Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 6488571, 9 pages
https://doi.org/10.1155/2022/6488571

Research Article

Deep Neural Network-Based Intrusion Detection System through PCA

Shoayee Dlaim Alotaibi,[1] Kusum Yadav,[1] Arwa N. Aledaily,[1] Lulwah M Alkwai,[1] Alaa Kamal Yousef Dafhalla,[1] Shahad Almansour,[1] and Velmurugan Lingamuthu[2]

[1]*College of Computer Science and Engineering, University of Ha'il, Kingdom of Ha'il, Saudi Arabia*
[2]*Department of Computer Science, School of Informatics and Electrical Engineering, Hachalu Hundesa Campus, Ambo University, Ambo, Ethiopia*

Correspondence should be addressed to Velmurugan Lingamuthu; velmurugan.lingamuthu@ambou.edu.et

Received 3 March 2022; Revised 2 April 2022; Accepted 8 April 2022; Published 9 May 2022

Academic Editor: Amandeep Kaur

Today, challenges such as a high false-positive rate, a low detection rate, a slow processing speed, and a big feature dimension are all part of intrusion detection. To address these issues, decision trees (DTs), deep neural networks (DNNs), and principal component analysis (PCA) are available. Through a higher detection rate and a lower false-positive rate, the research-based intrusion detection model DT-PCA-DNN increases the processing speed of intrusion detection systems (IDSs). To minimize the overall data volume and accelerate processing, DT is used to initially differentiate the data. Differentiate DTs save the temporary training sample set for intrusion data in order to retrain and optimize the DT and DNN, treat the DT judges as standard data, and delete the added average data. After signing, we should lower the dimension of the data using PCA and then submit the data to DNN for secondary discrimination. However, DT employs a shallow structure in order to prevent an excessive quantity of average numbers from being interpreted as intrusion data. As a result, additional DNN secondary processing cannot effectively increase the accuracy. DNN accelerates data processing by utilizing the ReLU activation function from the simplified neural network calculation approach and the faster convergence ADAM optimization algorithm. Class two and five trials on the NSL-KDD dataset demonstrate that the proposed model is capable of achieving high detection accuracy when compared to other deep learning-based intrusion detection approaches. Simultaneously, it has a faster detection rate, which effectively solves the real-time intrusion detection problem.

1. Introduction

Communication systems and network entrances are always faced with network attacks from the outside or even within their systems and are not like single attacks in the immature period of the network. Today, most of the intrusion behaviors are of various types and are developing in a mixed situation. Development is getting harder. According to the relevant literature, the Yahoo data breach caused a loss of 350 million US dollars and the "Bitcoin" breach caused a loss of about 70 million US dollars [1]. Based on intrusion behavior, the intrusion detection can be divided into network-based intrusion detection system (NIDS) and host-based intrusion detection system (HIDS) [2].

Various log files, disk resource information, and system information are used to detect intrusion behavior, while NIDS judges whether there is intrusion behavior by detecting the data packets in and out of the local network data flow. Machine learning, as a very popular algorithm tool in recent years, deserves experts and scholars to try its application in intrusion detection [3]. Especially in recent years, the application of machine learning in intrusion detection has appeared in people's field of vision; the support vector machine (SVM) to neural network (NN) to random forest (RF) has their applications in intrusion detection.

Machine learning has a long history in intrusion detection. In 2003, the literature [4] demonstrated that decision trees (DTs) could detect intrusions faster than the Snort detection engine [5] at the time. The proposed method of joint optimization of feature selection and SVM training model is demonstrated using the intrusion detection dataset. The results indicate that the joint optimization method outperforms the SVM in terms of performance and convergence speed. The literature [6] used support vector machines to identify intrusions and explored the real-time problem; however, the accuracy rate was low. The literature [7] suggested reducing dimensionality by principal component analysis (PCA) and then detecting intrusions with support vector machines (SVMs).

The self-optimization technology increases the classifier's accuracy and decreases training and testing time [8]. In 2019, the literature [9] developed an intrusion detection model based on an upgraded convolution neural network that has a high intrusion detection accuracy and true positive rate, while exhibiting a low false-positive rate. In the same year, Fernandez et al. proposed training an intrusion detection system using a feedforward fully connected deep neural network (DNN) (IDS). Due to the fact that DNN demonstrated robustness in the scenario of dynamic IP address allocation, the model they developed has a broader variety of real-world applications [10]. Still in 2019, the literature introduced the ICA-DNN intrusion detection model [11], which is based on ICA (independent component analysis) and DNN and has a higher feature learning capability than some shallow machine learning models. This has increased classification accuracy, but the algorithm's prediction time is not particularly tested and the model performs poorly in real time. According to the intrusion detection models proposed by the aforementioned scholars, it is discovered that the majority of research studies pay insufficient attention to real-time intrusion detection, while a few intrusion detection models with more in-depth real-time performance research suffer from low detection accuracy. To thoroughly investigate the real-time problem that is critical for intrusion detection and to ensure intrusion detection accuracy, this research offers the DT-PCA-DNN model. The trained DT is basically a series of if-else statements that handle huge batches of data quickly but with insufficient precision; the DNN network has a slow real-time performance but high precision when processing vast volumes of high-dimensional data. By combining the two, data are prefiltered using DT and then fed to DNN following PCA. The experimental results demonstrate that the model significantly accelerates training and detection while maintaining a high detection rate. Researchers are proposing different security protocols [12–15] for providing confidentiality and privacy against security attacks.

