Comparative Analysis of ML Classifiers for Network Intrusion Detection

Ahmed M. Mahfouz*, Deepak Venugopal, and Sajjan G. Shiva

The University of Memphis, Memphis TN 38152, USA
{amahfouz*, dvngopal, sshiva}@memphis.edu

Abstract. With the rapid growth in network-based applications, new risks arise, and different security mechanisms need additional attention to improve speed and accuracy. Although many new security tools have been developed, the fast-growth of malicious activities continues to be a severe issue, and the ever-evolving attacks create serious threats to network security. Network administrators rely heavily on Intrusion Detection Systems to detect such network intrusive activities. Machine learning methods are one of the predominant approaches to intrusion detection, where we learn models from data to differentiate between abnormal and normal traffic. Though Machine learning approaches are used frequently, a deep analysis of Machine learning algorithms in the context of intrusion detection is somewhat lacking. In this work, we present a comprehensive analysis of some existing Machine Learning classifiers regarding identifying intrusions in network traffic. Specifically, we analyze classifiers along various dimensions, namely, feature selection, sensitivity to hyper-parameter selection and class imbalance problems that are inherent to intrusion detection. We evaluate several classifiers using the NSL-KDD dataset and summarize their effectiveness using a detailed experimental evaluation.

Keywords: IDS, Machine Learning, Classification Algorithms, NSL-KDD Dataset, Network Intrusion Detection, Data Mining, Feature Selection, WEKA, Hyperparameters, Hyperparameter Optimization.

1 Introduction

Because of the massive volume of data on the network; the content of the network becomes vulnerable to a variety of attacks, and different intrusions are increasing day after day. Detecting intrusions is an essential step to stopping the intruders from breaking into or misusing network data. To defend against numerous network intrusions and malicious activities, many methods have been developed. Network Intrusion Detection is considered one of the most promising methods to protect the network from different dynamic intrusion behaviors. Intrusion Detection System differentiates intrusive behaviors from normal network activities by classifying data into various categories [1].

Several machine learning (ML) methods have been proposed to develop effective and intelligent Intrusion Detection [2]. However, there have been very few systematic studies that evaluate ML approaches for intrusion detection. Specifically, ML classifiers are typically used as a “black-box’’, where reported results may be obtained as a result of over-fitting the model for a specific dataset [3]. Thus, the results are not very generalizable and hard-to-replicate. Our main contribution in this paper is to provide a detailed experimental evaluation of a group of supervised ML methods in the context of intrusion detection. Specifically, we evaluate several well-known ML methods for intrusion detection along the following dimensions: Feature selection, sensitivity to hyperparameter tuning and effect all of the class imbalance. All these three dimensions are critical to effective use of ML methods in intrusion detection. Specifically, feature selection chooses the optimal subset of features to avoid building a complex classifier that may over-fit the data. A large hyperparameter sensitivity indicates that tuning the detection system optimally may be difficult for other datasets, and algorithms that handle the class imbalance problem more effectively are more viable in practice for intrusion detection.

The rest of the paper is organized as follows; we provide the background in section 2 where we talk about intrusion detection systems, machine learning and introduce one of the most popular networks traffic datasets. In section 2 also, we discuss the feature selection concept as well as the concept of hyperparameter optimization. The performance evaluation metrics are discussed in section 3. Related work is produced in section 4, and the experimental results are reported in section 5. Finally, in Section 6, we present the conclusion and mention the future work.

2 Background

2.1 Intrusion Detection System

An intrusion is a malicious activity that aims to compromise the confidentiality, the integrity, or the availability of any of the network components as an attempt to disrupt the security policy of the network [1].

Based on the analysis strategies and the detection methods, IDSs are categorized into Misuse Intrusion Detection systems (MID) and Anomaly Intrusion Detection systems (AID). Based on the data source, IDSs are categorized into network-based (NIDS) and host-based (HIDS) detection systems [4], [5].

