electronics

Article

Improvement of the Classification Performance of an Intrusion Detection Model for Rare and Unknown Attack Traffic

Sangsoo Han, Youngwon Kim and Soojin Lee *

Department of Computer Engineering, Korea National Defence University, 1040, Hwangsanbeol-ro, Yangchon-myeon, Nonsan-si 32010, Chungcheongnam-do, Korea; han19910130@gmail.com (S.H.); headsun21@gmail.com (Y.K.)
* Correspondence: cyberkma@gmail.com

Abstract: How to deal with rare and unknown data in traffic classification has a decisive influence on classification performance. Rare data make it difficult to generate validation datasets to prevent overfitting, and unknown data interferes with learning and degrades the performance of the model. This paper presents a model generation method that accurately classifies rare data and new types of attacks, and does not result in overfitting. First, we use oversampling methods to solve the data imbalance caused by rare data. We separate the test dataset into a training dataset and a validation dataset. A model is created using separate training and validation datasets. Furthermore, the test dataset is used only for evaluating the performance capabilities of classification models, in order to make the test dataset independent of learning. We also use a softmax function that numerically indicates the probability that the model's predictive results are accurate in detecting new, unknown attacks. Consequently, when applying the proposed method to the NSL_KDD dataset, the accuracy is 91.66%—an improvement of 6–16% compared to existing methods.

Keywords: intrusion detection; AI; GAN; softmax; validation; NSL_KDD

Citation: Han, S.; Kim, Y.; Lee, S. Improvement of the Classification Performance of an Intrusion Detection Model for Rare and Unknown Attack Traffic. *Electronics* **2021**, *10*, 2268. https://doi.org/10.3390/electronics10182268

Academic Editor: Arkaitz Zubiaga

Received: 24 August 2021
Accepted: 14 September 2021
Published: 15 September 2021

1. Introduction

In the real world, certain attacks are less numerous than others, and new types of attack continue to emerge. Therefore, the quality of the data in these cases is difficult to determine, and the datasets used in the field of network intrusion detection are unbalanced and lack volume.

NSL_KDD is a representative dataset that suitably reflects certain unbalanced characteristics of data, such as rare data appearing in the real world. There are two important characteristics of this dataset: First, to reflect the data imbalance, some data in the dataset are rare data, although there are relatively few instances, making the training dataset alone inadequate for sufficient learning. Second, we use the difference between the training dataset and the test dataset class configurations to reflect the existence of new attacks that are thus far unknown. In other words, there exist data that cannot be learned.

Owing to these characteristics, the NSL_KDD dataset has long been used in research, and although it is not possible to reflect all recent attack trends, it is still a common source of study.

Existing studies have sought to address the characteristics of data imbalances in the NSL_KDD dataset and differences in training/test dataset class configurations. For example, they either reconstruct the training dataset and the test dataset together, or may focus on how the test dataset is directly involved in model generation without constructing a validation dataset. These methods have resulted in factors that degrade model performance outcomes, such as overfitting and changes in dataset configurations, and the detection accuracy rates associated with the findings are in the low-to-mid 80% range.

Therefore, in this paper, we present a way to utilize these characteristics, rather than to eliminate them. First, we use oversampling methods to solve for the data imbalance issue caused by rare data. We use a generative adversarial network (GAN) to generate additional rare data and ensure the constancy of the amount of data while maintaining the construction of the dataset. Moreover, we learn the characteristics of the rare data fully during the neural network learning process, in order to improve the detection performance for the given class. In addition, the existing test dataset is separated into a training dataset and a validation dataset to generate models, with the training dataset used only for evaluating the performance of the classification models. In other words, we attempt to generate a model that is not overfitted by making the test dataset completely independent of learning. Second, we use a softmax function that numerically indicates the probability that the model's predictive results are accurate in detecting new, unknown attacks. Experiments confirm that new types of attack are often classified as normal, with ambiguous probabilities, because they are not trained. If the softmax score does not exceed a certain level, we classify traffic that is classified as normal traffic with ambiguous probabilities as attack traffic. The main contribution of this paper is that it overcomes the following challenges:

- We fully learn the characteristics of rare data using a GAN, and we make the test dataset independent of learning in order to prevent overfitting of the model;
- In order to detect new, unknown attacks using softmax, traffic classified as normal with ambiguous probabilities is classified as attack traffic;
- We show improved classification performance outcomes through comparisons with existing studies.

The goal of this paper is to classify rare data and new types of attacks accurately, and to present models that do not result in overfitting. When applying the proposed method to the NSL_KDD dataset, the accuracy rate is 91.66%, demonstrating an improvement of 6–16% compared to existing research.