2. Basic Theory

2.1. Principal Component Analysis. PCA is a very commonly used linear dimensionality reduction algorithm as shown in Figure 1. It is generally used to extract the main components of high-dimensional data and simplify them into low-dimensional data, but the integrity of the data can be adjusted according to the requirements. Specifically, PCA hopes to map the original feature space into another orthogonal space and hopes to use a hyperplane that satisfies the nearest reconstruction and maximum separability to properly describe all the data in the dataset.

The projections of different points on the dataset on this hyperplane should be as far away as possible.

2.2. Decision Tree. The decision tree model is a tree structure that describes the classification of instances. It consists of nodes and directed edges. There are two types of nodes. Internal nodes represent a feature or attribute [12]. The decision tree starts from the root node and continuously splits according to the characteristics of the data, until all the data reach the leaf nodes. The attributes used as the basis for splitting must be discrete attributes, and for continuous attributes, they can be discretized according to experimental requirements. A sales office wants to judge whether the surveyed object has a house purchase demand based on the object's identity information, age, and whether he is married. The survey results are shown in Table 1.

From this, the corresponding decision tree can be generated as shown in Figure 2. When an object "ω" is a company employee, is married, and is 29 years old, it can be known that he has a housing demand according to the decision tree.

In the above example, the feature selection order of the survey objects is different and different decision trees can be generated. According to the selection of different splitting features, there are three judgment bases, namely, information gain, gain rate, and Gini index. The core of the ID3 (iterative dichotomiser 3) algorithm is to apply the information gain criterion on each node of the decision tree to select features, the C4.5 algorithm uses the information gain ratio to select features, and the CART (classification and regression tree) uses the Gini index as the selection feature in accordance. The decision tree pruning and specific feature selection will not be repeated due to space reasons. Reference [16] is sufficient.

2.3. Deep Neural Networks. A neural network is a large parallel interconnected network composed of simple adaptive units that can be used to simulate the interactive response of the biological nervous system to real-world objects [17], where machine learning is used to interact with neural networks in a broader sense.

In a neural network, the most fundamental structure is the neuron model, which is a simple unit by definition. Each neuron in a biological neural network is connected to other neurons, and when it is "stimulated," it releases neurotransmitters to the related neurons, altering their potential. When a neuron's potential crosses a "threshold" and the neuron becomes activated, or "excited," and begins delivering neurotransmitters to its associated neurons [18].

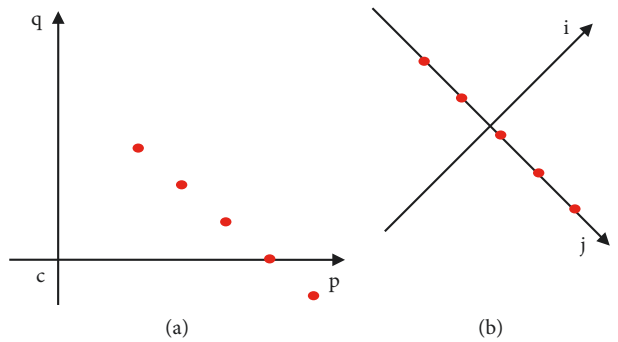

Figure 1: Diagram of data dimensionality reduction. (a) Before transformation. (b) After transformation.

Table 1: Respondent information and willingness.

Object	Identity	Age	Are you married	Whether to buy a house
A	Student	27	Yes	None
B	Staff	29	No	None
Γ	Staff	25	Yes	Have
Δ	Student	21	No	None
Λ	Staff	43	Yes	None

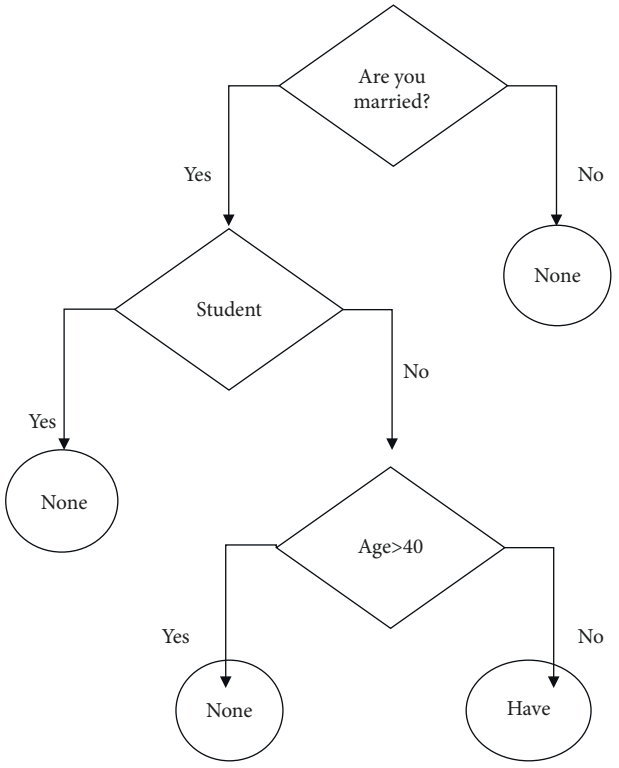

Figure 2: Decision chart of the respondent.