2.2 Machine Learning Classifiers

Several ML methods have been proposed to monitor and analyze network traffic for different anomalies. Most of these methods (classifiers) identify the anomaly by looking for variations from a basic normal traffic model. Usually, these models are trained with a set of attack-free traffic data that is collected over a long period. Any ML anomaly detection method is one of three broad categories that are Supervised, Unsupervised or Semi-supervised learning method. In this paper, we will focus on the supervised learning classifiers.

Supervised learning is the type of models that take both of input variable (X) and output variable (Y) to provide a learning basis to support future judgments by learning the mapping function Y = f(X). Supervised learning uses a training data which is a set of examples with paired input records and their desired outputs. In this learning the correct answer is known in advance, and the learner algorithm iteratively makes predictions on the training data and stops only when an acceptable level of performance is achieved. Thus, this method is appropriate when there is a specific target value. Supervised learning problem can be defined as either a classification problem or a regression problem. The output variable of the classification problem is a category, like "white" or "black" and "disease" or "no disease." On the other hand, the output variable of the regression problem is a real value, such as "the number of dollars" or "the height" [6]. The most famous supervised learning algorithms are Decision Trees (DT), Support Vector Machines (SVM), Artificial Neural Network (ANN), K-Nearest Neighbors (KNN), Logistic Regression (LR), Random Forests (RF), Naive Bayes (NB), etc.

2.3 Datasets

To support the assessment of different intrusion detection methods, researchers have introduced several network traffic datasets. These datasets are either public, private, or network simulation dataset. Most of these datasets were generated using several tools that helped in capturing the traffic, launching different types of attacks, and monitoring traffic patterns. In this paper, we use NSL-KDD dataset which is one of the most popular benchmark datasets in the domain of intrusion detection.

NSL-KDD. The NSL-KDD dataset [7] is a refined offline version of the well-known KDDcup99 dataset. Many researchers have carried out different types of analysis on the NSL-KDD and have employed different methods and tools to develop effective IDSs [8]. The NSL-KDD dataset has 41 attributes plus one class attribute. A full description of these attributes can be found in [9]. We present a statistical summary of the NSL-KDD dataset in section V.

2.4 Feature Selection

Feature selection is the process of selecting a subset of the original features so that the feature space is optimally reduced to the evaluation criterion [10]. A feature selection method selects a subset of relevant features. The relevance definition varies from technique to another. Based on its notion of relevance, a feature selection technique mathematically formulates a criterion for evaluating a set of features generated by a scheme that searches over the feature space. Kohavi et al. [11] define two degrees of relevance, strong and weak. A feature s is strongly relevant if removal of s deteriorates the performance of a classifier. A feature s is called weakly relevant if it is not strongly relevant and removal of a subset of features containing s deteriorates the performance of the classifier. A feature is irrelevant if it is neither strongly nor weakly relevant.

2.5 Hyperparameters

Parameters vs. hyperparameters. Model parameters are the set of configuration variables that are internal to the model and can be learned from the historical training data. The value of those parameters is estimated from the input data. Model hyperparameters are the set of configuration variables that are external to the model. They are the properties that govern the entire training process and cannot be directly trained from the input data. The model parameters specify how input data is transformed into the desired output, while, the hyperparameters define the structure of the model.

Hyperparameter optimization. Hyperparameter optimization (also known as hyperparameter tuning) is the process of finding the most optimal hyperparameters for the learning algorithm in ML. A set of different measures for a single ML model can be used to generalize different data patterns. This set is known as the hyperparameters set which should be optimized such that the ML model can solve the assigned problem as optimally as possible. The optimization process locates the hyperparameters tuple then produces a model that minimizes the predefined loss function on the given data. The objective function takes the hyperparameters tuple and returns the associated loss. The generalization performance is often estimated using Cross-validation [12]. Hyperparameter optimization techniques mostly use any one of optimization algorithms Grid Search, Random Search or Bayesian Optimization.

