Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **10**, Number **4**, November **2025**

Feature Selection Strategies and Their Application in the Prediction of Intrusion Detection

Rong Hu

School of Computer Science and Mathematics
Fujian University of Technology, Fuzhou 350118, China
Fujian Provincial Key Laboratory of Big Data Mining and Applications
Wuyi University, Wuyi 354300, China
The Key Laboratory of Cognitive Computing and Intelligent Information Processing of Fujian Education Institutions
896410521@qq.com

De-Qu Chen

School of Computer Science and Mathematics
Fujian University of Technology, 350118, China
Fujian Provincial Key Laboratory of Big Data Mining and Applications
dequchen@foxmail.com

Ming Zhang

School of Computer Science and Mathematics
Fujian University of Technology, 350118, China
Fujian Provincial Key Laboratory of Big Data Mining and Applications
772689671@qq.com

Long-Zheng Cai*

School of Computer Science and Mathematics
Fujian University of Technology, 350118, China
Fujian Provincial Key Laboratory of Big Data Mining and Applications
cailongzheng@163.com

*Corresponding author: Long-Zheng Cai

Received January 15, 2024, revised October 11, 2024, accepted July 21, 2025.

Abstract. *Feature selection is a pre-processing technique that can significantly influence the accuracy of model's classification and prediction. It is widely used in biological data processing, financial and medical applications and also in the network intrusion detection models. However, there is still lack of knowledge about how different feature selection strategies would affect the performance of models. In this paper, we explore the impact of different feature selection methods on model performance under two deep learning frameworks. We used four traditional feature selection methods (Analysis of Variance (ANOVA), Variance Permutation (VP), Correlation Analysis (CA), and Principal Component Analysis (PCA)), proposed three feature selection strategies (including the Mini-Sum, Mini-Index Grouping, and Intersection Feature Extractions) and compared them with the traditional and random selection methods. To evaluate the performance of these strategies on different models, we used the NSL-KDD dataset which is commonly used in network intrusion detection as the test set. The experimental results showed that although feature selection could generally improve model's classification performance, it is difficult to obtain an optimal order of features to be selected and the order also depends on the models used. We also conclude that the comprehensive feature selection strategies proposed in this paper could generally perform better than the existing ones and that the use of feature selection methods can at least cut down space and time complexities although sometimes they may not be possible to significantly improve model's performance.*
Keywords: Deep learning, Feature selection, Intrusion detection, Mini-Sum Strategy, Mini-Index Grouping

1. **Introduction.** The use of machine learning (ML) and deep learning (DL) technologies is often accompanied with multi-feature issues. However, it is difficult to find the optimal feature set from the original features. In any dataset, due to feature redundancy and dependence, inappropriate choice of features will lead to poor classification accuracy [1]. So, it is necessary to select the appropriate feature set from the initial wide range of features to improve the classification efficiency [2]. Especially in the field of ML, the correct selection of features can improve the training speed, generalization ability, or simplicity of induced models [3]. In addition, a small number of related features can reduce the cost of measurement and lead to a better understanding of the domain [4]. When the dimension of the dataset increases significantly, the sparsity of representative data in the relevant dataset also increases, making it more difficult to find statistically significant features [5]. Therefore, it is of great significance to investigate the effectiveness of new fast feature selection strategies on improving the performance of ML and DL algorithms.

In the realm of feature selection there are two primary categories: Wrapper and filter approaches [6]. This paper focuses on investigating filter-type feature selection methods. In traditional filter-based methods, the selection process typically involves evaluating each feature individually, identifying the most effective ones, and incorporating them into a feature subset that satisfies the desired constraints [7]. However, this process presents two challenges. Firstly, the evaluation methods often concentrate solely on the relationship between each feature and the class label, disregarding other pertinent information. Secondly, the suitability of evaluation algorithms across various data types remains uncertain, as not all selection algorithms are universally applicable.

To address these challenges, this paper proposes three novel feature selection strategies. Three strategies distinguish themselves from prior research in several aspects. They offer a universal idea that can be applied to a range of fundamental feature selection methods. The Mini-Sum Strategy (MSS) combines feature index and feature importance, and selects the top features by simple sum and sorting operations. The Mini-Index Grouping (MIG)

strategy purposefully groups the feature importance indices obtained by traditional feature selection methods. The Intersection Feature Extraction (IFE) strategy integrates the idea of intersection, performing intersection operations on the features obtained by various traditional feature selection methods according to the divide and conquer rule, and then prioritizing the extraction of features that are considered important by the intersection.

In this paper, the proposed strategy is compared with traditional feature selection methods, and delves into the effectiveness of different feature selection strategies and the impact of them on the performance of DL models. The AE and Multi-Layer Perceptron (MLP) are used for classification and comparison, the NSL-KDD dataset [8] is used as the benchmark dataset for comparative analysis of feature selection strategies with the intention of answering the follow 5 questions through extensive experiments.

1) whether the features selected by different algorithms perform the same in different models;

2) whether the difference in the importance order of features generated by different algorithms affects classification performance;

3) whether the features selected by well-designed algorithms are better than randomly selected ones;

4) whether the integrated application of different feature selection algorithms will perform better than any single ones; and

5) whether there is an optimal number of features to be selected.

The rest of this paper is organized as follows. Section 2 presents a review of related work in the field. Section 3 discusses the traditional feature selection methods and the DL models we used in this paper and introduces the new comprehensive feature selection strategies we proposed. Section 4 explores the performance of the strategies of feature selection in more detail by comparing the experimental results. In Section 5, we will discuss our findings in depth. Finally, conclusions are made in Section 6.

2. **Related works.** As early as 1997, Benediktsson and Sveinsson [9] verified the superiority of feature selection in data pre-processing by comparing the performance among the decision boundary feature selection method, traditional feature selection methods and the original data. In the field of hyperspectral data processing, Kumar et al. [10] proposed an optimal basic feature selection algorithm that reduces the dimension of hyperspectral data. The results showed that when using the data after feature selection, the accuracy of the classification results is increased by 5% to 10% compare with the original data. To explore changes in multidimensional unlabeled data, Kuncheva and Faithfull [11] used traditional PCA for feature selection on 35 datasets and compared the selected features with the original data. The results proved that feature selection was meaningful. Compared with the original data, the performance was improved by about 3%. Zhou et al. [12] proposed a method named as the Common and Individual Feature Extraction (CIFE) exploiting the linked nature of data and applied this method to the Extended Yale Database and PIE Database. In these two datasets, the average clustering accuracy of CIFE was 63.3% and 89.8%, respectively. Compared with PCA, GNMF, MMcut and so on, the accuracy of CIEF on Yale Database has been improved by 10% 20%. Similar improvement also occurred in the recognition accuracy of PIE database. Cheng et al. [13] proposed a feature selection method that combines subspace with sparse representation, and the experimental results improved by 3% to 30% compared to the original method in six datasets. When facing High-Impedance Faults (HIFs) of distribution network, Liu et al. [14] proposed a HIFs detection method using synchronous current information, in which the data was preprocessed by PCA, to achieve reliable detection of HIFs in noisy environments. To reduce the number of features, Jain and Jana [15] proposed an eXplainable Reasonably

Randomised Forest (XRRF) model to deal with the large-scale feature selection problem of network intrusion detection. The results showed that compared with traditional classification and prediction models for regression data sets the Predictive Mean Square Error (PMSE) of XRRF has reached the minimum (that is, the effect is optimal). Kumari and Singh [16] proposes a model based on deep learning and binary firefly optimization, which combines text and images to consider attack levels. The $F1 - Score$ of the proposed method is 11% higher than that of the original method.

On the other hand, the network activity observed in the past few years showed that there is a trend of a surge in cyber-attacks. The most common type of attack is the Denial-of-Service attack (DoS), which uses multiple connections to gain momentum and causes excessive economic and reputational losses to the victim [17]. Another multi-connection attack worth paying attention to is a probing attack, in which an attacker tries to obtain important information about the target computer. Probing attack itself may not cause any damage, but it is usually a precursor to other dangerous attacks [18], such as DoS, flood, user-to-root (U2R), remote-to-local (R2L) etc. To solve this problem, DL technologies were used to build the intrusion detection system (IDS). However, the massive amount of data and information transmitted in the computer network makes it too late for the intrusion detection system to complete the processing of a large amount of information, resulting in the detection system's untimely response or even failure [19]. Bellman called this phenomenon of massive amounts of data affecting DL technology a "dimensional disaster" [20]. To ease this kind of disaster, researchers put their sights on the pre-processing of data by filtering out some redundant features.

In response to this issue and to solve the problem of complex types of features in the Internet of Things (IoT), Pervez and Farid [21] proposed an optimizing accuracy by iteration method and used SVM to classify NSL-KDD dataset. Its accuracy was increased by 0.31% compare with that without feature selection. Yin et al. [22] introduced Recurrent Neural Networks (RNN) for the construction of intrusion detection system, the binary classification accuracy of NSL-KDD dataset reached 83.28%. Al-Qatf et al. [23] proposed a new intrusion detection system based on Self-Taught Learning (STL), which made full use of the advantage of Autoencoder (AE) and SVM. Its binary classification accuracy improved about 0.84% 9.40% in NSL-KDD dataset. Ieracitano et al. [24] proposed a way to remove features in the feature vectors having the number of zeros higher than 80% in the data to select useful features. The accuracy of binary classification in the NSL-KDD dataset reached 84.21%, which was 1.58% higher than previous studies. Almasoudy et al. [25] selected features based the Differential Evolution (DE) algorithm and employed Extreme Learning Machine (ELM) to work out the accuracy in NSL-KDD dataset, which reduced the number of features selected to 5 in binary classification problem and achieved a classification accuracy of 87.53%. Xu et al. [26] analyzed the NSL-KDD original data through PCA, intending to explore the relationship between normal and abnormal samples in the training dataset, and conducted an in-depth discussion on intrusion detection issues. The final binary classification accuracy was 90.61%. Zhao et al. [27] combined Weighted Stacking (WS) with Correlation-based Feature Selection Differential Evolution (CEF-DE) to improve classification performance in NSL-KDD dataset. Its $F1 - Score$ of the binary classification was 0.22% 7.05% higher than other methods.

