ENHANCING NETWORK INTRUSION DETECTION THROUGH AN EFFECTIVE COMBINATION OF TREE ALGORITHMS

Avinash Kumar
Department of Computer Science Engineering, Mangalayatan University, Beswan, Aligarh, UP, India
Dr. Meena Chaudhary
Mangalayatan University, Beswan, Aligarh, UP, India

Abstract

This paper established a unique combined classifiers model for identifying network breaches by using tree-based techniques. An enhanced edition of the old KDDCUP'99 information set, the NSL-KDD data set, was used for the assessment. Using 41 variables that described various trends within network automobile traffic, this algorithm's main goal was to determine whether arriving data from networks was legal or suggestive of an attack. By combining the NB Tree and random trees techniques with the total rule program, 89.24% precision for detection was obtained. This indicates that the combined classifier strategy is more successful than just the single random trees technique. The result showed the greatest degree of precision to date and set a new standard overall the whole NSL-KDD sample. A combined encoder strategy that utilizes a total rule technique may be used in the future to advance anomaly-based detection of intrusions, according to the encouraging findings.

Keywords: *NSL-KDD;Network Intrusion Detection; Classifier Combination Sensitivity and Specificity; Machine Learning Algorithms*

DOI Number: 10.48047/nq.2022.20.19.nq99492 **Neuroquantology 2022; 20(19):5219-5224**

Introduction

Computer system security is a critical process aimed at safeguarding three fundamental elements: confidentiality, integrity, and availability (CIA) **(Stallings, 2015).**The importance of secure network connections has increased due to our growing dependence on technology and software. The FBI and the Information Security Association performed a biennial privacy and security study in 2005, and the results showed that corporations lost 130 million dollars in revenue as a result of network breaches and assaults. **(Maza & Touahria, 2018).** Consequently, intrusion detection (ID) has emerged as a prominent research challenge in network security, dating back to the proposal of the ID concept by Anderson in 1980 **(Alalade, 2020).**

An "Intrusion Detection System (IDS)"is characterized as a security organization entrusted with keeping an eye constantly and looking into system happenings in order to spot any efforts by unapproved individuals to access the resources of the system. **(Vijay et al., 2021).** Intrusion, in this context, is broadly defined as any actions attempting to jeopardize a resource's access, secrecy, or authenticity. The underlying assumption of ID is that the behavior of intruders differs from that of legitimate users. The primary objective of IDS is to accurately identify attacks from intruders,

thereby securing internal networks **(Al-Mejibli&Alharbe, 2020).**

Network-based IDS plays a crucial role in detecting harmful behaviors in traffic over networks, whether known or unidentified, triggering alarms upon detecting suspicious behavior. Two fundamental approaches for detecting network intrusions are misuse detection and anomaly detection**(Alsarhan,. et al 2021).**Misuse recognition, often referred to as based on information or signature-based surveillance, depends on a database that contains characteristics of previous assaults. Basically, this method compares network-related data to specified incursion behaviors in order to identify and categorize operations as either typical or invasive. However, for recognized threats, abuse success shows excellent rates of detection, it struggles with novel or unknown attacks, necessitating continuous database updates.

On the other hand, anomaly or behavior-based IDS focuses on identifying changes in the behavior of network users. This approach assumes that intrusions can be detected by observing deviations from normal or expected behavior. Accurate profiling of the subject's normal behavior becomes crucial in anomaly detection. While both approaches have their strengths and limitations, the ongoing evolution of intrusion detection systems remains vital for addressing the dynamic landscape of network security threats

Examining different data records seen by procedures on the precise same connection is necessary for that reason (Tama, Comuzzi, & Rhee, 2019).One of the more crucial elements is the information contained in those recordings and the qualities that can be taken out of them. The system for detection of intrusions may more effectively distinguish between distinct user types if the input parameters include additional data on innocent and intruding individuals, enhancing the integrity of the machine or net under protection. Compared to misuse monitoring approaches, methods for recognizing anomalies have the benefit of being able to identify unknown assaults. Vendors of goods often choose misuse-based diagnosis because of its high degree of precision and repeatability. However, because intrusion prevention systems are deployed on our network, cybercriminals will constantly attempt to create and execute "new" assaults. The recognition of anomalies is usually thought of as a more effective approach in academic study that has to be carefully studied because of its theoretical ability to handle fresh threats. Several learning methods are often used to recognize anomalies in IDS.

This paper proposes a combined classifying model with several decision tree techniques. The proposed detection technique can distinguish between regular and malicious network data. A system for detecting intrusions must lower the proportion of false positives or alerts. It must simultaneously be proficient at stopping assaults and have a high detecting rate. The costs of IDS may grow unreasonably large, which is one of its primary issues. Another significant issue with the IDS is its recognition reaction time. Internet connections are dynamical in the sense that data and knowledge continually shift within computers. Thus, precisely and quickly identifying an incursion is essential, particularly in real-time systems for detection of intrusions.