Foreign scholars abstracted the above situation in 1943 into the simple model depicted in Figure 3, dubbed the "M-P neuron model" [19].

In this model, ai denotes the input from the ith neuron, wi is the ith neuron's connection weight, and θ is the threshold. The neuron receives input signals from n other neurons. The signals are transmitted via weighted

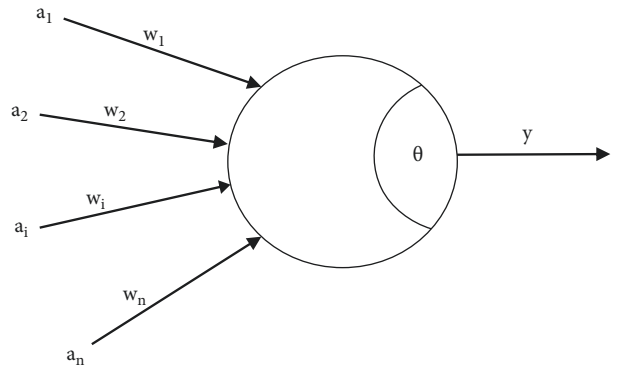

Figure 3: M-P neuron model.

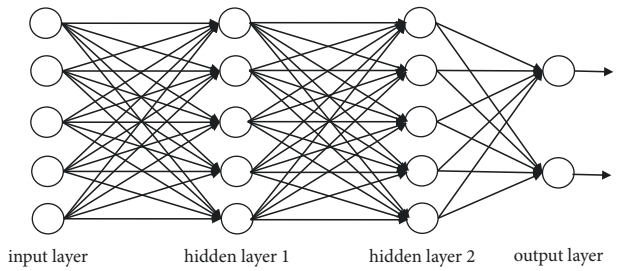

Figure 4: Double hidden layers of fully connected DNN.

connections, and neurons receive them. The resulting total input is compared, and the neuron's output y is obtained by processing the activation function $f(a)$ as specified in

$$o = f\left(\sum_{i=1}^{n} aiwi - \theta\right). \quad (1)$$

The sigmoid function, the tanh function, and the ReLU function are all frequently used activation functions [20]. A neural network is formed by connecting many neurons [21–23]. The term "deep neural network" refers to a neural network with more than two layers and more than two hidden layers. A neural network with input and output layers is only capable of solving linear problems. The hidden layer is introduced to address the nonlinear separable problem. Figure 4 illustrates a fully connected neural network, which means that any neuron in the preceding layer

must be connected to any neuron in the next layer. The neurons in the input layer accept only information and do not process functions. The neural network's learning process is fundamentally one of continuously adjusting the connection weights and thresholds of neurons in order to approach the output outcomes of the training samples. Among them, the most remarkable method is the error back-propagation (BP) algorithm, which is used for the majority of neural network training today.

3. System Design

The overall design of the system is shown in Figure 5. The first step is to preprocess the overall dataset. Data preprocessing first normalizes continuous data and secondly performs one-hot encoding on discrete valued data. The dataset after data preprocessing is divided into training dataset and test dataset.

The processed training dataset builds DT and trains DNN at the same time. PCA is unsupervised learning and does not need to be trained. After the DT and DNN training is completed, the DT-PCA-DNN model is established. At this time, the established intrusion detection model is tested with the preprocessed test dataset and the relevant parameters are adjusted and perfected. The trained DT is actually a series of if-else statements, processing large batches of data at high speed, but with insufficient processing precision; DNN network has poor real-time performance when processing large amounts of high-dimensional data, but high precision; PCA just happens to solve the problem caused by the high data dimension. Combining the three, firstly use DT to prefilter the data, then use PCA, and then send them to DNN for secondary classification, and use DT to filter out the intrusion data that is easy to judge and lighten the workload of DNN. PCA solves the problem that DNN network encounters high-dimensional data and slow training.

The three methods make up for the shortcomings of each other and ensure the accuracy of detection while having good real-time performance.

3.1. Data Processing. The data processing should be divided into two parts. First, the continuous data are normalized, and then the discrete valued data are encoded.

3.1.1. Data Normalization. In this study, the normalization process of the experiment adopts min-max normalization. The normalization method is to perform linear transformation on the original data and the transformed data falls within the [0, 1] interval. The transformation function used is as

$$x^* = \frac{x - \min}{\max - \min}. \quad (2)$$

Assume the dataset contains m items and each item has n-dimensional features, where x is the jth eigenvalue of the ith item prior to normalization and min is the jth dimension of the m items prior to normalization. min is the feature's minimum value, max is the feature's maximum value in the j-th dimension before normalization, and x^* is the feature's j-th dimension value in the ith piece of data after normalization.

