Hindawi
Security and Communication Networks
Volume 2021, Article ID 5564354, 9 pages
https://doi.org/10.1155/2021/5564354

Research Article

I^{2}DS: Interpretable Intrusion Detection System Using Autoencoder and Additive Tree

Wenfeng Xu, Yongxian Fan, and Changyong Li

School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

Correspondence should be addressed to Yongxian Fan; yongxian.fan@gmail.com

Received 14 January 2021; Revised 16 February 2021; Accepted 1 March 2021; Published 12 March 2021

Academic Editor: Petros Nicopolitidis

Intrusion detection system (IDS), the second security gate behind the firewall, can monitor the network without affecting the network performance and ensure the system security from the internal maximum. Many researches have applied traditional machine learning models, deep learning models, or hybrid models to IDS to improve detection effect. However, according to Predicted accuracy, Descriptive accuracy, and Relevancy (PDR) framework, most of detection models based on model-based interpretability lack good detection performance. To solve the problem, in this paper, we have proposed a novel intrusion detection system model based on model-based interpretability, called Interpretable Intrusion Detection System (I^{2}DS). We firstly combine normal and attack samples reconstructed by AutoEncoder (AE) with training samples to highlight the normal and attack features, so that the classifier has a gorgeous effect. Then, Additive Tree (AddTree) is used as a binary classifier, which can provide excellent predictive performance in the combined dataset while maintaining good model-based interpretability. In the experiment, UNSW-NB15 dataset is used to evaluate our proposed model. For detection performance, I^{2}DS achieves a detection accuracy of 99.95%, which is better than most of state-of-the-art intrusion detection methods. Moreover, I^{2}DS maintains higher simulatability and captures the decision rules easily.

1. Introduction

With the rise of technologies such as the Internet of Things and cloud computing and the advent of the era of big data, the network security environment has become even worse. The increasingly frequent network attacks have caused security researchers to refocus on network intrusion detection. At present, the network intrusion detection system can be divided into two types according to the detection method, one is the abnormal intrusion detection system and the other is the misuse of the intrusion detection system [1]. In terms of the ability to detect new attacks, anomalous intrusion detection systems have a more prominent effect than misuse of intrusion detection systems, which make anomaly intrusion detection systems necessary for intrusion detection [2]. At the same time, the important role of machine learning in anomalous network intrusion detection has made machine learning the mainstream of constructing anomalous intrusion detection systems.

The establishment of an effective intrusion detection system first requires a dataset that conforms to the current network environment. Since 1999, the KDDCUP99 dataset and the NSL-KDD dataset have been widely used in the construction of network intrusion detection systems. Many intrusion detection models use these two datasets, such as the ANN and fuzzy clustering model [3], and the model combined misuse and anomaly detection for intrusion detection [4]. However, a study explains the reasons why these two datasets cannot reflect the output performance of the network intrusion detection system [5]. (1) The attack types of these two datasets are only a small part of modern network attack methods. (2) These two datasets were established in 1999 and are quite different from modern network traffic benchmarks. (3) The different distribution of the training and test datasets in the data type will cause the deviation of the classifier and the accuracy will decrease.

In addition, building an intrusion detection system requires a dynamic detection model. In recent years, many powerful intrusion detection models have been proposed, which makes the detection effect better than before, especially when deep learning becomes the mainstream. For example, the detection accuracy of DL-IDS using Convolutional Neural Network (CNN) and Long Short-Term Memory Network (LSTM) reaches 98.67% in the CICIDS2017 dataset [6]. In addition, some hybrid detection models based on traditional machine learning are performing well, for example, the detection performance of IDS based on decision tree and rules-based concepts reaches 96.67% in the CICIDS2017 dataset [7]. However, although the predictive accuracy of intrusion detection models is very high, most of them lack great interpretability. Some of them achieve higher detection accuracy but use a black box model or lower interpretable model, and some of them use an effective model-based interpretable model such as a decision tree and rule-based model but have lower detection accuracy than some state-of-the-art methods.

In this paper, we have proposed a novel intrusion detection model based on model-based interpretability, which can achieve higher detection accuracy and interpretability. In the proposed model, AddTree, a model-based interpretable method is used as classifier that can provide remarkable predictive performance and obtain admirable interpretability, and AE is used as feature rebuilder then can improve the predictive accuracy. Our contributions to this research are as follows:

(1) We have proposed an intrusion detection model using AE and AddTree, in which we use AE to highlight the normal and attack features, respectively, and use AddTree and voting machine to detect whether a sample belongs to normal or attack.

(2) We have conducted comparative experiments on UNSW-NB15 datasets, illustrating that I^{2}DS outperforms most of the state-of-the-art methods.

