International Journal of Scientific Research and Engineering Development-– Volume 4 Issue 5, Sep- Oct 2021

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Performance Evaluation for Attack Detection in Intrusion Detection System

Mossa Ghurab[1], Reem Alshamy[1]*, Suad Othman[1]

[1]Department of Computer Science, Faculty of Computer and IT (FCIT), Sana'a University, Sana'a, Yemen

Email: reemalshamy2020@gmail.com

----------**********----------

Abstract:

An Intrusion Detection System (IDS) plays a vital role in cybersecurity to detect any threat in networks. In a dynamic environment such as a network that is vulnerable to various types of attacks, fast and robust solutions are needed to deal with rapidly changing threats. This paper evaluated different machine learning (ML) algorithms to detect types of attacks on the NSL-KDD dataset. In the experiment, we presented a comparison between K Nearest Neighbours (KNN), Random Forest (RF), Extra Trees (ET), and Gradient Boosting (GB) classifiers based on accuracy for each type of attack. The results of the experiment showed that the KNN algorithm achieved high accuracy for U2R compared with other classifiers.

Keywords **—Intrusion Detection, Imbalanced dataset, Anaconda, NSL-KDD dataset.**

----------**********----------

I. INTRODUCTION

Recently, the number of Internet users has grown, this has led to the created a large number of data and the emergence of various types of attacks. The big challenge facing network engineers and researchers today is to identify malicious activities in a host or over a network. The Cybersecurity research area looks at the ability to act proactively to mitigate or prevent attacks.

IDS is placed at a strategic point in the network where it monitors all traffic and analyses traffic to detect potential attacks [1]. IDS can be defined as an intrusion detection process which is to find events violation of security policies in computer networks, it is usually located within the network to monitor all internal traffics [2]. Mostly, IDS follows one of the two major detection methods: Signature-based IDS and Anomaly-based IDS. Additionally, many researchers have proposed hybrid methods. Signature-based detection is quite popular in commercial applications, it is designed to detect known attacks that are preloaded in the IDS datasets. Anomaly-based detection is limited to academics for research and development, it is an effective way to detect unknown attacks. A hybrid detection is combined two methods to overcome disadvantages in signature-based detection and obtain advantages for anomaly-based detection [3, 4]. To detect unknown attacks, researchers have paid great attention to introducing other techniques in detecting network intrusion, and machine learning (ML) techniques are one of the most techniques used.

In recent years, ML algorithms have been used in the field of intrusion detection, and some improvements have been achieved. However, as it is well known that there is each algorithm has its advantages and disadvantages. Some algorithms may perform well on one type of attack, but show poor performance on other types. In addition, many studies only focus on the overall accuracy, but the detection effect for small-scale data is often very

low. The proportion of real attack events in all data is imbalanced, so we need to focus on the detection ability of malicious attack for each type of attack.

This paper aims to use Anaconda in IDS that can reduce computation time and achieves accurate classification for each type of attack. For this purpose, we evaluated a classification algorithms for IDS using the NSL-KDD dataset. Firstly, the data preprocessing method is used to drop feature that has 0 value and to convert categorical data to numerical data, and then standardization in the dataset is done to improve classification efficiency. Secondly, ML classifiers is used to enhance the detection accuracy for each type of attack because the NSL-KDD dataset has a class imbalance problem. Additionally, we introduce a comparison between KNN, RF, ET, and GB classifiers based on accuracy for multi-class classification.

The remainder of this paper is structured as follows: A review of related work is introduced in the "Related work" section. In the "Methodology" section, we provided the performance evaluation for ML algorithms. Furthermore, each step in the methodology is introduced. Results and discussion are mentioned in the "Results and Discussion" section. Finally, conclusions and directions for future work are presented in the "Conclusion" section.

II. RELATED WORK

There are many methods proposed for IDS that used benchmark NSL-KDD dataset to improve detection attacks. Some researchers used traditional techniques and other big data techniques for analyzing and storing data in IDS. In this section, some previous related works that used big data techniques to solve the classification problems in IDS is summarized.

Bandyopadhyay et al. [5] applied Information Gain (IG) and Decision Tree classifiers. The IG reduced the features, and also produced a high detection. The overall performance comparison was evaluated on the NSL-KDD dataset in terms of accuracy, precision, recall, f1 measure, and construction time for building models for classification.

