


Article

AE-CGAN Model based High Performance Network Intrusion Detection System

JooHwa Lee  and KeeHyun Park *

Department of Computer Engineering, Keimyung University, Daegu 42601, Korea; yezi1004@gmail.com

* Correspondence: khp@kmu.ac.kr; Tel.: +82-10-7705-5266

Received: 4 September 2019; Accepted: 4 October 2019; Published: 10 October 2019



Abstract: In this paper, a high-performance network intrusion detection system based on deep learning is proposed for situations in which there are significant imbalances between normal and abnormal traffic. Based on the unsupervised learning models autoencoder (AE) and the generative adversarial networks (GAN) model during deep learning, the study aim is to solve the imbalance of data and intrusion detection of high performance. The AE-CGAN (autoencoder-conditional GAN) model is proposed to improve the performance of intrusion detection. This model oversamples rare classes based on the GAN model in order to solve the performance degradation caused by data imbalance after processing the characteristics of the data to a lower level using the autoencoder model. To measure the performance of the AE-CGAN model, data is classified using random forest (RF), a typical machine learning classification algorithm. In this experiment, we used the canadian institute for cybersecurity intrusion detection system (CICIDS)2017 dataset, the latest public dataset of network intrusion detection system (NIDS), and compared the three models to confirm efficacy of the proposed model. We compared the performance of three types of models. These included single-RF, a classification model using only a classification algorithm, AE-RF which is processed by classifying data features, and the AE-CGAN model which is classified after solving the data feature processing and data imbalance. Experimental results showed that the performance of the AE-CGAN model proposed in this paper was the highest. In particular, when the data were unbalanced, the performances of recall and F1 score, which are more accurate performance indicators, were 93.29% and 95.38%, respectively. The AE-CGAN model showed much better performance.

Keywords: NIDS; GAN; autoencoder; imbalanced data; deep learning; oversampling

1. Introduction

Highly accurate and targeted attacks are being made due to the increase in data collected from various sources. These attacks use malicious code to target a large network and destroy the authority of the administrator [1]. To defend against these attacks, a variety of security defense solutions are available, but the sophisticated methods used by cybercriminals in their attacks make the majority of solutions ineffective.

Network intrusion detection system (NIDS) is an important security defense technology, and the detection methods it uses can be divided into misuse detection method and abnormal behavior detection method [2]. As many new hacking methods and variations are emerging every day, any misuse detection system where the rules are made and maintained by experts will face limitations. In particular, the security of the network is a prerequisite in the current system, which must be based on the detection of abnormal behavior. The study of abnormal behavior detection techniques is essential and is the ultimate goal of the NIDS [3].

The abnormal behavior detection method is defined as an intrusion when events occur that cause a relatively rapid change, or that are less likely to occur, based on normal and average conditions [4].

Currently, research on abnormal behavior detection is actively being carried out using learning algorithms, but more recently, there have been a number of studies attempting to merge machine learning with deep neural network (DNN), which showed high performance in image recognition and voice recognition [5–7]. But there are some problems with these NIDS studies.

First, most data sets used in traditional NIDS studies, such as knowledge discovery in databases (KDD) 99 [8] and NSLKDD [9], are not reflective of the latest attacks, and lack the level of diversity and volume of traffic to reflect current attacks [10].

Second, if NIDS is actually used in a large network environment, it faces limitations in time and space complexity [11]. The essential reason is that it is data with high dimensions and nonlinear characteristics. As such, extracting only the important features from high-level data is an essential step in improving detection speed and detection performance, because it is dimensionally reduced.

Third, a common problem with high-capacity network traffic data is that it is often misinterpreted for sparse classes due to data imbalances [12]. Imbalanced data means that the ratio between data classes varies greatly.

Therefore, in this study, we intend to utilize autoencoder (AE) and generative adversarial nets (GAN), which are representative unsupervised deep learning models, using CICIDS 2017 datasets [13] that reflect the latest attacks. Reducing high-dimensional data to lower dimensions based on AE can improve intrusion detection performance and reduce classification time. To solve the imbalance of data, we propose a model that improves detection performance by balancing the data by oversampling the rare class of data using a conditional GAN (CGAN) [14] model applied to the original GAN model.

We compared the performance of three types of models. These included single-random forest (single-RF), a classification model using only a classification algorithm, AE-RF which is processed by classifying data features, and the autoencoder-conditional generative adversarial nets (AE-CGAN) model which is classified after solving the data feature processing and data imbalance. Thus, we confirm that feature processing and data imbalance problem-solving play an important role in NIDS.

The main contributions of this study are as follows:

- Using the CICIDS2017 dataset, which was collected in an environment similar to the real network and reflects the latest attacks, we propose a NIDS suitable for the actual situation.
- The AE-CGAN model is proposed to solve the degradation of detection performance due to high-dimensional features and data imbalances that occur in large network environments.
- AE and GAN are the most actively researched subjects in the area of deep learning and are being applied in a variety of fields ranging from image generation to voice and text, which have also been identified as contributing to performance improvement in NIDS.

2. Related Works

2.1. A Studies on Feature Extraction

Studies have been conducted on various approaches to feature selection and dimension reduction to address the problem of low detection rates and poor generalization capabilities when NIDS is used in large network environments

As a study on feature extraction, Harshal A. Sonamane [15] used the KDD99 dataset and, based on PCA (principal component analysis) technology, selected the characteristics of the dataset and compared the performance using the dataset's full characteristics. Comparisons show that using the full features increases accuracy but also increases delays in terms of time and memory. The PCA is not much higher than the autoencoder algorithm proposed in this paper because PCA projects the data in the dimension that maximizes the variance by using only linear properties without learning process.

