International Journal of Engineering

Journal Homepage: www.ije.ir

Feature Engineering Methods in Intrusion Detection System: A Performance Evaluation

F. Zare, P. Mahmoudi-Nasr*

Department of Compute Engineering, University of Mazandaran, Mazandaran, Iran

PAPER INFO

Paper history:
Received 19 March 2023
Received in revised form 24 April 2023
Accepted 06 May 2023

Keywords:
Feature Selection
Dimensions Reduction
Intrusion Detection System
Deep Neural Network
Security
Machine Learning

ABSTRACT

Today, the number of cyber-attacks has increased and become more complex with an increase in the size of high-dimensional data, which includes noisy and irrelevant features. In such cases, the removal of irrelevant and noisy features, by Feature Selection (FS) and Dimensions Reduction (DR) methods, can be very effective in increasing the performance of intrusion detection systems (IDS). This paper compares some FS and DR methods for detecting cyber-attacks with the best accuracy using implementation on KDDCUP99 dataset. A Deep Neural Network (DNN) is used for training and simulating them. The results show the filter methods are faster than wrapper methods but less accurate. Whereas the Wrapper methods have more accuracy but are computationally costlier. Embedded methods have the best output and maximum values, which is 99% for all the metrics, comparing to it the DR methods have shown a good performance and speed, among them Linear Discriminant Analysis (LDA) method even better than embedded method.

doi: *10.5829/ije.2023.36.07a.15*

1. INTRODUCTION

With the increasing use of internet, the threats have also increased and become more complicated, so the protection of networks in the modern world has become increasingly important. A key technology for ensuring the security of networks is intrusion detection systems [1], which are referred to as the first line of defense for securing networks [2].

In general, IDSs are divided into Signature-based and Behavior-based, with the former using Machine Learning algorithms (ML), which have been widely used by data scientists recently [3-5]. The first step in working with ML algorithms is feature engineering for reducing and cleaning the input data to make it faster and more accurate. Feature engineering techniques improve the detection performance by extracting relationships between data and removing irrelevant information [6].

Two important parts of feature engineering are Feature Selection (FS) and Dimensions Reduction (DR), which play an important role in machine learning classification problems [7]. Therefore, the detection process will be faster and more intensive, which in turn leads to a lower demand for computing resources. Some of the advantages of these methods are as follows:

1. Reduce the probability of overfitting the model and get a better generalization.
2. Give a better data visualization to get a good comprehension.
3. The unsupervised methods are useful when there is no labeled data or not enough labeled data.
4. Reduce the training time by removing the noisy and redundant data.
5. Improve the performance and accuracy of the model by removing the irrelevant data.

Since there are different types of FS and DR methods, it's necessary to select the best one. The purpose of this paper is to compare various methods of feature engineering.

Due to the increasing complexity and different types of attacks, this paper prefers to work on a behavior-based IDS, and due to the better performance of neural networks among the ML algorithms [8]. A Deep Neural Network (DNN) is selected as the final classifier. In this paper, we trained a DNN as a classifier for a behavior-based IDS to compare the performance of FS methods, i.e., filtering, wrapper, and embedded methods, as well as

Corresponding Author Institutional Email: P.mahmoudi@umz.ac.ir
(P. Mahmoudi-Nasr)

Please cite this article as: F. Zare, P. Mahmoudi-Nasr, Feature Engineering Methods in Intrusion Detection System: A Performance Evaluation, ***International Journal of Engineering, Transactions A: Basics***, Vol. 36, No. 07, (2023), 1343-1353

DR methods that categorize into linear and non-linear methods. As a classifier in FS methods, some ML algorithms like Support Vector Machine (SVM) [9] and Random Forest (RF) [10] are used to find the best features.

The rest of the paper is organized as follows: In section 2, the related works are presented. Details and explanations of the FS and DR methods are examined in section 3. The reports and experimental results are conducted in section 4 and the conclusion is given in section 5.

2. RELATED WORK AND CONTRIBUTIONS

Feature engineering is one of the most time-consuming parts in ML problems [11], FS trying to find the good features that are selected from the data and are not redundant, are related to the output, also cause more difference between classes and minimize the internal variance and maximize the external variance. FS Algorithms need to search in the data space, so four basic essential problems in their diversity have explained starting point, search organization, evaluation method and algorithms elimination metric data structure, model structure and data diversity make impression on FS. Each FS algorithm can give different results on an individual data set. FS is suitable to obtain the nature of the data, otherwise there are other methods such as feature extraction, which have become very popular in recent years.

Ghasemi and Esmaily [12] presented an IDS using KDDcup99 and NSL-KDD datasets based on machine learning algorithms. They pointed out that feature selection plays an important role in standard benchmark datasets. Therefore, the GA algorithm was used to select the optimal features. In this paper, it was shown that the data dimensions have an important effect on the performance of the algorithm, and finally the DT algorithm together with the feature selection method achieved the highest evaluation scores.

Venkatesh and Anuradha [13] have shown the importance of dimensionality reduction due to the increase of noisy data, which affects the performance of the algorithm. They have explained the FS methods and divided them into filter, wrapper and embedded methods. They have explained six stages for FS methods which consist of search direction, search strategy, evaluation criteria, stopping criteria and validation of results.

Biglari et al. [14] claimed that high-dimensional data posed a major challenge to the data mining problem. They presented a four-step feature selection method to improve the efficiency of machine learning algorithms on high-dimensional data. The proposed method was applied to two high-dimensional data and achieved a prediction accuracy of 0.92 and 0.99 (99%).

Kou et al. [15] tried 10 FS methods to get the best result on a text classification problem with 10 different data sets. The authors declared the evaluation of a FS method to be a Multiple Criteria Decision Making (MCDM) problem. They also used nine evaluation measures for binary classification and seven evaluation measures for multi-class classification. According to the results, it is obvious that MCDM-based methods are effective in evaluating the methods of FS.