Devan et al. [6] suggested XGBoost–DNN model for IDS using the NSL-KDD dataset. The XGBoost–DNN model contained three steps: normalization, feature selection, and classification. They used the XGBoost technique for feature selection and DNN for classification intrusion.

Gao et al. [7] suggested an adaptive ensemble learning (AEL) model. The main idea of this model is to use ensemble learning to gather the advantages of different algorithms. The NSL-KDD dataset was used to training and testing the proposed model by using Python language.

Nanda et al. [8] evaluated SVM, KNN, and DT algorithms over Normal, Dos, R2L, and U2R attacks on the NSL-KDD dataset. The features selection method was used by using a correlation-based method.

Alshamy et al. [9]proposed IDS-SMOTE-RF. They have used SMOTE Technique to deal with a class imbalanced problem and an RF classifier that has improved performance to detect types of attack. The results of the experiment showed that the IDS-SMOTE-RF model achieved high accuracy and efficient for Big Data.

Table 1 displays differences between related work based on the ML algorithms that were used as a classifier and the tool that was used to developing the work and the dataset that has been used to train and evaluate.

TABLE 1

SUMMARY OF RELATED WORK.

Work	Classifiers	Dataset	Tool
[5]	DT, RF, LR and NB	NSL-KDD	Anaconda
[6]	DNN, LR, SVM and NB	NSL-KDD	Anaconda
[7]	DT, RF, KNN, DNN and MultiTree	NSL-KDD	Anaconda
[8]	SVM, KNN and DT	NSL-KDD	Anaconda
[9]	RF, AB, LR and SVM	NSL-KDD	Anaconda

III. METHODOLOGY

In this section, we describe the methodology and techniques used in the experiment. The steps of methodology can be summarized as follows:

Step 1: Load the NSL-KDD dataset.
Step 2: Data preprocessing.
Step 3: split the NSL-KDD dataset into training data and testing data.
Step 4: Train by using the RF, ET, KNN, and GB classifiers on the NSL-KDD training data.
Step 5: Test on the NSL-KDD testing data.
Step 6: Results analysis.

The steps of the methodology for attack classification are shown in Figure 1.

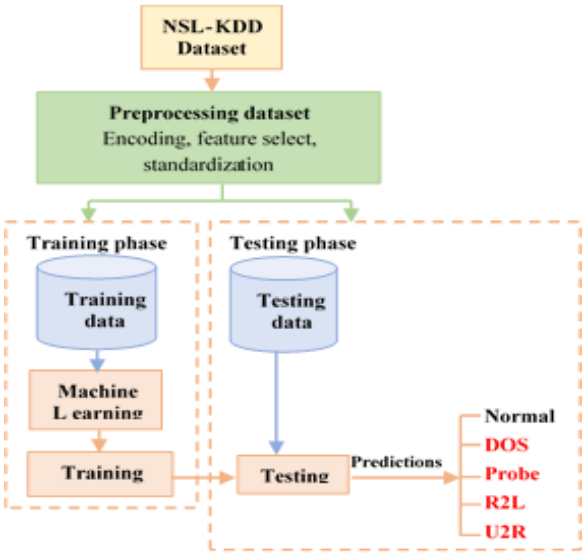

Fig.1 Steps of methodology

A. *Data description*

The NSL-KDD dataset is one of the most popular datasets that was used in training and evaluating IDS [10]. This is dataset contains 42 features [11]. Table 2 illustrates the features and types of features in the NSL-KDD dataset.

TABLE 2

THE NSL-KDD DATASET FEATURES.