3. **Methodology.**

In this section, we will introduce the traditional feature selection methods used in this paper, describe in detail the feature selection strategies we proposed, and finally introduce the DL models used.

3.1. **Traditional Feature selection Methods.** This section briefly introduces four traditional feature selection methods from mathematical perspectives, which include the Analysis of Variance (ANOVA) [28], variance permutation analysis (VP) [29], correlation analysis (CA) [30], and principal component analysis (PCA) [31, 32]. In the past, they were often used to select the useful features from the original feature set in the statistical analysis of various types of mathematical problems, but now they are also used in the feature analysis of DL technology [33].

3.1.1. *Analysis of Variance (ANOVA).* Analysis of Variance (ANOVA), also known as 'Variance Analysis' or 'F test', was invented by R.A. Fisher to test the significance of differences between two or more samples [28]. Its basic idea is to decompose the total variance of all data into several parts with each part representing the effect of a certain influencing factor or the interaction between influencing factors, compare the variance of each part with the variance of random errors, and make statistical inferences based on the F distribution to determine whether the effects of each factor or interaction are significant [34]. The relevant definition of ANOVA is to work out the following 'F test' statistic [35]:

$$F = \frac{MSB}{MSE} \tag{1}$$

where MSB and MSE stand for between-group mean square and within-group mean square errors, respectively. They are defined as follows:

$$MSB = n \times o_{it}^2 \tag{2}$$

$$MSE = \frac{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_l^2}{l} \tag{3}$$

where n is the number of samples, σ_μ^2 is the mean variance calculated using the data of all classes, $\sigma_1^2, \sigma_2^2, \cdots \sigma_l^2$ is the variance of each class, and l is the number of classes.

We conducted the ANOVA for each feature to obtain the F_i statistic and then arranged them in descending order. As the higher the F_i statistic, the better the feature could distinguish different classes, the features will be selected for subsequent training and detection in the descending order the F_i statistic and the remaining low contribution features will be removed to reduce the data dimensionality.

3.1.2. *Variance Permutation (VP).* In addition to the above method, we also calculated variance of each feature and sort them in descending order. As variance is a measure of how spread a random variable or a set of data is, and the greater the variance, the greater the spread [29], the features will be selected from the one with the largest variance to the ones with smaller variance for subsequent training and detection and the remaining low contribution features will be removed to reduce the data dimensionality.

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n} \tag{4}$$

where $\bar{x}$ is the average value of a feature.

3.1.3. *Correlation Analysis (CA).* Another method used in our selection is the correlation analysis, the role of which is to analyze the correlation between the features to eliminate the features that are strongly correlated [30]. It requires to calculate the correlation coefficient r_{ij} between each pair of features i, j:

$$r_{ij} = \frac{\sum_{k=1}^{n}(x_{ki} - \overline{x_i})(x_{kj} - \overline{x_j})}{\sqrt{\sum_{k=1}^{n}(x_{ki} + \overline{x_i})^2}\sqrt{\sum_{k=1}^{n}(x_{kj} + \overline{x_j})^2}}, i, j = 1, 2, \ldots, m \tag{5}$$

where x_{ki} and $\overline{x_i}$ are the element at kth row and ith column of data X and average of feature i, x_{kj} and $\overline{x_j}$ are the kjth element of data X and average of feature j, and n and m are the total number of samples and the number of features.

The pairwise correlation between any pair of features is calculated by Equation (5) to obtain an $m \times m$ correlation coefficients matrix with each element being r_{ij}. The larger the absolute value of r_{ij}, the stronger the correlation between features i and j (either positive correlation or negative correlation); if the correlation between two features is zero, it means that there is no correlation between them. Our objective is to select those features with as small r_{ij} as possible. As there are m correlation coefficients for each feature, we then take the mean of the absolute value of each row for each feature, i.e., work out the mean by the following equation:

$$r_{ij} = \frac{\sum_{j=1}^{m} |r_{ij}|}{m}, i = 1, 2, \cdots, m \tag{6}$$

Finally, the $\overline{r_i}$ is arranged in an ascending order and the features corresponding to smaller $\overline{r_i}$'s will be selected for the subsequent training and detection and the remaining high correlation features will be removed to reduce the data dimensionality. In this way, higher independent features will be selected to construct the model to improve the model's performance.

3.1.4. *Principal Component Analysis (PCA).* The three feature selection methods described above are the direct selecting ones. However, the principal component analysis (PCA) is a statistical method [32], in which a set of potentially correlated variables is transformed into a set of linearly uncorrelated variables through orthogonal transformation, and the transformed set of variables is called principal components [36]. Therefore, the original features have to be combined together to form a new set of principal components in this method.

In the transforming process, the original features $f_1, f_2, \ldots, f_m$ will be orthogonally transformed to a new set of principal components $z_1, z_2, \ldots, z_m$ with their variance satisfying $V(z_1) \geq V(z_2) \geq \cdots \geq V(z_{(z_m)})$. Therefore, z_1 is the one that carries the most information of the original data, and the m-dimensional main hyperplane $z_1, z_2, \ldots, z_m$ constitutes the subspace that retains the most information of the original data [37]. A subset of the principal components from $z_1, z_2, \ldots, z_m$ will be selected for subsequent training and detection in the order z_i and the remaining low variance components will be removed to reduce the data dimensionality.

3.2. **The Comprehensive Feature Selection Strategies.** The 'importance' of each feature calculated by each of the traditional mathematical methods listed in the previous subsection may not be the same. Take the NSL-KDD dataset (see subsection 4.1 for the detailed description of this dataset) as an example, there are 38 numerical features in this dataset, but we have deleted 3 'null features' (z_5, z_{17} and z_{18}) in which all the data are zero in the normal data. So, there are only 35 features remained (all the features are listed in Table 5). The order of feature importance sorted by each of the methods is listed in Table 1, in which the first row is the order of feature importance with the smaller number representing the feature having higher importance and the second to the fourth rows represent the feature index. We can see from this table that the same feature sorted by different method would show different importance (I-index means Important index). Although each of the feature selection methods has its own mathematical basis, as the order of feature importance obtained by each method is not the same, the classification results using the selected features may not be the same as well. Therefore, we will explore

some comprehensive strategies to combine the results obtained by these selection methods in order to improve the prediction accuracy.

Table 1. The order of part of the feature importance sorted by each of the methods

I-index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**...**	**33**	**34**	**35**
ANOVA	23	26	22	35	36	9	30	11	20	32	31	...	12	2	4
CA	2	15	4	6	14	8	3	7	19	10	11	...	36	31	26
VP	9	23	22	36	35	26	31	30	29	25	24	...	6	13	10

3.2.1. *The Mini-Sum Strategy (MSS).* The first strategy proposed is the Mini-Sum Strategy (MSS). The MSS will first add the importance index for each feature obtained by each of the three traditional mathematical methods and then sort the sum to select the features from the one having the smallest sum to the largest sum. In this way, the best features with smaller comprehensive sorting scores obtained by this strategy will be used as the input in the subsequent experiments. As for the NSL-KDD dataset, the original order of each feature sorted by Table 1 is listed in Table 2, in which the first row represents the index of each feature (the real feature names are presented in Table 5) and the second to fourth rows are the original order of features obtained by each method. Table 3 shows the specific operation of MSS according to Table 2, in which the importance indexes for each feature shown in Table 2 are added along each column and then the summation is sorted in ascending order to generate the ranking of features by the MSS.

Table 2. Part of the importance index sorted by the feature index using different method

F-index	**1**	**2**	**3**	**4**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**...**	**36**	**37**	**38**
ANOVA	27	34	32	35	29	24	23	6	30	8	33	...	5	20	16
CA	15	1	7	3	4	8	6	28	10	11	14	...	33	23	27
VP	22	32	31	27	33	24	26	1	35	23	25	...	4	13	12

Table 3. Illustration of the process of MSS

After Summing															
F-index	**1**	**2**	**3**	**4**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**...**	**36**	**37**	**38**
Order	64	67	70	65	66	56	55	35	75	42	72	...	42	56	55
After Sorting															
F-index	**23**	**9**	**22**	**35**	**11**	**36**	**26**	**19**	**30**	**33**	**24**	**...**	**12**	**10**	**13**
Order	34	35	36	41	42	42	43	44	44	46	48	...	72	75	78

3.2.2. *The Mini-Index Grouping (MIG) Strategy.* In addition to the MSS proposed in section 3.2.1, we also explore a method to group the features with the smallest index in each of the three traditional mathematical methods and call it as the Mini-Index Grouping (MIG) strategy. In this strategy, we first select the features with an importance index 1 in any of the three traditional methods, then select those with index 2, etc. until the required number of features have been selected for the subsequent training and detection. Take the order of the feature importance listed in Table 1 as an example, features 23, 2, and 9 will be first included in the input set, then it comes the features 26, and 15, etc. In this way, we are trying to include the best features sorted by each of the traditional methods into the input set. Of course, with the increase of the index, some of the features

may have already been included in the previous steps. For example, feature 23 has been included in step 1, hence, only features 26 and 15 will be included in the input set in the second step in the above example. Therefore, with the increase of the steps, the number of features included in the input set may be less than 3.

