**A Machine Learning Approach to Enhancing Intrusion Detection Systems**

Noah Hassett and David Truong

Graduate Assistant: Abiola Olanrewaju

Faculty Mentor: David Anyiwo

Department of Management Information Systems, Bowie State University

**Abstract**

Machine learning, which can be defined as enabling computers to make successful predictions using past experiences where machines are learning from their mistakes, has exhibited an impressive development recently with the help of the rapid increase in the storage capacity and processing power of computers and computer algorithms. Together with many other disciplines, machine learning methods have been widely employed in intrusion detection systems (IDSs). Currently, IDSs are widely used to protect networks and systems from increasing cyberattacks. However, their high false-alarm rate is becoming a huge problem to keep the users’ information safe. In this paper, we are going to use data mining software, WEKA 3.8.5, seven machine learning algorithms, J48, Random Forest, Random Tree, REP Tree, SMO, Naïve Bayes, and IBk, and two intrusion detection datasets, NSL-KDD and UNSW-NB15 to find the most accurate algorithm that could solve the problem of high false-alarm rate often associated with IDSs. From the tests we conducted, the Random Tree has the highest accuracy of all the seven algorithms. We expect that the findings of this study will contribute to the improvement of the performance of IDSs.

**Keywords**

Machine learning, intrusion detection systems, threat intelligence, cybersecurity.

**1. Introduction**

Cyber threats are evolving each day; the attackers are sharing their information with each other to increase the effectiveness of their attacks. Currently, the targets of cyberattacks are network environments like enterprises, governments, and countries (He 2021) [1]. Thus, using network security and threat intelligence, which is data that is collected, processed, and analyzed to understand attack behaviors to prevent cyberattacks, are necessary for them to protect their information from the attack.

These days, there are a number of types of cyberattacks, such as phishing, malware, and crypto mining, and those numbers keep increasing. To protect users from various attacks, a lot of security systems have been developed in the past years; one of the systems is intrusion detection systems (IDSs). IDS is a system that is used to detect unusual activity in a network of computer systems to identify if an activity is unfriendly or unauthorized in order to enable a response to that violation. The system has three main detection mechanisms: statistical method, data-mining method, and machine learning method. However, according to a computer scientist, Tadeusz Pietraszek (2004) [2], 99% of alerts reported by IDSs do not involve cybersecurity issues. This is causing real attacks to go often unnoticed. From the problem, our research team came up with this question: How effective are various machine learning algorithms in ensuring network security and enhancing IDSs?

Our research is going to focus on one of the most powerful mechanisms in IDS, the machine learning method, to solve the IDS problem. Machine learning is a field of study that gives computers the ability to learn and improve from experience without the need of being programmed. The machine learning-based IDS can achieve satisfactory detection levels when sufficient training data is available, and machine learning models have sufficient generalizability to detect attack variants and novel attacks. In addition, machine learning-based IDSs do not rely heavily on domain knowledge; therefore, they are easy to design and construct.

The goal of this research is to find the most accurate machine learning algorithm that can increase IDSs efficiency through automation. By achieving our research goal, we are hoping that our research will provide useful information to advance network security.

This paper is structured as follows. Section 2 presents some related works to our research. Section 3, the methodology, introduced how we conducted the research to achieve our goal. Section 4 shows the results of our research and discusses them. Finally, in section 5, we summarize our research and outlook on future work.

**2. Literature Review**

There are several existing studies that seek to improve cybersecurity by using intrusion detection systems (IDSs). This section is going to review a number of them.

He (2021) [1] mentions how there is a continuous development of information technology and that the network environment is becoming increasingly complex; both of these aspects have led to huge potential risks to network security. The author combined machine learning-related technologies to improve intrusion detection performance, alarm correlation automation, and investigate key technologies.

Pietraszek (2004) [2] analyzes the problem of false positives in IDSs by using a prototype implementation of the Adaptive Learner for Alert Classification (ALAC), which is a novel system for reducing false positives in intrusion detection. In the research, he estimated that up to 99% of alerts reported by IDSs do not involve cybersecurity issues due to the observed slight differences between normal and malicious activities.