Features	Type
1. duration lenght	Numeric
2. protocol_type	Categorical
3. service	Categorical
4. flag	Categorical
5. src_bytes	Numeric
6. dst_bytes	Numeric
7. land	Binary
8. wrong_fragment	Numeric
9. urgent	Numeric
10. hot	Numeric
11. num_failed_logins	Numeric
12. logged_in	Binary
13. lnum_compromised	Numeric
14. lroot_shell	Numeric
15. lsu_attempted	Numeric
16. lnum_root	Numeric
17. lnum_file_creations	Numeric
18. lnum_shells	Numeric
19. lnum_access_files	Numeric
20. num_outbound_cmds	Numeric
21. is_hot_login	Binary
22. is_guest_login	Binary
23. count	Numeric
24. srv_count	Numeric
25. serror_rate	Numeric
26. srv_serror_rate	Numeric
27. rerror_rate	Numeric
28. srv_rerror_rate	Numeric
29. same_srv_rate	Numeric
30. diff_srv_rate	Numeric
31. srv_diff_host_rate	Numeric
32. dst_host_count	Numeric
33. dst_host_srv_count	Numeric
34. dst_host_same_srv_rate	Numeric
35. dst_host_diffsrv_rate	Numeric
36. dst_host_same_src_port_rate	Numeric
37. dst_host_srv_diff_host_rate	Numeric
38. dst_host_serror_rate	Numeric
39. dst_host_srv_serror_rate	Numeric
40. dst_host_rerror_rate	Numeric
41. dst_host_srv_rerror_rate	Numeric
42. class	Categorical

The target class marks the attack type of the connection; the class including five categories one normal class and four attack classes. The four main categories of attacks are:

1) Denial of Service (DOS): intruders attempt to block system or network resources and services.
2) Probing: intruders collect the information about potential vulnerabilities of the target system that can be used later to launch attacks on those systems.
3) Remote to Local (R2L) attack: unauthorized ability to dump data to a remote system over a network and gain access either as user or root to carry out their unauthorized activity.

4) User to Root (U2R): intruders access the system as a normal user and break the vulnerabilities to gain administrative privileges.

This dataset consists of 23 classes that include 22 are attack and one is a normal. The 23 classes are grouped into five groups (Normal, DOS, Probe, R2L, and U2R). Table 3 demonstrates the different distributions of attack into one of the five main attacks.

TABLE 3

THE NSL-KDD ATTACK TYPES.

Class	Attack type
Normal	normal (1)
DOS	back, land, neptune, pod, smurf, teardrop (6)
Probe	ipsweep, portsweep, nmap, satan (4)
R2L	phf, guess_passwd, spy, warezmaster, ftp_write, warezclient, imap, multihop (8)
U2R	buffer_overflow, loadmodule, perl, rootkit (4)

Figure 2 displays the distributions of instances in different types of attacks for both training and testing datasets.

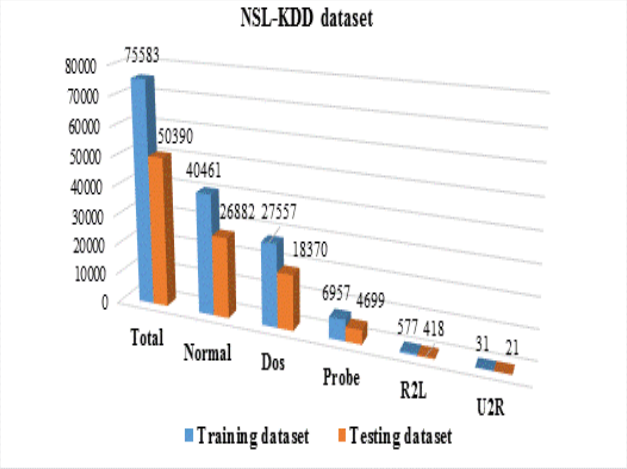

Fig. 2 Distributions of instances in the NSL-KDD dataset.

According to the distributions of instances in different types of attacks in Figure 2, it is found the distributions of various types of data are imbalance[12]. Normal has the highest proportion of total data, and U2R has the lowest proportion of total data, which may lead to inadequate training and misclassified data. Attackers often use Probe, R2L, and U2R as an advanced threat attack, so we should try our best to improve the classification accuracy detection of these types of attacks.

B. Data preprocessing

TheNSL-KDD dataset contains numeric, categorical, and binary features. The categorical features need to convert to numeric features. The categorical features are converted using a one-hot encoder. Through the analysis of features in the NSL-KDD dataset, we found the num_outbound_cmds feature has 0 value for all instances, so this feature is dropped. Several of the features of the NSL-KDD dataset have many values. In ML, the standardization of datasets is very significant for algorithms that use Euclidean distance [13]. In the experiment, the authors used StandardScaler. The StandardScaler uses a strict definition of standardization to standardize data [14].