3 Performance Evaluation

Various performance measurements have been proposed in the literature. Following are the most popularly used parameters in evaluating an ML model performance that can be used in ML-based IDS:

3.1 Confusion Matrix.

The efficiency of an ML model is usually determined by metrics called sensitivity and specificity measure. The sensitivity is referred to as the true positive rate (TPR), while specificity is known as a true negative rate (TNR). However, there is often a trade-off between these metrics in "real world" applications.

$$Sensitivity = \frac{(TP)}{(TP + FN)} \tag{1}$$

$$Specificity = \frac{(TN)}{(TN + FP)} \tag{2}$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{3}$$

3.2 Precision, Recall, and F-measure

Precision and recall are two recognized evaluation metrics in the information retrieval area. Precision refers to the portion of the relevant instances among the retrieved instances. Recall refers to the portion of relevant retrieved instances from the total number of the relevant instances. F-measure is the precision and recall harmonic mean.

$$Precision = \frac{TP}{(TP + FP)} \tag{4}$$

$$Recall = \frac{(TP)}{(TP + FN)} \tag{5}$$

$$F - measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \tag{6}$$

3.3 Receiver Operating Characteristic (ROC)

The use of ROC curves is a well-known evaluation measure that visualizes the relation between True Positive (TP) and False Positive (FP) rates of IDSs. It is also used to compare two or more ML classifiers regarding accuracy effectively.

3.4 K-Fold Cross-Validation

One of the most famous statistical methods in evaluating and comparing ML models is K-Fold Cross-Validation. It works by first separating the dataset into K equally sized folds (instances). K-1 folds are used to train the model, and the last one is left out for prepared model testing. The procedure is then reiterated so that every fold gets the chance to act as the test dataset. Finally, the capability of the model on the problem is estimated by averaging the performance measures across all folds. The K folds number is decided based on the size of the dataset, but the most used numbers are 3, 5, 7 and 10. The goal is to choose a number that makes a good balance between the size and representation of data in your train and test sets.

4 Related Work

G. Meera Gandhi [13], used DARPA-Lincoln dataset to evaluate and compare the performance of four supervised ML classifiers regarding detecting the four categories; DoS, R2L, Probe, and U2R attacks. Their results show that the J48 classifier outperforms the other three classifiers IBK, MLP, and NB in prediction accuracy. In [14], Nguyen et al. performed an empirical study to evaluate a comprehensive set of ML classifiers on the KDD'99 dataset to detect attacks from the four attack classes. Abdeljalil et al. [15] tested the performance of three ML classifiers namely J48, NN and SVM using the KDD'99 dataset and found that the J48 algorithm outperformed the other two algorithms.

Dhanabal, L et al. [9] analyzed and used the NSL-KDD dataset to measure the effectiveness of ML classifiers in detecting the anomalies in the network traffic patterns. In their experiment, 20% of the NSL-KDD dataset has been used to compare the accuracy of three classifiers. Their results show that with CFS is being used for dimensionality reduction, J48 outperforms SVM and NB regarding accuracy. In [18] Belavagi et al. tried to check the performance of four supervised ML classifiers namely SVM, RF, LR and NB for an intrusion detection over the NSL-KDD dataset. From the results, it was found that the RF classifier a 99% accuracy.

Unlike the above studies, our paper concentrates on evaluating and comparing the performance of a group of well-known, supervised ML classifiers over the full NSL-KDD dataset for intrusion detection along the following dimensions: Feature selection, sensitivity to hyperparameter tuning and class imbalance.

5 Experimental Results

In this section, we present the experimental setup and the results of comparing six ML classifiers regarding classification accuracy, TPR, FPR, precision, recall, f-measure, and ROC area. We selected the six classifiers from various classifier families and applied them to NSL-KDD dataset. The selected classifiers are Naïve Bayes, Logistic, MultilayerPerceptron (NN), SMO (SVM), IBK (KNN) and J48 (DT).