The structure of this paper is as follows: Section 2 describes related work. Section 3 analyzes the NSL_KDD dataset. Section 4 describes our approach. Section 5 shows the experimental and analytical results. Section 6 summarizes the results and presents future research directions.

2. Related Work

Starting with LeCun et al. [1], who proposed the concept, deep learning has been extensively applied to visual and speech recognition, as well as natural language processing. Attempts are also actively underway to leverage deep learning to improve intrusion detection performance outcomes. Mostafa A. Salama et al. [2] confirmed the performance of binary classification using intrusion detection imaging and a hybridization scheme method. Ugo Fiore et al. [3] implemented a semi-supervised anomaly detection system to implement the ability of the model to adapt to changes and generalize behavior to the network environment.

A convolutional neural network (CNN) is a representative deep learning algorithm that demonstrates superior performance on image classification tasks. Typically, a CNN is used as a method of classifying image classes after converting traffic from intrusion detection fields into images. A CNN offers the advantage of reducing the computations by algorithms by not having to perform the feature selection process early, but some sample properties are inevitably excluded, resulting in losses. Nevertheless, a CNN allows for effective classification of classes while using all of the properties of the sample. Vinayakumar R et al. [4] modeled network traffic in a predefined time range using a supervised learning method using a time series. Xuewen Zeng et al. [5] applied an expression learning approach to malicious program traffic classification using raw traffic data. Wei Wang et al. [6] used a CNN to learn low-level spatial features of network traffic and use long short-term memory (LSTM) networks.

One of the best examples of applying deep learning to intrusion detection was that by Chuanlong et al. [7], who applied a semi-supervised GAN (SGAN) to intrusion detection model generation tasks; they achieved a high binary classification accuracy rate of 84.75% on the NSL_KDD dataset. The discriminator of the original GAN is a binary classifier that determines whether the sample is extracted from a real dataset, and it is not capable of classifying sample classes [8]. On the other hand, the SGAN discriminator proposed by Augustus et al. [9] serves as a classification model, and can also distinguish between different classes of samples.

However, the above works ignored the effects of dataset imbalances and rare classes, and used a CNN and an SGAN for image classification. To address this problem, two studies [10,11] used a weighting method for each class of cost functions, and proposed an undersampling method [12]. The resulting classification model performance shows accuracy of approximately 80% based on the NSL_KDD dataset.

An important reason that the performances of intrusion detection models do not exceed a certain level is that they overlook the fact that the training dataset from the dataset and the detailed composition of the test dataset are different. Considerable amounts of rare data exist in the NSL_KDD dataset—a benchmark dataset used in intrusion-detection-related studies. In addition, there are many new types of attack traffic that are not in the training dataset, but exist only in test datasets. This highlights the continued emergence of new types of attack, or variant attacks. Classification models developed in existing studies misclassify most samples of R2L and U2R attack traffic classes into normal traffic classes.

Kuhn and Johnson note that when applying a prediction model, the model should be evaluated using samples not used for model construction and parameter tuning, with overfitting also not occurring in the model [13]. Russell and Norvig continue to reiterate the importance of the complete separation of the dataset used for final model performance evaluations [14].

3. Dataset

The NSL_KDD dataset is an improved version of KDD Cup '99, and is a benchmark dataset for intrusion detection research. Given that KDD Cup '99 contains a large number of duplicate data samples (78%), it interferes with the learning of the classification model. Therefore, NSL_KDD eliminates these duplicates, and keeps the ratio of normal traffic and attack traffic similar to the actual ratio. Each traffic sample has 41 forms of characteristic information in 4 categories, as shown in the example in Figure 1 [15], of which 32 are continuous values, 6 are discrete values, and 3 are categorical data types.

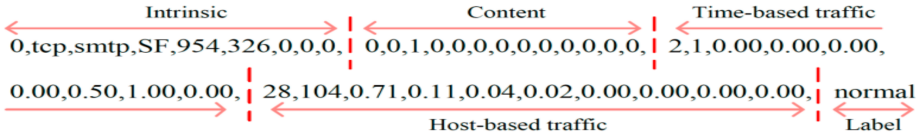

Figure 1. Characteristic of the NSL_KDD traffic sample.

The dataset is largely divided into five classes: Normal, DoS, Probe, R2L, and U2R. The compositions of the training dataset and test dataset are shown in Table 1.

Table 1. Class composition of NSL_KDD.