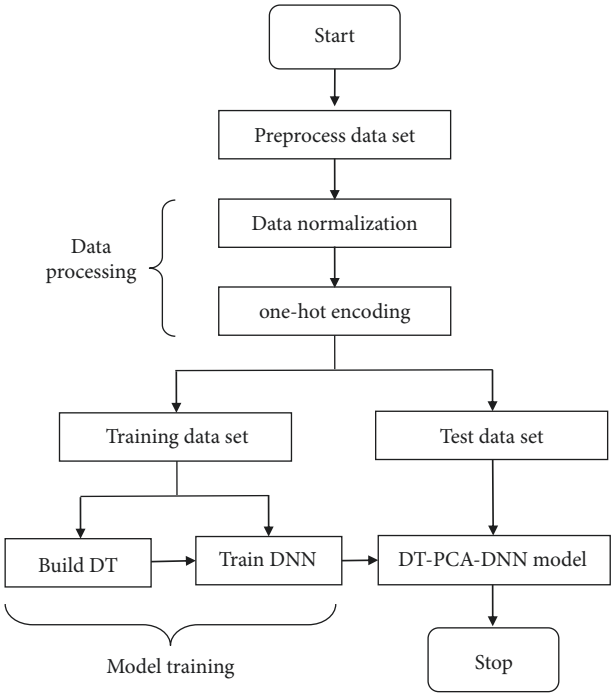

Figure 5: System flowchart.

3.1.2. One-Hot Encoding. For discrete data encoding, one-hot encoding, also known as one-bit efficient encoding, is utilised. N states are encoded using an N-bit status register; each state has its own register bit, and only one bit is valid at any time. Table 2 summarises the present sample set.

The sample feature dimension in Table 2 is 3, feature 1 has two values [0, 1], feature 2 has four values [0, 1, 2, 3], and feature 3 has three values [0, 1, 2].

Feature 1 has two values, so the encoding rule should be as follows:

(i) $0 \longrightarrow 10$

(ii) $1 \longrightarrow 01$

The corresponding feature 2 has four values, so the encoding rule should be as follows:

(i) $0 \longrightarrow 1000$

(ii) $1 \longrightarrow 0100$

(iii) $2 \longrightarrow 0010$

(iv) $3 \longrightarrow 0001$

The coding rule of feature 3 is the same as given above and will not be repeated.

The results of samples X, Y, and Z after one-hot encoding are shown in Table 3.

Table 2: Feature distribution of the sample set.

Sample	Feature 1	Feature 2	Feature 3
X	0	1	1
Y	1	0	2
Z	1	3	0

Table 3: One-hot encoded result of sample set.

Sample	Feature 1	Feature 2	Feature 3
X	10	0100	010
Y	01	1000	001
Z	01	0001	100

3.2. Model Training

3.2.1. Establishing DT. First, select the type of decision tree to be used because the information gain used by the ID3 algorithm has a preference for attributes with a large number of possible values, and the model used uses DT before the experimental data are dimensionally reduced. The data dimension is high, so this paper selects the ID3 algorithm. Followed by the depth of DT since the function of DT is not to identify as much intrusion data as possible but to misjudge average data as intrusion data as little as possible, the depth of DT should not be too deep. If the depth of DT is too deep, the accuracy of the first classification will be improved, but it has been judged as an entry. Invasive but average data will affect the final accuracy.

3.2.2. Training DNN. Because DNN requires a relatively large number of hidden layers to analyze high-dimensional data, the underfitting phenomena will be severe if the number of hidden layers is too low. As the number of hidden layers increases, the time spent training DNN grows exponentially, which is incompatible with the real-time needs of this work. When PCA pair data are introduced after the dimensionality reduction process, the connection between data feature dimensions and data redundancy is reduced, DNN training is faster, and DNN accuracy is ensured.

DNN uses the BP algorithm for training, ReLU as the activation function to simplify the calculation process of the neural network, and the "Adam" optimization algorithm, which occupies fewer resources and has a faster model convergence to shorten the training time.

3.2.3. DT-PCA-DNN Model Optimization. As shown in Figure 6, the preprocessed test dataset is first classified with the trained DT. Next, the data whose classification result is intrusion are judged as intrusion and stored in the temporary training sample. The information whose classification result is typical is removed from this DT classification. Given the label, prepare for the second judgment type of data. The DT layer is equivalent to a filter screen, which filters out the intrusion data easy to filter. Since the trained DT is a series of if-else statements, and the processing speed of large batches of data is extremely high, which significantly reduces the workload of DNN and improves the running speed of the algorithm.