5.1 Statistical Summary of NSL-KDD

Each record in the NSL-KDD dataset unfolds different features of the traffic with 41 attributes plus an assigned label classifying each record as either normal or attack. The features of the dataset are three types: Nominal, Numeric, and Binary. The nominal features are 2, 3, and 4, while the binary features are 7, 12, 14, 15, 21, and 22, and the rest of the features are a numeric type. Authors in [9] listed the details of those attributes that are the attributes names, description, and sample data.

Attack types in the dataset can be grouped into four main classes namely DoS, U2R, Probe, and R2L [17]. Table 1. maps different attack types with its attack class while Table 2. shows the number of occurrences for normal and different attack classes.

Table 1. NSL-KDD attack types and classes.

Attack Class	Attack Type	Sample Relevant Feature	Example
DoS	Apache2, Back, Pod, Process table, Worm, Neptune, Smurf, Land, Udpstorm, Teardrop	percentage of packets with errors - source bytes	Syn flooding
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint	source bytes - duration of the connection	Port scanning
R2L	Httptunnel, Snmpgetattack, Snmpguess, Guess_Password, Imap, Warezclient, Ftp_write, Phf, Multihop, Warezmaster, Spy, Xsnoop, Xlock, Sendmail	number of shell prompts invoked - the number of file creations	Buffer overflow
U2R	Buffer_overflow, Xterm, SQL attack, Perl, Loadmodule, Loadmodule, Ps, Rootkit	Service requested – connection duration – num of failed login attempts	password guessing

Table 2. shows that the number of attack records associated with the R2L and U2R attack classes in the dataset is very low compared to the normal and other attack clas-

ses, which leads to the imbalanced problem. Classification process for any imbalanced dataset is always a challenging issue for researchers. Most standard ML and data mining methods consider balanced datasets. When the methods are used with an imbalanced dataset, they produce biased results toward the samples from the majority classes. The classification accuracy for the majority classes is much higher than for the minority classes [18].

Table 2. No of samples for normal and attack classes.

Class	Training Set	Occurrences Percentage	Testing Set	Occurrences Percentage
Normal	67343	53.46 %	9711	43.08 %
DoS	45927	36.46 %	7460	33.08 %
Probe	11656	9.25 %	2421	10.74 %
R2L	995	0.79 %	2885	12.22 %
U2R	52	0.04 %	67	0.89 %
Total	125973	100.0 %	22544	100.0 %

5.2 Experimental Setup

Our experimental setup went through three phases. In the first phase, we compared the performance of the classifiers with their default settings and without any preprocessing for the dataset. We trained the classifiers on the training dataset provided by NSL-KDD using Stratified Cross-Validation of 10-folds and used the trained models with the testing dataset. The testing datasets were also provided by NSL-KDD to compare the performance. The results of this phase are summarized in Tables 3 and 4 where Table 3 summarizes performance metrics for the trained models and Table 4 summarizes the same performance metrics for the trained models on the test dataset.

In the second phase, the NSL-KDD dataset was preprocessed to reduce its dimension by selecting the most relevant features. We applied the InfoGainAttributeEval algorithm with Ranker which ranked the attributes by their evaluation and resulted in selecting 14 out of 41 features suggested by NSL-KDD. The selected features are 3, 4, 5, 6, 9, 12, 14, 25, 26, 29, 30, 37, 38 and 39. We also used CVParameterSelection to perform the hyperparameter optimization for each classifier. Finally, we compared the performance as we did in the first phase. The results of this phase are summarized in Tables 5 and 6.