Class	Total	Normal	DoS	Probe	R2L	U2R
KDD Train+	125,973	67,343	45,927	11,656	995	52
KDD Tset+	22,544	9711	7460	2421	2885	67

- Normal: Normal traffic;
- DoS: Traffic that attempts a denial-of-service attack;
- Probe: Traffic that attempts to acquire information for an attack, such as via a port search;
- R2L: Traffic in which an unauthorized external user attempts to obtain access rights;
- U2R: Traffic attempting to acquire administrator rights.

Each attack class consists of several attack categories. As shown in Table 2, 3750 samples of 17 types that do not appear in the training dataset newly appear in the test dataset. This is a high number, representing 29.2% of the total 12,833 attacks in the test dataset, having a decisive influence on the attack traffic classification performance of the intrusion detection system. For example, 996 instances of mscan in the Probe class are included only in the test dataset, and not in the training dataset. In this case, neural networks trained using the training dataset never learn the properties of mscan and, thus, cannot identify instances of mscan in the test dataset. Here, mscan accounts for 7.3% of all attacks on the test datasets—the third most common among all attacks. Detecting these can improve the overall intrusion detection accuracy. Another important problem relates to imbalances in training datasets, where most samples are biased towards certain types of attacks, resulting in multiple rare classes. For example, in the case of neptune of the DoS class, the number of samples is 41,214, which accounts for 70.3% of all attacks in the training dataset, whereas guess_password and warezmaster of the R2L class number 53 and 20, respectively, accounting for less than 0.1% of the training dataset.

Table 2. Class detail composition of NSL_KDD.

Set	Class	Category	Samples	Set	Class	Category	Samples
Train+	Normal	-	67,343	Test+	Normal	-	9711
	DoS	back	956		DoS	back	359
		land	18			land	7
		neptune	41,214			neptune	4657
		pod	201			pod	41
		smurf	2646			smurf	665
		teardrop	892			teardrop	12
		apache2	0			apache2	737
		udpstorm	0			udpstorm	2
		processtable	0			processtable	685
		worm	0			worm	2
		mailbomb	0			mailbomb	293
	Probe	6	45,927		Probe	11	7460
		satan	3633			satan	735
		ipsweep	3599			ipsweep	141
		nmap	1493			nmap	73
		portsweep	2931			portsweep	157
		mscan	0			mscan	996
		saint	0			saint	319
	R2L	4	11,656		R2L	6	2421
		guess_password	53			guess_password	1231
		ftp_write	8			ftp_write	3
		imap	11			imap	1
		phf	4			phf	2
		multihop	7			multihop	18
		warezmaster	20			warezmaster	944
		warezclient	890			warezclient	0
		spy	2			spy	0
		xlock	0			xlock	9
		xsnoop	0			xsnoop	4
		snmpguess	0			snmpguess	331
		snmpgetattack	0			snmpgetattack	178

Table 2. *Cont.*

Set	Class	Category	Samples	Set	Class	Category	Samples
	U2R	httptunnel	0		U2R	httptunnel	133
		sendmail	0			sendmail	14
		named	0			named	17
		8	995			13	2885
		buffer_overflow	30			buffer_overflow	20
		loadmodule	9			loadmodule	2
		rootkit	10			rootkit	13
		perl	3			perl	2
		sqlattack	0			sqlattack	2
		xterm	0			xterm	13
		ps	0			ps	15
		4	52			7	67
	Attack	22	58,630		Attack	37	12,833
	Total	-	125,973		Total	-	22,544

Although this dataset has long been available, new attacks continue to emerge, making it clear that attacks that are not present in the training dataset are meaningful in an analysis of the configurations present in the test dataset [16].

4. Our Approach

4.1. Structure of Our Model

The overall concept of the model scheme is shown in Figure 2. First, raw data from NSL_KDD are preprocessed, categorical data are one-hot encoded, and continuous data are adapted to the (0, 1) range via min–max normalization. The preprocessed data are transformed into 28 × 28 16-bit grayscale images suitable for inputs from artificial neural networks.

