ISSN 2616-5775 Vol. 7, Issue 12: 44-50, DOI: 10.25236/AJCIS.2024.071206

Research on Intrusion Detection Classification Methods Based on PCA-Decision Tree

Tengyao Wang*

School of Information Engineering, Xinyang Agriculture and Forestry University, Xinyang, 464000, China
**Corresponding author: wangtengyao666@qq.com*

Abstract: *This paper focuses on enhancing the performance of Intrusion Detection Systems (IDS) within the domain of network security through data preprocessing and algorithm optimization, addressing the increasingly complex landscape of network threats. This study utilizes the NSL-KDD dataset, addressing high dimensionality with Principal Component Analysis (PCA) and handling outliers via RobustScaler. Logistic regression and decision tree models were used for classification, with random oversampling applied to mitigate class imbalance. The main innovation of this study is the integration of PCA and decision tree models, which improves the detection accuracy in high-dimensional data, as well as the effective use of random oversampling to address class imbalance issues. The experimental results showed that the decision tree model surpassed logistic regression, achieving a prediction accuracy of 99.63% compared to 91%, confirming the superiority of nonlinear classification models for this datasets. Moreover, the balanced accuracy of the model improved after applying random oversampling, proving the efficacy of this method in addressing minor class imbalance. This study demonstrates that a combination of PCA and random oversampling techniques can significantly enhance IDS performance in detecting complex network threats. The results also indicate that nonlinear models such as decision trees are better suited for handling high-dimensional and imbalanced datasets, providing important insights for improving IDS performance. Future research may consider integrating more sophisticated deep learning methods and leveraging real-time big data analytics to further improve IDS generalization and responsiveness to emerging threats.*

Keywords: *Network Security, Intrusion Detection, Principal Component Analysis, Imbalanced Data Processing, Decision Trees*

1. Introduction

As the internet becomes more widely adopted and big data technologies advance rapidly, the volume of network traffic has surged, emphasizing the growing importance of network security. Intrusion detection technologies play a critical role in network defense systems. Most current Intrusion Detection Systems (IDS) rely heavily on misuse detection techniques, which focus on pattern matching. These systems create a feature library by collecting and storing known intrusion signatures, which are then used to compare with real-time network data to identify potential security threats. However, rule-based methods face challenges such as inefficiency when processing large volumes of data, high false-positive rates, and limited generalization capabilities.

Relevant studies provide valuable insights into addressing these challenges. For instance, Zhou Qi proposed a machine learning-based intrusion detection method, which optimizes feature selection using the improved ReliefF algorithm, significantly reducing feature dimensions and lowering the computational complexity of classifiers. This approach improved detection efficiency by approximately 57.8%[1]. Similarly, Shang Jiaqi introduced an intrusion detection model that integrates the DC-SMOTE algorithm, which combines local density and centrality with a triple-attention mechanism. This model addresses the class imbalance in the NSL-KDD datasets by first applying the DC-SMOTE algorithm to enhance the number of minority attack samples, followed by the use of a triple-attention mechanism for feature extraction and classification, effectively improving detection accuracy[2].

In light of these challenges, this study focuses on two key aspects: (1) Handling imbalanced datasetss to improve the accuracy and efficiency of the system's detection capabilities. (2) Utilizing Principal Component Analysis (PCA) for dimensionality reduction of high-dimensional data to reduce data redundancy, thereby improving system performance and response speed. The innovative aspect of

ISSN 2616-5775 Vol. 7, Issue 12: 44-50, DOI: 10.25236/AJCIS.2024.071206

this study is the combination of PCA with decision tree classifiers, a novel approach to improving detection accuracy on high-dimensional data, and the application of random oversampling to tackle the prevalent issue of class imbalance in IDS datasets. The primary research contributions and the technical approach of this paper are outlined in Figure 1 as follows:

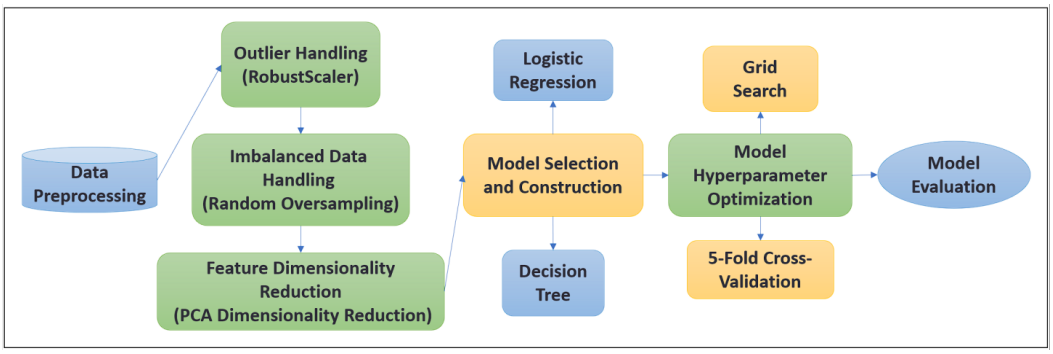


Figure 1: Technical Approach of This Paper

2. Data Exploration

2.1 Data Source and Distribution

The dataset used in this study originates from the [NSL-KDD datasets](https://www.kaggle.com), a revised version of the well-known KDD'99 datasets. The datasets contains a total of 125,971 samples, with each record consisting of 43 attributes. Of these, 41 attributes represent the network traffic input features, while the remaining two are the label (normal or attack) and a score indicating the severity of the network traffic. A subset of the dataset is shown in Table 1 below:

Table 1: Sample from the Original datasets

	duration	protocol_type	...	outcome	level
0	0	udp	...	normal	15
1	0	tcp	...	attack	19
...	...	...	...	...	...
125970	0	tcp	...	attack	20
125971	0	tcp	...	normal	21

In general network traffic, intrusion samples are far fewer compared to normal traffic samples. Figure 2 below visualizes the overall distribution of the NSL-KDD datasets, highlighting this imbalance between normal and attack samples:

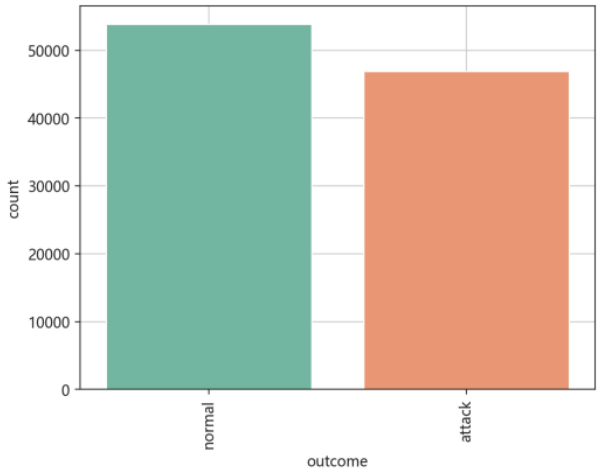


Figure 2: Distribution of Normal and Attack Samples in the NSL-KDD datasets

ISSN 2616-5775 Vol. 7, Issue 12: 44-50, DOI: 10.25236/AJCIS.2024.071206

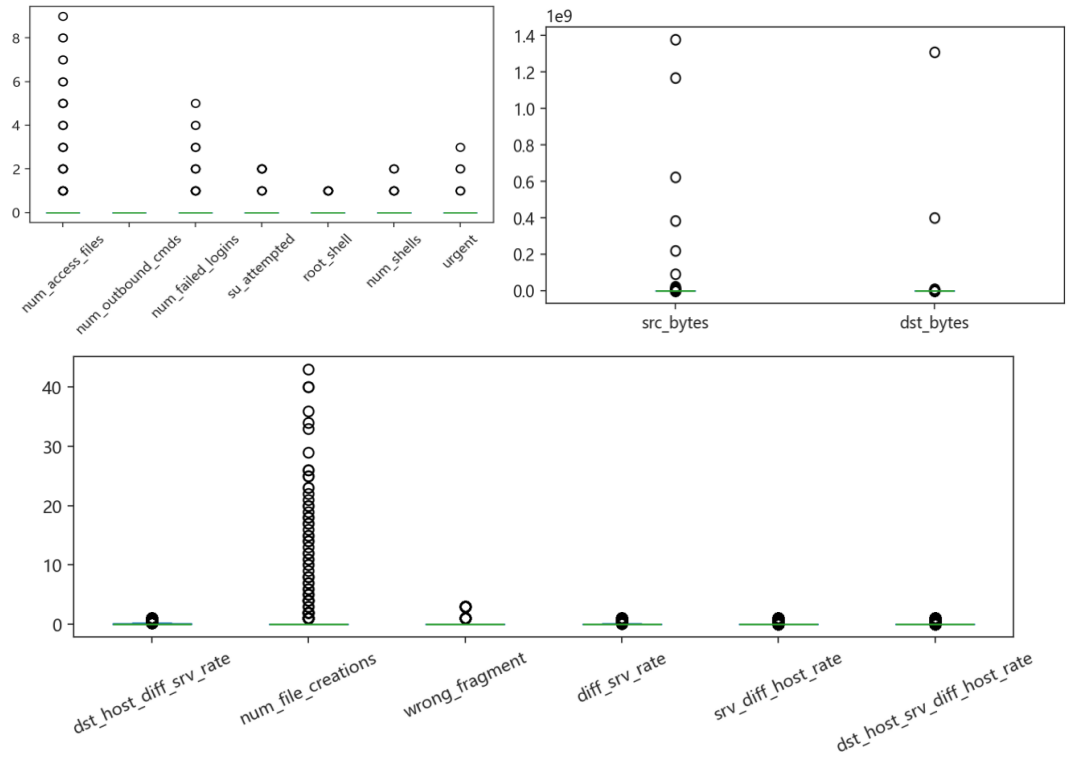


Figure 3: Boxplot of Input Features in the NSL-KDD datasets

Upon examining the continuous variables using boxplots (see Figure 3), a significant number of outliers was identified. These anomalies necessitate appropriate handling during the data preprocessing stage. Therefore, RobustScaler was applied to process the outliers. RobustScaler is a robust method for data standardization, particularly suited to datasets containing outliers. Unlike the StandardScaler, which is based on mean and standard deviation, RobustScaler relies on the median and interquartile range (IQR) for scaling. Specifically, it scales the data by subtracting the median and then dividing by the IQR, thus minimizing the impact of outliers during the normalization process. Compared to other methods like MinMaxScaler, RobustScaler demonstrates superior performance when faced with skewed distributions or extreme values[3].

2.2 PCA for Dimensionality Reduction

2.2.1 Principles of PCA

Principal Component Analysis (PCA) is a commonly employed statistical technique, particularly effective for reducing dimensionality and feature extraction in high-dimensional datasets. The core of PCA lies in transforming the original, potentially correlated variables into a set of new, linearly uncorrelated variables—known as principal components—thereby simplifying the complexity of the datasets without significantly losing information. This process follows three key steps:

1) Maximizing Variance: PCA seeks to find directions in the data where variance is maximized. The first principal component is chosen to capture the highest variance, while subsequent components are chosen orthogonally, each maximizing the remaining variance. These directions form the principal axes of the datasets.

2) Covariance Matrix and Eigenvalue Decomposition: To achieve the goal of maximizing variance, the data must first be standardized to remove the influence of different feature scales. A covariance matrix is then constructed using the formula$C = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T$, where C is the covariance matrix, m is the number of samples, x_i represents the i-th sample, and μ is the mean vector of all samples. The covariance matrix is then decomposed into eigenvalues and eigenvectors, with the size of the eigenvalues reflecting the amount of variance explained by each principal component.