Majjed Al-Qatf et al. [16] used NSL-KDD datasets to suggest learning methods based on SELL (self =taught learning) framework by combining SAE (sparse autoencoder) and SVM (support vector machine) to detect network intrusion. This approach dramatically reduces learning and testing time and effectively improves SVM's predictive accuracy. The method of dimension reduction using

autoencoder is the same as presented in this paper. However, the SVM classifier increases the learning time when using a large volume dataset. Therefore, in this study, the classifier uses random forest to reduce the learning time and the test time to process large network data.

In another impressive study using autoencoder, Dimitrois Paamartzivano et al. [17] suggested a deep learning self-adaptive use network input detection system. In this study, we propose an IDS (intrusion detection system) that can adapt itself to changes continuously in the network environment by making the best use of a sparse autoencoder (SAE) for unsupervised learning. AE can be used as a feature extraction method and can be employed to directly detect intrusions.

In recent NIDS, the deep learning approach has used models of deep neural network, self-learning lairing (STL), and current neural network (RNN). Clifford Green et al. [18] used autoencoder as models of STL and were able to classify attack types accurately. However, DNN or RNN relies heavily on the learning of models, so when there is an imbalance in the data, they are significantly less accurate, and the performance is inconsistent according to the hyper-parameters used for learning. In addition, in real-world network environments based on large amounts of data, there is a significant slowdown in detection rates when NIDS is used with deep learning.

NIDS is primarily aimed at improving the detection accuracy of attack data. Therefore, this study aims to extract features based on autoencoder, a deep learning model commonly used in NIDS, and to improve the detection accuracy of attack data by applying the latest deep learning model, the GAN model.

2.2. A Studies on Imbalanced Data

In order to solve the data imbalance problem, previous research has used random oversampling, random undersampling, and the SMOTE (synthetic minority oversampling technique) [19] techniques are available. In the case of random undersampling, an important feature may be missed, resulting in inaccurate results. Random oversampling may cause the problem of overfitting because the same data is randomly copied [20]. A key idea in SMOTE is to find data that is close to the entered sparse class data, and randomly sample data within the range.

Yan B. et al. [21] performed sampling using SMOTE considering data levels using NSL-KDD datasets to optimize data and improve performance using sampling techniques. The sampled datasets were combined with several classification algorithms such as SVM, RF, and backpropagation neural network (BPNN) to compare performance. The SMOTE technique is widely used to solve the data imbalance problem, but it has a problem of degrading the classification performance because classes overlap or make noise.

Sun Y. et al. [22] proposed an improved SMOTE-NCL (neighborhood cleaning rule) based on SMOTE. SMOTE-NCL calculates the ratio of each class, the average ratio calculated from it, the standard deviation of the class ratio, and the unbalanced scale divided by the class percentage, and uses SMOTE to sample sparse class data until the unbalanced scale exceeds the threshold. Finally, after sampling, a method was proposed to handle data considered to be noise through the neighborhood cleaning rule. The SMOTE-NCL improves the SMOTE. Its disadvantage, however, is that the performance of the classifier is greatly affected by the data sampled from the major data category when oversampling sparse data. The CGAN proposed in this study shows better performance because it is independent of the distance between data and is oversampled by the generation algorithm.

In another study on GAN, Zilong Lin et al. [23] proposed an IDSGAN using the GAN model to create an attack that could cheat and evade an intrusion detection system. IDSGAN leverages the generator to convert the original malicious traffic into a malicious traffic example. Considering that the internal structure of the detection system is unknown to attackers, examples of adversarial attacks perform the black-box attacks against the detection system. We used GAN to solve the data imbalance, and the above research differed in that it developed the IDS by making malicious data based on GAN.

3. Deep Learning-Based Feature Extraction and Generation Model

3.1. AutoEncoder

Autoencoder (AE) is a type of learning neural network with unsupervised learning using encoding and decoding processes, which is mainly used for dimension reduction and feature extraction. The structure of the AE consists of an input layer, a hidden layer, and an output layer as shown in Figure 1, and the size of the hidden layer must be smaller than the input layer [24].

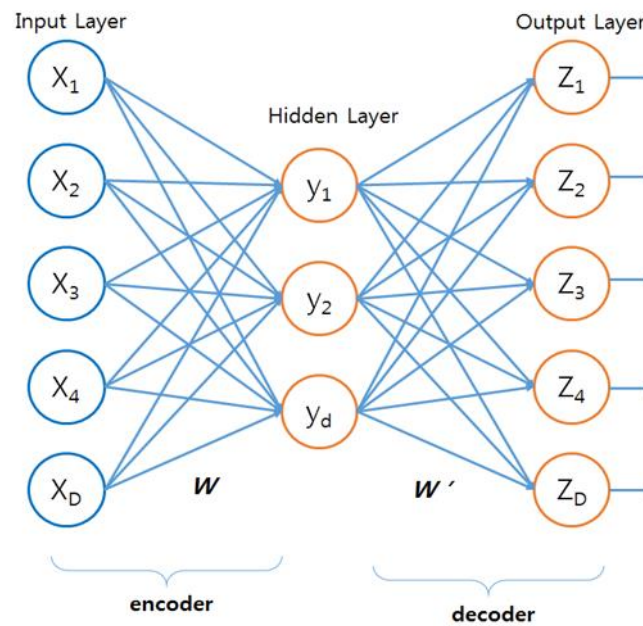


Figure 1. The structure of autoencoder.