Mohammadi et al. [16] have proposed a FS method to improve the performance of IDS using KDDcup99. It is a combination of a filtering method, where the linear correlation coefficient reduces the computational complexity, and a wrapper method, which is the Cuttlefish algorithm; in addition, the ID3 algorithm was used as a classifier. The results show a significant improvement in performance compared to using only the filter method or the wrapper method.

Meza and Touahria [17] have created a helpful review of FS methods to improve an IDS, they have explained many approaches in FS for IDS. They have proposed a new taxonomy of FS algorithms and presented their properties depending on different datasets, selection mechanisms, selection approaches, selection techniques, classifiers, selection features and multi-objective aspects.

Gündüz and Çeter [18] have conducted an experiment to improve the performance of IDS, which classifies attacks using four classification algorithms, namely Multi Layers Perceptron, Support Vector Machine (SVM), Decision Tree and a fuzzy-based algorithm on the KDDcup99 dataset. The FS is created first by correlation and then using the Best First Search (BFS) algorithm, where the 11 most important features were selected. The classification results show an improvement in performance after applying the FS.

Umar and Zhanfang [19] have tested five classification algorithms, i.e. Artificial Neural Network (ANN), SVM, Naïve Bayes (NB), K Nearest Neighbors (KNN) and Random Forest (RF) for improvement and IDS with two datasets, NSL-KDD and UNSW-NB15. The chosen FS. Method is wrapper-based. They compared the classification with and without FS, and the results show an increase in speed for all algorithms except ANN and a negligible decrease in accuracy. This shows the importance of FS in reducing execution time.

Zhao et al. [20] have proposed a novel model for intrusion detection using PCA and a classifier for the KDDcup99 dataset. To choose the best classifier, two algorithms were compared: soft max regression, an extended version of logistic regression, and K Nearest Neighbor. The experimental results show that the dimensions reduced by PCA contain negligible loss of useful information, but the redundant data are significantly reduced. For the classification algorithms, both achieved similar performance, with Soft max Regression having lower execution time.

Saranya et al. [21] have presented an investigation of ML algorithms to improve IDS for the Internet of Things (IoT) using the KDDcup99 dataset. They have implemented three algorithms: Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and the RF. Experimental results show an accuracy of 99.65% for RF, 98.1% for LDA and 98% for the CART algorithm. The work has also compared different ML algorithms, i.e. K-means, J.48, SVM, PCA, logistic regression, decision tree (DT) and ANN. The results show that ANN and DT are the classifiers with the highest accuracy (99.65%).

Considering the above context, the key contributions to this paper are as follows:

- Find out the best way to make intrusion detections faster and more accurate with less use of computational resources.
- A detailed comparison between most popular DR and FS methods from prediction performance point of view.
- Examine the application of SVM and DNN to the problem of FS and DR in IDS.

3. RESEARCH METHODOLOGY

The overall view of the methods evaluated and implemented in this research is shown in Figure 1.

3. 1. Filtering Methods Filtering methods [22] are independent of ML algorithms, so they are more optimal than other methods in terms of computational load. Figure 2 shows the overall structure of filtering methods. In these methods, FS is done according to the feature ranking by statistical characteristics such as Distance, Correlation, Information and Consisting.

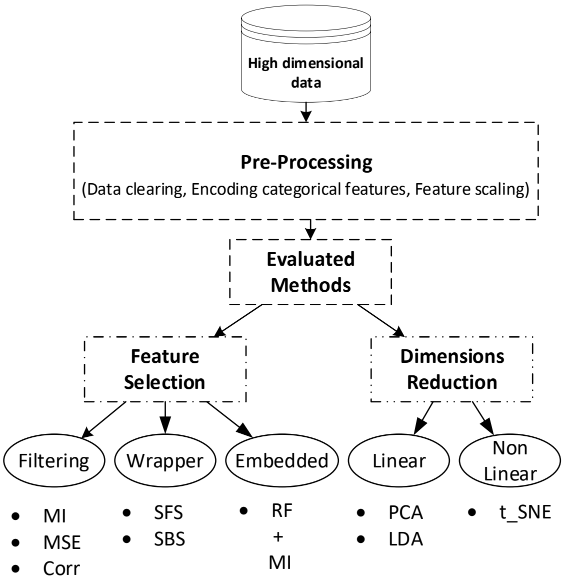

Figure 1. overall view of evaluated methods

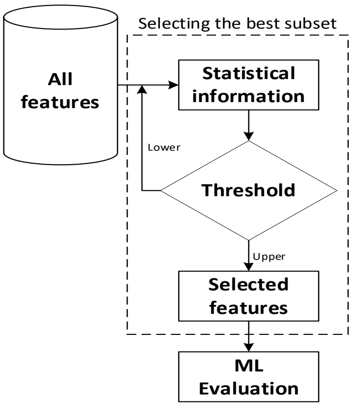

Figure 2. Overall structure of Filtering methods

3. 1. 1. Mutual Information (MI) In this method, the MI of each feature is calculated according to Equation (1) and the best features are selected according to the maximum value of MI.

$$MI(x_k, y) = Exy\left[log\,\frac{P(x,y)}{P(x)P(y)}\right] \quad (1)$$

If there is no relevance between x and y, they are considered independent and the value MI would be equal to 0.

3. 1. 2. Mean Squared Error (MSE) This method also provides a value called MSE according to Equation (2), which gives the mean squared error.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (2)$$

This method uses the output errors obtained with a ML algorithm that also requires a predefined threshold to select the best features.

3. 1. 3. Correlation The correlation metric selects the features that have the greatest relevance to the output, as the lower correlation indicates the separable and redundant features [23]. The correlation is determined according to Equation (3).

$$Correlation = \frac{Cov(\chi_k, d)}{\sqrt{Var(x_k)}\sqrt{Var(y)}} \quad (3)$$

A threshold value for the correlation value should be considered and the features with a correlation value below the threshold value should be removed. This metric works linearly and is not suitable for the nonlinear relevancies.