This work aims to evaluate the detection algorithm's performance using the same methodology that was used (Choudhary, &Kesswani, 2020). Using 41 distinct characteristics, we are going to employ judgment tree-based predictive techniques to determine when the incoming networks data is a legitimate request or an attack. There was no feature selection done. The NSL-KDD the data set regarded as an enhanced version from the previous KDDCUP'99 the data set is used to assess the usefulness of the proposed recognition system. Leveraging Weka is data mining software, students will use hybrid intelligence approaches to combine filters in order to improve the general accuracy of the resulting model.

Materials and methods

NSL-KDD dataset

The study utilized two generated datasets, namely “KDDTrain+ and KDDTest+”, comprising “125,973 and 22,544 records, respectively”. Additionally, the “KDDTest-21 dataset”, containing 11,850 records, excluded those correctly classified by all 21 learners

in the work by **(Choudhary and Kesswani 2020).** To enhance experimental diversity, In KDDTrain+, 20% of the documents were produced at random. Together with single class (battle or usual) or incident-type brands, such data sets, are publicly available.

Each network traffic pattern was characterized by 41 features, as outlined by **(Choudhary and Kesswani 2020).**For experimentation, a number of combination classification schemes were created, including votes by majority, max, min, sum, product, and max rules. Despite the fact that a few different strategies showed superior performance compared to individual classifiers, the reasons behind their relative effectiveness remained inadequately understood. Notably, the entire rule scored better than previous conjunction methods even though it was constructed under the strictest conditions. Sensitivity study clarified the experimental outcome and demonstrated the robustness of the average rule to estimating mistakes on test sets, in particular “KDDTest+ and KDDTest-21”.

The evaluation metrics extended beyond overall accuracy, encompassing sensitivity and specificity to assess the performance of the classification algorithms. Sensitivity gauged the algorithms' efficacy in detecting network attacks, while specificity measured false detections or the classifier's performance in identifying normal activities.

Three of the best-performing detection algorithms—NBTree, C4.5, and randomised tree—were chosen for in-depth examination. In addition, support vector machines, random woodlands, multilevel perceptrons, and Naive-Bayes were among the other techniques examined. The'seed' option was randomly assigned a value of 2, for C4.5, a fivefold decreased error cutting and a limit of 6 occurrences per plant were used. NBTree utilized default parameters throughout the experimentation process. This comprehensive methodology ensured a robust evaluation of the combining classifier model's performance in network intrusion detection.

Results and discussion

In order to prepare the classification engines during the testing period, we divided 20% of the data stored in the KDDTrain set into a training set. We then used models that were trained to assess how well they performed on both of the testing sets: “KDDTest+ and KDDTest-21”. The evaluation metrics extended beyond overall accuracy and included sensitivity and specificity to provide a comprehensive assessment of the classification algorithms.

Sensitivity, Important performance measures were particulars, which represented a count of incorrect identifications or the classifier's ability in distinguishing everyday behaviors, and efficacy, which showed how well the suggested methods detected network assaults.

Among the myriad of detection algorithms employed, three with the highest accuracies—random tree, C4.5, and NBTree—were singled out for detailed analysis. The comparative analysis also included other algorithms such as “Naive-Bayes, random forest, multilayer perceptron, and support vector machines”. To ensure reproducibility, two was selected as the random "seed" parameter. A by five reduced error clipping method and a requirement of 6 occurrences per leaf were used for the C4.5 procedure. Default settings were applied to NBTree during the experimental process.

The detection performance results for individual classifiers on the two test sets are summarized in Tables 1 and 2. Furthermore, the results for combining classifiers are presented in Tables 2 and 3. These tables provide a detailed insight into the effectiveness of both individual and combined classifier models in identifying network intrusions, offering a nuanced perspective on their sensitivity, specificity, and overall accuracy across the specified test sets

Table 1"Detection performance of combining classifiers models on KDDTest+"

"Combining classifiers"	Evaluation criteria		
	Sensitivity (%)	Specificity (%)	Accuracy (%)
"Random tree + NBTree"	83.9	96.2	89.24
"Random tree +C4.5"	83.0	96.5	88.81
"NBTree + C4.5"	76.7	95.6	84.98
"All three"	79.8	96.3	86.95

Table 2"Detection performance of combining classifiers models on KDDTest21"

Combining classifiers			
	Sensituation criteria		
"Random tree + NBTree"	78.8	Specificity (%)	Accuracy (%)
"Random tree + C4.5"	77.5	85.6	80.0
"NBTree + C4.5"	69.2	86.7	79.15
All three	73.3	81.8	71.46