The input vector is $X \in [0, 1]^D$, the hidden representation is $y \in [0, 1]^d$, and the constructed vector is $Z \in [0, 1]^D$. The coding process from input layer to hidden layer:

$$y = f_{\theta}(X) = s(WX + b) \quad (1)$$

Decoding process from hidden layer to output layer:

$$Z = g_{\theta'}(Y) = s(W'Y + b') \quad (2)$$

b and b' are the respective bias vectors of input layer and hidden layer and f_{θ} and g_{θ} are the active functions of hidden layer neurons and output layer neurons. In this study, the ReLU (rectified linear unit) function was used. The ReLU function can be expressed as $f(x) = \max(0, x)$, and if $x > 0$ the output is a straight line with a slope of 1, and if $x < 0$ the output value is always zero.

Adjusting the parameters of the encoder and decoder minimizes errors between the output data and the original data. The data output to the hidden layer is the optimal low-dimensional representation of the original data [25]. By using the extracted low-dimensional features as input for oversampling, we can improve the speed and performance of classification.

3.2. Generative Adversarial Networks

Generative adversarial networks (GAN) is a regression generation model published by Ian Goodfellow [26] in 2014 in Neural Information Processing Systems (NIPS), consisting of a model responsible for classifying (discriminator D) and a model responsible for generating regression. One neural network called G creates a new data instance, and the other, D , evaluates the authenticity

of the data. G creates a new image to pass to D. From G's perspective, he wants the fake image he created to look like it's real. The goal of G here is to generate an image that will actually differentiate the discriminator. D's goal, on the other hand, is to identify images delivered from G as fake.

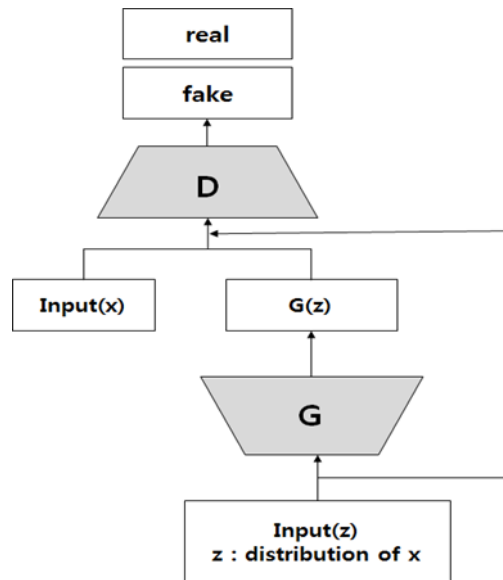


Figure 2. Structure of generative adversarial nets (GAN).

As shown in Figure 2, the operation steps of the GAN are applied to IDS as follows.

- Receives the network data generated by G taking a random number.
- These generated fake network data are delivered to D along with original network data from the actual data set.
- D identifies the actual network data and the fake network data and returns them as a probability value between 0 and 1. One (1) indicates real network data and 0 indicates fake network data.
- The above is repeated to create fake network data similar to the original network data.

D tries to reduce the probability of making mistakes and G tries to increase the probability of making mistakes. Therefore, this model is referred to as a 'min-max two-caliber game' or 'min-max proxy'.

$$V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (3)$$

The first term $E_{x \sim p_{data}(x)} [\log D(x)]$ is the actual data (x) and the second term $E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$ is the fake data G(z). As a result, a large value is output when a real image is put in, and a small value is output when a fake image is put in. As the learning progresses, the classification model continues to have robustness in the transformation of the data through fake data [27].

4. Proposed Method

4.1. The Framework

The NIDS based on the AE-CGAN model proposed in this paper is shown in Figure 3 and can be divided into feature extraction, data imbalance resolution, and classification. Autoencoder (AE) is used to extract low-dimensional features, and the CGAN model is used to oversample rare classes to solve data imbalances. Moreover, this is a structure that is classified as random forest by using oversampling total data.

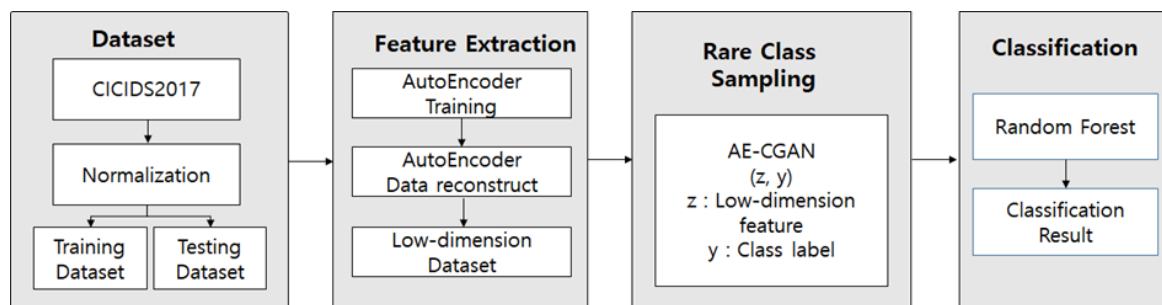


Figure 3. Structure of proposed model.

4.2. AE Based Low Dimensional Extraction of Feature

In real-world network environments, normal traffic accounts for the majority and abnormal traffic takes up a much smaller share. Detecting attacks that rarely occur among attack traffic is the most important element of a high-performance NIDS. For infrequent attacks—that is, rare attacks—it is difficult to find patterns to extract useful features. Therefore, in this study, AE was used to help extract the low-level features of sparse classes.

The CICIDS2017 dataset used in this study consists of a total of 77 features. In the process of encoding 77 features with the AE algorithm, only the core feature information of the data is learned in the hidden layer and the remaining information is lost. When the output value of the hidden layer is extracted during the decoding process, it is an approximation of the input value rather than the perfect value. By extracting the weights so that the output value is the same as the input value, we can extract the feature precisely. The experimental method extracts the optimal features by adjusting the number of hidden layers. When the optimal feature is extracted, the time and performance of detection can be improved as there is no need to learn unnecessary features.