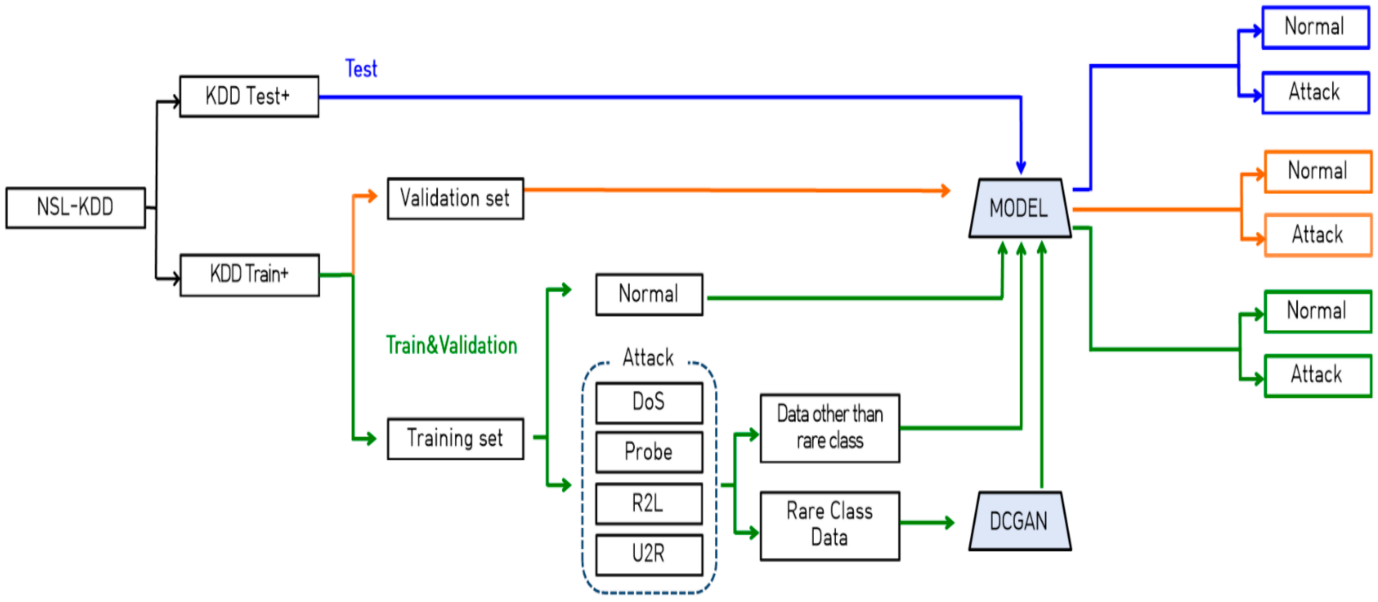

Figure 2. Structure of the model.

The training dataset is then constructed by dividing the training dataset (125,973) into data to be used for real training (100,778) and data comprising a validation (25,195) dataset to be used for validation at a ratio of 8:2, with the test (22,544) dataset used only for the final evaluation. That is, the ratio of the training, validation, and test datasets is 8:1:1.

Next, we select rare data from the training dataset and generate additional samples of that type using DCGAN. The generated samples are used for classification model training

with the original training dataset samples. The classification model is constructed based on SGAN, which showed outstanding performance during an NSL_KDD classification attempt study [7].

Once a model is created that achieves the best accuracy through the training and validation datasets, we use the model to check the softmax score for each sample on the test dataset. If this score does not exceed a certain level, we classify the sample as attack traffic.

The most important purpose of intrusion detection systems is to block attack traffic that penetrates the system when disguised as normal traffic. Therefore, it is more important to distinguish whether traffic is of the normal or attack type before designating the attack class in detail, and then to block attack traffic from penetrating the system.

In this paper, we reconstruct all four classes (DoS, Probe, R2L, and U2R) corresponding to attacks into attack classes in the NSL_KDD dataset, and attempt binary classification for attack and normal classes.

4.2. Data Preprocessing and Image Generation

Among the 41 features of the NSL_KDD dataset, 'num_outbound-cmds' was deleted as an attribute, as it did not affect the classification performance of the model because the values of all of the data were '0'. Because the three features of 'protocol_type', 'flag', and 'service' are categorical rather than numerical data, symbols are mapped to numbers through one-hot encoding. For example, three categories of protocol_type functions—TCP, UDP, and ICMP—are mapped to (1,0,0), (0,1,0), and (0,0,1), respectively. Through this process, 41-dimensional features of the NSL_KDD dataset are transformed into 121-dimensional shapes. In addition, due to the wide distribution of the dataset features, normalization is required so that certain features do not have an excessive impact on model learning; in this paper, all values are changed to be in the range of (0, 1) through min–max normalization [10].

Subsequently, the sample transformed into a 121-dimensional form through one-hot encoding is converted to an 11×11 matrix, which is then converted to a 16-bit grayscale image for reflection of the feature information into a single pixel of the image. Finally, as shown in Figure 3, each sample is scaled to a size of 28×28, which is suitable for input into an artificial neural network.

