

Intrusion Detection System with Supervised Learning Models

Jihan Zheng

University of Southern California Los Angeles, United States

tylerzhe@usc.edu

Abstract. Intrusion Detection Systems (IDSs) can analyze and detect abnormal network activity, which addresses potential attacks based on studying and analyzing past attacks. This paper uses four supervised machine learning methods, which are logistic regression, decision tree, support vector machine, and random forest, to detect these abnormal attacks. The dataset used in this paper is from KDDCUP'99, a publicly available dataset for network-based anomaly detection systems. Certain features and outcomes are first extracted from the dataset. The values in nominal features are converted into dummy variables, and the values in outcome are changed to either normal or attack. Then the training processes are performed with the four algorithms, and the models are tested to get the accuracy scores. According to the results, the logistic regression model has the highest accuracy score of 0.9415, and the other three models all have accuracy scores above 0.90. The accuracy scores of the decision tree, support vector machine, and random forest are 0.9317, 0.9374, and 0.9202, respectively. Our models turn out to be efficient in identifying the network anomaly with provided data.

Keywords: Intrusion Detection Systems (IDSs); network anomaly detection; machine learning algorithms; accuracy score; confusion matrix.

1. Introduction

Network security has become more important nowadays with the rapid development and wide usage of the Internet in daily lives. The number of computer network usage and applications has increased drastically over the past years and continues to grow. However, computer systems always suffer from security vulnerabilities due to technical limitations. If not handled properly, network attacks may lead to severe consequences. For example, in December 2011, the malware attacked Iran's oil ministry and successfully erased information on their hard disks [1]. Therefore, Intrusion Detection Systems (IDSs) have recently been introduced and become prevalent.

IDSs are software applications or devices that detect malicious computer network activities [2]. There are many types of IDSs, and among them, network intrusion detection systems (NIDS) and host intrusion detection systems (HIDS) are mainly used. Specifically, NIDS examines the traffic on the network from all devices, and HIDS analyzes the network traffic on a single device [3]. Generally, IDSs are studied to detect novel attacks in academic research [4]. By analyzing the features of past attacks, machine learning algorithms could be applied to discover possible hidden patterns among the networking attacks. The model could be used to test future network activities and predict whether it could be an attack. If the model has high accuracy, it could be considered an Intrusion Detection System, and the computer could prevent the attack. However, no commercial tools currently use IDS to detect network anomalies because experts still consider the technology immature [4]. Researchers have focused on anomaly detection and studied methods of learning and detection, training and test datasets, and evaluation.

This paper uses four supervised machine learning algorithms, including logistic regression, decision tree, support vector machine, and random forest, to detect the common features among the network activity and make a prediction model. The dataset we used is a publicly available network anomaly dataset from KDDCUP'99. It contains records of network activity and its different features, as well as whether the activity is normal or not. Certain features are examined from the dataset and split into train and test sets. Then we train the model and make predictions for the test dataset.

Each model's accuracy scores and confusion matrixes are generated and used to find the optimal model. From the results, logistic regression has the highest accuracy score of 0.9415, which means it

correctly predicts 94.15% of the activity from the test dataset. The accuracy scores of the decision tree, support vector machine, and random forest are 0.9317, 0.9374, and 0.9202, respectively. All of the models have relatively high prediction accuracy, meaning the features we used are influential in shaping the model.

2. The Proposed Method

This section describes the proposed method to detect abnormal attacks. The process consists of two parts: preparation of the dataset (Sec. A) and constructing the four machine learning models (Sec. B).

2.1. Dataset Pre-processing

We use the KDDCUP'99 dataset [4], a publicly available dataset for network anomaly detection. The original network activity dataset consists of 3 nominal features, 6 binary features, 32 numeric features, and one outcome column describing the type of the activity. To ensure the integrity and ease of the experiment, we only extracted the three nominal features and the outcome to perform the training, and we first checked for any missing values. Then we convert the nominal features into dummy variables for training. As for the outcome column, since we are focusing on binomial classification for the dataset, we make all the instances that are not 'normal' to be 'attacks.' Finally, we set normal to be 0 and attack to be 1 for the training.