Table 3Comparison table

Comparison table	Feature selection	Overall accuracy (%)	
		"KDDTest+"	"KDDTest-21"
"NBTree [10]"	No	82.02	66.16
"Decision tree [9]"	CDFTR	80.141	80.141
"Fuzzy classifier [21]"	No	82.74	-
"Random tree + NBTree (our approach)"	No	89.24	80.0

Only two further publications that used the same methodology—training and validating the recommended procedures as described in (Choudhary, &Kesswani, 2020)—were located by the writers. Table 3 allows us to compare our findings with the results reported in these studies. Using the KDDTest+, we can observe the our method significantly exceeds the others in terms of its total precision, with CDFTR and decision tree classifier coming in second and third, respectively, (Chouhan, & Khan, 2019). It is improbable that they recorded the same total precision for both of their test sets.

It is important to remember that the NBTree classifier used in Choudhary and Kesswani (2020) was developed in the Weka package with the values for the parameters set to default. Conversely, the decision tree classifier from Chouhan and Khan (2019) was created in MATLAB; however, the authors were unable to indicate wether the parameters were adjusted or left at their default settings. A method called genetic programming (GP) technique was utilized in (Batiha, & Krömer, 2021) to generate a fuzzy classifier; the work offered GP parameters. The setting in which the method was created and whether the values provided were typical ones weren't covered by the authors, nevertheless. Weka was also utilized to implement our strategy, however we used tailored values for the parameters since they produced an overall gain in accuracy above default values. Given the dearth of research that use the appropriate methodology for the training and evaluation of IDS and the observation that some studies fail to disclose whether parameter values have been changed (Chouhan & Khan, 2019), it seems that that the juxtaposition shown in Table 3 is appropriate and deserving of conversation.

Conclusion

In order to identify network intrusions, we created a combined classifier model in this article using tree-based methods. We assessed the detected algorithm's efficacy using the NSL-KDD dataset, which is a significantly enhanced variation of the basic KDDCUP'99 sample. Our monitoring procedure's job was to identify, using 41 attributes that described every trend of traffic over the network, when the coming network traffic was legitimate or an attack. We find that, while the contrary is also true, mixing classifier approaches that use an aggregate rule design may provide more favorable outcomes than each of the classifiers. The information in Tables 1, 2, and 3 makes this clear. We also draw the conclusion that the optimal combination for overall performance might not come from picking each of the best specific classifiers. In the not-so-distant future, it will be possible to assess how feature reduction affects both learning duration as well as detection reliability. In order to categorize the newly arriving network communication to be either normal or originating from one of four assault groups, we can additionally construct a system to identify it like to this.

Reference

Stallings, W. (2015). *Computer security principles and practice*. Pearson Education, Upper Saddle River

Maza, S., & Touahria, M. (2018). Feature selection algorithms in intrusion detection system: A survey. *KSII Transactions on Internet and Information Systems (TIIS)*, *12*(10), 5079-5099.

Alalade, E. D. (2020, June). Intrusion detection system in smart home network using artificial immune system and extreme learning machine hybrid approach. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)* (pp. 1-2). IEEE.

Vijay, R., Manoj, S., Ravikanth, V., Vikas, Y., & Priyadarshini, P. I. (2021). Augmenting network intrusion detection system using extreme gradient boosting (xgboost). *International Journal of Creative Research Thoughts*, *9*.

Al-Mejibli, I. S., &Alharbe, N. R. (2020). Analyzing and evaluating the security standards in wireless network: A review study. *Iraqi Journal for Computers and Informatics*, *46*(1), 32-39.

Kavitha, G., & Elango, N. M. (2020). An approach to feature selection in intrusion detection systems using machine learning algorithms. *International Journal of e-Collaboration (IJeC)*, *16*(4), 48-58.

Alsarhan, A., Alauthman, M., Alshdaifat, E. A., Al-Ghuwairi, A. R., & Al-Dubai, A. (2021). Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks. *Journal of Ambient Intelligence and Humanized Computing*, 1-10.

Tama, B. A., Comuzzi, M., & Rhee, K. H. (2019). TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. *IEEE access*, *7*, 94497-94507.

Choudhary, S., &Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. *Procedia Computer Science*, *167*, 1561-1573.

Batiha, T., & Krömer, P. (2021). Evolutionary fuzzy rules for intrusion detection in wireless sensor networks. In *Advances in Intelligent Networking and Collaborative Systems: The 12th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2020) 12* (pp. 149-160). Springer International Publishing.

Chouhan, N., & Khan, A. (2019). Network anomaly detection using channel boosted and residual learning based deep convolutional neural network. *Applied Soft Computing*, *83*, 105612.