C. Classifiers

There are four ML classifiers were used in the evaluation, we summarized as follow:

1) RF classifier

It is an ensemble learning algorithm [15]. RF is combining tree classifiers to predict new unlabelled data. The predictor depends on a constant that denotes the number of trees in the forest; the features are selected randomly, and each number of the set (trees) here, they represent one forest, and each one of these forests represents a prediction class. In this algorithm, random feature selection will be selected for each tree. RF architecture is displayed in Figure 4 [16].

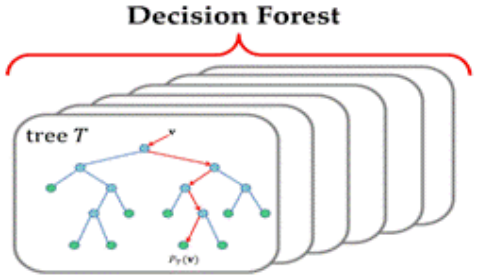

Fig. 3 Random Forest Architecture.

2) ET classifier

The ET classifier is an ensemble learning algorithm for classification and prediction. When you are growing a tree in a Random Forest, at each node only a random subset of the features is considered for splitting. It is possible to make trees even more random by also using random thresholds for each feature rather than searching for the best possible thresholds, like regular Decision Trees do. A forest of such extremely random trees is simply called an Extremely Randomized Trees ensemble or Extra-Trees for short. Once again, this trades more bias for a lower variance. It also makes Extra-Trees much faster to train than regular Random Forests since finding the best possible threshold for each feature at every node is one of the most time-consuming tasks of growing a tree. The predictor depends on a constant that denotes the number of trees in the forest; the features are selected randomly, and each number of the set (trees) here, they represent one forest, and each one of these forests represents a prediction class. In this algorithm, random feature selection will be selected for each tree [16].

3) KNN classifier

The KNN has been used to classify summarized data. The main idea of KNN classifier is that, in a sample space, if most of its k closest neighbors samples belong to a class, then the sample belongs to the same class. Note that, The Euclidian distance has been used to compute the closest k neighbors since the transformed data consists of one dimension. Finally, the KNN classifier has been chosen due to its low computation cost [17].

4) GB classifier

The Gradient-boosted decision trees are a popular method for solving prediction problems in both classification and regression domains. The approach improves the learning process by simplifying the objective and reducing the number of iterations to get to a sufficiently optimal solution. Gradient-boosted models have proven themselves time and again in various competitions grading on both accuracy and efficiency [18].

D. Evaluation metrics

The performance is evaluated in terms of accuracy. The accuracy measures the proportion of the total number of correct classifications instances [19]. It is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (1)$$

IV. RESULTS AND DISCUSSION

In this section, we show the results of ML algorithms that is used for attack classification. This algorithms were implemented on Windows 7 and using a personal computer, which the processor is Intel(R) Core (TM) i5 2.60 GHz, the RAM is 4 GB, and the main programming language was Python using Jupyter Notebook on Anaconda platform [20].

The dataset of the experiment contained 125973 instances; 60% of instances were used as a training data, and 40% of instances were used as a testing data. The distributions of instances in different attacks for both training and testing data that have been used in the experiment were shown in Figure 2. After training the classifiers, the evaluated performance was performed on testing data to classify data into one of five classes (Normal, Probe, R2L, and U2R). Moreover, the authors compared the performance of classifiers for each type of attack based on accuracy. Table 4 illustrates the accuracy for each individual class using four ML classifiers.

TABLE 3

ACCURACY FOR EACH ATTACK USING THE NSL-KDD DATASET.

Model	Normal	DOS	Probe	R2L	U2R
RF	**99.97**	**99.97**	**99.61**	95.45	38.09
ET	99.92	99.96	99.40	93.30	38.09
KNN	99.86	99.86	98.87	90.66	**42.85**
GB	99.84	99.97	98.68	**96.88**	14.28

As observed from the comparisons in Table 4 of four ML classifiers for multi-class classification, the RF classifier achieved high accuracy for Normal, DOS, Probe classes, the GB classifier achieved high accuracy for R2L, and the KNN classifier achieved high accuracy for U2R class.