4.3. CGAN Based Rare Class Oversampling

In this paper, we propose a CGAN model for oversampling rare classes. The original GAN model has the disadvantage that mode-collapse phenomenon that outputs only one kind of output occurs instead of evenly imitating the entire distribution when the distribution of real data is multi-modal. And because of learning instability optimal sampling is not possible. The CGAN may generate data by reflecting a feature desired by the user. The difference with GAN is that the features can be learned together with the distribution.

In our CGAN model, the input data distribution and the feature to learn are the labels of each class. In order to solve the shortcomings of the original GAN model, in which the effect of oversampling according to the distribution of data is deteriorated, optimal oversampling is possible by learning an additional label.

As shown in Figure 4, the vector used as input (x) is a low-level feature compressed by AE. For oversampling of rare classes, 10,000 additional data were generated for ‘Bot’, ‘Infiltration’, and ‘Heartbleed’, which are less than 0.1% of CICIDS2017 dataset.

4.4. Classification

Random forest (RF) [28], a leading machine learning algorithm, was used to check the performance of NIDS after solving sparse class problems. The means of learning several models in machine learning to predict better values than a single model using the predictions of those models is called ensemble learning, and a prime example is RF. RF is an algorithm that creates several decision trees (DT) [29] and then predicts the most selected class of values predicted in each tree.

It is difficult to generalize DT because the tree generated by learning data is very different. In addition, the hierarchical approach is not an appropriate classification method in the network environment because the error propagates to the next step when an error occurs in the middle. Because

RF is composed of trees with slightly different characteristics due to randomness, the predictions of each tree are uncorrelated and consequently improve generalization performance.

For example, if the "Bot" of the rare class is said to be oversampled, the generator will generate the fake data of the bot by entering any noise and the label of the class. Fake data that is first generated will have a different shape than the real data. The discriminator then compares the actual bot data with the fake data generated. Repeated learning of the fake data and real data using the min-max problem of the GAN creates data similar to the actual bot data.

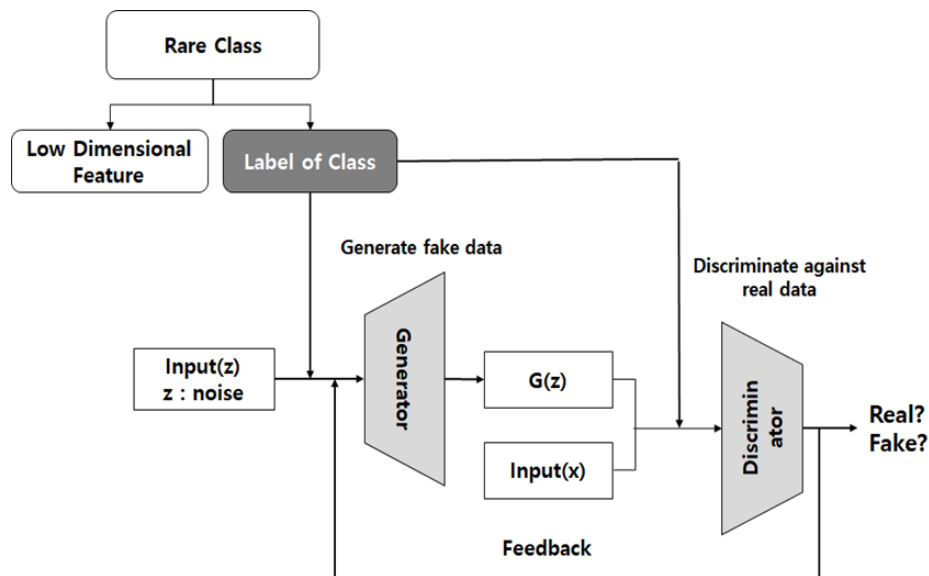


Figure 4. Structure of autoencoder-conditional generative adversarial networks (AE-CGAN).

5. Experimental Environment

5.1. Dataset

There are not many open datasets used for NIDS, and they are mainly based on KDD99 dataset [8], NSL-KDD dataset [9], Kyoto2006 dataset [30], and ISCX2012 dataset [31]. Researchers' assessment of existing datasets shows that they are mostly old and unreliable, and some lack traffic diversity and volume. There are also problems that do not reflect the current tendency to attack.

Among the recently released datasets, the UBSW-VB15 dataset [32] reflects the latest attacks. However, it is not suitable as the dataset of this study because the types of attacks are smaller than those of CICIDS2017 and the number of features is small.

For this reason, this study uses the CICIDS2017 dataset with normal and most recent attacks similar to actual data. It consists of normal traffic and 15 types of attacks. Similar to the actual network, normal data accounts for more than 80% of the data and includes rare attacks such as Infiltration and Heartbleed attacks. Table 1 shows the types and ratios of the classes in the CICIDS2017 dataset.

Dataset uses maxima-minimum normalization method to normalize [33] the characteristic value of the CICIDS2017 dataset to facilitate a comparison of results. X_{max} and X_{min} represent the maximum and minimum values of the original characteristic values, respectively.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (4)$$

Conduct an experiment on a dataset that has been normalized by dividing it into 60% and 40% for training and testing, respectively.

Table 1. Canadian institute for cybersecurity intrusion detection system (CICIDS)2017 dataset.