3.2.3. *The Intersection Feature Extraction (IFE) Strategy.* The next strategy we use is the Intersection Feature Extraction (IFE) strategy [15]. In this strategy, the features are first sorted by each of the three traditional methods. Then in order to select the most important features, we first select the same number of features with high important indexes obtained from each of the traditional methods, then work out the intersection of these three sets. The process is shown in Figure 1, in which we use the key feature set (KFs), important feature sets (IFs) and elementary feature sets (EFs) to represent the features intersected among three sets, between two sets, and without intersection. At last, only the features in any of the IFs or KFs are selected as the input set for the subsequent training and prediction. Take the order of the feature importance listed in Table 1 as an example, when 2 most important features are used from each traditional method to work out the intersection, only feature 23 can be selected. But when 10 most important features are used, the number of features in the KFs is 0, while the numbers of features in the IFs are 0 for ANOVA & CA, 7 for ANOVA & VP (features 9, 22, 23, 26, 30, 35 and 36), and 0 for CA & VP. In this case, features 7, 9, 22, 23, 26, 30, 35 and 36 are selected for the input. Therefore, when more original features are used in the selection set, more features are selected. But the total number of features selected is always fewer or at most equal to the number of features used.

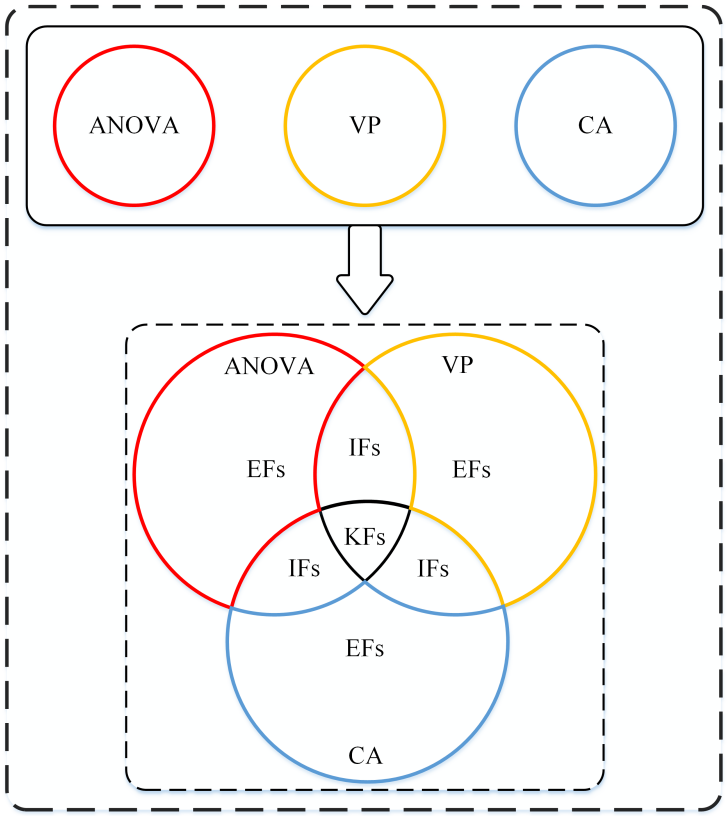


FIGURE 1. A Venn diagram to show the intersection of the features selected by the ANOVA, VP and CA methods that constitute the feature subsets of KFs, IFs and EFs

3.2.4. *Randomly Selection Strategy (RSS).* The above feature selection methods and strategies are all based on some kind of mathematical algorithms and each of them would to some extend consider itself as the 'best' method. However, as the order of features selected by each of them is not the same, at least some of them are not the best. In order to verify the significance of the feature selection methods and strategies, we just randomly select a certain number of features as the input for the training and prediction. Then, to reduce the randomness in the experiments, we run each RSS 5 times in each experiment and take the mean of the 5 classifications as the final experimental result for comparison.

3.3. **Classification.** As deep learning classifiers are widely used in intrusion detection related work [38], in the subsequent experiments, two DL models (AE and MLP) are used to detect normal and abnormal classes on the NSL-KDD dataset. The AE and MLP classifiers used in this paper are described below.

3.3.1. *Autoencoder (AE).* AE is a class of Artificial Neural Networks (ANNs) used in semi-supervised learning and unsupervised learning [39,40]. An AE consists of two parts: an encoder and a decoder [41] and its function is to represent the input information by taking the input information as the learning target.

The role of the encoder is to encode the high-dimensional input X into a low-dimensional hidden variable h, thereby forcing the neural network to learn the most informative features; while the role of the decoder is to restore the hidden variable h in the hidden layer to the original dimension, and the best state is that the output of the decoder can perfectly or approximately restore the original input $X \approx X^R$ [41, 42]. Figure 2 shows the AE model used in this paper with a five-layer structure. The relationships between layers i and i+1 can be represented as:

$$X_{i+1} = g_i(X_i) = \sigma_i(W_i X_i + b_i), i \in [0, 3] \tag{7}$$

where X_i is the input to layer i, W_i represents the weight matrix between layer i and layer i+1, bi is the bias vector, and σ_i represents the activation function. In this way, the original input X=X_0 and the reconstructed vector X^R=X_4.

The AE framework shown in Figure 2 takes the structure of AE [m:32:5:32:m], which is the same as that in [26]. Specifically, the AE encodes an m-dimensional feature set (X) into a 32-dimensional vector, which will then be converted to a 5-dimensional vector (h). Then the vector h will be converted back to a 32-dimensional vector and finally back to the reconstructed vector X^R with the same dimension as the input space. In this study, the AE [m:32:5:32:m] is trained for 120 epochs using Adam. Also, the tanh and ReLU activation function in TensorFlow is used in the compression and reconstruction operations.

In our experiments, only the normal samples are used to train the AE and then the final training loss is calculated. Through a number of preliminary experiments, we find that the best classification performance was obtained when the threshold is set as 1.3 times of the training loss. Thus, in the classification process, 1.3 times of the training loss is used as a threshold for classification judgment. When the loss of a test sample is smaller or equal to this threshold, it will be classified as a normal, otherwise, it will be classified as an intrusion.

3.3.2. *Multi-Layer Perception (MLP).* Multi-layer perceptron (MLP) is a feedforward neural network trained using supervised learning algorithms with an input layer, some hidden layers, and an output layer [43,44]. Figure 3 illustrates the MLP classifier used in this paper, which consists of two hidden layers, one with 32 neurons and the other with 5 neurons in order to be comparable to the AE model described above. Finally, there

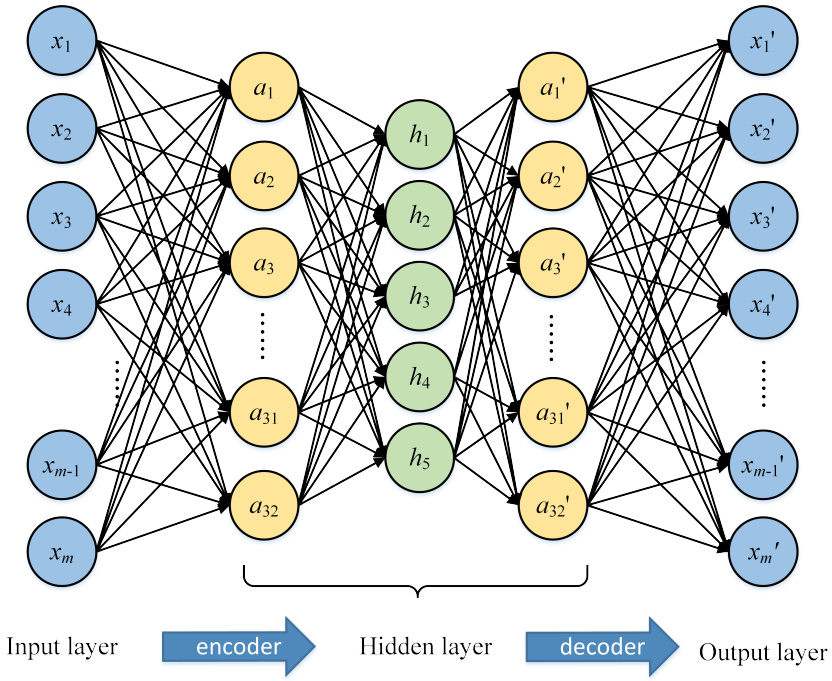


FIGURE 2. The Autoencoder model used in this paper

are two output neurons for binary classification. Besides, the ReLU activation function is applied to train the MLP for 240 epochs using Adam.