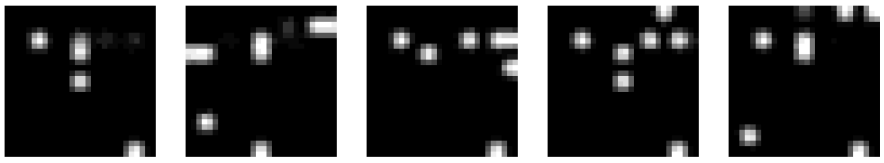

Figure 3. Sample images for each class (Normal, DoS, Probe, R2L, and U2R).

4.3. Resampling

Looking at the structure of the NSL_KDD training dataset, 13 of the 22 sub-attack types in the attack class, including land in the DoS class, correspond to rare data, at less than 1% of the total number of samples. After dividing the existing training dataset into training and validation datasets at 8:2, using the DCGAN generator on a real training dataset generates 1000 additional samples for each of the 13 attack types, increasing the number of attack traffic samples in the training dataset from 46,904 to 59,904.

4.4. Classification Model Training

For classification model training, as shown in Figure 4, normal and attack traffic images from the training dataset, images generated from DCGAN, and fake images generated by the generator of SGAN are used as inputs. One of the goals of SGAN is to use unlabeled images for learning, but NSL_KDD uses all labels in an intact manner, because samples are already designated according to class, and with more labels used, better classification accuracy results.

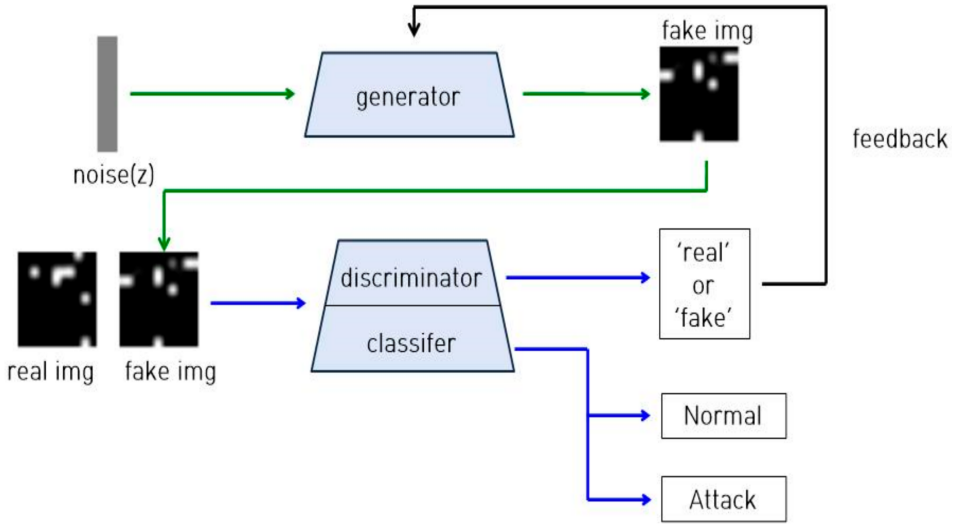

Figure 4. Training of the classification model.

4.5. The Softmax Score and Reclassification

Table 3 is part of the softmax scores for the KDDTest+ process in the classification model. For example, for Sample 1, the probability of attack traffic is 100%, and the probability of normal traffic is 0%. On the other hand, samples 56 and 114 were misclassified as normal traffic, with a low probability of 64.750% and 55.047%, respectively, although they were samples representing actual attack traffic. Therefore, we reclassify samples classified with these ambiguous probabilities as attack traffic.

Table 3. The KDDTest+ softmax score.

Real Class	Sample Number	Softmax Score		Predicted Class
		Attack	Normal	
Attack	0	1.00000	0.00000	Attack
	1	0.98548	0.01452	Attack
	2	1.00000	0.00000	Attack
	...	...	...	...
	56	0.35250	0.64750	Normal
	...	...	...	...
	114	0.44953	0.55047	Normal
	115	0.99716	0.00284	Attack
Normal	22,539	0.00024	0.99976	Normal
	22,540	0.00000	1.00000	Normal
	22,541	0.40185	0.59815	Normal
	22,542	0.00000	1.00000	Normal
	22,543	0.00002	0.99998	Normal

4.6. The Performance Evaluation Index

To evaluate the performance capabilities of the classification models, we typically use metrics such as accuracy, precision, recall, and F1 scores. However, ACC (accuracy), DR (detection rate), and FAR (false alarm rate) are used in the study of intrusion detection systems. In other words, in this paper we use ACC, DR, and FAR as performance evaluation metrics. The relationships between elements of the error matrix, and binary classification to illustrate them, are shown in Figure 5 [4].