2.1.1. Preparation of Feature Data

We first select the three nominal features from the original dataset: 'protocol_type,' 'service,' 'flag,' and 'outcome.' Then we check for any missing values for each column, and there are no missing values for all. By examining the dataset, there are 3 distinct values in protocol_type, 70 different values in service, and 11 distinct values in flag. As for the outcome, other than normal, there are 40 other abnormal types of activities.

After that, we separated the dataset into x and y . In the subsets, x contains the features and y contains the outcome. We transformed the x into dummy variables (0 or 1) for training. After converting the values, x is a table with 84 columns, and y is only one column. The first five rows and the first five columns of x are shown in Table I.

Table 1. The first five rows and columns of x

protocol_type_icmp	protocol_type_tcp	protocol_type_udp	service_IRC	service_X11
0	1	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	0	0	0
0	1	0	0	0

2.1.2. Preparation of Outcome Data

Since we are only detecting whether the activity is normal, we changed the values in y that are not 'normal' to be attacks at first, and then we converted 'normal' to be 0 and 'attack' to be 1 for the training.

2.2. Model Construction

We perform four supervised learning algorithms on our dataset. We first divide the dataset into train and test, then we develop models using the train data and test the models' accuracy using the test data. We also plot the confusion matrixes of each model for the test data to visualize the accuracy of the prediction.

3. Experiment Settings

This section describes the experimental settings in the experiment. We first described the extracted dataset used in our experiment (Sec. A). Then we discuss the baselines that are used in the experiment (Sec. B). Finally, we use the accuracy score and confusion matrix to compare and evaluate the models (Sec. C).

3.1. Dataset

The dataset is from KDDCUP'99 [4] and is publicly available on Kaggle. It consists of 148,517 instances of network activity and their 41 different features. We only need the three nominal features and the outcome, so we extract these four columns. The first five rows of the extracted dataset are shown in Table II.

The three nominal features are `protocol_type`, `service`, and `flag`. `Protocol_type` tells which protocol is employed during the connection; `service` identifies the utilized network service and its destination; and `flag` shows whether the connection is normal or error. Other than 'normal,' all different types could be considered attacks.

Table 2. The First Five rows of extracted data

protocol_type	service	flag	outcome
tcp	ftp_data	SF	normal
udp	other	SF	normal
tcp	private	S0	neptune
tcp	http	SF	normal
tcp	http	SF	normal

3.2. Baselines

We use four supervised machine learning algorithms as our baseline models.

3.2.1. Logistic Regression

Logistic regression is frequently used in binary classification [5]. It predicts whether one event will happen or not based on its particular parameters [6]. In our experiment, logistic regression classifies the network activity into either normal or attack.

3.2.2. Decision Tree

Decision tree is an analysis technique uses known probabilities from prior events to assess the likelihood that an event's expected value will be larger or equal to zero [7].

The decision tree model analyzes feature importance, which measures the system's reduction of mean squared error by a particular feature and its splitting value [8]. It helps us determine which features are majorly used in categorizing the instance.

3.2.3. Support Vector Machine

Support vector machine is commonly used to solve regression and classification problems. The algorithm can categorize the data points into groups by locating a hyperplane in the space [9].

3.2.4. Random Forest

Random forest is a widely-used algorithm that essentially integrates the outcomes of numerous decision trees to produce a singular result. It's frequently utilized to solve classification and regression problems [10].

3.3. Evaluation Metrics

We use accuracy scores and confusion matrixes of the models to evaluate the performances of the four algorithms.

3.3.1. Accuracy Score

Scikit-learn provides an accuracy score that helps to compute subset accuracy, which is the percentage of the correctly predicted labels [11].

3.3.2. Confusion Matrix

In binary classification, a confusion matrix categorizes the prediction results into four groups: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). It is true positive if the event occurs and the prediction is accurate; true negative if the event does not occur and the prediction is accurate; false positive if the event does not occur and the prediction is inaccurate; and true negative if the event occurs and the prediction is inaccurate [12].