(3) According to the PDR framework, our proposed model has outstanding predictive accuracy and higher descriptive accuracy and relevancy. AddTree, as a kind of decision tree, has superior simulatability because of its high prediction performance and user-friendly and visual decision process. Combining with AE, I^{2}DS achieves prominent predictive performance.

In this paper, the other sections are as follows: Section 2 proposes a review of related work to intrusion detection and interpretability of machine learning and its application. Section 3 supplies the details of our proposed detection model. Section 4 presents experimental details and the comparison of our model and other machine learning algorithms and the interpretability analysis of our model. Section 5 indicates the conclusion and an overall review of our research results.

2. Related Work

2.1. Intrusion Detection. A number of studies have been conducted on intrusion detection from the network traffic perspective (Table 1). Gharaee and Hosseinvand proposed IDS based on genetic algorithm (GA) and support vector machines (SVM) [8]. They built new fitness function based on true positive rates, false positive rates, and computation time using SVM. Using KDDCUP-99 and UNSW-NB15 datasets, the detection performance of the model reached 99.26% and 93.24%, respectively. Huang and Lei proposed a novel Imbalanced Generative Adversarial Network (IGAN) and applied it into intrusion detection system [9]. The IGAN module was composed of CNN and fully connection network (FCN), and it was used to balance the percentage between normal and attack samples. For NSL-KDD, UNSW-NB15, and CICIDS2017 datasets, the detection rates of the model were 84.45%, 82.53%, and 99.79%, respectively. Liu et al. proposed a detection system based on feature representation and data augmentation [10]. The model was mainly divided into three parts: feature extraction, data augmentation, and detection. First, the data set is converted into an image set through the steps of feature encoding, feature reduction, standardization, and recirculation pixel permutation strategy. Then, least squares generative adversarial network was used to balance the training dataset and the convolutional neural network was used as classifier. NSL-KDD and UNSW-NB15 datasets were used to evaluate the proposed model. The model accuracy reached 98.80% and 94.90%, respectively. Zhang et al. proposed an intrusion detection model based on conditional Wasserstein Generative Adversarial Network (CWGAN) and cost-sensitive stacked autoencoders (CSSAE) [11]. CWGAN was used to generate the minority samples to reduce the class imbalance. CSSAE was used to extract deep feature representation of the data and detect attacks by utilizing cost-sensitive loss function. KDDTest+, KDDTest-21, and UNSW-NB15 datasets were used to evaluate the proposed model. The accuracy of model achieved 90.34%, 80.78%, and 93.27%. Ferrag et al. proposed rules and decision tree-based intrusion detection system [12]. This model consisted of three classifiers. The first classifier was REP tree, a binary classifier used to detect normal and attack. The second classifier was Jrip, detecting benign and one of the different categories of attacks. The third classifier was Forest PA, and its input was the result of the first two classifiers and the entire training set. For CICIDS2017 and BoT-IoT datasets, the detection rate of RDTIDS achieved 96.66% and 96.99%, respectively.

2.2. Interpretability of Machine Learning and its Applications. In recent years, deep learning models have very splendid performance in many fields (Table 2), such as face recognition, image classification, and natural language processing, but this performance is more dependent on the model's highly nonlinear and parameter adjustment technology [16]. From the perspective of human beings, if the decision-making process of the model is incomprehensible, the model is unexplainable [17]. Therefore, interpretability can be defined as users have enough understandable information to understand the decision-making process and the decision result of the model. In machine learning, interpretability is divided into two categories: (1) model-based interpretability that constructs a model which is interpretable in nature and

Table 1: Summary of related work of intrusion detection system.

Reference	Algorithm	Dataset	Accuracy
[8]	GA + SVM	KDDCUP-99, UNSW-NB15	99.26% (average), 93.24% (average)
[9]	IGAN + DCNN	NSL-KDD, UNSW-NB15, CICIDS2017	84.45%, 82.53%, 99.79%
[10]	NADS-RA	NSL-KDD, UNSW-NB15	98.80%, 94.90%
[11]	CWGAN-CSSAE	KDDTest+, KDDTest-21, UNSW-NB15	90.34%, 80.78%, 93.27%
[12]	RDTIDS	CICIDS2017, BoT-IoT	96.66%, 96.99%

Table 2: Summary of related work of interpretability.

Reference	Main work in interpretability
[13]	Using the decision tree encodes CNN's decision pattern as the quantitative basis of each prediction that can explain CNN's prediction at the semantic level
[14]	Using gradient boosting to replace gini to improve the predictive performance of the decision tree
[15]	Explaining and redefining interpretability from three aspects, predictive accuracy, descriptive accuracy, and relevancy

(2) post hoc interpretability that applying an interpretability method after training a black box model.