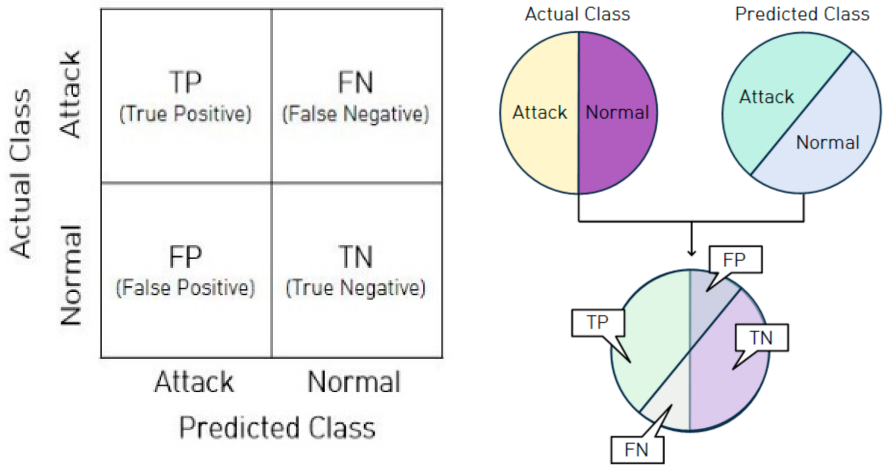

Figure 5. Relationships between TP, FP, FN, and TN.

The three evaluation indicators can be defined as follows: (1) ACC = (TP + TN)/(TP + FN + FP + TN); (2) DR = TP/(TP + FN); (3) FAR = FP/(FP + TN). In addition, ACC and DR should be improved without excessively increasing FAR, in order to improve the performance of the model.

5. Experiment and Evaluation

All experiments in this paper were conducted using Python and Keras on a PC with the Windows 10 Home 64-bit operating system, an AMD Ryzen 7 4800H CPU, 16.0 GB of RAM, and a NVIDIA GeForce GTX 1650i graphics card.

The KDDTest+ confusion matrix in the classification model shown in Figure 6 indicates that most of the normal traffic is correctly classified, while 3531 out of a total of 12,833 instances of attack traffic are incorrectly classified as normal traffic. As shown in Figure 7, the training results achieved the highest accuracy in Epoch 57, with rates of 99.51% for the training dataset and 99.51% for the validation dataset, and 83.14% accuracy was later achieved by conducting experiments on the test dataset using this model.

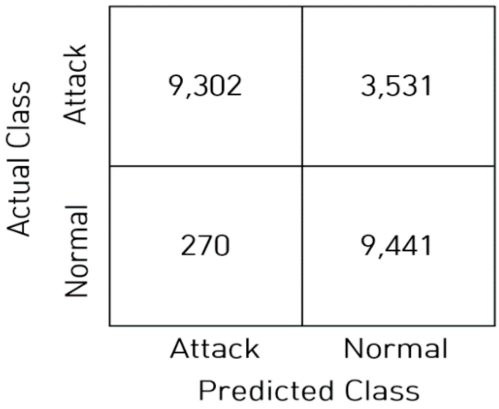

Figure 6. Confusion matrix for the KDDTest+.

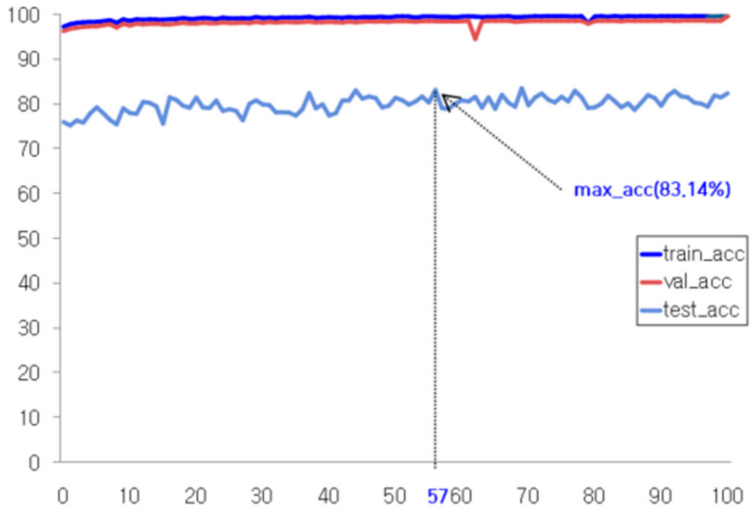

Figure 7. Accuracy of the classification model.