In the third phase, we worked on mitigating the dataset imbalance problem by under-sampling the dominant classes and over-sampling the minority classes. For under-sampling, we used WEKA's Resample filter which takes a random subsample. By setting the bias toward the uniform class to 1, we ensured that the output subsample was balanced. For oversampling, we used WEKA's SMOTE filter. SMOTE stands for Synthetic Minority Over-Sampling Technique. It works by generating synthetic instances based on the existing instances in the minority class to balance the data. The synthetic instances are generated by taking random points along a line between an existing minority instance and its nearest neighbors.

Table 3. Classifiers trained models accuracy metrics of phase 1.

Classifier	Accuracy	TPR	FPR	Precision	Recall	F-Measure	ROC Area
NB	80.20 %	0.841	0.087	0.890	0.841	0.852	0.968
Logistic	91.55 %	0.955	0.039	0.952	0.955	0.952	0.983
MLP	94.98 %	0.969	0.026	0.973	0.969	0.970	0.992
SMO	91.81 %	0.957	0.027	0.972	0.957	0.973	0.977
IBK	94.62 %	0.986	0.002	0.996	0.996	0.996	0.988
J48	94.74 %	0.987	0.002	0.997	0.997	0.997	0.987

Table 4. Classifiers trained models accuracy metrics on the test dataset of phase 1.

Classifier	Accuracy	TPR	FPR	Precision	Recall	F-Measure	ROC Area
NB	76.12 %	0.916	0.337	0.673	0.916	0.776	0.837
Logistic	75.60 %	0.928	0.380	0.649	0.928	0.763	0.653
MLP	77.60 %	0.929	0.393	0.642	0.929	0.759	0.886
SMO	75.39 %	0.926	0.393	0.641	0.926	0.758	0.625
IBK	79.35 %	0.927	0.353	0.665	0.927	0.775	0.802
J48	81.69 %	0.972	0.318	0.698	0.972	0.813	0.818

Table 5. Classifiers trained models accuracy metrics of phase 2.

Classifier	Accuracy	TPR	FPR	Precision	Recall	F-Measure	ROC Area
NB	90.41 %	0.898	0.083	0.947	0.898	0.922	0.957
Logistic	95.48 %	0.965	0.064	0.958	0.965	0.961	0.975
MLP	96.50 %	0.973	0.051	0.965	0.973	0.969	0.973
SMO	95.73 %	0.966	0.060	0.960	0.966	0.963	0.953
IBK	97.83 %	0.979	0.022	0.979	0.979	0.979	0.978
J48	97.89 %	0.979	0.022	0.979	0.979	0.979	0.979

Table 6. Classifiers trained models accuracy metrics on the test dataset of phase 2.

Classifier	Accuracy	TPR	FPR	Precision	Recall	F-Measure	ROC Area
NB	78.15 %	0.782	0.083	0.821	0.782	0.794	0.889
Logistic	81.51 %	0.815	0.142	0.851	0.815	0.832	0.889
MLP	78.15 %	0.782	0.173	0.818	0.782	0.799	0.889
SMO	79.83 %	0.798	0.161	0.832	0.798	0.814	0.856
IBK	84.35 %	0.824	0.134	0.860	0.824	0.841	0.838
J48	82.67 %	0.807	0.157	0.837	0.807	0.821	0.883

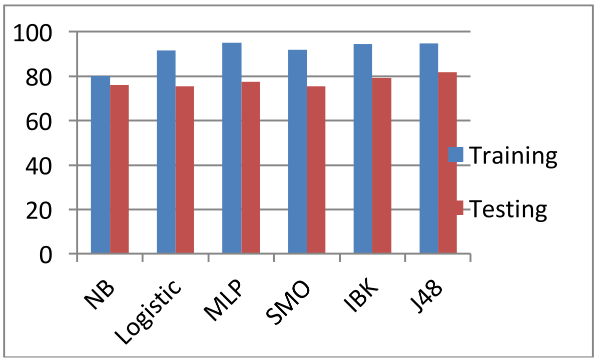


Fig. 1. Training Vs. Testing Accuracy of Phase 1.