There are lots of researches that focus on it. Zhang et al. used the decision tree to quantitatively explain the logic of deep neural network prediction at the semantic level [13]. They used the decision tree to interpret the neural network prediction results for each input image, determined which parts of the object are used for prediction, and quantified the contribution of each object part to the prediction. Luna et al. proposed a new decision tree model called Additive Tree [14]. They combined Gradient Boosting with Classification and Regression Trees (CART) to build a more accurate tree that improved the predictive performance and maintained the characteristic of the decision tree. Murdoch et al. proposed a PDR framework for interpretability [15]. They defined interpretability of three categories: predictive accuracy, descriptive accuracy, and relevancy.

3. Materials and Methods

3.1. Overview of Approach. The main contribution of the proposed model is to detect network traffic and label them as benign or attack. This detection process can be divided into two phases (Figure 1): (1) we do data preprocessing for data sets, including data standardization and data cleaning. (2) We use the proposed model to detect the dataset and then get the detection result.

3.2. Dataset. In this paper, the dataset we use is UNSW-NB15, which is developed by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security [18]. The Argus and Bro-IDS tools are used, and twelve algorithms are developed to generate totally 49 features with the class label. The created features can be classified into five categories: flow features, basic features, content features, time features, and additional generated features (Table 3). The dataset contains 9 attack types, including Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms [19]. However, the attributes in UNSW-NB15 training and testing dataset (UNSW_NB15_training-set.csv and UNSW_NB15_testing-set.csv) [20] contain only 44 features, including 42 attributes and 2 categories. Table 4 illustrates the description of training-set and testing-set of UNSW-NB15.

3.3. Data Processing. In this section, the UNSW-NB15 data processing operations are as follows:

(1) We split UNSW-NB15 into two datasets by label category

(2) We remove the columns named "id" and "attack_cat"

(3) We encode the columns named "proto," "service" and "state"

(4) We normalize the data with the minimum-maximum normalization method which is defined as $x_i - \min(x)/\max(x) - \min(x),\ i = 1, 2, \ldots, k$

3.4. Proposed Model. In this section, we describe I^{2}DS we proposed dealing with problem of network intrusion detection. It combines an unsupervised approach using two AEs with a supervised machine learning model using Additive Tree. The architecture of the proposed model is described in Figure 2.

3.5. Feature Selection Using Autoencoder. Different from classical multilayer perceptron (MLP), an autoencoder (AE) is a particular neural network which tries to copy the input to the output. In other words, the task of AE is an attempt to the best to make the output content the same as the input content. In particular, AE contains two parts: encoder $h = f(x)$ and decoder $r = g(h)$. For input data x, AE can make the x be approximately equal to $g(h(x))$.

In this paper, we use two AEs to learn features of attack and normal, respectively. The loss function is defined as $\mathrm{MSE} = 1/MN \sum_{j=1}^{M} \sum_{k=1}^{N} (x_{j,k} - x'_{j,k})^2$. The architecture of AE is described in Table 5. In our proposed model, AE reestablishes normal and attack samples, respectively, which