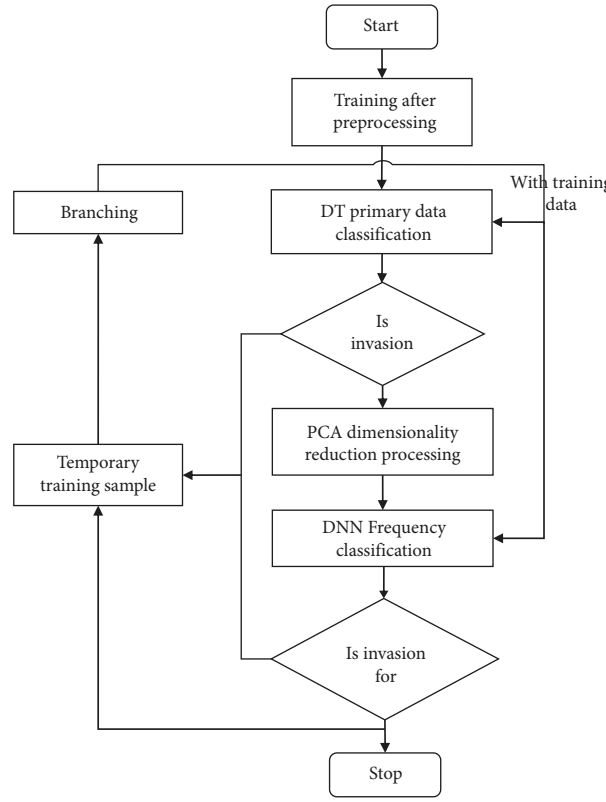

Figure 6: DT-PCA-DNN model optimization.

The second step is to perform PCA dimensionality reduction on the data judged by DT to be expected, but the labels have been removed. The trained DNN classifies the low-dimensional data output after PCA processing for the second time. If the classification result is an intrusion, the intrusion label is added, and the temporary training sample is stored. The standard classification result adds standard labels and stores temporary training samples. Since DT and DNN belong to supervised learning, the tags assigned to the data need to be used when using the brief training sample set for retraining. Since the intrusion detection process is carried out one by one, the method can quantify the comparison results between the original data type of the test dataset and the label added to the corresponding data during the detection process. After the quantified value is accumulated to the threshold set, the collected data can be used. Then, do a retraining fine-tuning of DT and DNN. After several fine-tuning, the designed model reaches the optimum.

4. Experimental Simulation

4.1. Dataset. The NSL-KDD [24] dataset was employed in this experiment, which augments the KDD 99 dataset. In comparison to the KDD 99 dataset, this dataset is devoid of duplicate records. The number of selected records is inversely related to the percentage of records in the original dataset, allowing for more efficient evaluation of the

developed model [25]. The training set has 125,937 items while the test set contains 22,544 items. Table 4 details the various data kinds.

The NSL-KDD dataset contains 41 features, which are classified into four major feature categories: basic TCP connection features, operating features on hosts, time-based network traffic statistical features, and host-based network traffic statistical features. The first three of the 41 features are described in the following. The types are distinct features and include protocol type (there are three types of protocols: TCP, UDP, and ICMP), service (there are 70 values in the training set and 64 in the test set), take two (big value), and flag (connection normal or error status, a total of 11 values). After normalizing the data, one-hot encoding is conducted. The data dimension increases to 43 after encoding the protocol type, then to 112 after encoding the service, and finally to 122 after encoding the flag. After one-hot encoding, the final NSL-KDD dataset has a data dimension of 122.

4.2. Metrics. First, the confusion matrix is listed as shown in Table 5. TP represents the number of data pieces whose real data type is normal. The model prediction result is still normal; TN represents the number of data pieces whose real data type is an intrusion. The model prediction result is also intrusion. FP indicates the number of data pieces whose real data type is an intrusion. Still, the model prediction result is normal. FN represents the number of data pieces whose real data type is normal, but the model prediction result is an intrusion. Of course, the size of different data bars is not enough as a standard for evaluating the experimental results. Therefore, a relatively reasonable evaluation standard is established based on the above parameters: the accuracy rate (AC), detection rate (DR), precision rate (PR), and false alarm rate (FAR), and the definitions are as follows:

$$\begin{aligned} \mathrm{AC} &= \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}, \\ \mathrm{DR} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \\ \mathrm{FAR} &= \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}, \\ \mathrm{PR} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \end{aligned} \tag{3}$$

4.3. Parameter Setting. After preprocessing the data, including one-hot encoding and normalization, all data values are located in the interval [0, 1]. After discretizing each dimension of the data with 0.5 as the standard, we use DT to perform the first test on all training data. For the secondary screening, the main parameters of the DT used are shown in Tables 6–8 and then the PCA dimensionality reduction is carried out. Because the designed system is linear, all parameters can be obtained one by one by fixing other parameters to obtain the optimal parameters.

Table 4: Data distribution of NSL-KDD dataset.

Type		The amount of training set data	Test set data volume
Normal		66945	9641
Attack	DoS	44623	6758
	U2R	48	187
	R2L	886	2247
	Probe	10649	2148
Total		123151	20981

Table 5: Data confusion matrix.

Confusion matrix		Actual value	
		Normal data	Intrusion data
Predictive value	Normal data	TN	FN
	Intrusion data	FP	TP

Table 6: DT main parameters.

Main parameters	Value
Criterion	Entropy
Splitter	Best
max_depth	5
random_state	392

Table 7: PCA main parameters.

Main parameters	Value
*n*_components	11
Whiten	True
svd_solver	Auto

Table 8: DNN main parameters.