3. 2. Wrapper Methods As shown in Figure 3, wrapper methods use ML evaluation algorithms to select a subset of features. These methods require more processing time than filtering methods because the evaluation algorithm is run multiple times and each time a subset of features is selected and then performance is examined according to the predefined learning model

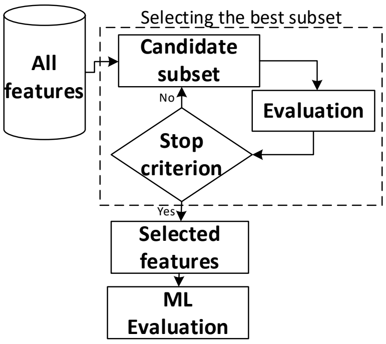

Figure 3. Overall structure of Wrapper Methods

with an evaluation metric. Since the wrapper methods naturally operate during model training, they cause a heavy computational load when processing big data.

These methods divide into greedy and non-greedy. This section presents the greedy methods, which select the features that seem best at the time and are usually trapped in a local minimum. Some of the greedy methods are as follows:

3. 2. 1. Best Individual *n* Features In this method, to select *n* features, the cost function is calculated *t* times, then the features with the best value are selected. So, the training is repeated *t* times to find *n* best features. This method works blindly and has the disadvantage that the duplicate features are not considered and a good repeated feature appears several times in the output features.

3. 2. 2. Sequential Forward Search (SFS) At the beginning, there is a single feature that is updated in each step. First the best single feature is selected, in the next step a feature is selected that is best related to the previous feature, and so on until the end of all features. It goes forward in the same way through all features to find *n* number features, but this has the following weaknesses:

1. The selection of the first feature follows the method of the *best individual feature* and therefore has its disadvantages.
2. The features selected in the next steps are based on the previously selected features, so the removed features have no chance to serve as the main component of the feature set.
3. Each selected feature is frozen and remains until the end, even if it could not form the best feature set.

3. 2. 3. Sequential Backward Search (SBS) This method starts with a totality of all features as a set and in each step one of them is removed. In the initial state, the cost function has the maximum value and is reduced by the elimination in each step, so the feature chosen for removal should have the least influence on the cost function. This method has 2 weaknesses:

1. After deleting a feature, there is no way to select it in the next steps.

2. Unlike SFS, it starts with a large number of features, which reduces the reliability of the cluster, which is why the SFS method is more popular than SBS.

3. 3. Embedded Methods Filtering methods do not use clusters reduce performance, and wrapper methods are also computationally intensive. Thus, embedded methods proposed to use the clusters to determine the criteria during training and usually use for specific ML algorithm. As can be seen in Figure 4, in these methods, the search for the optimal subset of features would occur in the cluster design phase and can be viewed as a search in a combined space of subsets and hypotheses.

Random Forest is a very powerful model for both regression and classification, which can also provide its own interpretation of feature importance. Each tree of the random forest can calculate the importance of an attribute according to its ability.

The higher the importance of the feature, the more appropriate feature to choose, and according to the importance of each feature, feature selection is done.

3. 4. Comparison Between Feature Selection Methods Table 1 summarizes the comparison between the above methods. Filtering methods are appropriate when the speed of FS is more important, and wrapper methods are appropriate for systems that are delay tolerant and have the ability to provision the computational resources, and for systems that care about both, embedded methods are good.

As it is mentioned in Table 1, interaction with the classifier can be an advantage. The filtering methods select the features just according to statistical criteria to score the correlation or dependence between the input variables and determine the relationship between them, but the wrapper and embedded methods work with the classifier and select the features according to the main problem and evaluate and categorize effective features and introduce them to the model.

3. 5. Dimensions Reduction (DR) As can be seen in Figure 5, the DR methods are generally divided into linear and nonlinear methods. These methods change the

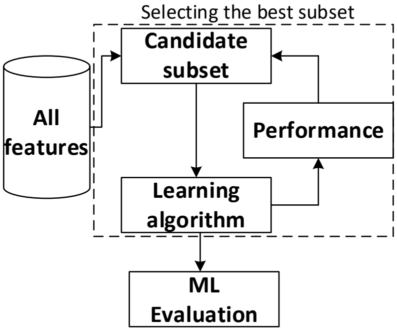

Figure 4. Overall structure of Embedded Methods

TABLE 1. Methods Comparison

Methods	Advantages	Disadvantages
Filtering	Independent of classifier, less computational cost, short running time,	No interaction with classifier
Wrapper	Interaction with classifier, Recording attribute dependency	High computational cost, Overfitting feasibility, dependent on classifier
Embedded	Interaction with classifier, less computational cost, Recording attribute dependency	Dependent on classifier

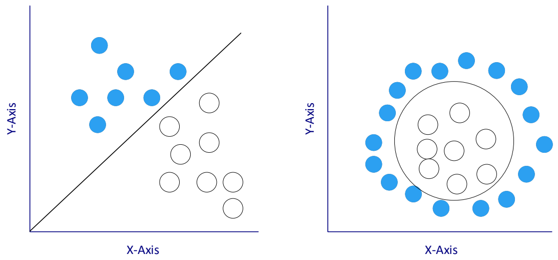

Figure 5. Structure of linear and none linear methods

distribution of data points to the distribution in which the data can be visually classified.

As part of this category, Principal Component Analysis (PCA) [24] and Linear Discriminant Analysis (LDA) [25] as linear methods, and t-distributed stochastic neighbor embedding (t-SNE) as nonlinear technique are described.

3. 5. 1. Principal Component Analysis To reduce computational costs and process complexity when using high-dimensional data, features must be reduced and combined rather than deleted, although the combination should contain the information of all features.