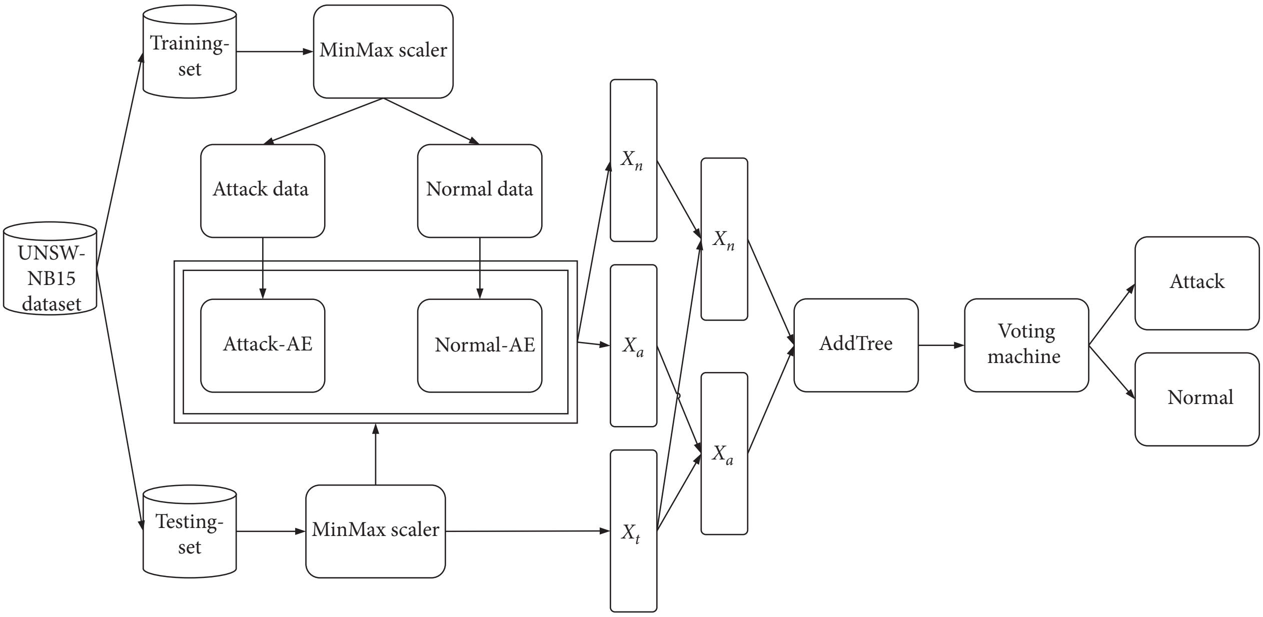


Figure 1: Overview of our proposed analysis framework.

Table 3: Feature description of UNSW-NB15 training set and testing set.

Categories	Features	Categories	Features
Flow features	Proto	Basic features	State
	Rate		dur
Content features	swin		sbytes
	dwin		dbytes
	stcpb		Sttl
	dtcpb		dttl
	smeansz		sloss
	dmeansz		dloss
	trans_depth		Service
	res_bdy_len		sload
Additional generated features	is_sm_ips_ports	Time features	dload
	ct_state_ttl		spkts
	is_ftp_login		dpkts
	ct_ftp_cmd		sjit
	ct_srv_src		djit
	ct_srv_dst		sintpkt
	ct_dst_ltm		dintpkt
	ct_src_ ltm		tcprtt
	ct_src_dport_ltm		synack
	ct_dst_sport_ltm		ackdat
	ct_dst_src_ltm	Class	Label

Table 4: Dataset description, including attributes, total samples, normal samples (and their percentage on the total samples), and attack samples (and their percentage on the total samples).

Dataset	Attributes	Total	Normal (%)	Attack (%)
UNSW-NB15train	43	82,332	37,000 (44.9%)	45,332 (55.1%)
UNSW-NB15test	43	175,341	56,000 (31.9%)	119,341 (68.1%)

can emphasize the normal or attack features. The benefit of this is that the original data can enhance the main features and weaken unrelated features by combining with data generated by AE.

3.6. Binary Classifier Using Additive Tree. In order to build a decision tree model with higher predictive accuracy, Luna et al. proposed a new decision tree model, called Additive Tree (AddTree) [14]. The Additive Tree walks like CART but learns like Gradient Boosting. In other words, it is an algorithm that builds a single decision tree, similar to CART, but the training is similar to boosting stumps (a stump is a tree of depth 1). More specifically, the steps of AddTree are similar to those of Gradient Boosting. Its iterative steps are as follows: (1) calculate the negative gradient, (2) fit the weak learner by minimizing the product of the square error and the instance weight, (3) find the optimal scale by minimizing the product of the instance weight and the square of the difference between the estimated function, the weak learner factor, and the original data, (4) update the current function estimate, and (5) finally calculate the weights of the left and right subtrees, until the node field and the classifier partition are empty. In other words, unlike CART, AddTree uses weight to measure the partitioning effect of the dataset not the Gini coefficient. This improves accuracy without compromising interpretability [14].

3.7. Model Design. The algorithm steps of our proposed model are as follows:

(1) We use the attack-set and normal-set obtained in data processing to train the AE (attack-AE and normal-AE).

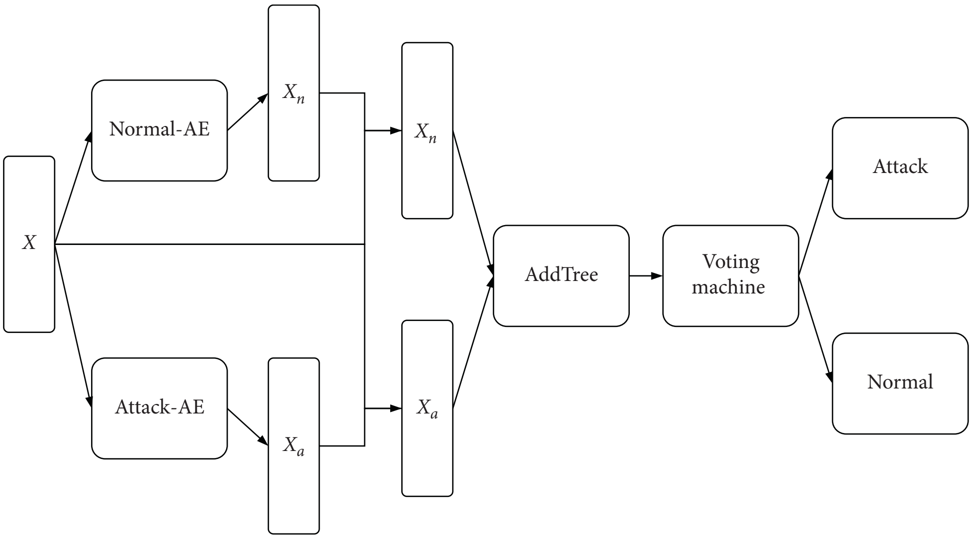


Figure 2: Building intrusion detection model using AddTree and AE.