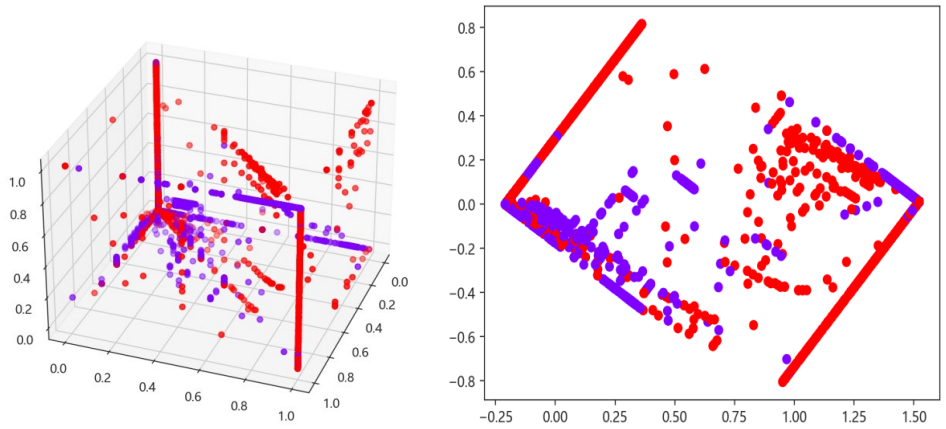
3) Dimensionality Reduction and Feature Selection: After obtaining the eigenvalues and eigenvectors, dimensionality reduction is achieved by selecting the eigenvectors associated with the

ISSN 2616-5775 Vol. 7, Issue 12: 44-50, DOI: 10.25236/AJCIS.2024.071206

largest eigenvalues. The data is then projected onto the subspace formed by these eigenvectors, significantly reducing dimensionality while retaining crucial information. This process helps uncover the intrinsic structure of the data and improves the efficiency of subsequent analysis or model training[4].

2.2.2 PCA Results Example

To demonstrate the effectiveness of PCA in dimensionality reduction, this paper first plots a 3D space consisting of the features srv_rerror_rate, rerror_rate, and dst_host_srv_rerror_rate, as shown in Figure 4(a). The corresponding 2D plane after applying PCA is illustrated in Figure 4(b). By comparing the two, it is observed that PCA performs well in reducing dimensionality while preserving the important information in the datasets.



(a)3D Feature Plot Before Dimensionality Reduction (b) 2D Feature Plot After PCA

Figure 4: Example of PCA Dimensionality Reduction

The NSL-KDD datasets contains 41 input features, making it a high-dimensional datasets. Some of these features may contain redundant or irrelevant information, leading to sparse and difficult-to-process data. High-dimensional data also increases the complexity of the model and its susceptibility to overfitting. Thus, PCA was applied in this study as a classic dimensionality reduction technique to preprocess the data.

3. Model Construction

3.1 Introduction to Model Principles

3.1.1 Logistic Regression

The fundamental principle of logistic regression is to estimate the likelihood of a sample belonging to a specific class by learning a linear combination of its features. The model assumes that the probability $P(y = 1 \mid x)$ can be expressed as:

$$P(y = 1 \mid x) = \frac{1}{1 + \exp(-w^T x)} \tag{1}$$

where w is the weight vector, x is the input feature vector, and $\exp(\cdot)$ represents the exponential function. This is commonly known as the sigmoid function, which maps the linear combination $w^T x$ to the range (0,1), giving the probability of the sample belonging to class 1.In logistic regression, the objective is to estimate the parameter w by maximizing the log-likelihood function. For a given set of training data $(x_i, y_i)\}_{i=1}^n$, the log-likelihood function is:

$$\ell(w) = \sum_{i=1}^n [y_i \log P(y_i = 1 \mid x_i) + (1 - y_i) \log(1 - P(y_i = 1 \mid x_i))] \tag{2}$$

Logistic regression maximizes this log-likelihood function in order to determine the optimal weight w, thereby effectively classifying new samples. Although logistic regression employs a linear model, its non-linearity comes from the sigmoid function, which allows it to handle nonlinear boundaries to some extent. However, it remains fundamentally a linear classifier, which limits its effectiveness in handling

ISSN 2616-5775 Vol. 7, Issue 12: 44-50, DOI: 10.25236/AJCIS.2024.071206

intricate datasets[5].

3.1.2 Decision Tree

A decision tree is a non-linear model that uses a hierarchical structure to classify data. It works by iteratively dividing the dataset into subsets until each subset is mostly homogeneous. Decision trees are effective for both classification and regression, leveraging feature values to split data and making predictions at leaf nodes. The main benefits of decision trees are their interpretability, ability to manage non-linear data, applicability to mixed data types, minimal need for preprocessing, and resilience to missing data and noise. However, decision trees often overfit, are sensitive to data variations, show bias toward features with more unique values, and can be computationally expensive to build and scale for larger datasets[6].