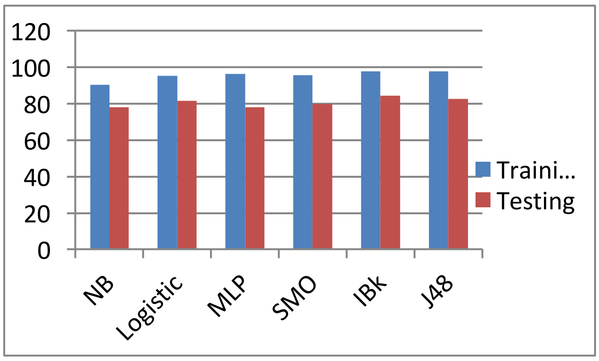


Fig. 2. Training Vs. Testing Accuracy of Phase 2.

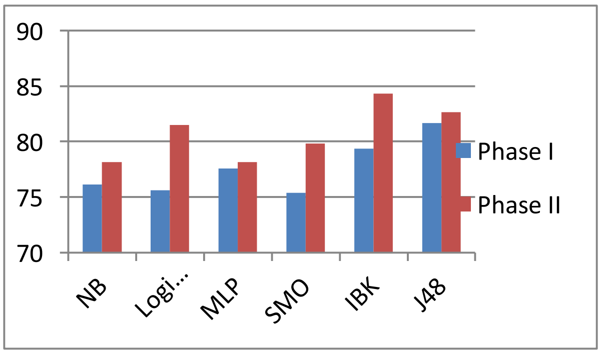
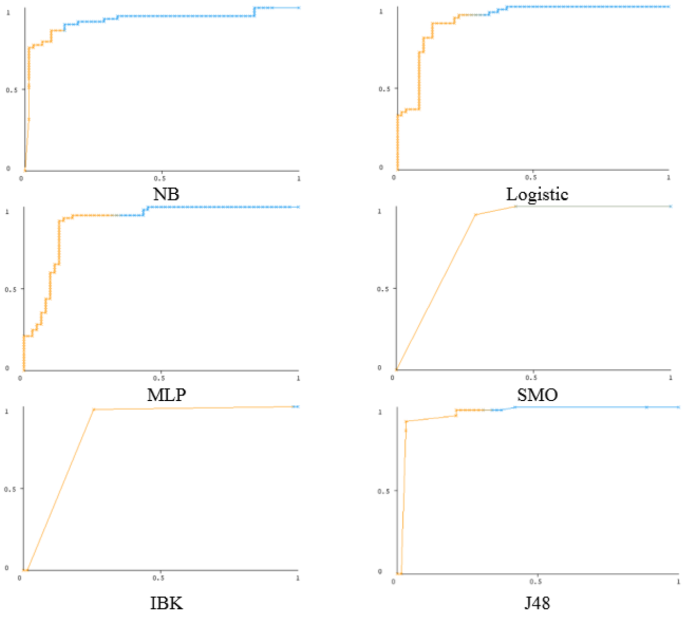


Fig. 3. Testing Accuracy of Phases 1 and 2.

All experiments have been carried out using WEKA [19], a data mining tool running on a PC with Intel(R) CORE(TM) i5-6600K CPU @ 3.50GHz, 3.50 GHz, 8 GB RAM installed and running a 64-bit Windows 10 OS, x64-based processor.



