusion detection datasets.

Large language model (LLM) is a model trained to understand and produce human-like language. Network logs often contain textual information that describes events. LLMs can then process these logs. Therefore, LLM-based feature extraction can obtain more comprehensive features from information extracted from IDS target data. In future work, we

will consider applying LLM techniques, such as transformers, to develop an anomaly detection model that can identify patterns that significantly deviate from normal behavior.

References

[1] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer Networks*, vol. 31, no. 8, pp. 805-822, Apr. 1999. Article (CrossRef Link)

[2] M. Aljanabi, et al., "Intrusion detection systems, issues, challenges, and needs," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 560-571, 2021. Article (CrossRef Link)

[3] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences*, vol. 9, no. 20, 4396, pp. 1-28, 2019. Article (CrossRef Link)

[4] P. Mishra, V. Varadharajan, U. Tupakula and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686-728, First quarter 2019. Article (CrossRef Link)

[5] A. Rashid, M. J. Siddique, and S. M. Ahmed, "Machine and deep learning based comparative analysis using hybrid approaches for intrusion detection system," in *Proc. of the 2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*, Lahore, Pakistan, pp. 1-9, 17-19 Feb. 2020. Article (CrossRef Link)

[6] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42-57, Jan. 2013.

[7] A. Azab, M. Alazab, and M. Aiash, "Machine learning based botnet identification traffic," *IEEE Trustcom/BigDataSE/ISPA*, Tianjin, pp. 1788-1794, 2016. Article (CrossRef Link)

[8] A.M. Mahfouz, D. Venugopal, and S.G. Shiva, "Comparative analysis of ML classifiers for network intrusion detection," *Fourth International Congress on Information and Communication Technology*, vol. 1027, pp. 193-207, 2020.

[9] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," *Expert Systems with Applications*, vol. 29, no. 4, pp. 713-722, Nov. 2005. Article (CrossRef Link)

[10] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, "Anomaly-based network intrusion detection: techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18-28, 2009. Article (CrossRef Link)

[11] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," *IEEE Transactions on Computers*, vol. 63, no. 4, pp. 807-819, Apr. 2014.

[12] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proc. of the 35th International Conference on Machine Learning*, PMLR, vol. 80, pp. 4393-4402, 2018. Article (CrossRef Link)

[13] A. Torkaman, G. Javadzadeh, and M. Bahrololum, "A hybrid intelligent HIDS model using two-layer genetic algorithm and neural network," in *Proc. of 5th Conference on Information and Knowledge Technology (IKT)*, pp. 92-96, 28-30 May 2013. Article (CrossRef Link)

[14] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525-41550, Apr. 2019. Article (CrossRef Link)

[15] E. Eskin, A. Arnold, M. Prerau, L. Portnoy and S. Stolfo, "A geometric framework for unsupervised anomaly detection," *Applications of Data Mining in Computer Security*, vol. 6, pp. 77-101, 2002. Article (CrossRef Link)

[16] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, no. 2, pp. 233-243, Feb. 1991. Article (CrossRef Link)

[17] F. T. Liu, K. M. Ting and Z. Zhou, "Isolation forest," in *Proc. of Eighth IEEE International Conference on Data Mining*, pp. 413-422, Pisa, Dec. 2008. Article (CrossRef Link)