Class		Number	Percentage
Web Attack	Benign	2,273,097	80.3004%
	Distributed Denial-of-service (DDoS)	128,027	4.5227%
	Port Scan	158,930	5.6441%
	Bot	1,966	0.0695%
	Infiltration	36	0.0013%
	Brute Force		
	Structured Query Language (SQL) Injection	2,180	0.0770%
	Cross-site Scripting (XSS)		
	File Transfer Protocol (FTP)–Patator	7,938	0.2804%
	Secure Shell (SSH)–Patator	5,897	0.2083%
	Denial-of-service (DoS) GoldenEye	10,293	0.3636%
	DoS Hulk	231,073	8.1630%
	DoS Slowhttptest	5,499	0.1943%
	DoS Slowloris	5,796	0.2048%
	Heartbleed	11	0.0004%

5.2. Evaluation

This paper is based on the confusion matrix [34] for measuring results. The definition of the confusion matrix is shown in Table 2.

Table 2. Confusion matrix.

		Actual	
		Normal	Attack
Predicted	Normal	TP	FP
	Attack	FN	TN

The experimental performance evaluation measured precision, recall, and F1 score. Methods for measuring performance are as follows.

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

Accuracy is a rating indicator that can most intuitively represent the performance of a model, but it is not a meaningful indicator when the class is imbalanced data. The indicator to be supplemented is F1 score, which is the harmonic mean of precision and recall. The F1 score is an important performance evaluation factor because it can accurately evaluate the performance of a model when using imbalanced data. Therefore, in this paper, precision, recall, and F1 score are used as indicators, with accuracy excluded.

5.3. Model Parameters

The hardware experimental environment was tested on a desktop with an Intel(R) core i9-7900X CPU of 3.30GHz, 64GB RAM, and Linux Ubuntu 16.04 operating system.

Experiment simulations were performed using Tensorflow and scikit-learn, the most commonly used of the machine learning frameworks, and Python was chosen as the programming language.

CICIDS2017 dataset consists of 15 classes in total, but in this paper, brute force, Structured Query Language (SQL) injection, and XSS are grouped into web attack classes, making up a total of 13 classes.

In AE for dimension reduction of features, 77 features are used as inputs to find the optimal parameter while adjusting the number of hidden layers. The parameters of AE, AE-CGAN, and random forest (RF) are as shown in Table 3.

Table 3. Parameters of the models.

Model	Parameters	Value
AE	Batch size	500
	AE Pre-training learning rate	0.001
	Epoch	20
AE-CGAN	Hidden nodes	96
	Num_noise	256
	Batch size	10
	Epoch	20
	Learning rate	0.001
RF	Random state	1
	N estimators	100

6. Experiment and Results

As shown in Figure 4, the vector used as input is a low-level feature compressed by AE, and we learn by adding a label as a condition.

First, we extract an AE-based optimal feature. We measure the performance of the AE-RF model classified as random forest using low-dimensional features according to the number of neurons in hidden layer. The performance here is taken as the average of multi-classifications.

Second, we measure the performance of the AE-CGAN-RF model trained by using the most optimal feature found in AE and using it as input of GAN and adding class label as condition.

Third, the performance of single-RF classified without oversampling, AE-RF extracted and classified in low dimensions based on AE, and AE-CGAN-RF proposed in this study are compared.

Fourth, the SMOTE technique, which has been widely used to solve the data imbalance problem, is compared with the performance of AE-CGAN.

6.1. AE-Based Optimal Low Dimensional Feature Extraction

To extract the AE-based optimal low feature, we find the optimal low dimension feature by adjusting the number of neurons in the hidden layer. This compares the performance when 77 features are used as input layers and the number of neurons in the hidden layer is compressed to (5), (15), (30), (40), (50), (60), and (70). Performance here means the average performance of multi-classification classified by the random forest algorithm using feature compressed by encoding. If the number of hidden neurons is small, the learning time may be short, but the best compression is not achieved, and the performance is low. As shown in Figure 5, the best performance is shown when the number of neurons in the hidden layer is 40, and even when the number of neurons in the layer is greater than 40, perfect feature extraction does not occur, indicating that the performance is poor.

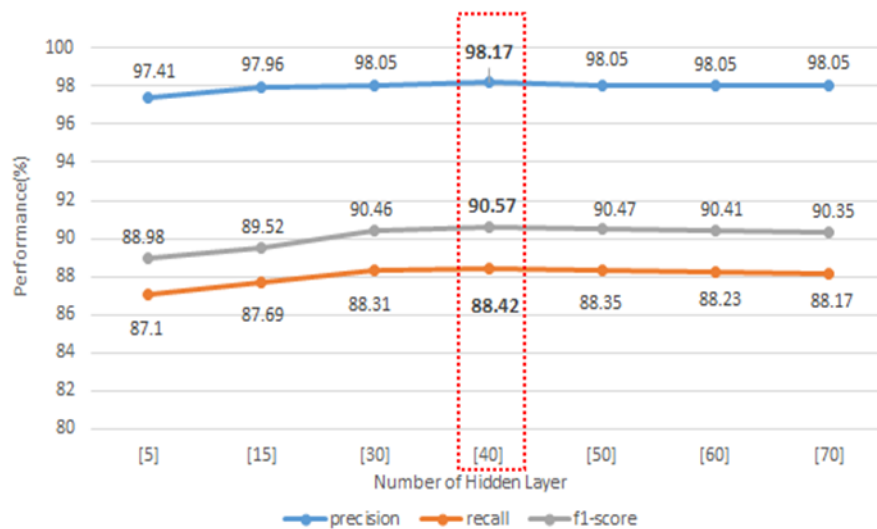


Figure 5. Performance comparison by number of neurons in hidden layer.