There are many methods to do this, but PCA is one of the first and most important methods in the field DR. PCA doesn't consider the data labels, which is why it's also called an unsupervised method. When projecting data, some data information is inevitably lost. Therefore, the PCA method selects the axis that preserves the most information, and when projecting data with this axis, the loss of data information is the least.

PCA requires preprocessing of the data before the dimensions are reduced, which can be done using formula 4.a, or to compensate for the deviation of the data points according to Equation (4.b), the standard deviation can also be part of the preprocessing.

$$\text{X}=x_j^i-\mu_j \tag{4.a}$$

$$\text{X}=\frac{x^i_{j-\mu_j}}{\sigma_j} \tag{4.b}$$

X refers to input data with *m*n* dimensions (*m* is the number of samples and *n* is the number of features).

3. 5. 2. Linear Discriminant Analysis The LDA algorithm is a kind of counterparty to PCA, since it uses the data labels and falls into the category of supervised methods. In the following, this method is studied as a binary and multiclass method.

3. 5. 3. Multi-Classes For multiple class data needed to determine the median, it must be determined for each class and for the entire classes in general according to Equations (5) and (6), respectively.

$$\mu_i=\frac{1}{n_i}\sum\nolimits_{x_i\in c_i}x_i \tag{5}$$

$$\mu=\frac{1}{n}\sum x_i \tag{6}$$

The variance is calculated in two parts, first the variance between classes (SB) and then the variance within classes (SW) according to Equations (7) and (8), respectively.

$$S_B=\sum\nolimits_{i=1}^{c}n_i(\mu_i-\mu)(\mu_i-\mu)^{\mathsf{T}} \tag{7}$$

$$S_w=\sum\nolimits_{i=1}^{c}s_i;\, s_i=\sum\nolimits_{i=1}^{c}\sum\nolimits_{x_k\epsilon c_i}(x_k-\mu_i)(x_k-\mu_i)^{\mathsf{T}} \tag{8}$$

The eigenvalues and vectors are determined using Equation (9).

$$S_Bv=\alpha s_wv\ ;\ s_w^{-1}s_Bv=\alpha v \tag{9}$$

3. 5. 4. t-distributed Stochastic Neighbor Embedding (t-SNE) LDA and PCA reduce the data by finding a linear relationship between the data, but if there are too many features in the data, it's better to use a nonlinear method, which is called one of the most famous nonlinear methods t_SNE and is newer than PCA and LDA. The t_SNE algorithm is a complicated calculation [26].

The t_SNE method doesn't use labels, so it's also unsupervised and reduces data dimensions by extracting a nonlinear relationship. This section contains a brief explanation of its steps.

Step 1: The similarity rate between data points in high dimensions is calculated according to Equation (10) and the similarity for each data point is determined using the Guassian distribution.

$$\text{P}_{\text{j|i}}=\frac{exp\left(-\|x_i-x_j\|^2\right)/2\sigma^2}{\sum_{k\neq i}exp(-\|x_i-x_k\|^2)/2\sigma^2} \tag{10}$$

Step 2: This step is a repeat of the previous step, but using a different distribution called Student's t_distribution with freedom of 1, called the Cauchy distribution, so that the Qj|i for each data point are calculated according to Equation (11).

$$\text{Qj|i}=\frac{exp\left(-\|x_i-x_j\|^2\right)}{\sum_{k\neq i}exp(-\|y_i-y_k\|^2)} \tag{11}$$

Step 3: in this step, the Kull-Back-Leibler divergence metric (KL) plays the role of the cost function. Each distribution tries to keep the parameters as small as possible by making the best use of the gradient decent.

3. 5. 5. Comparison Between Feature Selection Methods

PCA and LDA are linear methods and can not handle complex and high dimensional data, but t_SNE is non-linear and suitable for high dimensional data. More over the LDA versus PCA and t_SNE is supervised and requires labeled data. On the speed discussion according to the done experiments, the fastest algorithm is LDA and the lowest one is t_SNE.

4. IMPLEMENTATION

Implementation is done using python 3.9 and on a on a machine with Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz -2.30 GHz, 1 NVIDIA GP108 PCIe 2GB and 12GB RAM, using the Visual Studio Code environment with Keras and SKlearn library.

As can be seen in Figure 6, this paper attempts to find the best way to reduce the data before training a Deep Neural Network (DNN) for a classification problem. The DNN works as a supervised method and uses the labels of the data to be trained. Feature selection methods (FS) are also evaluated using Support Vector Machine (SVM) and the results can be seen in Table 2.

4. 1. Dataset And Performance Evaluation

The dataset that has been chosen for evaluation and FS is KDDcup99, which is still working as an useful dataset [21, 27]. This dataset is selected base on: (i) lack of public benchmark datasets for network-based intrusion detection, (ii) popularity and frequent use of KDDcup99 dataset by many researchers as a good bench mark dataset, (iii) various number of attack classes in this dataset, (iv) arrangement of records in such a way that there is no need to randomly select a part of the dataset to well train a model.

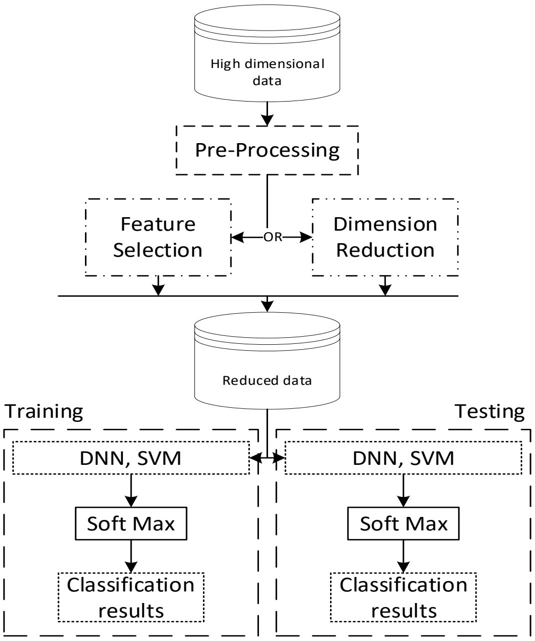

Figure 6. Implementation Structure

TABLE 2. Evaluation results for feature selection methods by SVM (Acc, Pre stand for Accuracy and Precision respectively)