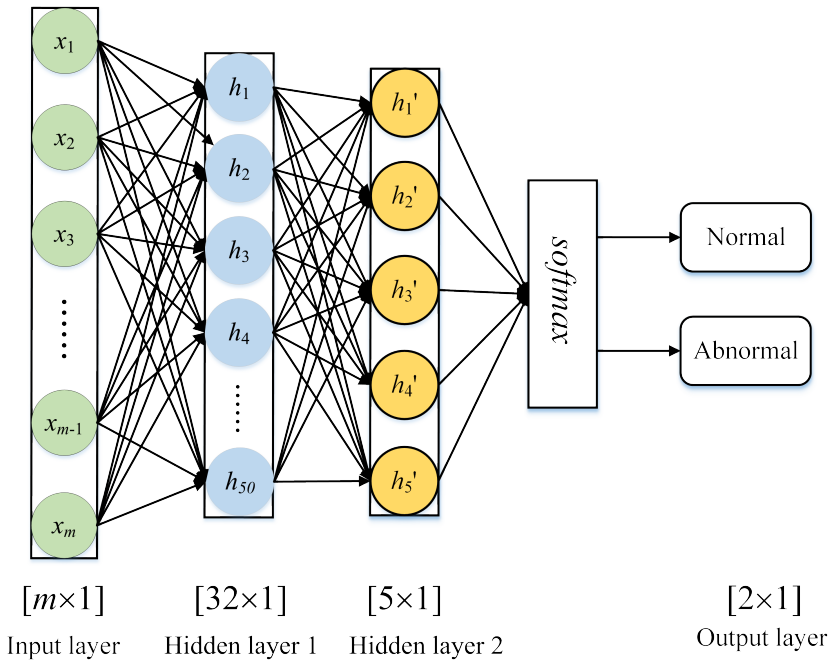


FIGURE 3. The MLP architecture used in this paper

4. **Experiment.** In this section, we first introduce two experimental datasets, then describe some data pre-processing methods and evaluation criteria used in this work, next present in detail the changes in model performance after feature selection, and finally

verify the effectiveness and feasibility of the feature selection methods and strategies used in this work.

4.1. **Dataset Analysis.** To evaluate the feature selection methods and strategies proposed in this study, we use NSL-KDD dataset for the test. The NSL-KDD is a subset of the original KDD99 dataset and it is widely used as the benchmark in several intrusion detection systems. In fact, the NSL-KDD dataset removed some of the redundant and duplicate data that exist in the classes of the KDD-99 dataset having large amount of samples [24].

NSL-KDD is used as a valid benchmark dataset to help researchers compare different intrusion detection methods. It has been divided into training and test datasets, denoted here KDD_{Train+} and KDD_{Test+}, which include 125973 and 22544 instances, respectively. Its details are shown in Table 4. As the setting of training and test sets is reasonable and fixed, the evaluation results of different research works will be consistent and comparable. Like the KDD-99 dataset, it contains 41 features (38 numerical and 3 categorical features). The detailed feature information is shown in Table 5. The three categorical features are transformed into numerical values using the one-hot encoding technique, and Min-Max normalization method is used to map the numeric features into the range of [0-1]. Then the Z-scores method is used to drop out outliers from the normal samples in the training set. In addition, we notice that the normal samples after removing outliers with 3 features being all 0. Therefore, we actively removed the features of z_5, z_{17} and z_{18} and only kept 35 features for feature selection.

Table 4. Details of the NSL-KDD dataset

NSL-KDD	Total	Normal	Dos	Probe	R2L	U2R
KDDTrain+	125973	67343	45927	11656	995	52
KDDTest+	22544	9711	7458	2421	2754	200

4.2. **Data Preprocessing.**

4.2.1. *One-hot Encoding.* We convert the three categorical features to numerical ones using one-hot encoding. One-hot encoding can effectively solve the disorder of classification data, adapt to the requirements of machine learning algorithms, and improve the performance of classification model [24]. For example, the s_1 feature (protocol type) has three attributes: TCP, UDP, and ICMP. Using the one-hot encoding technique, they are converted to binary vectors [1,0,0], [0,1,0] and [0,0,1], respectively. Similarly, s_2 and s_3 features (service and flag) are also converted to the one-hot encoding vectors (s_2 is converted to a 70-dimensional binary vector, and s_3 is converted to an 11-dimensional binary vector). Overall, the original 41-dimensional features are mapped to the 122-dimensional features (38 continuous and 84 binary associated features) [24].

4.2.2. *Data Normalization.* Normalization reduces the effect of different scales on features, thereby improving the model's prediction accuracy. Min-Max normalization is applied after removing outliers, which is shown as follows:

$$x'_{ij} = \frac{x_{ij} - min_(x_j)}{max(x_j) - min(x_j)} \tag{8}$$

where $max(x_j)$ and $min(x_j)$ represent the maximum and minimum of the numeric feature x_j and x'_{ij} are normalized values in range [0-1].

TABLE 5. Features of NSL-KDD dataset: 38 numeric (or continuous, cont.) and 3 categorical (or symbolic, symb.) features

No.	Features	Types	No.	Features	Types
z_1	duration	cont.	z_{19}	is_guest_login	cont.
s_1	protocol_type	symb.	z_{20}	count	cont.
s_2	service	symb.	z_{21}	srv_count	cont.
s_3	flag	symb.	z_{22}	serror_rate	cont.
z_2	source_bytes	cont.	z_{23}	srv_serror_rate	cont.
z_3	destination_bytes	cont.	z_{24}	rerror_rate	cont.
z_4	land	cont.	z_{25}	srv_rerror_rate	cont.
z_5	wrong_fragment	cont.	z_{26}	same_srv_rate	cont.
z_6	urgent	cont.	z_{27}	diff_srv_rate	cont.
z_7	hot	cont.	z_{28}	srv_diff_host_rate	cont.
z_8	num_failed_logins	cont.	z_{29}	dst_host_count	cont.
z_9	logged_in	cont.	z_{30}	dst_host_srv_count	cont.
z_{10}	num_compromised	cont.	z_{31}	dst_host_same_srv_rate	cont.
z_{11}	root_shell	cont	z_{32}	dst_host_diff_srv_rate	cont.
z_{12}	su_attempted	cont.	z_{33}	dst_host_same_src_port_rate	cont.
z_{13}	num_root	cont.	z_{34}	dst_host_srv_diff_host_rate	cont.
z_{14}	num_file_creations	cont.	z_{35}	dst_host_serror_rate	cont.
z_{15}	num_shells	cont.	z_{36}	dst_host_srv_serror_rate	cont.
z_{16}	num_access_files	cont.	z_{37}	dst_host_rerror_rate	cont.
z_{17}	num_outbound cmds	cont.	z_{38}	dst_host_srv_rerror_rate	cont.
z_{18}	is_host_login	cont.			

4.2.3. *Outlier Removal.* In this work, outliers are removed to reduce the degree of data sample imbalance. We first convert the data into the $Z-scores$ using the following formula:

$$Z_{ij} = \frac{(x_{ij} - \overline{x}_j)}{\sigma_j} \tag{9}$$

where $\overline{x}_j$ and σ_j are the mean and standard deviation of feature j, and x_{ij} is the attribute of the ith sample in feature j. Then we have chosen 2σ as the outlier screening criterion, which includes about 95% of the samples in normal distribution. This means that if $|z_{ij}|$ is greater than 0.95, the corresponding sample is considered an outlier and will be removed from the training or test set.

4.2.4. *Evaluation Criteria.* The performance of the proposed classifiers is evaluated using the following traditional metrics: $Precision$, $Recall$, $Accuracy$ and $F1-Score$:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{12}$$

where TP (True Positive) is the number of correctly detected abnormal instances; TN (True Negative) is the number of correctly detected normal instances; FP (False Positive) is the number of normal instances misclassified as abnormal; FN (False Negative) is the

number of abnormal instances that are incorrectly identified as normal [45]. In addition, $F1 - Score$ is used to evaluate the quality of classification, which is defined as:

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{13}$$

As $F1 - Score$ takes the balance between $Precision$ and $Recall$, we will mainly use this metric to evaluate the experimental results. In order to estimate the ability of the classifier to correctly detect normal and abnormal attacks, we also work out the prediction results for normal samples and abnormal attacks by the above formulas, respectively.

4.3. **Results of Experiment.**

4.3.1. *Comparison of Traditional Methods.* In order to further discuss the basis and principle of feature selection, we compare and analyze the importance of each feature sorted by each mathematical method using the NSL-KDD data set. The differences are shown in Figure 4. We can see from this figure that there are obvious differences in the selection of different features by different mathematical methods. Features such as 'num_compromised' and 'diff_srv_rate' are only regarded as more important in ANOVA; 'logged_in' is the most important feature in VP. Similarly, there are differences in the evaluation criteria for the importance of different features by other feature selection methods. In fact, we can draw a clear conclusion by combining the results presented in Figure 4 and Figure 5 that the features selected by different feature selection methods are not the same, which is the key reason that makes the classification performance of DL vary.