6.2. Rare Class Over-Sampling with CGAN

AE-CGAN is a model applying condition-GAN. It uses the AE model as a generator and discriminator to give compressed low-dimensional features as input and to further learn class labels as conditions. The experimental method gave the optimal low dimension feature obtained in the experiment of Section 5.1 as the input of GAN and measured its performance. Figure 6 shows the confusion matrix in the actual experiment.

The horizontal axis is the actual number of classes, and the vertical axis is the class prediction. For example, the BENIGN class actually consists of 908,528 data sets, with an exact forecast of 907,719 and incorrectly predicting two as DDoS. The rare class Infiltration has 10 actual data points and 10 predicted numbers, so accuracy and precision are 100%. However, recall and score were 66.67% and 80.00%, respectively, because the number of other BENIGN class was incorrectly predicted as Infiltration. Therefore, in order to measure the performance of the rare class, it is necessary to measure the performance of recall and F1 score.

	Actual												
	BENIGN	DDoS	PortScans	Bot	Infiltration	Web Attack	FTP-Patator	SSH-Patator	DoS GoldenEye	DoS Hulk	DoS Slowhttptest	DoS Slowloris	HeartBleed
Prediction													
BENIGN	907719	33	11	357	5	43	5	6	10	234	11	6	0
DDoS	2	51170	0	0	0	0	0	0	0	1	0	0	0
PortScans	390	0	63494	0	0	2	0	0	0	1	1	1	0
Bot	83	0	0	426	0	0	0	0	0	0	0	0	0
Infiltration	0	0	0	0	10	0	0	0	0	0	0	0	0
Web Attack	0	0	3	0	0	827	0	0	0	1	0	1	0
FTP-Patator	0	0	0	0	0	0	3169	0	0	0	0	0	0
SSH-Patator	0	0	0	0	0	0	0	2353	0	0	0	0	0
DoS GoldenEye	10	0	0	0	0	0	0	0	4095	11	2	1	0
DoS Hulk	310	7	14	0	0	0	0	0	12	91802	0	0	0
DoS Slowhttptest	14	0	0	0	0	0	0	0	1	0	2177	7	0
DoS Slowloris	0	0	0	0	0	0	0	0	0	0	9	2303	0
HeartBleed	0	0	0	0	0	0	0	0	0	0	0	0	5

Figure 6. Confusion matrix of the proposed model.

6.3. Performance Comparison of Single-RF, AE-RF and AE-CGAN-RF

To verify the high performance of the AE-CGAN model proposed in this paper, the performance of the single-RF model that is classified only as RF without oversampling and the AE-RF model that classifies the low-dimension feature extracted based on AE is compared.

As shown in Table 4, AE-RF model showed an improvement in performance compared to single-RF model. These results show that important features were well compressed based on AE. In addition, the performance of AE-CGAN-RF with sparse class oversampled based on low-level compressed features has been further enhanced.

Particularly when oversampling the number of rare attack classes such as Bot, Infiltration, and Heartbleed, you can see that not only has the performance of rare classes improved, but also the performance of regular classes. In addition, the F1 score, which is used as an indicator of performance when data is unbalanced, has significantly improved performance compared to other metrics. As such, it was confirmed that performance was improved when class was balanced and classified using oversampling.

Table 4. Performance comparison of single-RF, AE-RF, and AE-CGAN-RF.

Type	Precision			Recall			F1 Score		
	Single-RF	AE-RF	AE-CGAN-RF	Single-RF	AE-RF	AE-CGAN-RF	Single-RF	AE-RF	AE-CGAN-RF
Benign	99.62	99.76	99.92	99.52	99.59	99.91	99.57	99.67	99.92
DDoS	99.98	99.91	99.99	99.86	99.74	99.92	99.92	99.82	99.96
Port Scan	98.52	94.75	99.38	99.89	97.35	99.96	99.20	96.03	99.67
Bot	100.00	85.76	83.69	20.69	66.92	54.41	34.29	55.89	65.94
Infiltration	100.00	100.00	100.00	40.00	13.33	66.67	57.14	42.82	80.00
Web Attack	99.62	99.06	99.40	91.40	96.56	94.84	95.33	97.79	97.07
FTP-Patator	100.00	99.97	100.00	99.50	99.84	99.84	99.75	99.91	99.92
SSH-Patator	100.00	99.66	100.00	99.81	98.94	99.75	99.40	99.30	99.87
DoS Goldeneye	95.73	99.20	99.42	97.55	98.98	99.44	96.63	99.09	99.43
DoS Hulk	95.33	99.67	99.63	96.73	99.84	99.73	96.03	99.75	99.68
DoS Slowhttptest	88.22	99.05	99.00	79.00	99.00	89.95	83.36	99.02	98.98
DoS Slowloris	99.55	99.48	99.61	85.38	99.31	99.31	91.92	99.40	99.46
Heartbleed	100.00	100.00	100.00	80.00	80.00	100.00	88.78	88.89	100.00
Average	98.20	98.17	98.46	83.79	88.42	93.29	87.79	90.57	95.38

6.4. Comparison with Other Data Imbalanced Resolution Algorithms

In previous studies, SMOTE techniques have often been used to address the problem of data imbalance. Therefore, we compared the performance of the SMOTE-RF model, which classifies oversampled data as RF using the SMOTE technique, and the AE-CGAN-RF model proposed in this study. In addition, the feature was extracted with AE, oversampled with SMOTE, and compared with AE-SMOTE-RF. This is to check whether the vector extracted by AE is well oversampled in SMOTE.

As shown in Figure 7, the performance of the proposed AE-CGAN-RF model is higher than that of SMOTE-RF and AE-SMOTE-RF.