Method	Selected Features	Acc	Pre	Recall	F1-Score
MI	2,4,22	99%	98%	99%	98%
Correlation	9,15,28	79%	78%	79%	71%
MSE	2,4,22	99%	98%	99%	98%
SFS	2,3,4	99%	99%	99%	99%
SBS	2,4,29	99%	98%	99%	99%
Embedded	-	**99.9%**	**99.9%**	**99.9%**	**99.9%**

This dataset has 41 features which three of them are object type (Protocol_type, Service and Flag). Totally there are four types of attacks in this dataset:

- **DoS:** Denial-of-Service, e.g. syn flood
- **R2L:** Remote to Local, unauthorized access from a remote machine, e.g. guessing password
- **U2R:** User to Root, unauthorized access to local superuser (root) privileges, e.g., buffer overflow
- **Probe:** surveillance and other probing, e.g., IP sweeping.

The dataset consists of totally 1,072,992 records which divided into 812,814 normal, 247,267 DoS, 13,860 Probe, 999 R2L and 52 U2R records.

In this work, the results and the selected features are denoted by their indices from 0 to 41.

The performance scores used in this work are **accuracy**, **precision**, **Recall** and **F1 score**, which are given in the Equations (12)-(15), respectively.

$$accuracy = TP + TN \big/ TP + TN + FP + FN \tag{12}$$

$$precision = TP \big/ TP + FP \tag{13}$$

$$recall = TP \big/ TP + FN \tag{14}$$

$$F1-score = \frac{2 * precision * recall}{precision + recall} \tag{15}$$

4. 2. Pre-processing

Before start working with the algorithms of ML, it's necessary to preprocess the input data. The 3 main steps of preprocessing are as follows:

1. *Missing value*: the selected KDD is complete and doesn't need this phase.

4. 2. 1. Encoding the Categorical Data In this paper the label encoding method was used for the protocol_type, service, flag and label columns.

4. 2. 2. Feature Scaling In this phase, the standard scaler is used to set the values between -1 and 1. The standardization was performed according to Equation (16).

$$Standard\ x = \frac{x_j - \overline{x_j}}{\sigma_j} \tag{16}$$

4. 3. Filtering Methods In this method, FS is based on the classification of features in terms of their statistical properties. To perform the filtering methods such as MI, MSE and correlation, *n* (number of selected features) was set to 3, so that the 41 original features of KDD are reduced to 3. In this section, a SVM was implemented to validate the methods.

4. 3. 1. Mutual Information (MI) This method calculates the MI and then selects the features with the most MI. Then, depending on the importance of accuracy or speed, the *n* numbers of the best selected features must be selected.

As can be seen in Figure 7, according to the result of the MI method, the 3 best features (No. 2, No. 4 and No. 22) were selected. To validate the selected features, they are applied to a SVM as input, and according to Table 2, the results show satisfactory values for the validation metrics.

4. 3. 2. MSE Decision tree was selected as a ML algorithm to evaluate each feature to predict the target, and finally the *n* number of features with the minimum MSE are selected.

According to the plot in Figure 8, which shows the results of applying the MSE method to the KDD, the top three features (No. 2, No. 4, and No. 22) are selected to be scored with SVM. As can be seen in Table 2, the metrics scores are very similar to the method MI.

4. 3. 3. Correlation The correlation of a variable indicates the degree of its relationship with the corresponding target. If the correlation is equal to "1", it means that they are strongly correlated, and if it is equal to "-1", it means that there is an inverse relationship between them. Then, a threshold of α must be set and a number of *n* must be selected for the remaining features.

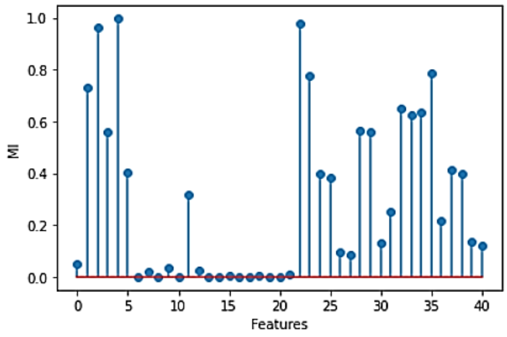

Figure 7. Features based on MI

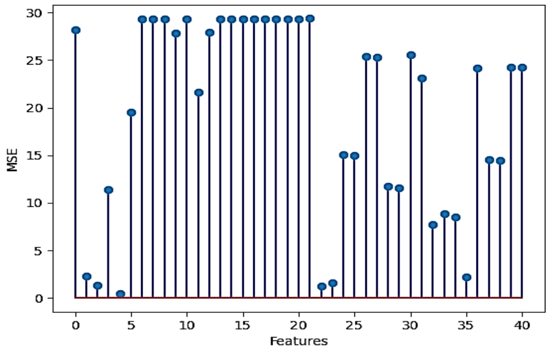

Figure 8. features based on MSE

In this work, the correlation between the features is calculated first, and the features whose correlation is greater than 'α" are selected, which is 0.8 in this implementation. As can be seen in Table 3, seven groups with correlation greater than 0.8 were found, from each of which one feature was selected and the others were removed. By running the Random Forest (RF) for each feature group, the features with the highest significance are selected (Table 4). Next, the three best features (No. 9, No. 15, and No. 28) have been examined with SVM, as shown in Table 2.