Table 5: The architecture of AE.

Layer	Shape	Parameter
Input	(None, 42)	0
Dense_encoder_1	(None, 30)	1290
Dense_encoder_2	(None, 20)	620
Dense_decoder_1	(None, 30)	630
Dense_decoder_2	(None, 42)	1302

(2) We put the UNSW-NB15 testing-set into attack-AE and normal-AE to obtain reconstructed datasets (re-attack-set and re-normal-set).

(3) We combine UNSW-NB15 Testing-set with re-attack-set and re-normal-set, respectively (attack-addtree-set and normal-addtree-set).

(4) We use attack-addtree-set and normal-addtree-set to train AddTree and then put the results into voting machine to get the final detection result.

4. Results and Discussion

4.1. Performance Metrics. To evaluate our proposed model, we use recall, precision, F_1-score, and accuracy as the primary metrics. Accuracy is the proportion of all correct predictions to the total. Precision refers to the proportion of true correctness that is positive for all predictions, while recall refers to the proportion that is really correct and accounts for all the actual positives. F_1 score is the harmonic mean of recall and precision, considering both the classifying ability and the detection rate.

Four statistical standards are defined as follows:

$$\begin{aligned}
\text{recall} &= \frac{\text{true positive}}{\text{true positive } + \text{false negative}},\\
\text{precision} &= \frac{\text{true positive}}{\text{true positive} + \text{false positive}},\\
F_1 - \text{score} &= \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}},\\
\text{accuracy} &= \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{false negative} + \text{true negative} + \text{false positive}}.
\end{aligned} \tag{1}$$

4.2. Predictive Accuracy. In this section, we compare our proposed model I^2DS with IDS-AddTree which not use AE (Table 6). It can see from the results that, in the primary metrics, I^2DS is better than IDS-AddTree.

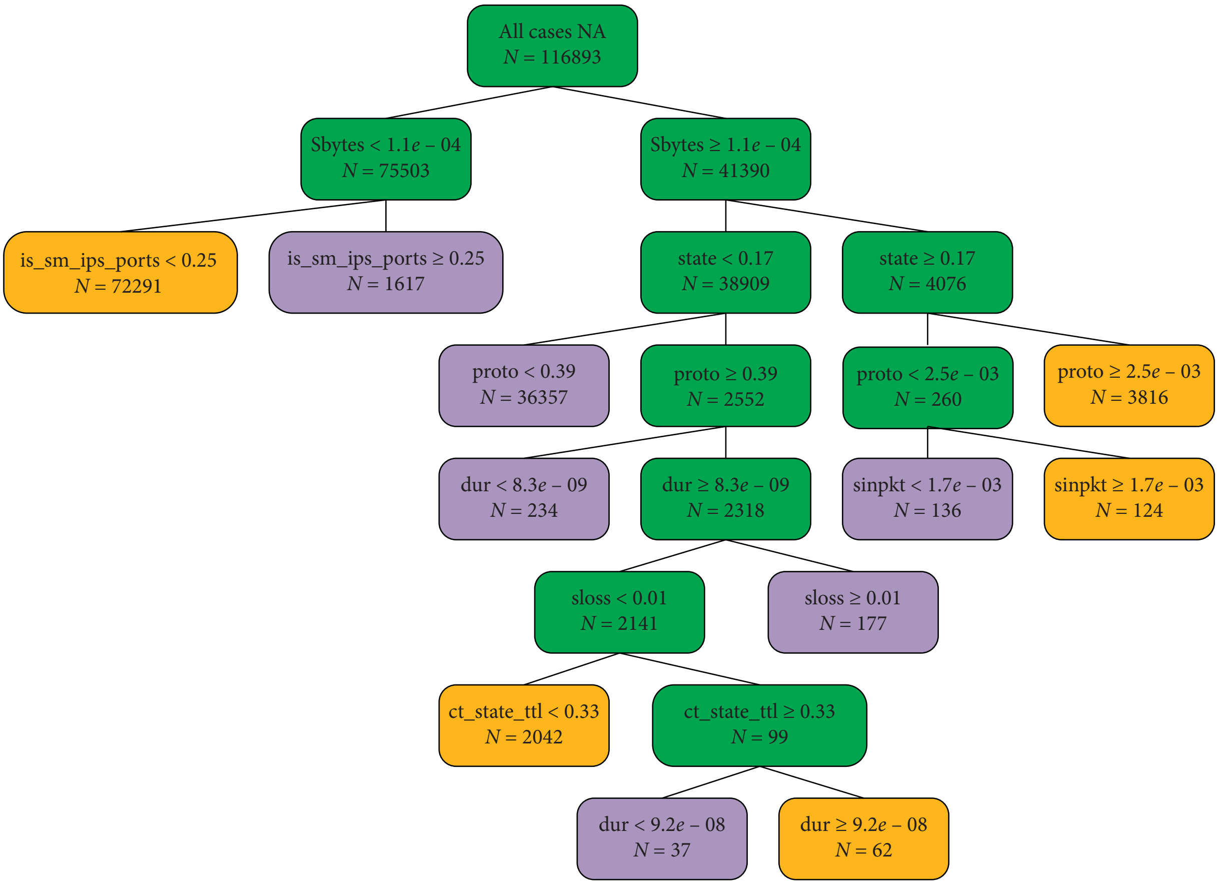
In addition, we also compare our proposed model with other state-of-the-art classifiers (Table 7). It can be seen that the accuracy of our proposed model is the best, achieving 99.95%. For other evaluation metrics, the precision, recall, and the F_1-score of our proposed model are higher than those of other classifiers. Therefore, the proposed method achieves satisfactory performance across evaluation metrics when compared to other classifiers using the UNSW-NB15 dataset.