NB

Logistic

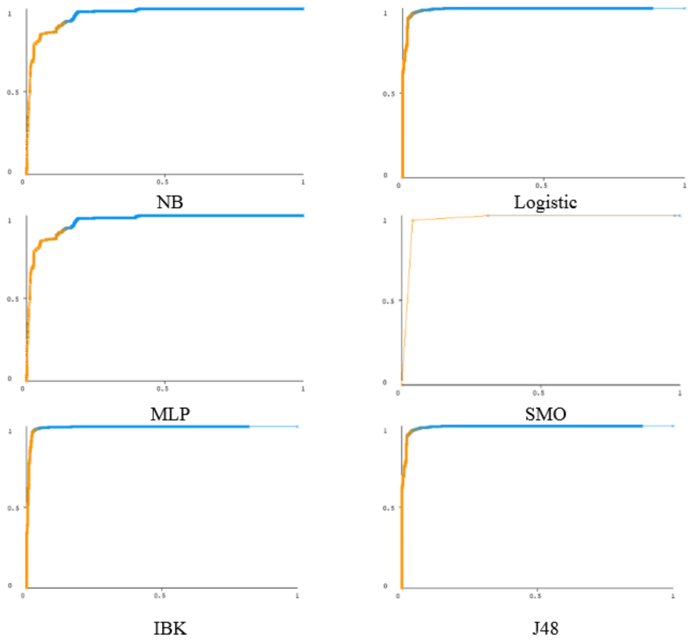
MLP

SMO

IBK

J48

Fig. 4. ROC Curves for Phase 1.



NB

Logistic

MLP

SMO

IBK

J48

Fig. 5. ROC Curves for Phase 2.

Table 8. Classifiers accuracy detection for different classes of attacks.

Classifier	Class	Phase I	Phase II	Phase III
NB	Normal	76.1 %	86.0 %	89.9 %
	DoS	75.2 %	83.8 %	91.8 %
	Probe	76.1 %	81.8 %	83.9 %
	R2L	10.1 %	26.7 %	39.0 %
	U2R	30.3 %	30.8 %	32.1%
Logistic	Normal	75.6 %	84.4 %	96.4 %
	DoS	74.9 %	90.7 %	96.7 %
	Probe	75.1 %	69.6 %	88.7 %
	R2L	00.0 %	00.0 %	26.2 %
	U2R	22.3 %	26.7 %	53.2 %
MLP	Normal	77.6 %	82.4 %	97.5 %
	DoS	80.5 %	86.3 %	97.4 %
	Probe	68.9 %	63.2 %	93.9 %
	R2L	00.0 %	00.0 %	60.6 %
	U2R	08.9 %	09.7 %	30.2 %
SMO	Normal	75.3 %	83.3 %	96.7 %
	DoS	74.7 %	91.9 %	97.5 %
	Probe	55.4 %	60.0 %	87.6 %
	R2L	00.0 %	00.0 %	02.7 %
	U2R	00.0 %	00.0 %	04.9 %
IBK	Normal	79.3 %	86.8 %	99.4 %
	DoS	80.5 %	90.7 %	99.5 %
	Probe	71.8 %	76.2 %	99.0 %
	R2L	00.0 %	00.0 %	53.2 %
	U2R	00.0 %	00.0 %	41.5 %
J48	Normal	81.6 %	84.8 %	99.5 %
	DoS	80.1 %	89.2 %	99.2 %
	Probe	67.9 %	63.2 %	91.6 %
	R2L	18.9 %	18.2 %	55.1 %
	U2R	00.0 %	00.0 %	39.3 %

5.3 Experimental Results

For the evaluation purpose of each of the classifiers, we considered Stratified Cross-Validation more important. The evaluation is performed using the training dataset, Stratified Cross-Validation of 10-fold and the testing dataset provided by NSL-KDD.

Figures 1, 2, 3, 4, and 5 summarize the performance of the tested classifiers according to Accuracy, and ROC Area in the first two experimental phases. Tables 3, 4, 6, and 7 present a comprehensive comparison of the classifiers regarding classification accuracy, Precision, Recall, TPR, FPR, F-Measure, and ROC Area. Table 8 shows the accuracy of each classifier in classifying different types of attacks.