SMOTE is a method of creating virtual fractional class data on a straight line between the fractional class data and the randomly selected data among closest to the data. SMOTE works by adding points that are moved slightly in consideration of nearest neighbors. If other classes are adjacent, the other classes are overlapped. As a result, data imbalance cannot be completely solved. However, the CGAN proposed in this study shows that the effect of oversampling can be maximized because the data is replicated similarly by the generation algorithm. In addition, the AE-SMOTE-RF model has lower performance than the SMOTE-RF, and the data reduced by AE are not suitable for the model that moves exact points like SMOTE.

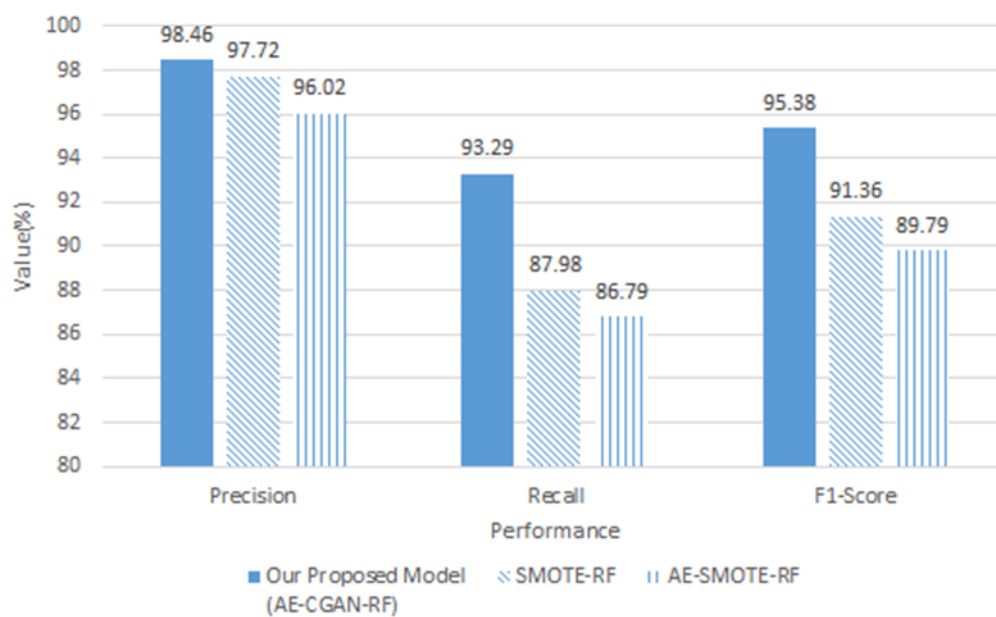


Figure 7. Performance comparison of the proposed model with synthetic minority oversampling technique (SMOTE).

7. Conclusions

In this paper, the AE-CGAN model for high performance intrusion detection is proposed in situations where normal and abnormal traffic occur disproportionately. The AE-CGAN model is a model based on autoencoder and GAN, which are representative generative deep learning models, and a method that utilizes AE to use compressed features in low dimensions as input to CGAN, and learn by adding a label of class.

The AE-CGAN model was able to show improvements in performance compared to other models, particularly in F1 score, a performance indicator for imbalanced data. You can also see that other classes of false detection as well as rare classes have decreased. Therefore, our proposed AE-CGAN model could be used to sample unbalanced data almost like existing data to create a high-performance NIDS with an improved taxonomy.

In addition, through this experiment, we were able to confirm that the generative model of deep learning not only develops in image, voice, and text, but also contributes to performance improvement in NIDS.

Intrusion detection in an internet of things (IoT) environment is intended for future research and study. We will prepare for the cyber threats of IoT through research that identifies incoming data similar to normal data in the IoT environment using a deep learning generation model.

Author Contributions: Conceptualization, K.P. and J.L.; methodology, K.P. and J.L.; software, J.L.; validation, K.P. and J.L.; investigation, K.P. and J.L.; resources, J.L.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, K.P.; supervision, K.P.; project administration, K.P.; funding acquisition, K.P.