Table 4 presents the results after reclassifying the traffic as attack traffic when the softmax score is below a certain level, e.g., reclassifying samples with a softmax score of 0.990 or less as attack traffic increases ACC by 7.94% and DR by 16.61% compared to the original model. As a result, the maximum softmax score increases the ACC, DR, and FAR levels. At this time, a high FAR can be analyzed by administrators, but low ACC and DC outcomes do not properly block attack traffic, causing serious damage within the system, implying that it is important to increase ACC and DR even if FAR rises.

Table 4. The result of the reclassification.

Max Softmax Score	ACC (%)	DR (%)	FAR (%)
Original	83.140	72.485	2.780
<0.990	91.084	89.098	6.292
<0.991	91.173	89.387	6.467
<0.992	91.266	89.683	6.642
<0.993	91.284	89.901	6.889
<0.994	91.341	90.205	7.157
<0.995	91.457	90.587	7.394
<0.996	91.637	91.218	7.806
<0.997	91.656	91.576	8.238
<0.998	91.581	91.943	8.897
<0.999	91.115	92.543	10.462

The ACC score records the maximum values at the point where samples with a maximum softmax score of 0.997 or less are reclassified as attack traffic, increasing ACC by 8.52% and DR by 19.09%. The accuracy and confusion matrix of the model are shown in Figures 8 and 9, respectively.

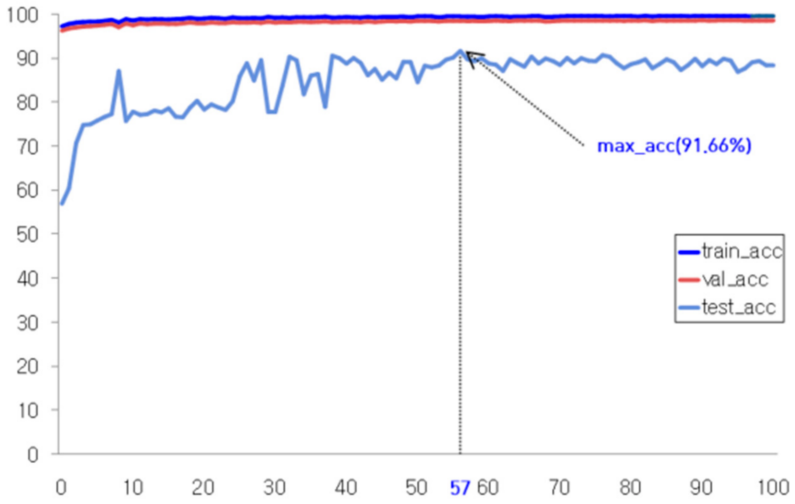

Figure 8. Accuracy of our model.

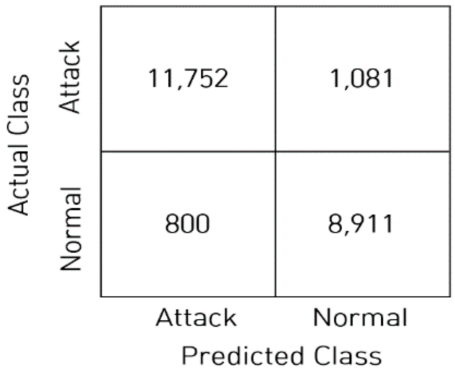

Figure 9. Confusion matrix for our model.

Table 5 shows the results of a comparison of the model method in this study with that in an earlier aforementioned study. Compared to this earlier work, we can say that the effectiveness of the model method proposed here is sufficient, as it outperformed the previous method by 6.63–14.84% for ACC.

Table 5. The performance comparison.

Model	ACC (%)
DNN-4 [17]	82.74
RNN [17]	77.00
CNN4 [17]	80.00
RF [17]	77.00
SVM [17]	78.00
Random tree+NBTree [18]	89.24
TSE-IDS [18]	85.79
SAVAER-DNN [18]	89.36
OUR MODEL	91.66

6. Conclusions

In the real world, certain attacks are less numerous than others, and new types of attack continue to emerge. Therefore, the quality of the data in these cases is difficult to determine, and the datasets used in the field of network intrusion detection are unbalanced and lack volume. The purpose of the intrusion detection system is to prevent attempts to penetrate the system as if attack traffic was normal. Therefore, we reclassify four attack classes (DoS, Probe, R2L, U2R) from the NSL_KDD dataset into attack classes, attempting

binary classification for attack and normal classes in this dataset. The main contribution of this paper is that it overcomes the following challenges: We fully learn the characteristics of rare data using a GAN, and we make the test dataset independent of learning in order to prevent overfitting of the model. In order to detect new, unknown attacks using softmax, traffic classified as normal with ambiguous probabilities is classified as attack traffic. We show improved classification performance outcomes through comparisons with existing studies.