Table 6: Comparison between I^{2}DS (using AE) and IDS-AddTree (not using AE).

Classifier	Accuracy	Precision	Recall	F_1-score
IDS-AddTree	0.9592	0.9623	0.9786	0.9703
I^{2}DS	0.9995	0.9994	0.9999	0.9996

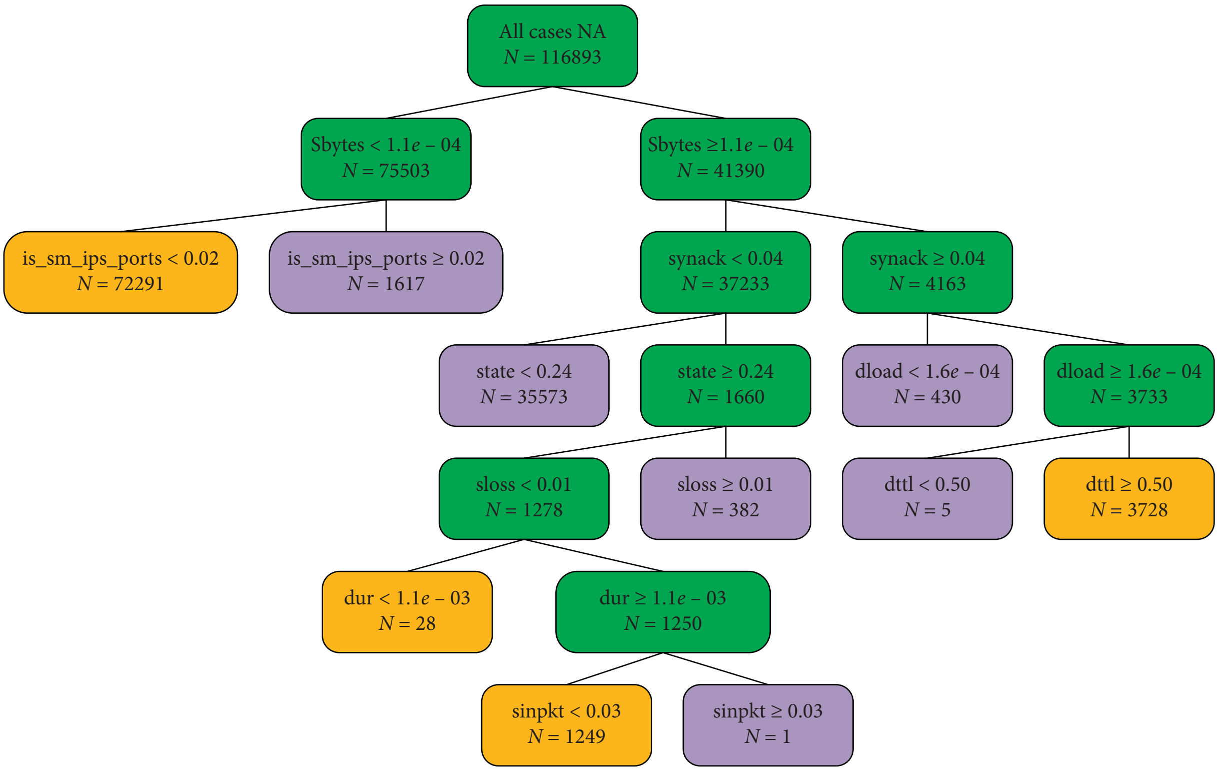
Table 7: Comparison of the proposed model and other classifiers with UNSW-NB15.