V. CONCLUSIONS

In this paper, we evaluated the performance of four ML algorithms for attack classification in IDS. In the experiment, we used the KNN, RF, ET, and GB algorithms to classify data into normal or specific attack (Dos, Probe, R2L, and U2R) using Anaconda platform and the NSL-KDD dataset. The results of the experiment showed that the RF has high accuracy for majority class and the KNN has high performance for minority class. In future work, we can improve the accuracy detection rate for each type of attack and reduced computation time by apply feature selection techniques.

REFERENCES

[1] A. Ju, Y. Guo, Z. Ye, T. Li, and J. Ma, "HeteMSD: A Big Data Analytics Framework for Targeted Cyber-Attacks Detection Using Heterogeneous Multisource Data," *Security and Communication Networks,* vol. 2019, 2019.

[2] K. Kim, M. E. Aminanto, and H. C. Tanuwidjaja, *Network Intrusion Detection Using Deep Learning: A Feature Learning Approach*: Springer, 2018.

[3] S. M. Othman, F. M. Ba-Alwi, N. T. Alsohybe, and A. Y. Al-Hashida, "Intrusion detection model using machine learning algorithm on Big Data environment," *Journal of Big Data,* vol. 5, p. 34, 2018.

[4] R. Alshamy and M. Ghurab, "A Review of Big Data in Network Intrusion Detection System: Challenges, Approaches, Datasets, and Tools," *Journal of Computer Sciences and Engineering,* vol. 8, pp. 62-75, 2020.

[5] S. Bandyopadhyay, R. Chowdhury, P. Banerjee, S. D. Dey, and B. Saha, "A Decision Tree Based Intrusion Detection System for Identification of Malicious Web Attacks," 2020.

[6] P. Devan and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system," *Neural Computing and Applications,* pp. 1-16, 2020.

[7] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access,* vol. 7, pp. 82512-82521, 2019.

[8] N. B. Nanda and A. Parikh, "Network intrusion detection system: classification, techniques and datasets to implement," *Int. J. Future Revol. Comput. Sci. Commun. Eng,* vol. 4, pp. 106-109, 2018.

[9] R. Alshamy, M. Ghurab, S. Othman, and F. Alshami, "Intrusion Detection Model for Imbalanced Dataset using SMOTE and Random Forest Algorithm," in *Third International Conference on Advances in Cyber Security*, Malaysia, 2021.

[10] M. Ghurab, G. Gaphari, F. Alshami, R. Alshamy, and S. Othman, "A Detailed Analysis of Benchmark Datasets for Network Intrusion Detection System," *Asian Journal of Research in Computer Science,* vol. 7, pp. 14-33, 2021.

[11] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," *Expert Systems with Applications,* vol. 141, pp. 112963-112963, 2020.

[12] R. Alshamy, M. Ghurab, S. Othman, and F. Alshami, "Intrusion Detection Model for Imbalanced Dataset using
SMOTE and Random Forest Algorithm," presented at the Third International Conference On Advances In Cyber Security, 2021.

[13] T. K. Tunduny, "A HIV/AIDS viral load prediction system using artificial neural networks," Strathmore University, 2017.

[14] A. C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*: " O'Reilly Media, Inc.", 2016.

[15] A. Araar and R. Bouslama, "A COMPARATIVE STUDY OF CLASSIFICATION MODELS FOR DETECTION IN IP NETWORKS INTRUSIONS," *Journal of Theoretical & Applied Information Technology,* vol. 64, 2014.

[16] I. Obeidat, N. Hamadneh, M. Alkasassbeh, M. Almseidin, and M. AlZubi, "Intensive pre-processing of kdd cup 99 for network intrusion classification using machine learning techniques," 2019.

[17] M. Riyadh, B. J. Ali, and D. R. Alshibani, "IDS-MIU: An Intrusion Detection System Based on Machine Learning Techniques for Mixed type, Incomplete, and Uncertain Data Set."

[18] Y. M. Banadaki, J. Brook, and S. Sharifi, "Design of the network intrusion detection systems for the internet of things infrastructure using machine learning algorithms," *Design of Intrusion Detection Systems on the Internet of Things Infrastructure using Machine Learning Algorithms,* 2021.

[19] K. Ye, "Key feature recognition algorithm of network intrusion signal based on neural network and support vector machine," *Symmetry,* vol. 11, pp. 380-380, 2019.

[20] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," pp. 86-93.