Consequently, the accuracy score in this paper is determined using the formula below:

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

4. Results and Discussion

This section provides the results of the accuracy scores (Sec. A) and confusion matrixes (Sec. B) of each model. Then in the decision tree model, we further investigate the feature importance and the number of nodes (Sec. C).

4.1. Accuracy Scores

The accuracy scores of all four models are shown in Table III. From the accuracy scores, logistic regression outperforms other models with an accuracy score of 0.941489. All four models give relatively high accuracy scores (greater than 0.90), which shows that the models are most accurate in determining the abnormal network activity.

Table 3. Accuracy Scores of All Models

Model	Accuracy Score
Logistic Regression	0.941489
Decision Tree	0.931659
Support Vector Machine	0.937449
Random Forest	0.908474

4.2. Confusion Matrix

Using the built-in functions, the confusion matrixes of each model are plotted and shown in the figures below.

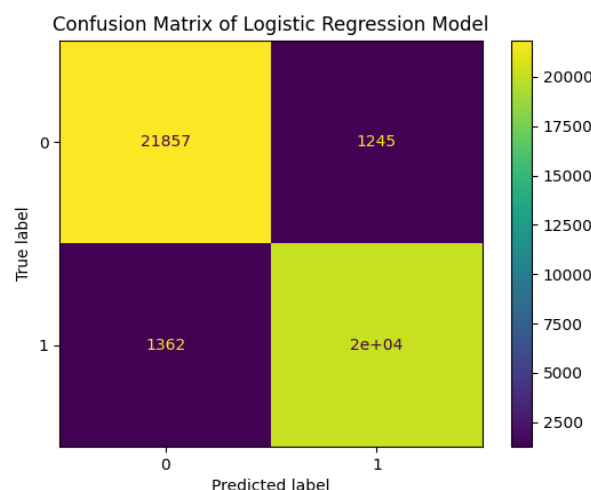


Figure 1. Logistic Regression Confusion Matrix

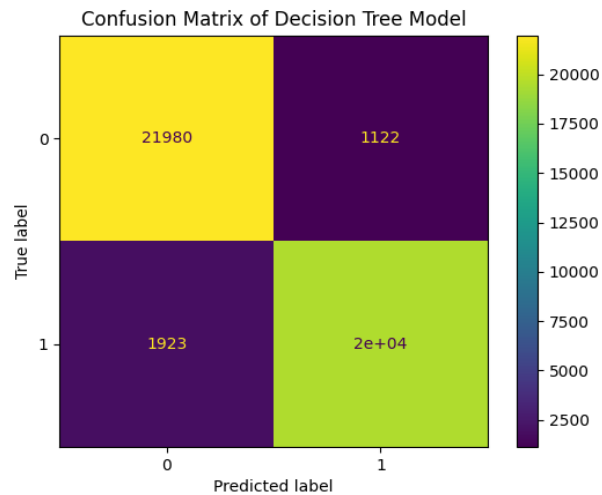


Figure 2. Decision Tree Confusion Matrix

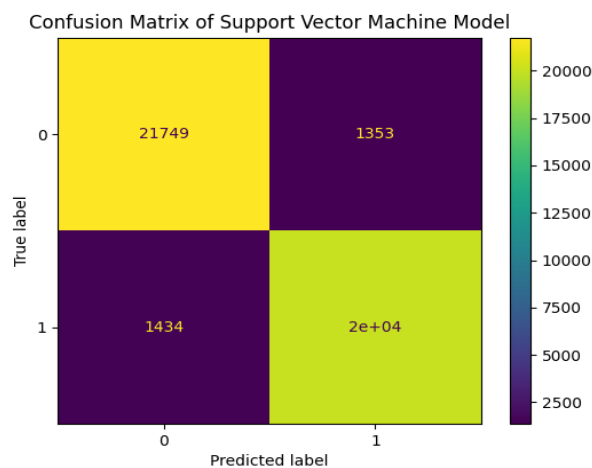


Figure 3. Support Vector Machine Confusion Matrix

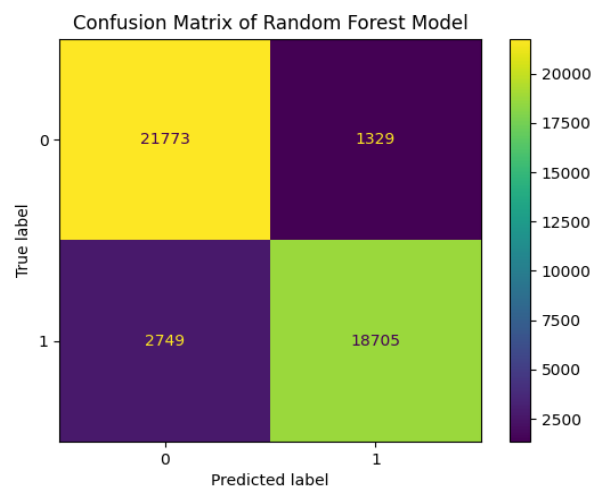


Figure 4. Random Forest Confusion Matrix

The confusion matrixes agree with the accuracy scores in Section A. The values of the true positive and true negative are the lowest from the random forest model, which explains its lowest accuracy score among the four.