In this paper, to implement the proposed model and technique, the raw dataset of NSL_KDD is preprocessed with one-hot encoding and normalization, and transformed into 28 × 28 16-bit grayscale images. We then divide the training dataset into training and validation datasets to make the test dataset fully independent. Rare data from the training dataset are generated using DCGAN and learned using a model based on the SGAN classification model. We also propose a method that reclassifies traffic as attack traffic when the softmax score is below a certain level.

The experimental results show that the accuracy rate is 91.66%, which is 6~16% better than the results of previous research.

Various neural networks and datasets will be used in future studies, and the method here will be extended beyond binary classification to applicable methods for multi-class classification problems. Furthermore, we will study the performance improvement of the output layer function calculated with ambiguous probability.

Author Contributions: Methodology, S.H. and Y.K.; Project administration, S.H.; Supervision, S.L.; Validation, Y.K. and S.L.; Writing—original draft, S.H.; Writing—review & editing, S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
2. Salama, M.A.; Eid, H.F.; Ramadan, R.A.; Darwish, A.; Hassanien, A.E. *Hybrid Intelligent Intrusion Detection Scheme, Soft Computing in Industrial Applications*; Springer: Berlin, Germany, 2011; pp. 293–303.
3. Fiore, U.; Palmieri, F.; Castiglione, A.; De Santis, A. Network anomaly detection with the restricted Boltzmann machine. *Neurocomputing* **2013**, *122*, 13–23. [CrossRef]
4. Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Applying convolutional neural network for network intrusion detection. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1222–1228.
5. Wang, W.; Zhu, M.; Zeng, M.; Ye, X.; Sheng, Y. Malware traffic classification using convolutional neural network for representation learning. In Proceedings of the 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 11–13 January 2017; pp. 712–717.
6. Wang, W. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access* **2018**, *6*, 1792–1806. [CrossRef]
7. Chuanlong, Y.; Yuefei, Z.; Shengli, L.; Jinlong, F.; Hetong, Z. Enhancing network intrusion detection classifiers using supervised adversarial training. *J. Supercomput.* **2019**, *76*, 6690–6719.
8. Ian, G.; Jean, P.; Mehdi, M.; Bing, X.; David, W.; Sherjil, O.; Aaron, C.; Yoshua, B. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
9. Augustus, O. Semi-supervised learning with generative adversarial networks, international conference on machine learning (ICML). *arXiv* **2016**, arXiv:1606.01583.
10. Kehe, W.; Zuge, C.; Wei, L. A novel intrusion detection model for a massive network using convolutional neural networks. *IEEE Access* **2018**, *6*, 50850–50859.
11. Peng, L.; Zhang, H.; Chen, Y.; Yang, B. Imbalanced traffic identification using an imbalanced data gravitation-based classification model. *Comput. Commun.* **2016**, *102*, 177–189. [CrossRef]
12. Liu, Y.; Liu, S.; Zhao, X. Intrusion Detection Algorithm Based on Convolutional Neural Network. In Proceedings of the 4th International Conference on Engineering Technology and Application (ICETA), Nagoya, Japan, 29–30 June 2017; pp. 9–13.
13. Kuhn, M.; Johnson, K. An introduction to feature selection. In *Applied Predictive Modeling*; Springer: New York, NY, USA, 2013; pp. 487–519.
14. Russell Stuart, J.; Norvig, P. *Aritificial Intelligence: A Modern Approach*; Prentice Hall: Hoboken, NJ, USA, 2009; p. 538.

15. Dhanabal, L.; Shantharajah, S.P. A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. *Int. J. Adv. Res. Comput. Commun. Eng.* **2015**, *4*, 446–452.
16. Varnika, E. A review on machine learning models used for anomaly detection, SSRN. *J. Innov. Dev. Pharm. Tech. Sci.* **2021**, *4*, 9–15.
17. Gao, M.; Ma, L.; Liu, H.; Zhang, Z.; Ning, Z.; Xu, J. Malicious network traffic detection based on deep neural networks and association analysis. *Sensors* **2020**, *20*, 1452. [CrossRef] [PubMed]
18. Yang, Y.; Zheng, K.; Wu, B.; Yang, Y.; Wang, X. Network intrusion detection based on supervised adversarial variational auto-encoder with regularization. *IEEE Access* **2020**, *8*, 42169–42184. [CrossRef]