Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM

ISSN: 2788-7421

Ensemble Machine Learning Techniques for Attack Prediction in NIDS Environment

T. Sreenivasula reddy[1,*], R. Sathya[2]

[1]Department of Computer Science & Engineering, Annamacharya Institute of Technology & Sciences Tirupathi, Andhra Pradesh-517520, India

[2]Department of Computer Science & Engineering, Annamalai University, Annamalai Nagar, Chidambaram, Tamil Nadu-608002, India

*Corresponding Author: T. Sreenivasula reddy

DOI: https://doi.org/10.52866/ijcsm.2022.02.01.008

Received January 2022; Accepted March 2022; Available online March 2022

ABSTRACT: The need for network intrusion detection systems (NIDS) to protect against different attacks grows as the scale of cyber attacks increases. The main areas of cyber attack research are its detection and prevention. Traditional ma- chine learning (ML) algorithms with low accuracy are used by the current NIDS, but it is not suitable for newer anonymous cyber attacks. In this paper, an NIDS model with ensemble ML methods, which can detect and prevent different types of attacks compared with traditional ML methods, is proposed. Our specific sys- tem detects known attacks and blocks unknown attacks. The selected system uses four different machine learning methods, including data processing techniques for data preprocessing and data labeling. The entire NSL-KDD database is used to evaluate the performance of various ML classifiers based on different param- eters. The simulation analysis shows that the developed NIDS system is better than the existing single ML methods. The detection accuracy rate of intrusion detection system (IDS) is increased by the model, which is essential for NIDS.

Keywords: attacks; machine learning; network intrusion detection systems; NSLKDD dataset; data labeling

1. INTRODUCTION

To secure information systems, the important component is intrusion detection system (IDS). Network intruders try to access unauthorized sources within the network. Monitoring and analyzing user performance and system behavior are important [1, 2]. By changing the configuration of system parameters, system behavior may become er- ratic. Therefore, the system must be equipped with the features of periodic monitoring and its behavior for routine and extraordinary activities. IDS has two types based on direct deployment and detection systems [3]. Based on screening, IDS is classified into host-based IDS (HIDS) and network intrusion detection systems (NIDS), where the inner workings of a computer system are monitored by HIDS. Live network traffic logs are monitored by HIDS to discover network intrusions using appropriate detection al- gorithms [4, 5]. Based on the detection mechanism, IDS is categorized into detection of abuse, fault, and hybrid IDS.

To detect known attacks, abuse detection uses predefined rules or signatures. Disorder detection creates a default functional profile to verify that the system state is different from the default installed functional profile and to detect unknown attacks. Known and unknown attacks are detected by hybrid IDS [6–8]. Nowadays, intrusions are detected by using data mining techniques, which are used by all kinds of IDSs. Most current NIDS attacks are detected using all features configured from network traffic [9]. However, not all features are required to detect attacks. Detection rate is increased and high computation time is decreased by minimizing the number of features [10]. In this work, the filter-based approach is combined with machine learning (ML) synchronization techniques for selecting appropriate features for IDS detection. The task is to reduce the number of features with better performance for an uncompromising detection rate. Although NIDS has a variety of techniques based on the selection of features and classifiers in the literature, the specific

task focuses on data preprocessing and tagging using Python and Jupiter notebook codes. In addition, not only a single ML technique is used similar to the existing technique, but the research work also focuses on ensemble ML techniques for final classification.

The rest of this paper is organized as follows: Section 2 summarizes relevant works in the literature. Section 3 describes the database with the proposed functionality for intrusion detection. Section 4 presents the findings and discussions. Section 5 reveals the final notes.

2. LITERATURE REVIEW

In [11], the authors designed an SVM to select and classify the features. The test results showed that the method achieved 99% accuracy on the NSL-KDD Cup 99 of the Intrusion Detection Database. A KNN classifier was developed in [12] for IDS. The simulation results proved that the KNN classifier performed better than the existing techniques on the NSL-KDD database.

In [13], the authors proposed a new framework for combining abuse and malpractice detection using the RF algorithm. The algorithm achieved 2% false positive rate and 94.7% detection accuracy. In [14], the artificial neural network evaluated the performance of the NSL-KDD database. The detection rate of the IDS task and attack type classification for the NSL-KDD database were 81.2% and 79.9%, respectively.

In [15], an end-of-tree (DT)-dependent intrusion detection method was proposed. The feature test results using the relevant Feature Selection Subcommittee evaluation method showed that the DT-based IDS has a high accuracy. However, these approaches require complex feature engineering for large-scale preprocessing and transfer data. The massive IDS classification problem cannot be solved using individual ML methods.

In [16], self-learning in the NSL-KDD database was used to navigate the net- work attack. The test results showed that their average rating out of 5 was 75.76%. The authors from [17] designed an IDS using the deep belief network (DBN) and probability neural network (PNN). The experimental results from the KDD CUP 99 database showed that this method performs better than conventional PNN, PCA-PNN, and mod- ified DBN-PNN.