TABLE 3. Correlated Features (F stands for Feature)

Group1			Group 4		
F 1	**F 2**	**Correlation**	**F 1**	**F 2**	**Correlation**
38	25	0.999334	32	33	0.973634
38	37	0.998142	32	28	0.898427
38	24	0.997839	32	2	0.867102
38	28	0.857570	-	-	-
-			**Group 5**		
-	-	-	**F 1**	**F 2**	**Correlation**
-	-	-	21	9	0.83892
Group 2			**Group 6**		
F 1	**F 2**	**Correlation**	**F 1**	**F 2**	**Correlation**
26	27	0.994817	35	23	0.944650
26	39	0.986782	35	1	0.860319
26	40	0.984970	35	22	0.860243
Group 3			**Group 7**		
F 1	**F 2**	**Correlation**	**F 1**	**F 2**	**Correlation**
15	13	0.995016	3	28	0.851775

TABLE 4. Important Feature of each group

Group number	Most Important Feature	Importance
1	28	0.576205
2	40	0.426946
3	15	1.0
4	2	0.541133
5	9	1.0
6	22	0.527443
7	28	1.0

4. 4. Wrapper Methods This category includes several methods. To implement the wrapper methods, the variable *n* was set to 3. The implemented wrapper methods are as follows:

4. 4. 1. Sequential Forward Search (SFS) The SFS method attempts to eliminate the redundant features and selects a number of *n* remaining features to achieve satisfactory accuracy and speed. In this step, useful algorithms can be used to select the best features.

For simplicity, the correlation matrix is used in this implementation and the features with high correlation were removed. About 14 features were removed and the number of features was reduced to 27. Then, K Neighbors was used to find the 3 best features.

After removing 14 features through the correlation matrix, the selected algorithm for the SFS method found the three best features according to the accuracy metric. As can be seen in Table 2, the SFS selected the three best features (No. 2, No. 3, and No. 4) with accuracy greater than 99%.

4. 4. 2. Sequential Backward Search (SBS) Backward FS starts with all features and builds a model that deletes one feature at each step to get the best result. In this algorithm, the termination metric is reaching a certain number of features, so the *n* numbers of features needed can be determined at this stage.

To implement this algorithm, the RF algorithm is used to find the 3 best features according to the accuracy. Table 2 shows that the three best features (No. 2, No. 4 and No. 29) selected by the SBS method have an accuracy of more than 99%. As can be seen, the results are too close to the SFS method.

4. 5. Embedded Method The results of this method depend on the chosen machine learning algorithm (ML) that FS uses during the training phase. After that, another method such as MI is applied to the results of the algorithm to capture the 'n" number of features needed.

4. 5. 1. RF In this phase, the features are clustered using RF as the classification algorithm and ordered by MI.

As can be seen in Figure 9, which shows the feature group evaluated by the RF algorithm, the top three features with the most MI were selected. Table 2 and a comparison between the presented methods show that the embedded method using RF and MI received the most values for the validation metrics and outperformed the filtering and wrapper methods.

4. 6. Dimensions Reduction In this section, a DNN was implemented to validate the methods. Figure 10 shows the structure of implemented DNN. The implemented DNN has only 2 neurons in the first layer, 10 neurons in the second layer, and finally 23 neurons for the last layer due to the 23 types of attacks that exist in the KDD dataset.

ReLU was chosen for the activation function in the first layer, *tanh* was chosen for the second layer, and *SoftMax* was chosen for the last layer, and *dropout* was also applied between layers to suppress the overfitting of the model.

Due to the existence of different types of neural networks (LSTM, CNN, TCN, DNN...), it can be claimed that there are different types of feature extraction and the most suitable network should be selected according to the type of problem. Therefore, apart from the fact that neural networks are able to extract features, when it comes to performance comparison and in all comparison modes the number of training epochs is fixed, the model in the mode that receives inputs that pass through the FS or DR stages can make better use of the limited number of epochs to be trained better than a model that is faced with raw input data with the same number of epochs. So, in the first phase of implementation, the DNN is trained and tested with all 41 features of KDD, and the results are

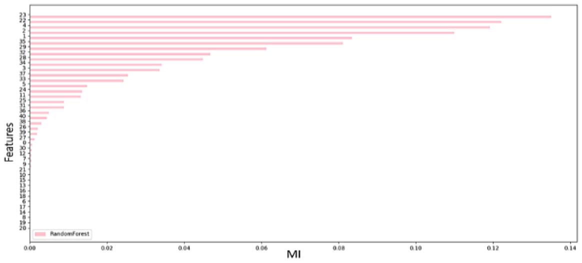

Figure 9. Random Forest Output

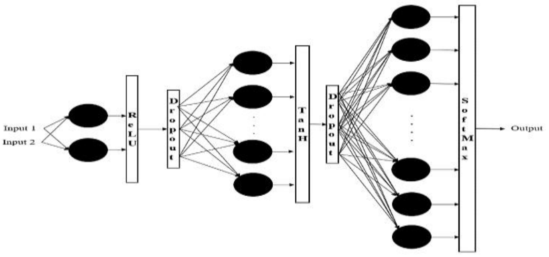

Figure 10. Structure of Deep Neural Network

shown in Table 5. As can be seen, the accuracy of the model is about 78%, which is very low compared to the other implementations.

In the next stage, Dimensions Reduction (DR) methods such as PCA, LDA and t_SNE are applied to the data.

4. 6. 1. PCA In order to perform PCA as a linear method DR, the number *n* of required dimensions must be defined, which can be determined, for example, according to the number of neurons in a Deep Neural Network.

The PCA applied to KDD is set to *n*=2, so that all 41 features are mapped into 2 dimensions and passed as input to the DNN. The validation results can be seen in Table 5. As can be seen, the DNN with the PCA method outperformed the validation without applying any FS method.

4. 6. 2. LDA The LDA method works as a linear method, and to reduce the dimensions of the data, the *n* numbers for the dimensions of the output data must first be specified. These may vary depending on computational resources, speed required, or accuracy needed.

To compare the results of the methods of DR, the output dimensions of LDA were also set to 2 and the mapped data was used as DNN input. As can be seen in Table 5, the LDA method outperformed PCA.