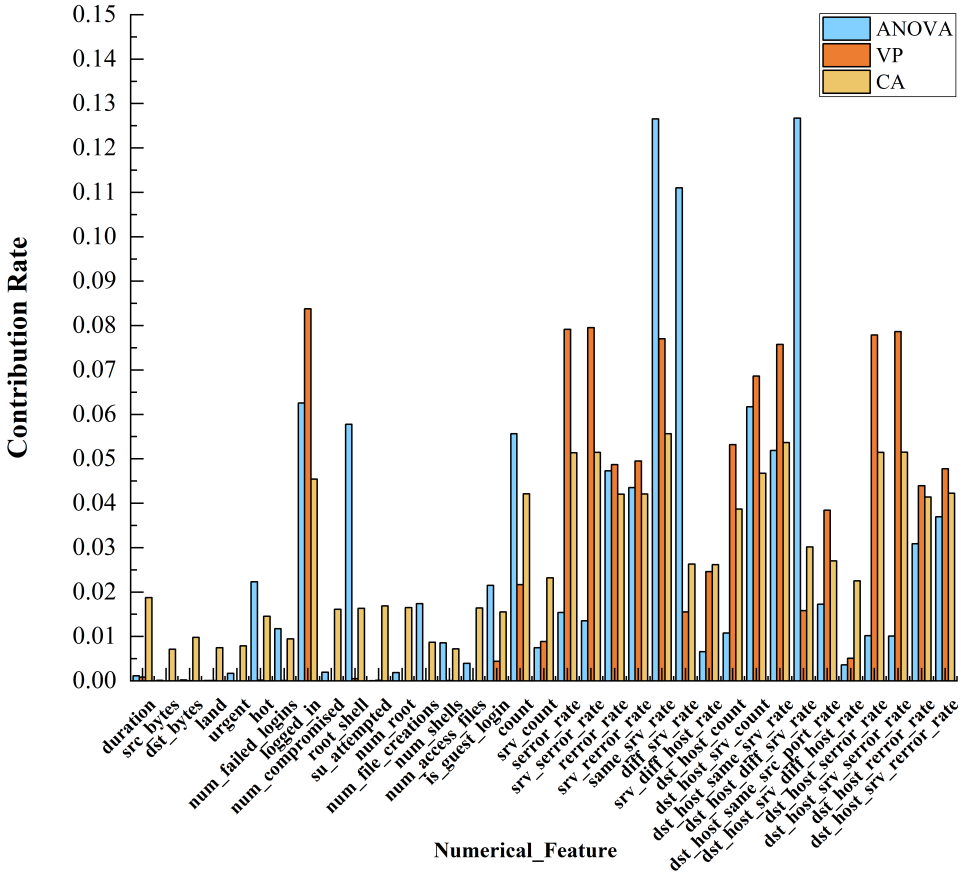


FIGURE 4. Bar chart of the 35 numeric relative features importance obtained by ANOVA, VP and CA methods

We also test the performance of the feature selection methods and strategies described in subsections 3.1 and 3.2 with both AE and MLP models. In the training process, we set the learning rate of AE and MLP as $2 * 10^{-4}$ and 10^{-5}, respectively. The batch size of AE is 120 and MLP is 500. At the same time, we adopt the dropout to avoid overfitting, whose values are set within the range of [0.3, 0.4]. In the AE model, we only use the 'normal' samples to train the model and when the loss of a test sample is greater than 1.3 times of the training loss, it is classified as an abnormal.

In order to make the outlier processing rigorous, we obtained data which only contain normal samples for data screening. In other words, we only analyzed the 'normal' samples to remove the outliers. In the experiments, we use each method to select features based on the feature importance index. The number of features selected are 0, 2, 4, $\cdots$, 34, 35, respectively. Since the values of features z_5, z_{17} and z_{18} are all 0 after removing the outliers, the features z_5, z_{17} and z_{18} are removed. Therefore, only 35 features are used for experimental exploration. Figure 5 shows the 8 results in box plot before and after the removal of outliers by each method.

We can see from Figure 5 that the models are more stable and perform better after the outliers are removed. At the same time, removing outliers provides a guarantee for the subsequent discussion of the performance of different models. Figure 6 shows the influence of traditional feature selection methods on the F1-Score of the classifier when different numbers of features are included in accordance with the feature importance.

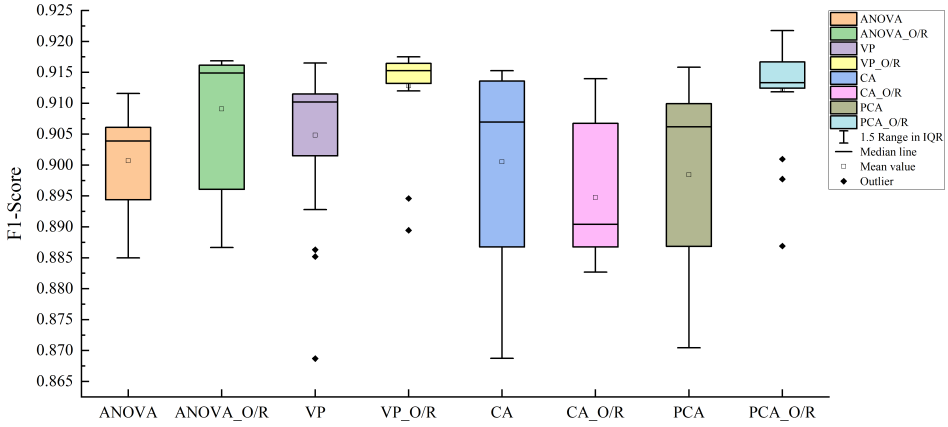
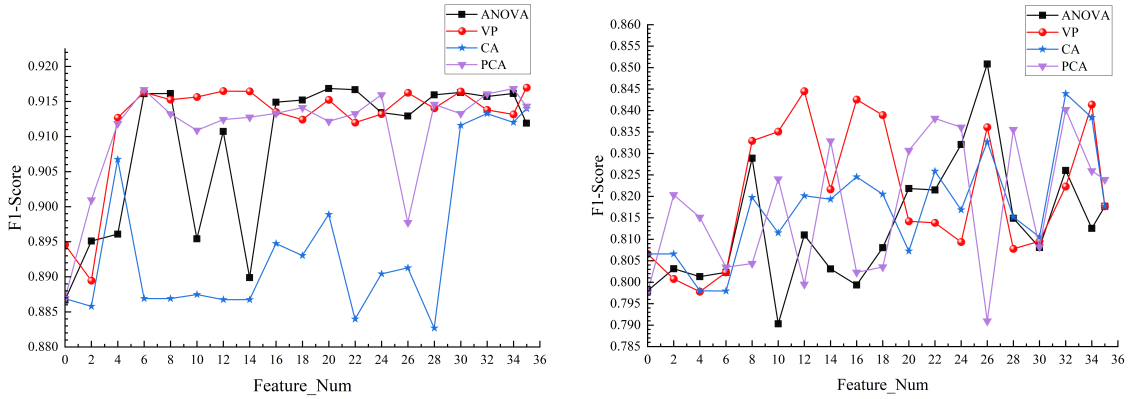


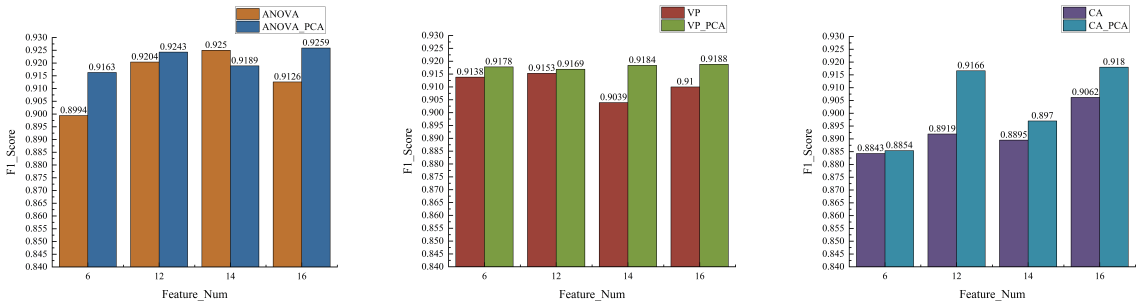
FIGURE 5. Boxplots of the F1-Score obtained from the AE model before and after the removal of outliers for each mathematical feature selection method using the NSL-KDD dataset, where O/R means outliers are removed

Generally, it is believed that most of the datasets potentially have a best set of important features, which means that there is a set of the most important features in the dataset that enables the classifier to achieve the optimal performance [46]. However, it is not difficult to find in Figure 6 (a) and (b) that the prediction performance of both models fluctuates with the inclusion of the number of features. We speculate that this is because any single feature selection method would not be the optimal to improve the learning ability of the classifier. In this regard, we perform another PCA based on the number of features selected. That is, we do not use the total number of features to construct the PCA but only use those selected by one of the traditional methods based on the importance order. Figure 7 shows the performance comparison results in which the number of features selected by one of the traditional methods is the same as that used to construct the PCA.