Classifier	Accuracy	Precision	Recall	F_1-score
IGAN-IDS [9]	0.8253	0.8486	0.8445	0.8286
NADS-RA [10]	0.9490	0.9820	0.9250	0.9530
CWGAN-CSSAE [11]	0.9327	0.9259	0.9543	0.9399
HNGFA [21]	0.9024	0.9277	0.8248	0.8536
RHF-ANN [22]	0.9760	0.9550	0.9990	0.9770
GA + SVM [23]	0.9610	0.9830	0.9820	0.9820
SVM + EML + K-means [24]	0.9450	0.9480	0.9970	0.9720
Wrapper + neurotree [25]	0.9710	0.9500	0.9830	0.9660
HC-IBGSA + SVM [26]	0.9847	—	—	—
MINDFUL [27]	0.9340	—	—	0.9529
I^{2}DS (our proposed model)	**0.9995**	**0.9994**	**0.9999**	**0.9996**



(a)

Figure 3: Continued.



(b)

FIGURE 3: The tree of (a) attack and (b) normal classes.

4.3. Interpretability. Predictive accuracy, descriptive accuracy, and relevancy (PDR) framework is a method to define interpretable machine learning [15]. According to the PDR framework, if a model has exceptional interpretability, this model should have higher predictive accuracy, descriptive accuracy, and relevancy. Decision tree is considered as an interpretable model because it has higher simulatability. Human beings can use input data and decision tree model parameters in the appropriate time to make predictions through each calculation step [28]. In the decision tree, each leaf node of the tree represents a class and is interpreted by the path from the root node to the leaf node in terms of a rule such as "If A1 and A2 and A3, then class C1," where A1, A2, and A3 are the clauses involving the attributes and C1 is the class label [29]. However, although the traditional decision tree has good simulatability, it is very poor in prediction performance. Therefore, AddTree, which is proposed by Luna et al., makes up for the poor prediction performance of traditional decision tree [14]. It maintains the advantages of the decision tree that it can be visualized directly which means it can help to understand the influence of different variables on decision-making results by using the criteria of information theory and provides the same great predictive performance as ensemble learning and neural networks.

In terms of predictive accuracy, I^{2}DS reaches 99.95% and outperforms most advanced intrusion detection models. In terms of descriptive accuracy and relevancy, AddTree, as a classifier to classify a certain sample, can be visualized directly and easily (Figure 3), and in the AE and voting machine, the architecture is based on fully connect network, which can be seen as the process of matrix linear transformation. In conclusion, in the model-based interpretability, I^{2}DS obtains higher interpretability.

5. Conclusion

This paper proposes a more accurate and interpretable Intrusion detection model using AutoEncoder and Additive Tree, called I^{2}DS. In our proposed model, two autoencoders are learned from normal and attack flows, respectively. They can highlight the main features of traffic flows during reconstruction. AddTree is used as classifier to classify which class a certain sample belongs to. Additionally, the UNSW-NB15 dataset is used to evaluate the proposed model.

In terms of predictive accuracy, we use recall, precision, F_1-score, and accuracy as the primary metrics. The results demonstrate that the primary metrics of I^{2}DS is better than most of state-of-the-art intrusion detection methods. In terms of descriptive accuracy and relevancy, AddTree of I^{2}DS maintains the characteristic of a decision tree that can be easily visualized and capture the interactive information between features. Autoencoder of I^{2}DS is composed of fully connect network, which means the AE can be regarded as linear matrix transformation.

In general, according to the PDR framework, the interpretability of I^{2}DS can be said to be super-excellent. I^{2}DS has the excellent predictive performance and the high-level descriptive accuracy and relevancy. We come to the conclusion that I^{2}DS provides a first-class predictive performance and credible interpretation of intrusion detection system.

Data Availability

The dataset can be obtained in the Kaggle (https://www.kaggle.com/wenfengxu/i2ds-for-addtree), and the code can be obtained in the Github (https://github.com/Xuwenfeng-GUET/I2DS).

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61762026 and 61462018, in part by Guangxi Natural Science Foundation under Grant 2017GXNSFAA198278, and in part by the Innovation Project of GUET Graduate Education under Grant 2019YCXS056.

References

[1] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proceedings of the 1999 IEEE Symposium Security and Privacy*, pp. 120–132, Oakland, CA, USA, May 1999.