Despite the fact that the decision tree model's false negative value (21,980) is marginally higher than the logistic regression model's (21,857), the value of false negative from the logistic regression model (1,362) is much smaller than that of decision tree model (1,923), which explains why logistic regression has a higher overall accuracy score than decision tree.

4.3. Decision Tree Criteria and Results

The feature importance, as well as the criteria of our decision tree model, are investigated.

4.3.1. Feature Importance

The three features in the decision tree model with the highest feature importance are located using the built-in function. They are shown in Table IV.

Table 4. Top Three Features in Decision Tree Model

Feature Name	Feature Importance
flag_SF	0.6309
protocol_type_icmp	0.1624
service_http	0.0986

Based on the feature importance, we learn that the most crucial feature is flag_SF with an importance of 0.6309, which means whether the flag feature (status of the connection) of network activity is SF or not contributes the most in determining the anomaly network activity. Then the other two important features are protocol_type_icmp and service_http. Combining three of them, they have total feature importance of 0.8919. We can identify which features to focus on in the future dataset and prediction by extracting the feature importance.

4.3.2 Decision Tree Model Criteria

We plot the decision tree of our model and show it in Fig .5 below. From Fig. 5, we can see that it corresponds with the results from the feature importance. The first node is based on flag_SF. When flag_SF is 0, meaning the flag of the activity is not SF, it will be classified as No (not attack); vice versa, when flag_SF is 1, the activity will be classified as Yes (attack).