4. 6. 3. t_SNE This method is useful for complicated data with too many features. Since it is a nonlinear method, it takes more time to reduce the dimensions of the data. The number of training epochs may be different, so it takes time to determine the number of epochs.

The t_SNE method was applied with two different iterations, first with 500 and then with 1000, and the output dimension was set to 2. According to Table 5, the t_SNE method did not perform well on KDD, it might work better on datasets with many more features.

TABLE 5. Evaluation results for dimension reduction and feature selection methods by DNN

Method	Accuracy	Precision	Recall	F1-Score
Without FE	0.7853	1.0000	0.5685	0.7209
PCA	0.9401	0.9454	0.9400	0.9426
LDA	**0.9776**	**0.9783**	**0.9774**	**0.9778**
t-SNE (500 iter)	0.6749	0.8536	0.5218	0.6446
t-SNE (1000 iter)	0.6917	0.8203	0.5934	0.6865
MI (22, 4)	0.9466	0.9567	0.9180	0.9366
Correlation	0.5698	0.5733	0.5698	0.5698
MSE (4, 22)	0.7854	0.7854	0.7854	0.7854
SFS (3, 4)	0.7852	0.7852	0.7852	0.7852
SBS (2, 29)	0.7852	0.9184	0.9183	0.9183
Embedded (22, 23)	**0.9701**	**0.9704**	**0.9698**	**0.9701**

5. DISCUSSION

For comparison and further evaluation, three best features were selected from those chosen by the above methods and the performance of each was evaluated using the SVM algorithm on the KDDCUP99 dataset. As can be seen from Table 2, the SFS and SBS methods are close to each other, although the running time of SBS is much longer than that of SFS. Among the algorithms in Table 2, the correlation method has the lowest accuracy but also the shortest running time. Using MI method in embedded methods leads to maximum performance and satisfactory running speed.

So, the 3 different methods such as PCA, LDA, and t_SNE were applied before training the DNN to see the difference in performance. To have a complete comparison, the DNN model was also trained and tested using the selected features by FS methods.

The comparison results can be found in Table 5. As can be seen, the LDA method has the best accuracy, even compared to the embedded method, which was the best among the FS methods.

6. CONCLUSION

In this paper, the methods of Feature Selection (FS) and Dimensions Reduction (DR) are presented and compared using a DNN. The importance of DR for Big Data was shown as it increases performance, and a comparison was made between FS methods using the implementation on the KDDCUP99 dataset. The wrapper methods have higher accuracy but are more computationally expensive. Embedded methods had the best results and maximum values that are 99% for all metrics.

This paper also compares the methods of DR. Based on the implementation results, it can be seen that LDA has the best performance among the mentioned methods, even ahead of the embedded method. The t_SNE method is also very accurate, it can achieve better results on data sets with very high dimensions.

The experimental results of this paper show that:

Among the FS methods:

1. The filtering methods have the minimum run time.
2. The wrapper methods have the best accuracy.
3. The embedded methods present a trade-off between run time and accuracy.

Among the DR methods:

4. The LDA has the best value.
5. The t-SNE methods takes too long to response, it may show better results on very high dimensional data.

6. The PCA is also very close to LDA, but still among them all, the LDA has shown the best results.

In future works, the FS method will be discussed by heuristic algorithms on a more complicated data with more dimensions. Deep Learning methods that go forward by feature extraction will be also studied.