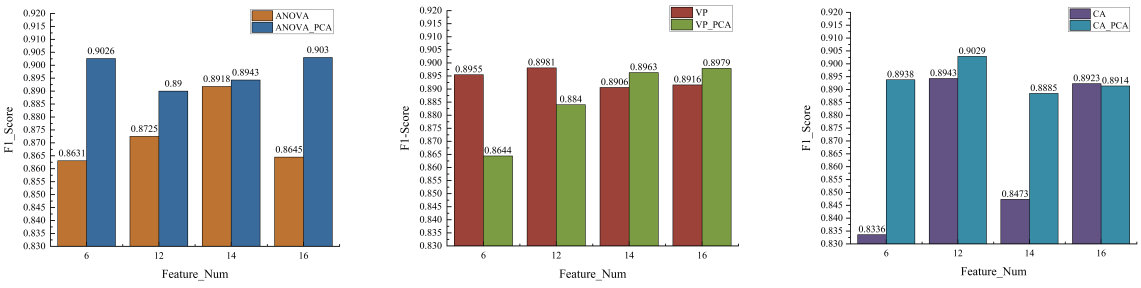


(a) AE classifier (b) MLP classifier

FIGURE 6. Change of the F1-Score with the numbers of features included according to the feature importance after outliers in normal samples are removed in the NSL-KDD dataset using (a) the AE classifier and (b) the MLP classifier



(a) AE classifier



(b) MLP classifier

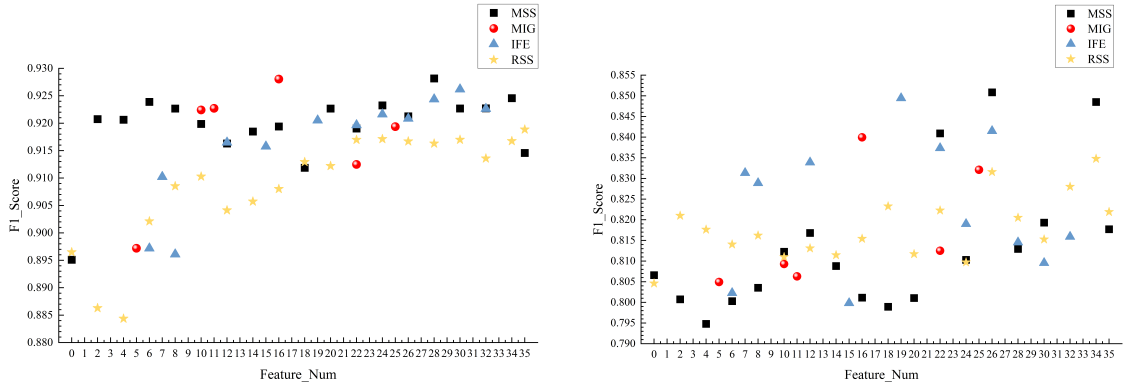
FIGURE 7. Comparison of performance obtained by different numbers of features selected by one of the traditional methods and the recombination of them by the PCA

We can see from Figure 7 that if the AE classifier is used, the appropriate recombination of the original features by the PCA method would generally lead to a better performance. However, when the MLP classifier is used, the recombination of the original features by the PCA method would not always lead to a better performance. Combining the results shown in Figure 6 with those shown in Figure 7, we find that no matter whether all of the original features or part of them are used to construct PCA, the performance achieved by

the PCA would not always be better. Therefore, some new kinds of combination strategies should be explored.

4.3.2. *Comparison of Comprehensive Strategies.* As the features selected by the simple traditional methods are not able to achieve a better result, we are now exploring how the combined strategies of MSS, MIG, IFE and the RSS proposed in Section 3.2 perform in this subsection. The aim of introducing the random selection is to verify whether the traditional or combined methods could perform better than the random one. In order to eliminate the randomness in the experiments, we run each random combination of features 5 times and take the mean of them as the report.

Figure 8 shows the performance comparison of our proposed strategies, in which the number of feature combinations selected by the MIG strategy is 6 (the first 5, 11, 12, 16, 22 and 25 features are selected respectively) and that by the IFE strategy is 12 (6, 7, 8, 12, 15, 19, 22, 24, 26, 28, 30 and 32 features are selected) as some of the combinations do not exist in these two strategies. However, for the MSS and RSS strategies, the number of feature combinations from 0 to 35 could be used.



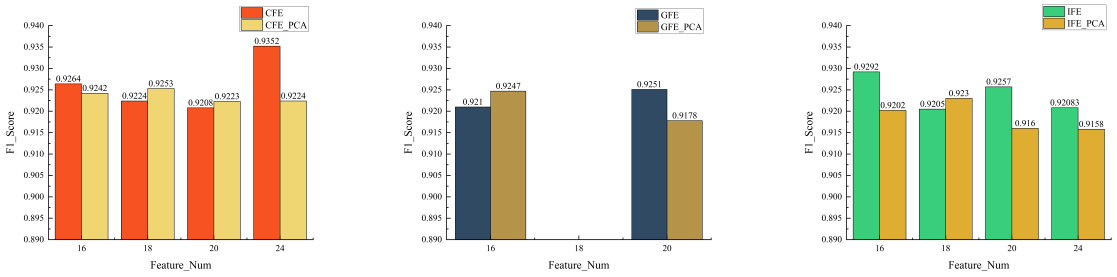
(a) AE classifier (b) MLP classifier

FIGURE 8. Change of the F1-Score with the numbers of features included according to the feature importance after outliers in normal samples are removed in the NSL-KDD dataset using (a) the AE classifier and (b) the MLP classifier

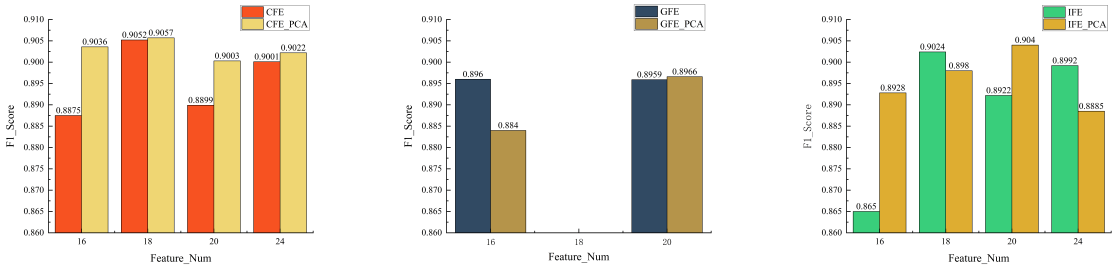
We can see from Figure 8 that AE classifier performed better and tended to be more stable to the change of the number of features included than MLP. However, the performance of both the AE and MLP classifiers fluctuates with the numbers of features included, which means that there may not exist the best number of features to be selected. In addition, we find that when 28 features are selected by the MSS strategy in the AE classifier, the F1-Score is the best, reaching 92.82. At the same time, when the MIG strategy is used, the optimal F1-Score 92.81% is associated with 16 features included. In contrast, the optimal F1-Score is only 85.08% when the MLP model is used, which is achieved by MSS when 26 features are included. In terms of the performance of each feature selection method, the comprehensive selection strategies are generally superior to the random selection method in AE classifier. But this advantage is less evident in MLP classifier.

We also notice from Figure 8 that the performance of the comprehensive strategies also fluctuates with the increase of the number of features included in the two DL classifiers. Therefore, we conduct another PCA based on the number of features selected. The

numbers of features used in PCA are 6, 10, 12, 16, and 22 that are selected by each strategy. Figure 9 shows the performance comparison.



(a) AE classifier



(b) MLP classifier

FIGURE 9. Comparison of the F1-Score obtained by the feature combinations selected by our proposed feature selection strategies and those after PCA conversion

We can see from Figure 9 (a) and (b) that when the best number of features have been selected by a selection algorithm, the recombination of these features by PCA will generally not give better results. For example, the F1-Score obtained from the first 6 features selected by MSS is 92.39% using AE, but when the PCA is applied to the above 6 features, the F1-Score is reduced to 91.66%, decreased by 0.73%. A similar situation has occurred in MIG, in which the F1-Score obtained by the originally selected first 16 features is 92.81% using AE, which is decreased to 92.17% after PCA. For MLP, the F1-Score obtained by the first 12 features selected by MSS method is significantly improved by PCA, which is increased from 81.68% to 85.94%, increased by 4.26%.

Tables 6 and 7 compare the best performance of three measurement indicators obtained by each method and strategy under different classifiers. Table 8 compares the best accuracy of each method and strategy separately. In these tables, the digit followed after each method represents the best number of features selected by that method. For example, MSS (28) in Table 6 represents that the best classification performance is achieved by the MSS when the first 28 features are used and classified by AE classifier.

We can see from these tables that the comprehensive strategies MSS, MIG, and IFE generally perform better than the traditional methods ANOVA, VP, and CA, which are in turn better than the RSS. For example, In AE classifier, the best F1-Score 92.82% is obtained by MSS (28), and the second worst F1-Score 91.71% is obtained by RSS (24), which is only slightly better than the worst F1-Score 91.33%. This is also reflected in the MLP classifier, in which the MSS (26) gets the best F1-Score of 85.06%, but RSS (34) only gets 83.48%, the difference is 1.6%. In addition, the best classification performance achieved by different classifiers using the same feature selection method is

Table 6. Performance (Precision, Recall, F1-Score) comparison of each method using AE classifier

AE							
Metrics	MSS (28)	MIG (16)	IFE (30)	ANOVA (20)	VP (18)	CA (32)	RSS (24)
Precision	90.35%	91.04%	91.07%	88.48%	89.52%	89.19%	90.27%
Recall	95.42%	94.64%	94.22%	95.13%	94.03%	93.57%	93.21%
F1-Score	92.82%	92.81%	92.62%	91.68%	91.72%	91.33%	91.71%

Table 7. Performance (Precision, Recall, F1-Score) comparison of each method using MLP classifier

MLP							
Metrics	MSS (26)	MIG (16)	IFE (19)	ANOVA (26)	VP (12)	CA (32)	RSS (34)
Precision	85.03%	84.13%	84.97%	85.03%	84.5%	84.44%	83.47%
Recall	85.14%	83.86%	84.92%	85.14%	84.40%	84.34%	83.49%
F1-Score	85.08%	83.99%	84.94%	85.08%	84.45%	84.39%	83.48%

Table 8. Accuracy of each method for AE and MLP

AE		MLP	
Methods	Accuracy	Methods	Accuracy
MSS (28)	90.93%	MSS (26)	83.59%
MIG (16)	90.90%	MIG (16)	82.07%
IFE (30)	90.68%	IFE (19)	84.92%
ANOVA (20)	90.57%	ANOVA (26)	83.59%
VP (18)	90.30%	VP (12)	82.71%
CA (32)	90.60%	CA(32)	82.67%
RSS (24)	90.58%	RSS (34)	81.93%

also very different. For example, the best F1-Score obtained by the AE classifier using MSS is 92.82%, while the best F1-Score obtained by the MLP classifier using MSS is only 85.08%.