[18] M. M. Breunig et al., “LOF: Identifying density-based local outliers,” in *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas Texas, USA, pp. 93-104, May 15-18, 2000. Article (CrossRef Link)
[19] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” in *Proc. of Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2009. Article (CrossRef Link)
[20] A. F. M. Agarap, “A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data,” in *Proc. of the 10th International Conference on Machine Learning and Computing (ICMLC 2018)*, pp. 26-30, NY, USA, Feb. 2018. Article (CrossRef Link)
[21] M. S. M. Pozi, M.N. Sulaiman, N. Mustapha, and T. Perumal, “Improving anomalous rare attack detection rate for intrusion detection system using support vector machine and genetic programming,” *Neural Processing Letters*, vol. 44, no. 2, pp. 279-290, 2016. Article (CrossRef Link)
[22] A. Binbusayyis, and T. Vaiyapuri, “Identifying and benchmarking key features for cyber intrusion detection: an ensemble approach,” *IEEE Access*, vol. 7, pp. 106495-106513, 2019. Article (CrossRef Link)
[23] T. T. Bhavani, M. K. Rao, and A. M. Reddy, “Network intrusion detection system using random forest and decision tree machine learning techniques,” in *Proc. of First International Conference on Sustainable Technologies for Computational Intelligence*, vol. 1045, pp. 637-643, 2020.
[24] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, “A survey of deep learning-based network anomaly detection,” *Cluster Computing*, vol. 22, pp. 949-961, Jan. 2019. Article (CrossRef Link)
[25] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, “An intrusion detection model based on feature reduction and convolutional neural networks,” *IEEE Access*, vol. 7, pp. 42210-42219, Mar. 2019. Article (CrossRef Link)
[26] Z. Yong, C. Xu, G. Da, S. Mei, T. Yinglei, and W. Xiaojuan, “PCCN: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows,” *IEEE Access*, vol. 7, pp. 119904-119916, Aug. 2019. Article (CrossRef Link)
[27] X. Zhang, J. Chen, Y. Zhou, L. Han, and J. Lin, “A multiple-layer representation learning model for network-based attack detection,” *IEEE Access*, vol. 7, pp. 91992-92008, July 2019. Article (CrossRef Link)
[28] H. Zhang, C. Q. Wu, S. Gao, Z. Wang, Y. Xu, and Y. Liu, “An effective deep learning based scheme for network intrusion detection,” in *Proc. of 24th International Conference on Pattern Recognition (ICPR)*, pp. 682–687, 20-24 Aug. 2018. Article (CrossRef Link)
[29] 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, Oct. 2017. Article (CrossRef Link)
[30] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *Journal of Machine Learning Research*, vol. 2, no. 1, pp. 139-154, Mar. 2002. Article (CrossRef Link)
[31] K. Sadaf and J. Sultana, “Intrusion detection based on autoencoder and isolation forest in fog computing,” *IEEE Access*, vol. 8, pp. 167059-167068, Sept. 2020. Article (CrossRef Link)
[32] P. Bergmann et al., “Improving unsupervised defect segmentation by applying structural similarity to autoencoders,” in *Proc. of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pp. 372-380, 2019. Article (CrossRef Link)
[33] S. Pidhorskyi, R. Almohsen, D. A. Adjeroh and G. Doretto, “Generative probabilistic novelty detection with adversarial autoencoders,” in *Proc. of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018)*, pp. 6823-6834, Dec. 2018. Article (CrossRef Link)
[34] M. S. Elsayed et al., “Network anomaly detection using LSTM based autoencoder,” in *Proc. of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp. 37-45, Alicante, Spain, Nov. 2020. Article (CrossRef Link)

[35] N. Japkowicz, "The class imbalance problem: significance and strategies," in *Proc. of the International Conference on Artificial Intelligence (IC-AI' 2000)*, vol. 1, pp. 111-117, 2000.
[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, June 2002. Article (CrossRef Link)
[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139-144, Nov. 2020. Article (CrossRef Link)
[38] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, pp. 1-27, Mar. 2019. Article (CrossRef Link)
[39] Y. Zhu, J. Liang, J. Chen, and Z. Ming, "An improved NSGA-III algorithm for feature selection used in intrusion detection," *Knowledge-Based Systems*, vol. 116, no. 15, pp. 74-85, Jan. 2017. Article (CrossRef Link)
[40] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network," *Sensors*, vol. 19, no. 11, pp. 1-20, June 2019. Article (CrossRef Link)
[41] I. Ullah, and Q. Mahmoud, "A two-level hybrid model for anomalous activity detection in IoT networks," in *Proc. of 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1-6, Las Vegas, NV, USA, 11-14 Jan. 2019. Article (CrossRef Link)
[42] G. Karatas, O. Demir, and O.K. Sahingoz, "Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset," *IEEE Access*, vol. 8, pp. 32150-32162, Feb. 2020. Article (CrossRef Link)
[43] H. Zhang, L. Huang, C.Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Computer Networks*, vol. 177, Aug. 2020. Article (CrossRef Link)
[44] Isolation Forest (IF). [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html
[45] Local Outlier Factor (LOF). [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html
[46] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114v11*, Dec. 2022. Article (CrossRef Link)
[47] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843-52856, Sept. 2018. Article (CrossRef Link)

Kangseok Kim was born in Seoul, South Korea. He received the B.S. degree in Mathematics from Gachon University, Korea, the M.S. degree in Computer Engineering from Syracuse University, Syracuse, NY, USA, and his Ph.D. in Computer Science from Indiana University at Bloomington, IN, USA. He is currently an associate professor in the Department of Cyber Security, College of Computing and Informatics at Ajou University, Suwon, Korea. His main research interests include applied security using machine learning and deep learning in the field of big data.