[2] A. S. A. Aziz, A. T. Azar, A. E. Hassanien, and S. E.-O. Hanafy, "Continuous features discretization for anomaly intrusion detectors generation," in *Soft Computing in Industrial Applications*, vol. 223, pp. 209–221, Springer, Cham, Switzerland, 2014.

[3] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6225–6232, 2010.

[4] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014.

[5] N. Moustafa and J. Slay, "Creating novel features to anomaly network detection using DARPA-2009 data set," in *Proceedings of the 14th European Conference on Cyber Warfare and Security ECCWS-2015*, Hatfield, UK, July 2015.

[6] P. Sun, "DL-IDS: extracting features using CNN-LSTM hybrid network for intrusion detection system," *Security and Communication Networks*, vol. 2020, Article ID 8890306, 11 pages, 2020.

[7] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, "A novel hierarchical intrusion detection system based on decision tree and rules-based models," in *Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 228–233, Santorini, Greece, May 2019.

[8] H. Gharaee and H. Hosseinvand, "A new feature selection IDS based on genetic algorithm and SVM," in *Proceedings of the 2016 8th International Symposium on Telecommunications (IST)*, pp. 139–144, Tehran, Iran, September 2016.

[9] S. Huang and K. Lei, "IGAN-IDS: an imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks," *Ad Hoc Networks*, vol. 105, Article ID 102177, 2020.

[10] X. Liu, X. Di, Q. Ding et al., "NADS-RA: network anomaly detection scheme based on feature representation and data augmentation," *IEEE Access*, vol. 8, pp. 214781–214800, 2020.

[11] G. Zhang, X. Wang, R. Li, Y. Song, J. He, and J. Lai, "Network intrusion detection based on conditional Wasserstein generative adversarial network and cost-sensitive stacked autoencoder," *IEEE Access*, vol. 8, pp. 190431–190447, 2020.

[12] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, "RDTIDS: Rules and decision tree-based intrusion detection system for internet-of-things networks," *Future Internet*, vol. 12, no. 3, p. 44, 2020.

[13] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu, "Interpreting CNNs via decision trees," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6254–6263, Long Beach, CA, USA, June 2019.

[14] J. M. Luna, E. D. Gennatas, L. H. Ungar et al., "Building more accurate decision trees with the additive tree," *Proceedings of the National Academy of Sciences*, vol. 116, no. 40, pp. 19887–19893, 2019.

[15] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Definitions, methods, and applications in interpretable machine learning," *Proceedings of the National Academy of Science*, vol. 116, no. 44, pp. 22071–22080, 2019.

[16] Y. Hua, D. Zhang, and S. Ge, "Research progress in the interpretability of deep learning models," *Journal of Cyber Security*, vol. 5, no. 3, pp. 1–12, 2020.

[17] W. Fei, L. Binbing, and H. Yahong, "Interpretability for deep learning," *Aero Weaponry*, vol. 26, pp. 39–46, 2019.

[18] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, November 2015.

[19] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Systems Security*, vol. 25, no. 1–3, pp. 18–31, 2016.

[20] M. Nour, "UNSW_NB15 Dataset," *IEEE Dataport*, 2019.

[21] R. Elhefnawy, H. Abounaser, and A. Badr, "A hybrid nested genetic-fuzzy algorithm framework for intrusion detection and attacks," *IEEE Access*, vol. 8, pp. 98218–98233, 2020.

[22] F. E. Ayo, S. O. Folorunso, A. A. Abayomi-Alli, A. O. Adekunle, and J. B. Awotunde, "Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection," *Information Security Journal: A Global Perspective*, vol. 29, no. 6, pp. 267–283, 2020.

[23] B. M. Aslahi-Shahri, R. Rahmani, M. Chizari et al., "A hybrid method consisting of GA and SVM for intrusion detection system," *Neural Computing & Applications*, vol. 27, no. 6, pp. 1–8, 2016.

[24] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296–303, 2017.

[25] S. S. S. Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Expert Systems with Applications*, vol. 39, no. 1, pp. 129–141, 2012.

[26] M. R. Gauthama Raman, N. Somu, S. Jagarapu et al., "An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm," *Artificial Intelligence Review*, vol. 53, no. 5, pp. 3255–3286, 2019.

[27] G. Andresini, A. Appice, N. D. Mauro, C. Loglisci, and D. Malerba, "Multi-channel deep feature learning for intrusion detection," *IEEE Access*, vol. 8, pp. 53346–53359, 2020.

[28] Z. C. Lipton, "The mythos of model interpretability," *Communications of the ACM*, vol. 61, no. 10, 2016.

[29] J. Basak and R. Krishnapuram, "Interpretable hierarchical clustering by constructing an unsupervised decision tree," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 1, pp. 121–132, 2005.