5.4 Discussion

Our experimental results show that J48 outperforms other classifiers with the best accuracy in the first phase while IBK performs better in the second phase. Figure III shows that the best performance improvement when applying the feature selection methods is for SMO, Logistic, and IBK classifiers. Moreover, the results shown in Table 8 indicate that all the classifiers give good accuracy for the dominant classes, while it is not the case for the R2L and U2R classes. It also shows that the imbalance mitigation method improves limitations in detecting R2L and U2R attacks.

6 Conclusion & Future Work

Our analysis results for the performance of the six different classifiers on the NSL-KDD dataset shows that J48 and IBK are the best two classifiers in terms of accuracy detection but IBK is much better when applying feature selection techniques. For future work, we propose to carry out an exploration on how to employ optimization techniques to develop an intrusion detection model with a better accuracy rate.

References

1. D. B. Roy, R. Chaki, State of the art analysis of network traffic anomaly detection, in Applications and Innovations in Mobile Computing (AIMoC), 2014, IEEE, 2014, pp. 186–192.
2. Buczak, Anna L., and Erhan Guven. "A survey of data mining and machine learning methods for cyber security intrusion detection." IEEE Communications Surveys & Tutorials 18.2 (2016): 1153-1176.
3. Papernot, Nicolas, Patrick McDaniel, and Ian Goodfellow. "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples." arXiv preprint arXiv:1605.07277 (2016).
4. Alkasassbeh, Mouhammd. "An empirical evaluation for the intrusion detection features based on machine learning and feature selection methods." arXiv preprint arXiv:1712.09623 (2017).
5. Potluri, Sasanka, and Christian Diedrich. "High Performance Intrusion Detection and Prevention Systems: A Survey." ECCWS2016-Proceedings fo the 15th European Conference on Cyber Warfare and Security. Academic Conferences and publishing limited, 2016.
6. Fabris, Fabio, João Pedro De Magalhães, and Alex A. Freitas. "A review of supervised machine learning applied to ageing research." Biogerontology 18.2 (2017): 171-188.
7. NSL-KDD dataset [online] available: http://www.unb.ca/cic/datasets/nsl.html Accessed on 10/21/2018.

8. Ingre, Bhupendra, and Anamika Yadav. "Performance analysis of NSL-KDD dataset using ANN." Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on. IEEE, 2015.
9. Dhanabal, L., and S. P. Shantharajah. "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms." International Journal of Advanced Research in Computer and Communication Engineering 4.6 (2015): 446-452.
10. Karimi, Zahra, Mohammad Mansour Riahi Kashani, and Ali Harounabadi. "Feature Ranking in Intrusion Detection Dataset using Combination of Filtering Methods." International Journal of Computer Applications 78.4 (2013).
11. Kohavi, Ron, and George H. John. "Wrappers for feature subset selection." Artificial intelligence 97.1-2 (1997): 273-324.
12. Claesen, Marc; Bart De Moor (2015). "Hyperparameter Search in Machine Learning". arXiv:1502.02127.
13. MeeraGandhi, G. "Machine learning approach for attack prediction and classification using supervised learning algorithms." Int. J. Comput. Sci. Commun 1.2 (2010).
14. Nguyen, Huy Anh, and Deokjai Choi. "Application of data mining to network intrusion detection: classifier selection model." Asia-Pacific Network Operations and Management Symposium. Springer, Berlin, Heidelberg, 2008.
15. Jalil, Kamarularifin Abd, Muhammad Hilmi Kamarudin, and Mohamad Noorman Masrek. "Comparison of machine learning algorithms performance in detecting network intrusion." Networking and Information Technology (ICNIT), 2010 International Conference on. IEEE, 2010.
16. Belavagi, Manjula C., and Balachandra Muniyal. "Performance evaluation of supervised machine learning algorithms for intrusion detection." Procedia Computer Science 89 (2016): 117-123.
17. Revathi, S., and A. Malathi. "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection." International Journal of Engineering Research and Technology. ESRSA Publications (2013).
18. H. He, E.A. Garcia, "Learning from Imbalanced Data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284,2009.
19. Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.