Funding: This research was funded by the Basic Science Research Programs through the National Research Foundation of Korea (NRF), grant number funded by the Ministry of Education, Science and Technology (No.NRF-2018R1D1A1B07043982) and The APC was funded by the National Research Foundation of Korea (NRF-2018R1D1A1B07043982).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Marir, N.; Wang, H.; Feng, G.; Li, B.; Jia, M. Distributed Abnormal Behavior Detection Approach Based on Deep Belief Network and Ensemble SVM Using Spark. *IEEE Access* **2018**, *6*, 59657–59671. [\[CrossRef\]](#)
- Amoli, P.V.; Hämmäläinen, T. A real time unsupervised NIDS for detecting unknown and encrypted network attacks in high speed network. In Proceedings of the 2013 IEEE International Workshop on Measurements & Networking (M&N), Naples, Italy, 7–8 October 2013; pp. 149–154.
- Lee, C.H. *A Study on the Normal Data Extraction Algorithm for Network Intrusion Detection System Learning Data Optimization*; ETRI: Daejeon, Korea, 2002.
- Bitaab, M.; Hashemi, S. Hybrid Intrusion Detection: Combining Decision Tree and Gaussian Mixture Model. In Proceedings of the 2017 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC), Shiraz, Iran, 6–7 September 2017; pp. 8–12.
- Ahmad, I.; Bashari, M.; Iqbal, M.J.; Rahim, A. Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection. *IEEE Access* **2018**, *6*, 3789–33795. [\[CrossRef\]](#)
- Yin, C.; Zhu, Y.; Fei, J.; He, X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access* **2017**, *5*, 21954–21961. [\[CrossRef\]](#)
- Lin, W.; Lin, H.; Wang, P.; Wu, B.; Tsai, J. Using Convolutional Neural Networks to Network Intrusion Detection for Cyber Threats. In Proceedings of the IEEE International Conference on Applied System Innovation 2018, Chiba, Japan, 13–17 April 2018; pp. 1107–1110.
- KDD Cup 1999 Data. Available online: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> (accessed on 5 June 2018).
- NSL-KDD Dataset. Available online: <https://www.unb.ca/cic/datasets/nsl.html> (accessed on 10 June 2018).
- Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, Singapore, 8–10 August 2018; pp. 108–116.
- Yan, B.; Han, G. Effective Feature Extraction via Stacked Sparse Autoencoder to Improve Intrusion Detection System. *IEEE Access* **2018**, *6*, 41238–41248. [\[CrossRef\]](#)
- Divekar, A.; Parekh, M.; Savla, V.; Mishra, R.; Shirole, M. Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives. In Proceedings of the 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, Nepal, 25–27 October 2018; pp. 1–8.
- Intrusion Detection Evaluation Dataset (CICIDS2017). Available online: <https://www.unb.ca/cic/datasets/ids-2017.html> (accessed on 20 June 2018).
- Zhu, J.; Mu, J.; Wei, D.; Feng, B.; Wang, Y.; Yin, K. A spatial correlation-based hybrid method for intrusion detection. In Proceedings of the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, China, 6–8 May 2017; pp. 1097–1102.
- Sonawane, H.A.; Pattewar, T.M. A comparative performance evaluation of intrusion detection based on neural network and PCA. In Proceedings of the 2015 International Conference on Communications and Signal Processing (ICCS), Melmaruvathur, India, 2–4 April 2015; pp. 841–845.
- Al-Qatf, M.; Lasheng, Y.; Al-Habib, M.; Al-Sabahi, K. Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection. *IEEE Access* **2018**, *6*, 52843–52856. [\[CrossRef\]](#)
- Papamartzivanos, D.; Marmol, F.G.; Kambourakis, G. Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems. *IEEE Access* **2019**, *7*, 13546–13560. [\[CrossRef\]](#)
- Green, C.; Lee, B.; Amaresh, S.; Engels, D.W. Comparative Study of Deep Learning Models for Network Intrusion Detection. *SMU Data Sci. Rev.* **2018**, *1*, 1–13.
- Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [\[CrossRef\]](#)
- Qazi, N.; Raza, K. Effect of Feature Selection, SMOTE and under Sampling on Class Imbalance Classification. In Proceedings of the International Conference on Computer Modelling and Simulation, Cambridge, UK, 28–30 March 2012; pp. 145–150.
- Yan, B.H.; Han, G.D.; Sun, M.D.; Ye, S.Z. A Novel Region Adaptive SMOTE Algorithm for Intrusion Detection on Imbalanced Problem. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 1281–1286.

22. Sun, Y.; Liu, F. SMOTE-NCL: A Re-Sampling Method with Filter for Network Intrusion Detection. In Proceedings of the 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 14–17 October 2016; pp. 1157–1161.
23. Lin, Z.; Shi, Y.; Xue, Z. IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection. *arXiv* **2019**, arXiv:1809.02077v3.
24. Kunang, Y.N.; Nurmaini, S.; Stiawan, D.; Zarkasi, A.; Jasmir, F. Automatic Features Extraction Using Autoencoder in Intrusion Detection System. In Proceedings of the 2018 International Conference on Electrical Engineering and Computer Science (ICECOS), Pangkal Pinang, Indonesia, 2–4 October 2018; pp. 219–224.
25. Wang, X.; Wang, L. Research on Intrusion Detection Based on Feature Extraction of Autoencoder and the Improved K-Means Algorithm. In Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 9–10 December 2017; Volume 2, pp. 352–356.
26. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Neural Information Processing Systems Conference, Montreal, QC, CA, 8–13 December 2014; pp. 2672–2680.
27. Usama, M.; Asim, M.; Latif, S.; Qadir, J.; Fuqaha, A.A. Generative Adversarial Networks for Launching and Thwarting Adversarial Attacks on Network Intrusion Detection Systems. In Proceedings of the The 15th International Conference on Wireless Communications and Mobile Computing Machine learning Workshop, Tangier, Morocco, 24–28 June 2019; pp. 78–83.
28. Park, K.; Song, Y.; Cheong, Y. Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm. In Proceedings of the 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (Big Data Service), Bamberg, Germany, 26–29 March 2018; pp. 282–286.
29. Mizianty, M.; Kurgan, L.; Ogiela, M. Comparative Analysis of the Impact of Discretization on the Classification with Naïve Bayes and Semi-Naïve Bayes Classifiers. In Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications, San Diego, CA, USA, 11–13 December 2008; pp. 823–828.
30. Description of Kyoto University Benchmark Data. Available online: http://www.takakura.com/Kyoto_data/BenchmarkData-Description-v5.pdf (accessed on 7 September 2018).
31. Intrusion Detection Evaluation Dataset (ISCXIDS2012). Available online: <https://iscxdownloads.cs.unb.ca/iscxdownloads/ISCX-IDS-2012/#ISCX-IDS-2012> (accessed on 20 August 2018).
32. The UNSW-NB15 Dataset Description. Available online: <https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/> (accessed on 8 April 2019).
33. Othman, Z.A.; Eljadi, E.E. Network anomaly detection tools based on association rules. In Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, 17–19 July 2011; pp. 1–7.
34. Confusion Matrix. Available online: https://en.wikipedia.org/wiki/Confusion_matrix (accessed on 12 March 2018).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).