3. PROPOSED SYSTEM

The proposed system of our research work is depicted in Figure 1.

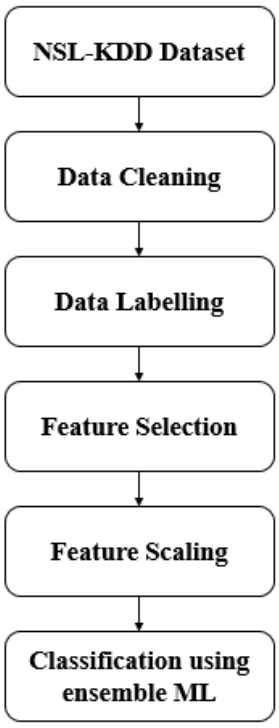

FIGURE 1. Workflow of Proposed Methodology

3.1 DATA EXTRACTION

NSL-KDD of Canadian Network Traffic Database is used for this research work. This database is an updated version of KDD CUP 99 database that fixes several vulnerabilities. When the database is unbalanced, classifying by category label is diffi- cult. This database presents five categories, namely, R2L, normal, U2R, probe, and DoS. Some attacks are not in the training package but are in the test package, which makes them very realistic.

3.2 DATA CLEANING

After extracting the data, data cleaning is performed by using the Python programming language. A total of 43 features are extracted from the Canadian NSL-KDD database. Two features are removed with some fixed values instead of sorted data. Fixed value features are also removed. Finally, 40 features are used, where one feature is considered a target class feature.

3.3 DATA LABELING

Among the features, three features, namely, "class," "aware," and "type_protocol," are non-numeric. For example, the protocol type called tcp, udp, and icmp from the features are modified and marked as 0, 1, and 2, respectively. Likewise, "class" has five types of attributes, and "flag" has 11 types of attributes. Similarly, all non-numeric values are converted to numeric values after naming.

3.4 FEATURE DATA SELECTION

Among the feature data selection's different types of filtering methods, the Fast Approach is the best. In this paper, the probable value of the features used by ScikitLearn is considered. The p-value is used to identify statistically significant features, within which statistical data can be used for identifying the features/physical parameters that show irregularity in network traffic. The parameters are significant when the value of p is small. After selecting the feature, the selected feature is measured using the stand- ard ScikitLearn function.

3.5 CLASSIFICATION USING ENSEMBLE ML TECHNIQUES

According to Bayes theory, Naïve Bayes (NB) is a simple, effective classifi- cation technique. It requires independence between prophets, that is, attributes or traits should not be correlated or interdependent. Despite the bias, all of these traits or traits freely contribute to the possibility, which is why it is called naive.

RF is a commonly tracked ML algorithm, which is best for classification tasks and regression. As the name implies, RF considers several final trees before a release is released. Thus, it is a group of the last trees. To make right decision, this technique is based on the belief of trees. For classification, it uses a rating system and then selects the category, whereas in regression, it takes the mean of each final tree and all the outputs. It works well on vast amount of datasets with large volumes.

The KNN technology is the most basic and effective classification techniques. With minimal or no prior knowledge of the data distribution, no assumptions about data are made by this technology and used for taxonomic tasks. This method involves finding the data points closest to k and assigning the average value of the data points found in the data point exercise where the target value is not available.

DT is a tracking learning tool, which is used for classification problems. It runs smoothly with continuous, varied features. According to the important predictions, this algorithm divides the population into two or more similar groups. Each attribute's entropy is initially calculated by DT algorithm, where the data set is separated with the help of variables or predictors with maximum data gain or low entropy.

4. RESULTS AND DISCUSSION

All tests are conducted on an Ubuntu 14.0.4 LTS with Python. ScikitLearn is used to implement all ML algorithms. The following evaluation metrics are examined based on the above given terms. Here, TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

$$Accuracy = \frac{TN + TP}{TP + TN + FN + FP} \times 100 \tag{1}$$

$$F - measure = \frac{2TP}{(2TP + FP + FN)} \times 100 \tag{2}$$

$$Precision = \frac{TP}{(FP+TP)} \times 100 \tag{3}$$

$$Recall = \frac{TP}{(FN+TP)} \times 100 \tag{4}$$

4.1 PERFORMANCE EVALUATION OF PROPOSED MODEL

The proposed evaluation is segregated into major parts such as binary classification and multiclass classification. Binary classification detects the attack or normal communica- tion. Multiclass classification detects the various types of attack, which is presented in the dataset.

A. Binary Classification

The detailed results for the binary classification of several ML classifiers are reported in this section.