Main parameters	Value
hidden_layer_sizes	[140, 70]
Activation	ReLU
Solver	Adam

For Criterion (attribute segmentation criterion), the value is a string type. There are two criteria to choose from: "gini" and "entropy." For Splitter (segmentation point), the value is a string type; there are two standards to choose from, "best" and "random," where "best" means that in all features finding the optimal segmentation point in the "random" means to find the optimal segmentation point in the randomly selected part of the features—max_depth (the constructed decision tree), which can be an integer or none. max_features is the number of features to consider when finding the best segmentation. random_state (multiple states used to generate random numbers) can be an integer, an instance of random state, or none. Experiments have shown that the best effect is achieved when the value is 392.

*n*_components (feature dimension after dimension reduction) can be the number of dimensions reduced or the percentage of data retained; whiten (whether whitening) reduces the correlation between features and all features

have the same variance; svd_solver (singular value decomposer) is a string when its value is "auto," and certain conditions are met, the complete singular value decomposition function is called hidden_layer_sizes (hidden layer sizes), tuple type. The number of hidden layers and the number of neurons in the hidden layer are determined by adjusting this value. Two hidden layers are introduced here, with 140 neurons in the first layer and 70 neurons in the second layer; activation is the activation function; solver (weight optimization function) is selected by selecting different strings of the corresponding weight optimization function.

4.4. Experimental Results. The experiment uses a Windows10 system and 64 bit operating system, the processor version is Intel® CoreTMi7-9750H CPU@2.60 GHz, the total physical memory is 16.0 GB, the development language is Python3.5, and the software package used is sklearn.

4.4.1. Experiment 1. This experiment mainly studies the detection time of binary classification, the real-time problem of detection under binary classification. This experiment primarily compared the two-class prediction accuracy and training time of FC [13], DT, PCA-DNN, EDF [20], CNN [26], and DT-PCA-DNN models. To reflect the characteristics of DT-PCA-DNN, the test data used are all the data in the NSL-KDD test dataset. For the convenience of observation, Figure 7 is obtained from Table 9. In the figure, because the FC training time is too long, the impact on the selected vertical axis interval is too significant, so it is not listed.

Observing Figure 7, we can see that the accuracy AC of PCA-DNN and FC is the same, but the training time of FC is much higher than that of PCA-DNN, the prediction time is slightly longer, and the real-time detection is poor. On the other hand, the training time of the EDF algorithm is marginally more extended than that of PCA-DNN, and the accuracy AC is the same as that of PCA-DNN.

The training time of the CNN algorithm is as long as 90 s, and the accuracy rate is slightly lower than that of EDF and PCA-DNN, which is inferior to both. On the other hand, although the training speed of DT is breakneck, the accuracy rate is four percentage points lower than that of EDF and PCA-DNN.

Compared with PCA-DNN without DT, DT-PCA-DNN takes 1.32 s longer to train 125973 pieces of data and takes about 10 ms more to predict 22544 pieces of data, but the accuracy AC has improved by nearly ten percentage points remarkably. The introduction of DT has minimal impact on training time and prediction time but significantly improves prediction accuracy.

4.4.2. Experiment 2. This experiment mainly studies the five-category detection time of the DT-PCA-DNN model, the real-time detection problem under five categories. Mark normal samples as 0, DoS samples as 1, probe samples as 2, U2R samples as 3, and R2L samples as 4. The analysis in

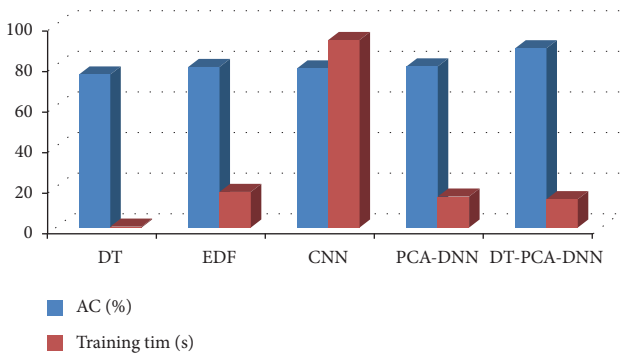

Figure 7: Results of experiment 1.

Table 9: Results of experiment 1.

Algorithm	AC%	DR%	Training time per second	Prediction time/ms
FC	79.49	N/A	1856.31	221.05
DT	75.68	65.74	0.67	58.84
EDF	79.32	64.33	17.79	N/A
CNN	78.63	68.77	92.64	N/A
PCA-DNN	79.61	66.69	15.55	47.84
DT-PCA-DNN	88.64	84.56	14.15	57.86

Table 10: Total results of experiment 2.

Algorithm	Five classification training time/s	Total accuracy/%
DT	0.67	78.82
EDF	41.31	87.28
CNN	93.39	87.01
PCA-DNN	14.48	77.06
DT-PCA-DNN	17.36	83.29

Table 11: Five classification results of experiment 2.