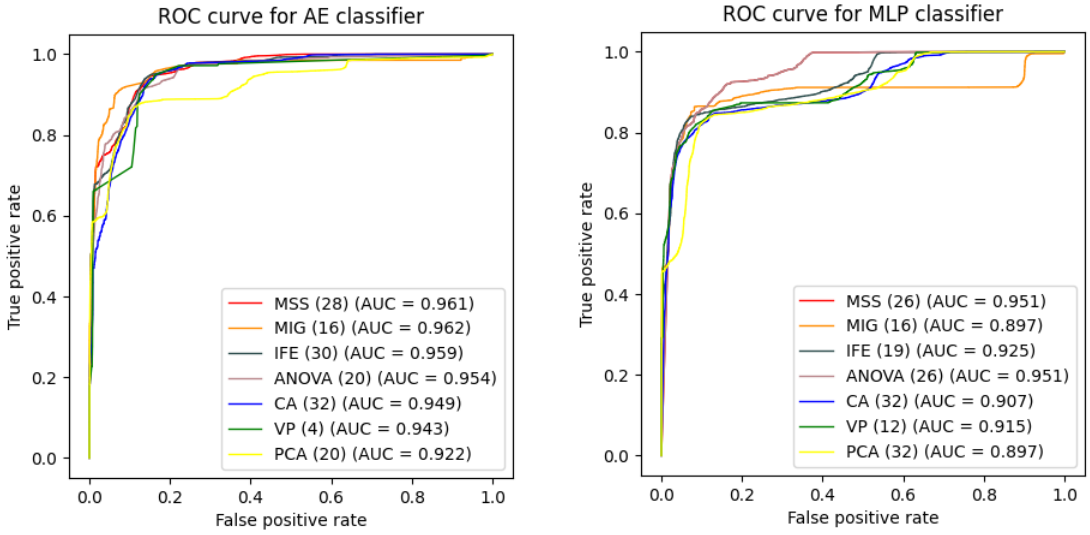
The efficiency of the proposed method is estimated by ROC analysis, and the ROC curves of each method are shown in Figure 10. In fact, among the two classifiers applied in this experiment, the proposed three methods achieved some performance improvement under both classifiers, with the AUC scores obtained by MSS being 96.1% and 95.1%, respectively. (MIG 96.2% and 89.7%, IFE 95.9% and 92.5%, respectively). In addition, we also compare the training and testing time of the features obtained by each selection strategy, and the results are presented in Tables 9 and 10 (in seconds: s). In fact, we find that the training and testing time decreases as fewer features are selected, with some exceptions. For example, 30 features are selected for PCA (training time 60.585), and 16 features are selected for MIG (training time 61.961), but the training time of PCA screening is less than MIG, in Table 9. Similarly, the number of features selected by MSS was 28 (training time 63.157) and the number of features selected by ANOVA was 20 (training time 64:058), but MSS selected features using a shorter training time than ANOVA, in Table 10. The testing time has a similar story. Therefore, we believe that the number of features does not completely determine the training time, and only selecting the truly meaningful features can help to reduce the training time and testing time.

TABLE 9. The training time and testing time after the selection operation by each feature selection method in AE

Methods (number of features)	Training time	Testing time
MSS (28)	63.157	0.532
MIG (16)	61.961	0.503
IFE (30)	64.227	0.486
ANOVA (20)	64.058	0.538
CA (30)	63.796	0.439
VP (4)	58.944	0.380
PCA (30)	60.585	0.517

TABLE 10. The training time and testing time after the selection operation by each feature selection method in MLP

Methods (number of features)	Training time	Testing time
MSS (26)	131.359	0.091
MIG (16)	103.821	0.081
IFE (19)	111.472	0.088
ANOVA (26)	133.412	0.092
CA (32)	123.446	0.066
VP (12)	110.103	0.068
PCA (32)	134.531	0.092



(a) AE classifier (b) MLP classifier

FIGURE 10. Change of the F1-Score with the numbers of features included according to the feature importance after outliers in normal samples are removed in the NSL-KDD dataset using (a) the AE classifier and (b) the MLP classifier

Comparing the results shown in above tables and figures, we find that different classifiers have different optimal number of features corresponding to the same feature selection method. For example, in terms of the F1-Score measurement, the best number of features selection by the IFE strategy for the AE classifier is 30, while for the MLP classifier is 19. This shows that none of the traditional feature selection methods or comprehensive

strategies with fixed number of features can be superior to others over different ML classifiers.

5. **Discussion.** Analyzing the results presented in Section 3 we find that 1) Although the features selected by a specific algorithm generally performed slightly better than those selected randomly, different feature selection methods would give different rankings of feature importance, which shows that it is difficult to say which feature selection algorithm is the best; 2) Recombination of the selected features by PCA would not always give a better result, which indicates that if the features selected by a specific algorithm are not the best, no matter how they are recombined together, they cannot lead to a better result; 3) The prediction performance does not always increase monotonously with the increase of the number of features included, which reveals that although the features have been sorted by a kind of importance index, some features are not important at all in teams of improving the model's performance, the addition of which is only to decease the model's performance instead of improving it; 4) For the same combination of the selected features, different classification models would perform differently, which would further add to the complexity of feature selection issues.

Additionally, observing the results obtained by the comprehensive feature selection strategies MSS, MIG and IFE, we find that 1) Although the optimal performance of the comprehensive strategies was slightly improved compared to that of traditional methods, they did not increase much and in some cases their performance was slightly worse. This means that, if the original order of features is not the best, no matter how they are combined together, it is still difficult to produce a result that is much better than the original ones. The reason is that comprehensive strategies may convey both the advantages and disadvantages from the traditional methods to the new one and hence the final result may be affected by the original negative effects in some ways; 2) Combining the comprehensive strategies with PCA, the results were in some cases better but in other cases worse with no clear trend. This may be because that PCA also includes all the feature information obtained from the original strategies and there may be some interference features among these features; 3) Classifiers and feature selection strategies would jointly affect the final classification result. Comparing the classification performance of AE and MLP using either traditional or comprehensive feature selection strategies, we believe that AE classifier is better than MLP classifier in the field of intrusion detection. But whether this conclusion is still applicable to other fields remains to be revealed; 4) Each method has its own limitations and may not be able to find the 'perfect' order of features. This is because there may be some latent relationships between features and such relationships are very difficult to be found by a single algorithm or any combinations of them.

Table 11. Performance comparison with other approaches on KDDTest+.

Model	Methods	F1-Score	Precision	Recall	Accuracy	Authors
SVM	Iterations	-	-	-	82.68%	Pervez and Farid [21]
RNN	None	-	-	-	83.28%	Yin et al. [22]
AE	Cut 80% features	81.98%	87.00%	80.37%	84.24%	Ieracitano et al. [24]
STL	None	85.28%	96.23%	76.57%	84.96%	Al-Qatf et al. [23]
ELM	DE	75.74%	87.68%	67.20%	87.53%	Almasoudy et al. [25]
AE	PCA	92.26%	86.83%	98.43%	90.61%	Xu et al. [26]
WS	CEF-DE	88.25%	89.09%	87.44%	87.44%	Zhao et al. [27]
MLP	**MSS-PCA**	**85.94%**	**85.73%**	**86.15%**	**84.85%**	**Ours**
AE	**MSS**	**92.82%**	**90.35%**	**95.42%**	**90.93%**	**Ours**

Finally, in order to further evaluate the performance of the feature selection strategies proposed in this paper, we also compare our proposed strategies with the relevant results in intrusion detection using dataset NSL-KDD. Four indicators are used to evaluate the performance, namely *Accuracy*, *Precision*, *Recall*, and $F1 - Score$, and the results are listed in Table 11. We find that the results not only depend on feature selection method, but also on the classifiers used. Generally, AE performed better than other classifiers. However, we can see from Table 11 that our method (AE with MSS) achieved the best results in most of the cases. This shows that the comprehensive feature selection strategies proposed in this paper have advantages in improving classification performance.

6. **Conclusion.** In this paper, extensive experiments were carried out to verify the effectiveness of various feature selection methods and strategies. Experimental results have answered the 5 questions proposed in section 1:

1) The features selected by different algorithms will perform differently in different models, which has added the complexity to feature-selecting issues;

2) Different feature selection algorithms may generate completely different order of feature importance, which leads to difficulty in determining which features are the best to be selected;

3) A well-designed feature selection algorithm generally performs better than the randomly selected ones as well as no selections, which means that it is worth of using a feature selection algorithm to improve the model's classification performance;

4) If the original selection algorithms are not good enough, the performance could not be improved much no matter how they are recombined together;

5) It is difficult to find the optimal number of features being selected, let alone different selection algorithms would lead to different numbers and the number would also depend on the model used. This means that it is impossible to find the optimal number of features being selected in practice.

However, despite the conclusions made above, it is still worth applying a kind of feature selection algorithm to select some important features from a total feature set, especially when the total number of features is very large. In doing so, not only can it lead to some better classification or prediction results, but also reduce the dimension of data and cut down computational complexity. Therefore, it is worth using them to both improve model's performance and at the same time cut down space and time complexities.