7. REFERENCES

1. Li, X., Chen, W., Zhang, Q. and Wu, L., "Building auto-encoder intrusion detection system based on random forest feature selection", ***Computers & Security***, Vol. 95, (2020), 101851. https://doi.org/10.1016/j.cose.2020.101851
2. Kasongo, S.M. and Sun, Y., "A deep learning method with wrapper based feature extraction for wireless intrusion detection system", ***Computers & Security***, Vol. 92, (2020), 101752. https://doi.org/10.1016/j.cose.2020.101752
3. MR, G.R., Somu, N. and Mathur, A.P., "A multilayer perceptron model for anomaly detection in water treatment plants", ***International Journal of Critical Infrastructure Protection***, Vol. 31, (2020), 100393. https://doi.org/10.1016/j.ijcip.2020.100393
4. ur Rehman, S., Khaliq, M., Imtiaz, S.I., Rasool, A., Shafiq, M., Javed, A.R., Jalil, Z. and Bashir, A.K., "Diddos: An approach for detection and identification of distributed denial of service (ddos) cyberattacks using gated recurrent units (GRU)", ***Future Generation Computer Systems***, Vol. 118, (2021), 453-466. https://doi.org/10.1016/j.future.2021.01.022
5. Abdelaty, M., Doriguzzi-Corin, R. and Siracusa, D., "Daics: A deep learning solution for anomaly detection in industrial control systems", ***IEEE Transactions on Emerging Topics in Computing***, Vol. 10, No. 2, (2021), 1117-1129. DOI: 10.1109/TETC.2021.3073017
6. Butcher, B. and Smith, B.J., *Feature engineering and selection: A practical approach for predictive models: b*y Max Kuhn and Kjell Johnson. Boca Raton, FL: Chapman & Hall/CRC Press, (2019), https://doi.org/10.1080/00031305.2020.1790217
7. Tran, M.-Q., Liu, M.-K. and Elsisi, M., "Effective multi-sensor data fusion for chatter detection in milling process", ***ISA Transactions***, Vol. 125, (2022), 514-527. https://doi.org/10.1016/j.isatra.2021.07.005
8. Chalapathy, R. and Chawla, S., "Deep learning for anomaly detection: A survey", ***Computer Science***, (2019). https://doi.org/10.48550/arXiv.1901.03407
9. Guo, Y., Zhang, Z. and Tang, F., "Feature selection with kernelized multi-class support vector machine", ***Pattern Recognition***, Vol. 117, (2021), 107988. https://doi.org/10.1016/j.patcog.2021.107988
10. Nazir, A. and Khan, R.A., "A novel combinatorial optimization based feature selection method for network intrusion detection", ***Computers & Security***, Vol. 102, (2021), 102164. https://doi.org/10.1016/j.cose.2020.102164
11. Chio, C. and Freeman, D., "Machine learning and security: Protecting systems with data and algorithms, " O'Reilly Media, Inc.", (2018).
12. Ghasemi, J. and Esmaily, J., "A novel intrusion detection systems based on genetic algorithms-suggested features by the means of different permutations of labels' orders", ***International Journal of Engineering*, *Tansactions A: Basics,*** Vol. 30, No. 10, (2017), 1494-1502. DOI: 10.5829/ije.2017.30.10a.10
13. Venkatesh, B. and Anuradha, J., "A review of feature selection and its methods", ***Cybernetics and Information Technologies***, Vol. 19, No. 1, (2019), 3-26. https://doi.org/10.2478/cait-2019-0001
14. Biglari, M., Mirzaei, F. and Hassanpour, H., "Feature selection for small sample sets with high dimensional data using heuristic hybrid approach", ***International Journal of Engineering*, *Tansactions B: Applications*** Vol. 33, No. 2, (2020), 213-220. DOI: 10.5829/IJE.2020.33.02B.05
15. Kou, G., Yang, P., Peng, Y., Xiao, F., Chen, Y. and Alsaadi, F.E., "Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods", ***Applied Soft Computing***, Vol. 86, (2020), 105836. https://doi.org/10.1016/j.asoc.2019.105836
16. Mohammadi, S., Mirvaziri, H., Ghazizadeh-Ahsaee, M. and Karimipour, H., "Cyber intrusion detection by combined feature selection algorithm", ***Journal of Information Security and Applications***, Vol. 44, (2019), 80-88. https://doi.org/10.1016/j.jisa.2018.11.007
17. Maza, S. and Touahria, M., "Feature selection algorithms in intrusion detection system: A survey", ***KSII Transactions on Internet and Information Systems***, Vol. 12, No. 10, (2018), 5079-5099. https://doi.org/10.3837/tiis.2018.10.024
18. Gündüz, S.Y. and ÇETER, M.N., "Feature selection and comparison of classification algorithms for intrusion detection", ***Anadolu University Journal of Science and Technology A-Applied Sciences and Engineering***, Vol. 19, No. 1, (2018), 206-218. https://doi.org/10.18038/aubtda.356705
19. Umar, M.A. and Zhanfang, C., "Effects of feature selection and normalization on network intrusion detection", (2020).
20. Zhao, S., Li, W., Zia, T. and Zomaya, A.Y., "A dimension reduction model and classifier for anomaly-based intrusion detection in internet of things", in 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), IEEE. (2017), 836-843. DOI: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.141
21. Saranya, T., Sridevi, S., Deisy, C., Chung, T.D. and Khan, M.A., "Performance analysis of machine learning algorithms in intrusion detection system: A review", ***Procedia Computer Science***, Vol. 171, (2020), 1251-1260. https://doi.org/10.1016/j.procs.2020.04.133
22. Zhang, Y., Yang, C., Yang, A., Xiong, C., Zhou, X. and Zhang, Z., "Feature selection for classification with class-separability strategy and data envelopment analysis", ***Neurocomputing***, Vol. 166, (2015), 172-184. https://doi.org/10.1016/j.neucom.2015.03.081
23. El Bilali, A., Taleb, A. and Brouziyne, Y., "Groundwater quality forecasting using machine learning algorithms for irrigation purposes", ***Agricultural Water Management***, Vol. 245, (2021), 106625. https://doi.org/10.1016/j.agwat.2020.106625
24. Shlens, J.J.a.p.a., "A tutorial on principal component analysis", arXiv preprint arXiv:1404.1100, (2014). https://doi.org/10.48550/arXiv.1404.1100
25. Izenman, A.J., Linear discriminant analysis, in Modern multivariate statistical techniques. 2013, Springer. 237-280.
26. Van der Maaten, L. and Hinton, G.J.J.o.m.l.r., "Visualizing data using t-sne", Vol. 9, No. 11, (2008).
27. Ravipati, R.D. and Abualkibash, M., "Intrusion detection system classification using different machine learning algorithms on kdd-99 and nsl-kdd datasets-a review paper", ***International Journal of Computer Science & Information Technology***, Vol. 11, No. 3, (2019). http://dx.doi.org/10.2139/ssrn.3428211

Persian Abstract

چکیده

امروزه تعداد حملات سایبری با افزایش حجم داده‌های با ابعاد بالا که شامل ویژگی‌های نویزدار و نا مرتبط است، افزایش یافته و پیچیده‌تر شده است. در چنین مواقعی حذف ویژگی های نامرتبط و نویزی می تواند در افزایش عملکرد سیستم های تشخیص نفوذ بسیار موثر می‌باشد. این مقاله برخی از روش‌های انتخاب ویژگی و کاهش ابعاد را برای تشخیص حملات سایبری با استفاده از پیاده‌سازی روی مجموعه داده KDDCUP99 مقایسه می‌کند. یک شبکه عصبی عمیق برای آموزش و پیاده سازی آنها نیز استفاده می شود. نتایج نشان می‌دهد که روش‌های فیلتر سریع‌تر از روش‌های بسته بندی هستند اما دقت کمتری دارند. در حالی که روش های بسته بندی دقت بیشتری دارند اما از نظر محاسباتی پرهزینه تر هستند. روش‌های تعبیه شده، بهترین خروجی را دارند که برای همه معیارها به میزان ۹۹ درصد رسیده است، در مقایسه با آن روش‌های کاهش بعد، عملکرد و سرعت خوبی از خود نشان داده‌اند که در میان آنها روش تحلیل تفکیک خطی، بهتر از روش های تعبیه شده می‌باشد.