Table 1. Comparative analysis of binary class on ensemble ML algorithms

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
NB	87.73	88.10	89.47	90.23
DT	91.52	90.83	90.16	91.07
KNN	92.28	92.01	91.07	92.43
RF	94.02	95.65	93.24	94.26

Table 1 clearly proves that the RF technique achieves better accuracy (94.02%), precision (95.65%), recall (93.24%), and F-score (94.26%), and considers the pro- posed model. The other techniques, namely, DT and KNN, achieves nearly 91%– 92% accuracy, precision, recall, and F-score, where NB achieves nearly 88%–90% accuracy, precision, recall, and F-score on binary data classification. Compared with all techniques, NB provides low results in all parameters.

BMulticlass Classification

The detailed results for multiclass classification of proposed RF systems are reported in this section. Table 2 shows the performance analysis of the proposed model on multiclass data classification. RF shows better performance on binary classification. Hence, it is only considered for this classification.

Table 2. Comparative analysis of multiclass onproposed RF classifier

Category	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
Normal	80.18	81.57	92.61	90.37
DoS	81.54	90.06	88.44	84.62
Probe	83.40	75.87	77.12	86.78
U2R	81.03	70.14	75.07	93.04
R2L	84.91	73.72	70.14	82.85

In the normal category, the proposed RF method achieves 80.18% accuracy, 81.57% precision, 92.61% recall, and 90.37% F1-measure. Compared with other categories on recall experiments, the proposed RF technique achieves high performance on the normal category only. Similarly, the proposed method achieves high precision (i.e., 90.06%) on the DoS category and high F1-measure (i.e., 93.04%) only on the U2R category. In other categories such as Probe, U2R, and R2L, the proposed method achieves nearly 70%–74% precision, 70%–77% recall, and 81%–84% accuracy, whereas the RF technique achieves a lower recall (i.e., 70.14%) on the R2L category only.

5. CONCLUSION

The traditional basic level ML algorithms are not sufficient for the accurate detection of an intrusion attack. Moreover, DL methods do not produce better results. In this case, a collective ML system can achieve better detection rates for potential model development. In our work, ScikitLearn notebook, Jupyter for reducing measurements, increasing detection accuracy, and reducing false positive ratios that are important functions of data processing technologies for intrusion detection are used. Most of the current network IDS based on the NSL-KDD database fail and use 41 features. In this work, the appropriate features (41) are selected from the total features for network intrusion detection, and ensemble ML

classifiers are used to predict attacks. The test results in the NSL-KDD database indicate that the RF model achieves the highest accuracy. Comparisons shows that the RF sample results are quite reliable than some ML classifiers. Therefore, the problem of IDS can be effectively solved by the proposed method, which is a powerful tool.

ACKNOWLEDGEMENT

The first author would like to thank the reviewers for providing useful suggestions, allowing for the improved presentation of this paper.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] K. A. Taher, B. M. Y. Jisan, and M. M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," *2019 International conference on robotics, electrical and signal processing techniques (ICREST)*, pp. 643–646, 2019.
[2] S. K. Biswas, "Intrusion detection using machine learning: A comparison study," *International Journal of pure and applied mathematics*, vol. 118, no. 19, pp. 101–114, 2018.
[3] S. Otoum, B. Kantarci, and H. T. Mouftah, "On the feasibility of deep learning in sensor network intrusion detection," *IEEE Networking Letters*, vol. 1, no. 2, pp. 68–71, 2019.
[4] M. Belouch, S. E. Hadaj, and M. Idhammad, "Performance evaluation of intrusion detection based on machine learning using Apache Spark," *Procedia Computer Science*, vol. 127, pp. 1–6, 2018.
[5] K. A. D. Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. D. Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, 2019.
[6] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, pp. 322–322, 2019.
[7] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.
[8] Y. Chang, W. Li, and Z. Yang, "Network intrusion detection based on random forest and support vector machine," *2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC)*, vol. 1, pp. 635–638, 2017.
[9] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *applied sciences*, vol. 9, no. 20, p. 4396, 2019.
[10] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 277–000282, 2017.
[11] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," *Proc. 8th Int. Conf. Softw., Knowl*, pp. 1–6, 2014.
[12] H. Shapoorifard and P. Shamsinejad, "Intrusion detection using a novel hybrid method incorporating an improved KNN,," *Int. J. Control Automat*, vol. 173, no. 1, pp. 5–9, 2017.
[13] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *Man, Cybern. C, Appl. Rev*, vol. 38, no. 5, pp. 649–659, 2008.
[14] B. Ingre and A. Yadav, "2015 international conference on signal processing and communication engineering systems (spaces," *Proc. Int. Conf. Signal Process*, pp. 1–15, 2015.
[15] B. Ingre, A. Yadav, and A. K. Soni, "Decision tree based intrusion detection system for NSL-KDD dataset," *Proc. Int. Conf. Inf. Commun. Technol. Intell. Syst*, pp. 207–218, 2017.
[16] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *Proc. 9th EAI Int. Conf. BioInspired Inf*, pp. 21–26, 2016.
[17] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network,," *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE), IEEE Int. Conf. Embedded Ubiquitous Comput. (EUC)*, pp. 639–642, 2017.