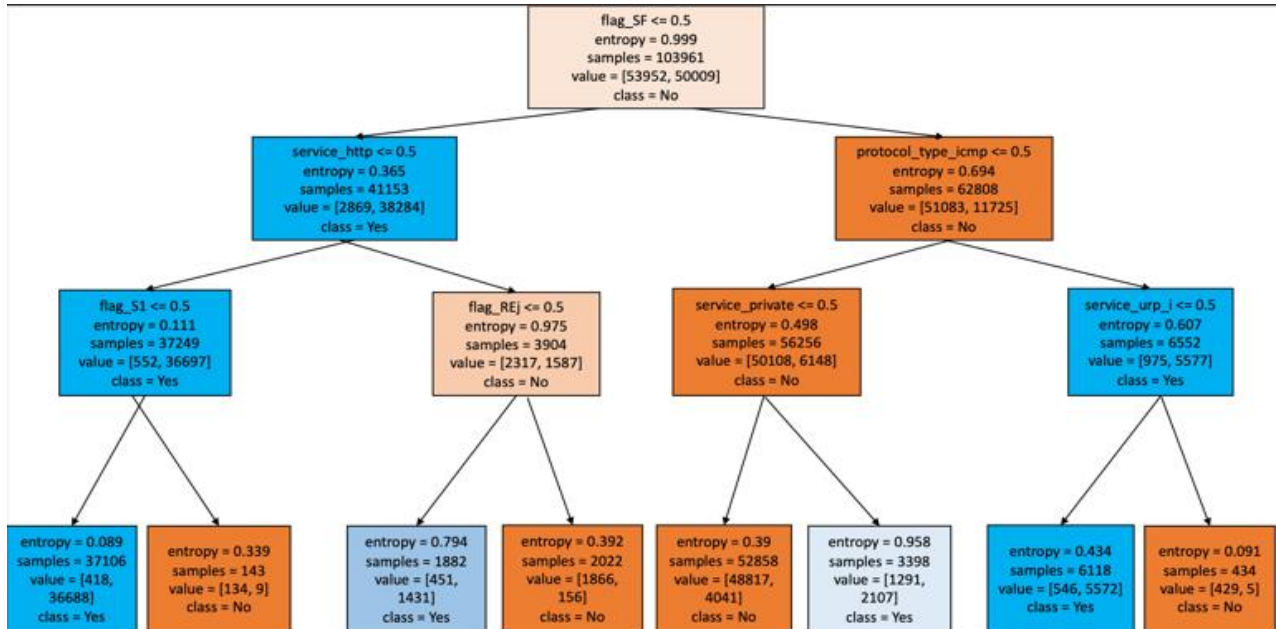


Figure 5. Decision Tree Plot

In our decision tree model, we set the max_depth in our function to be 3 for easy code operation. In the decision tree, max_depth determines the number of nodes the tree will use in the classifying process [13]. If it is not set, the nodes will expand until all leaves are pure.

Our decision tree model has an accuracy score of 0.931659, which is relatively high in machine learning prediction. However, we can try to further increase the accuracy score by adjusting the max_depth of the model in future experiments.

5. Conclusion

This paper focuses on building and selecting the optimal model that could help detect potential network anomalies based on their features. Our experimental results suggest that the logistic regression model has the highest accuracy in identifying abnormal activities, while all the other models are relatively accurate. The results indicate that it is highly feasible to build an IDS with machine learning algorithms by studying the features of the network anomaly.

In the future, the models can be further improved with more data and columns of features. Furthermore, the algorithms could be mixed and integrated to build more accurate and complicated models.

References

- [1] Erdbrink, T. (2012). Iranian Oil Sites Go Offline Amid Cyberattack. The New York Times.
- [2] What is an Intrusion Detection System (IDS)?. Check Point. <https://www.checkpoint.com/cyber-hub/network-security/what-is-an-intrusion-detection-system-ids/>
- [3] Hoffman, J. Different Types of Intrusion Detection System (IDS). WisdomPlexus. <https://wisdomplexus.com/blogs/different-types-of-intrusion-detection-systems-ids/>
- [4] Anush. Network Anomaly Detection. Kaggle. <https://www.kaggle.com/datasets/anushonkar/network-anamoly-detection?resource=download>
- [5] Allison, P. D. (1999) Logistic regression using the sas system: theory and application. SAS Publishing.
- [6] Tolles, Juliana; Meurer, William J (2016). "Logistic Regression Relating Patient Characteristics to Outcomes". JAMA. 316 (5): 533–4. doi:10.1001/jama.2016.7653. ISSN 0098-7484. OCLC 6823603312. PMID 27483067.
- [7] Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, V. Vapnik. (1995) Comparison of learning algorithms for handwritten digit recognition. In: International Conference on Artificial Neural Networks. Paris.
- [8] Bujokas, E. (2022) Feature Importance in Decision Trees. Towards Data Science. <https://towardsdatascience.com/feature-importance-in-decision-trees-e9450120b445>
- [9] Gandhi, R (2018). Support Vector Machine — Introduction to Machine Learning Algorithms. <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
- [10] IBM Cloud Education. Random Forest. IBM. <https://www.ibm.com/cloud/learn/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems>
- [11] Scikit-learn. Accuracy Score. Sklearn. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html
- [12] Scikit-learn. Confusion Matrix. Sklearn. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
- [13] Scikit-learn. Decision Tree Classifier. <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>