In the future, we will explore introducing ideas from different fields into the field of feature selection and propose more effective feature selection methods. Apply the expected method to other feature selection problems in various fields and test them in more classification models to dig out more useful feature selection rules for rapid and accurate recognition and classification tasks.

Acknowledgments. This work was supported in part by the Open Project Program of the Key Laboratory of Cognitive Computing and Intelligent Information Processing of Fujian Education Institutions, Wuyi University (Grant No. KLCCIIP202303); the Natural Science Foundation of Fujian Province (Grant No. 2022J01953); the Fujian Provincial Transportation Science and Technology Demonstration Project (Grant No. FJJT-KJSF2021-01).

REFERENCES

[1] B. Hallajian, H. Motameni, and E. Akbari, "Ensemble feature selection using distance-based supervised and unsupervised methods in binary classification," *Expert Systems with Applications*, vol. 200, p. 116794, 2022.

[2] J. Meng and F. Zhu, "Seek for commonalities: Shared features extraction for multi-task reinforcement learning via adversarial training," *Expert Systems with Applications*, vol. 224, p. 119975, 2023.
[3] Y.-S. Lee, E. Choi, M. Park, H. Jo, M. Park, E. Nam, S.-M. Yi, Kim, and J. Young, "Feature extraction and prediction of fine particulate matter (pm2. 5) chemical constituents using four machine learning models," *Expert Systems with Applications*, vol. 221, p. 119696, 2023.
[4] V. Bolón-Canedo and A. Alonso-Betanzos, "Ensembles for feature selection: A review and future trends," *Information Fusion*, vol. 52, pp. 1–12, 2019.
[5] J. Kim, D.-K. Ghosh, and Y.-J. Jung, "Event-based video deblurring based on image and event feature fusion," *Expert Systems with Applications*, vol. 223, p. 119917, 2023.
[6] M. Abd Elaziz, S. Ouadfel, and R.-A. Ibrahim, "Boosting capuchin search with stochastic learning strategy for feature selection," *Neural Computing and Applications*, vol. 35, no. 19, pp. 14 061–14 080, 2023.
[7] J. Ba, P. Wang, X. Yang, H. Yu, and D. Yu, "Glee: A granularity filter for feature selection," *Engineering Applications of Artificial Intelligence*, vol. 122, p. 106080, 2023.
[8] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. Ieee, 2009, pp. 1–6.
[9] J. Benediktsson and J. Sveinsson, "Feature extraction for multisource data classification with artificial neural networks," *International Journal of Remote Sensing*, vol. 18, no. 4, pp. 727–740, 1997.
[10] S. Kumar, J. Ghosh, and M.-M. Crawford, "Best-bases feature extraction algorithms for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 7, pp. 1368–1379, 2001.
[11] L.-I. Kuncheva and W.-J. Faithfull, "Pca feature extraction for change detection in multidimensional unlabeled data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 69–80, 2013.
[12] G. Zhou, A. Cichocki, Y. Zhang, and D.-P. Mandic, "Group component analysis for multiblock data: Common and individual feature extraction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 11, pp. 2426–2439, 2015.
[13] D. Cheng, S. Zhang, X. Liu, K. Sun, and M. Zong, "Feature selection by combining subspace learning with sparse representation," *Multimedia Systems*, vol. 23, pp. 285–291, 2017.
[14] Y. Liu, Y. Zhao, L. Wang, C. Fang, B. Xie, and L. Cui, "High-impedance fault detection method based on feature extraction and synchronous data divergence discrimination in distribution networks," *Journal of Modern Power Systems and Clean Energy*, 2022.
[15] N. Jain and P.-K. Jana, "Xrrf: An explainable reasonably randomised forest algorithm for classification and regression problems," *Information Sciences*, vol. 613, pp. 139–160, 2022.
[16] K. Kumari and J.-P. Singh, "Multi-modal cyber-aggression detection with feature optimization by firefly algorithm," *Multimedia Systems*, vol. 28, no. 6, pp. 1951–1962, 2022.
[17] R. Hu, Z. Wu, Y. Xu, and T. Lai, "Multi-attack and multi-classification intrusion detection for vehicle-mounted networks based on mosaic-coded convolutional neural network," *Scientific Reports*, vol. 12, no. 1, p. 6295, 2022.
[18] S. Bhattacharya and S. Selvakumar, "Multi-measure multi-weight ranking approach for the identification of the network features for the detection of dos and probe attacks," *The Computer Journal*, vol. 59, no. 6, pp. 923–943, 2016.
[19] A. Naskar, R. Pramanik, S.-S. Hossain, S. Mirjalili, and R. Sarkar, "Late acceptance hill climbing aided chaotic harmony search for feature selection: An empirical analysis on medical data," *Expert Systems with Applications*, vol. 221, p. 119745, 2023.
[20] R. Song and L. Liu, "Event-triggered constrained robust control for partly-unknown nonlinear systems via adp," *Neurocomputing*, vol. 404, pp. 294–303, 2020.
[21] M.-S. Pervez and D.-M. Farid, "Feature selection and intrusion classification in nsl-kdd cup 99 dataset employing svms," in *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*. IEEE, 2014, pp. 1–6.
[22] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21 954–21 961, 2017.
[23] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with svm for network intrusion detection," *IEEE Access*, vol. 6, pp. 52 843–52 856, 2018.
[24] C. Ieracitano, A. Adeel, F.-C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, 2020.

[25] F.-H. Almasoudy, W.-L. Al-Yaseen, and A.-K. Idrees, "Differential evolution wrapper feature selection for intrusion detection system," *Procedia Computer Science*, vol. 167, pp. 1230–1239, 2020.
[26] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, and F. Sabrina, "Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset," *IEEE Access*, vol. 9, pp. 140 136–140 146, 2021.
[27] R. Zhao, Y. Mu, L. Zou, and X. Wen, "A hybrid intrusion detection system based on feature selection and weighted stacking classifier," *IEEE Access*, vol. 10, pp. 71 414–71 426, 2022.
[28] L. St and S. Wold, "Analysis of variance (anova)," *Chemometrics and Intelligent Laboratory Systems*, vol. 6, no. 4, pp. 259–272, 1989.
[29] M. Neuhäuser and B.-F. Manly, "The fisher-pitman permutation test when testing for differences in mean and variance," *Psychological Reports*, vol. 94, no. 1, pp. 189–194, 2004.
[30] D.-R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.
[31] F. Anowar, S. Sadaoui, and B. Selim, "Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne)," *Computer Science Review*, vol. 40, p. 100378, 2021.
[32] B.-M.-S. Hasan and A.-M. Abdulazeez, "A review of principal component analysis algorithm for dimensionality reduction," *Journal of Soft Computing and Data Mining*, vol. 2, no. 1, pp. 20–30, 2021.
[33] H. Shi, H. Li, D. Zhang, C. Cheng, and X. Cao, "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," *Computer Networks*, vol. 132, pp. 81–98, 2018.
[34] L. Graefe, S. Hahn, and A. Mayer, "On the relationship between anova main effects and average treatment effects," *Multivariate Behavioral Research*, vol. 58, no. 3, pp. 467–483, 2023.
[35] Q. Liu and L. Wang, "t-test and anova for data with ceiling and/or floor effects," *Behavior Research Methods*, vol. 53, no. 1, pp. 264–277, 2021.
[36] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.
[37] L.-C. Lee and A.-A. Jemain, "On overview of pca application strategy in processing high dimensionality forensic data," *Microchemical Journal*, vol. 169, p. 106608, 2021.
[38] R. Hu, Z. Wu, Y. Xu, T. Lai, and C. Xia, "A multi-attack intrusion detection model based on mosaic coded convolutional neural network and centralized encoding," *Plos One*, vol. 17, no. 5, p. e0267910, 2022.
[39] X. Li, W. Chen, Q. Zhang, and L. Wu, "Building auto-encoder intrusion detection system based on random forest feature selection," *Computers & Security*, vol. 95, p. 101851, 2020.
[40] A. Kurani, P. Doshi, A. Vakharia, and M. Shah, "A comprehensive comparative study of artificial neural network (ann) and support vector machines (svm) on stock forecasting," *Annals of Data Science*, vol. 10, no. 1, pp. 183–208, 2023.
[41] A. Abhaya and B.-K. Patra, "An efficient method for autoencoder based outlier detection," *Expert Systems with Applications*, vol. 213, p. 118904, 2023.
[42] G. Zhang, Y. Liu, and X. Jin, "A survey of autoencoder-based recommender systems," *Frontiers of Computer Science*, vol. 14, pp. 430–450, 2020.
[43] X. Zhang, X. Cao, J. Wang, and L. Wan, "G-unext: a lightweight mlp-based network for reducing semantic gap in medical image segmentation," *Multimedia Systems*, vol. 29, no. 6, pp. 3431–3446, 2023.
[44] N. Tathawadekar, N.-A.-K. Doan, C.-F. Silva, and N. Thuerey, "Modeling of the nonlinear flame response of a bunsen-type flame via multi-layer perceptron," *Proceedings of the Combustion Institute*, vol. 38, no. 4, pp. 6261–6269, 2021.
[45] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121–2133, 2022.
[46] J.-P. Horwath, D.-N. Zakharov, R. Mégret, and E.-A. Stach, "Understanding important features of deep learning models for segmentation of high-resolution transmission electron microscopy images," *npj Computational Materials*, vol. 6, no. 1, p. 108, 2020.