3.1.3 Random Oversampling

Random oversampling is a widely used technique for addressing imbalanced datasets by boosting the number of minority class samples in the training set, thus balancing the data distribution. This approach involves randomly replicating minority class instances to prevent the classifier from being biased towards the majority class, thereby enhancing its ability to correctly identify minority instances. Due to its simplicity and efficiency, random oversampling is a favored method for managing class imbalance. However, it can contribute to overfitting of the minority class and inflate the dataset size, leading to increased computational costs. Moreover, since it merely duplicates existing samples without adding new information, its effectiveness may be limited in certain scenarios[7].

3.2 Model Evaluation Metrics

Model performance is assessed using several important metrics, which include:

Accuracy: Represents the ratio of correct predictions (both positive and negative) among the total number of predictions, calculated as:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{3}$$

where TP (True Positives) denotes the count of correctly identified positive instances, TN (True Negatives) denotes correctly identified negative instances, FP (False Positives) refers to instances wrongly classified as positive, and FN (False Negatives) refers to instances wrongly classified as negative.

Precision: The ratio of true positive predictions to the total predicted positive cases, calculated as:

$$\mathrm{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4}$$

Recall: The ratio of correctly classified positive instances to the total actual positive instances, calculated as:

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

Balanced Accuracy: This metric accounts for imbalanced data by calculating the average recall for each class. It is a more effective measure than traditional accuracy when dealing with imbalanced datasets. Balanced accuracy is calculated as:

$$\text{Balanced Accuracy} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{Recall_i} \tag{6}$$

The term N represents the number of classes. Recall is defined as the proportion of true positive predictions within each class relative to the total actual positives. Balanced accuracy is calculated by averaging the recall across all classes, ensuring that the classifier's performance is not skewed by the predominance of a majority class, which may otherwise overshadow the classification of minority classes. This makes balanced accuracy an effective metric for evaluating classifier performance, particularly on imbalanced datasets.

ISSN 2616-5775 Vol. 7, Issue 12: 44-50, DOI: 10.25236/AJCIS.2024.071206

3.3 Model Hyperparameters

In this study, a combination of cross-validation and grid search was used to enhance the generalization ability of the model and to identify the optimal hyperparameter configuration. Specifically, a 5-fold cross-validation was employed, where the dataset was divided into five parts. During each iteration, four subsets were used for training and one for validation, and this process was repeated five times. This approach effectively mitigates overfitting and allows for assessing model performance across different data partitions.

Simultaneously, grid search was employed to comprehensively explore all potential parameter combinations within a predefined parameter space to find the optimal configuration for enhancing model performance. By combining these two techniques, the model's stability across different datasets was ensured, resulting in the best possible performance. Table 2 below presents the 5-fold cross-validation accuracy for each model across the five iterations.

Table 2: The 5-fold cross-validation accuracy for each model across the five iterations

Model	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Logistic Regression	0.9005	0.8977	0.9001	0.9092	0.9044
Oversampling-Logistic Regression	0.9023	0.9012	0.9034	0.9087	0.9076
Decision Tree	0.8901	0.8923	0.8912	0.8898	0.8909
Oversampling-Decision Tree	0.8804	0.8812	0.8834	0.8843	0.8856

Through the 5-fold cross-validation and grid search, the optimal hyperparameters for each model were determined, as shown in Table 3. Other parameters not listed here follow the default settings of the respective models.

Table 3: Optimal Hyperparameters for Each Model

Model	Hyperparameter
Logistic regression	C=1.14
Oversampling Logistic Regression	C=0.05
Decision Tree	criterion=entropy
Oversampling Decision Tree	criterion=entropy

Table 3 shows that the Decision Tree model outperforms the Logistic Regression model, highlighting the advantage of non-linear classification models over linear ones. This indicates that the relationship between the input features and predicted classes in this study is inherently non-linear. Furthermore, applying random oversampling to the same model enhances its balanced accuracy, demonstrating the effectiveness of this method in addressing data imbalance.