Algorithm	Evaluation criteria	Normal	DoS	Probe	R2L	U2R
DT	PR	67.01	91.84	26	71.49	52
	DR	95.31	84.18	0.41	61.99	4
	FAR	34.38	3.15	0.16	2.62	0.04
EDF	AC	91.62	98.9	88.15	59.59	15.39
CNN	AC	92.56	98.47	92.58	35.5	26.56
PCA-DNN	PR	64.78	95.62	96.67	82.45	0
	DR	98.19	75.48	16.08	67.5	0
	FAR	39.24	1.51	0.08	1.70	0
DT-PCA-DNN	PR	71	92.21	91.09	78.92	0
	DR	98.81	83.26	36.95	61.66	0
	FAR	26.11	4.1	0.39	1.5	0

Table 10 shows that the speed advantage of DT-PCA-DNN in the five classifications is undeniable. Still, the overall accuracy is slightly inferior to EDF and CNN, and we compare DT and PCA-DNN. However, the training time is relatively short. The overall accuracy is low, and the performance is poor. Comparing PCA-DNN and DT-PCA-

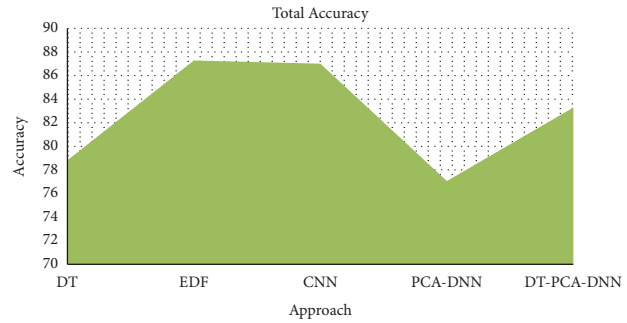

Figure 8: Comparison of accuracy of the model.

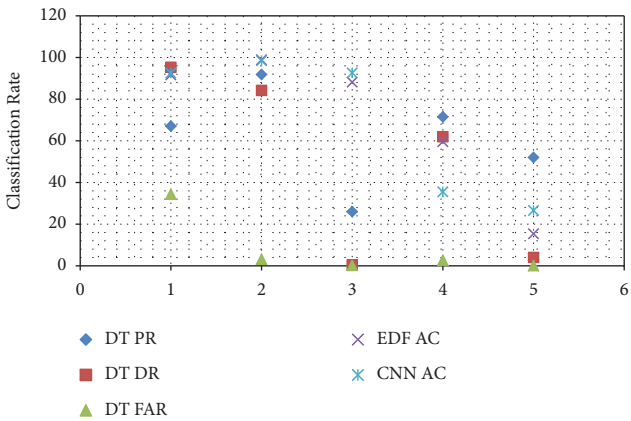

Figure 9: Classification performance over different attack scenarios for benchmark algorithm.

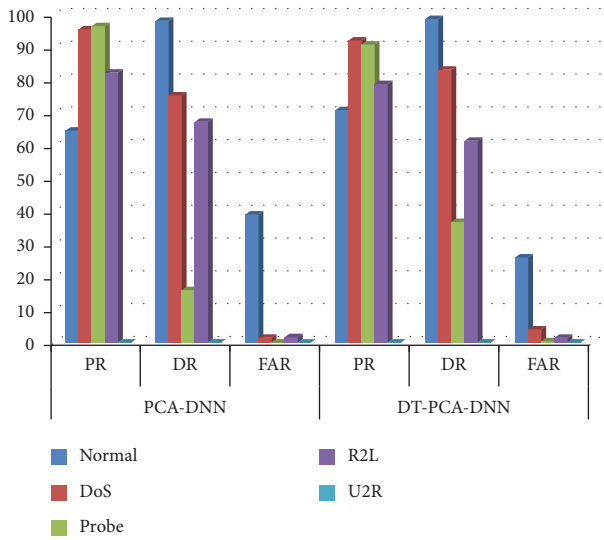

Figure 10: Classification performance over different attack scenarios for the proposed algorithm.

DNN in the five-category experiment, the training time is 3 s longer. The accuracy rate is improved by six percentage points, proving that the introduction of DT does not cause much time while ensuring the accuracy rate loss.

The analysis of Table 11 and Figures 8–10 shows that DT-PCA-DNN may have processed part of the data during DT prescreening, resulting in no U2R results (small U2R sample size).

The advantages of DT-PCA-DNN are mainly reflected in the ability to recognize R2L, but because of its small proportion in the dataset.

As a result, the overall accuracy rate is lower than EDF and CNN. At the same time, the model has a relatively high false alarm rate for normal data when the detection rate is relatively high. This is a problem that needs to be pointed out.

5. Conclusion

The DT-PCA-DNN intrusion detection model described in this study greatly improves the training and detection speed while preserving accuracy. The model employs DT to do a preliminary screening of the preprocessed data to be detected before employing PCA as an input to perform a secondary judgment via DNN. The addition of DT causes a small increase in training time but a large boost in accuracy. Simultaneously, DT prescreening reduces future DNN burden, which has a considerable effect on overall training pace. The following study focus is mostly on overcoming the issue of the DT-PCA-DNN model having a high false alarm rate for normal data in the five-classification experiment while also increasing the DT-PCA-DNN model's five-classification capability. [27].

Data Availability

The data used to support the findings of this study are available from the author Kusum Yadav upon request (kusumasyadav0@gmail.com).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Shoayee Dlaim Alotaibi proposed the method, Kusum Yadav wrote the paper, Arwa Naser Mohammed Aledaily developed the methodology, Luluah Dhaifullah Al-Quwai proofread the paper, Alaa Kamal Yousef Dafhalla simulated the work, and Shahad Almansour and Velmurugan Lingamuthu analyzed the results.