By comparing the results of the four algorithms, the Oversampling Decision Tree model is identified as the optimal one. Thus, this model is selected as the final classification model. The confusion matrix for this model is presented in Figure 5, with the classification report detailed in Table 4.

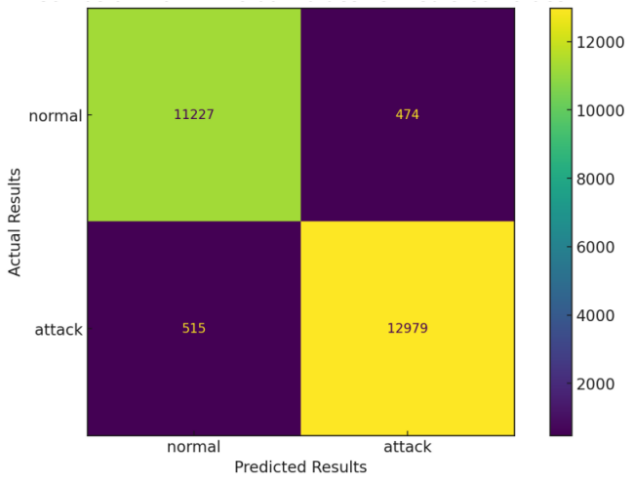


Figure 5: Confusion Matrix: Actual Values vs Predicted Values

ISSN 2616-5775 Vol. 7, Issue 12: 44-50, DOI: 10.25236/AJCIS.2024.071206

Table 4: Classification Report for the Oversampling Decision Tree Model

	precision	recall	f1-score	support
attack	0.9561	0.9595	0.9578	11701
normal	0.9648	0.9618	0.9633	13494
accuracy			0.9607	25195
macro avg	0.9604	0.9607	0.9606	25195
weighted avg	0.9608	0.9607	0.9608	25195

The results from the Oversampling Decision Tree model show minimal accuracy differences between the Attack and Normal classes, demonstrating the effectiveness of random oversampling in achieving balanced performance.

4. Conclusion

This study systematically investigated how data preprocessing and algorithm optimization can improve Intrusion Detection Systems (IDS) performance against evolving network threats. By applying outlier processing (RobustScaler) and dimensionality reduction (PCA) to the NSL-KDD dataset, the study ensured model stability and efficiency. Logistic Regression and Decision Tree models were compared, with the Decision Tree showing superior performance, especially when combined with random oversampling to address class imbalance. The oversampling Decision Tree model was identified as the optimal approach, offering enhanced balanced accuracy and robust performance.

As network attack methods continue to evolve, IDS must improve both detection accuracy and real-time responsiveness. Future research should explore advanced deep learning methods, such as Generative Adversarial Networks (GANs) and adaptive neural networks, to further enhance model generalization and threat detection capabilities. Additionally, integrating real-time big data analytics and automated response systems will be key to detecting and mitigating emerging threats. Addressing persistent issues like data imbalance and model overfitting will be crucial to developing more intelligent and efficient network security defenses.

References

[1] Zhou Qi. Research on Intrusion Detection Model Based on Feature Selection and Machine Learning Algorithms [D]. Gannan Normal University, 2023.
[2] Shang Jiaqi. Research on Intrusion Detection Models for Imbalanced Data [D]. Liaoning: Liaoning Technical University, 2023.
[3] Dong Yifan. Study on the Robustness of Intrusion Detection Methods in Adversarial Environments [D]. National University of Defense Technology, 2018.
[4] Li Yinghao. Research on PCA-based Feature Subset Selection and Network Intrusion Detection Algorithm [D]. Tianjin: Tianjin University of Technology, 2021.
[5] Ma Zehui. Research on Webshell Detection Method Based on Logistic Regression Algorithm [J]. Information Security Research, 2019, 5(4):298-302.
[6] Xu Ying, Qu Danqiu. Design and Implementation of a Computer Network Intrusion Detection System Based on Decision Tree Classification Algorithm [J]. Information Recording Materials, 2024, 25(6): 137-139.
[7] Jian Shijie. Research on Intrusion Detection Method Based on Oversampling and Traffic Data Dimensionality Reduction [D]. University of Chinese Academy of Sciences, 2021.