References

[1] I. Sembiring, "Implementation of honeypot to detect and prevent distributed denial of service attack," in *Proceedings of the 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 345–350, Semarang, Indonesia, October 2016.

[2] A. Phadke, M. Kulkarni, P. Bhawalkar, and R. Bhattad, "A review of machine learning methodologies for network intrusion detection," in *Proceedings of the2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 272–275, Erode, India, March 2019.

[3] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A survey on machine learning techniques for cyber

security in the last decade," *IEEE Access*, vol. 8, pp. 222310–222354, 2020.

[4] G. Vigna, E. Jonsson, and C. Kruegel, Eds., *RAID 2003, LNCS 2820*, pp. 173–191, c Springer-Verlag, Berlin Heidelberg, 2003.

[5] M. Crosbie and G. Spafford, "Defending a Computer System Using Autonomous Agents," Technical Report No. 95-022, Department of Computer Sciences, Purdue University, West Lafayette, Indiana, March 1996.

[6] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Systems with Applications*, vol. 39, no. 1, pp. 424–430, 2012.

[7] N. P. Moorthi and V. Mathivanan, "Hybrid optimization for feature selection in opinion mining," *International Journal of Engineering & Technology*, vol. 7, no. 1.3, p. 112, 2017.

[8] G. Liu, Z. Yi, and S. Yang, "A hierarchical intrusion detection model based on the PCA neural networks," *Neurocomputing*, vol. 70, no. 7-9, pp. 1561–1568, 2007.

[9] J. Cao, C. Wu, L. Chen, H. Cui, and G. Feng, *Hindawi Computational Intelligence and Neuroscience*, vol. 2019, Article ID 2060796, 12 pages, 2019.

[10] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, pp. 239–247, 2021.

[11] Y. Gu, B. Zhou, and J. Zhao, "PCA-ICA ensembled intrusion detection system by pareto-optimal optimization," *Information Technology Journal*, vol. 7, no. 3, pp. 510–515, 2008.

[12] M. Soni and D. K. Singh, "privacy preserving authentication and key management protocol for health information system," *Data Protection and Privacy in Healthcare: Research and Innovations*, CRC Publication, vol. 37, , 2021.

[13] M. Soni and D. K. Singh, "LAKA: Lightweight Authentication and Key Agreement Protocol for Internet of Things Based Wireless Body Area Network," *Wireless Personal Communication*, 2021.

[14] M. Soni and D. K. Singh, "Blockchain Implementation for Privacy Preserving and Securing the Healthcare Data," in *Proceedings of the 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 729–734, Bhopal, India, June 2021.

[15] M. Soni, G. Dhiman, B. S. Rajput, R. Patel, and N. K. Tejra, "Energy-Effective and Secure Data Transfer Scheme for Mobile Nodes in Smart City Applications," *Wireless Pers Commun*, 2021.

[16] R. Nian, G. Ji, and M. Verleysen, "An unsupervised Gaussian mixture classification mechanism based on statistical learning analysis," in *Proceedings of the 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 14–18, Shandong, China, October 2008.

[17] R. Aggarwal and Y. Song, "Artificial neural networks in power systems. Part 1: general introduction to neural computing," *Power Engineering Journal*, vol. 11, no. 3, pp. 129–134, June 1997.

[18] Z.-H. Zhou, *Machine Learning*, Springer Nature Singapore Pte Ltd, Singapore, 2021.

[19] V. R. Sargsyan, "Formation of human higher nervous activity and new biological theories, HSOA journal of brain & neuroscience research, sargsyan VR," *J Brain Neursci*, vol. 2, p. 004, 2018.

[20] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, in *Proceedings of the 2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, Antalya, Turkey, August 2017.

[21] V. K. Gupta, S. K. Shukla, and R. S. Rawat, "Crime tracking system and people's safety in India using machine learning approaches," *International Journal of Modern Research*, vol. 2, no. 1, pp. 1–7, 2022.

[22] P. K. Vaishnav, S. Sharma, and P. Sharma, "Analytical review analysis for screening COVID-19 disease," *International Journal of Modern Research*, vol. 1, no. 1, pp. 22–29, 2021.

[23] I. Chatterjee, "Artificial intelligence and patentability: review and discussions," *International Journal of Modern Research*, vol. 1, no. 1, pp. 15–21, 2021.

[24] UNB, "NSL-KDD Data Set," https://www.unb.ca/research/iscx/dataset/iscx-NSL-KDDdataset.html.

[25] T. Brugger, "KDD Cup'99 Dataset (Network Intrusion) Considered Harmful," KDNuggets, Sept. 2007, https://www.kdnuggets.com/news/2007/n18/4i.html.

[26] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82512–82521, 2019.

[27] Q. Qin, K. Poularakis, K. K. Leung, and L. Tassiulas, "Line-speed and scalable intrusion detection at the network edge via federated learning," in *Proceedings of the 2020 IFIP Networking Conference (Networking)*, pp. 352–360, Paris, France, June 2020.