Fang et al. (2020) [3] describe how intrusion detection technology has become an important means to ensure network security and the problem of false alarm rates in IDSs. The research uses intrusion detection of robust support vector machines (SVM) neighbor classification to solve the problems of network security. The results were that the intrusion detection based on robust SVM neighbor classification could achieve an 87.3% detection rate, and when the false alarm rate was 2.8%, the detection rate was 100%.

Tungjaturasopon and Piromosopa (2018) [4] mention what are the proper machine learning algorithms for intrusion detection in various environments by checking the processing time. They tested three types of machine learning algorithms, decision tree, support vector machine (SVM), and multi-layer perceptron (MLP), by using an intrusion detection dataset, the NSL-KDD dataset. As a result, they discovered that decision trees and SVM use similar process time and faster than MLP to build a model, but the SVM performs slower than decision trees for larger workloads.

**3. Methodology**

This section is divided into two parts: procedure, and analysis for our research.

**3.1 Procedure**

This research used the data from the digital libraries, Galileo, ACM, and IEEE, operated by organizations. The resources should contain some if not all of the following keywords: cybersecurity, machine learning, threat intelligence, network security, and intrusion detection system/intrusion prevention system. By using the resources, we collected information on intrusion detection (ID) datasets and selected a couple of ID datasets that contained at least 100,000 records of two types, normal and malicious, but not more than 300,000 records of them. This is to avoid the need of selecting a small portion of the record randomly when we operate an experiment by choosing a dataset with too many records. The datasets also have to contain more than three types of attacks records and each dataset should contain at least one type of the same type of attack record. After we selected some datasets, we used Microsoft Excel to understand, clean, and organize the datasets. This will help to find and remove duplicate records that could affect our research.

**3.2 Analysis**

This research used a data analysis tool, WEKA 3.8.5, to achieve our research goal. WEKA is a visualization tool that can be used for data preparation, classification, regression, clustering, and association. It is developed by the University of Waikato, New Zealand. WEKA has a number of machine learning algorithms that can be used in a dataset. In this research, we used seven algorithms as follows: four tree-based algorithms, one support vector machine algorithm, the Naïve Bayes algorithm, and one K-nearest neighbors algorithm. The datasets we chose will be converted to ARFF file format by using WEKA’s tool, ArffViewer, to load the datasets in WEKA.

**3.2.1 Tree-Based Algorithms**

A tree-based algorithm is a type of algorithm that uses a tree structure flowchart that follows an IF-THEN rule to generate predictions. All tree-based algorithms can be used to solve classification and regression challenges. The four tree-based algorithms we picked are below:

• **J48** [5] - An open-source java implementation of the C4.5 algorithm which was developed by J. Ross Quinlan. It split every aspect of information into minor subsets to solve a problem.

• **Random Forest** [6] - A supervised learning classifier that has multiple decision trees. It searches for the best feature among a random subset.

• **Random Tree** [7] - A supervised learning classifier that builds a tree that selects a K number of attributes randomly at each node.

• **REP Tree** [7] **-** A machine learning algorithm that creates multiple trees in different iterations. It is a fast decision tree learner.

**3.2.2 Support Vector Machines (SVM)**

SVM [8] is a supervised learning method that can be used for both classification and regression challenges. It is based on the idea of finding the best hyperplane that divides a dataset into two classes. A hyperplane is a subspace that has one less dimension than its ambient space; for two-dimensional space, the hyperplane will be a one-dimensional space. The SVM algorithm we picked is sequential minimal optimization (SMO).

• **Sequential Minimal Optimization (SMO)** [9] - A machine learning algorithm that was invented by American computer scientist, John Platt. The algorithm can handle large datasets and is one of the fastest SVM algorithms.

**3.2.3 Naïve Bayes**

Naïve Bayes [10] is a supervised machine learning algorithm that uses the Bayes theorem, which is used to calculate a conditional probability. The algorithm assumes that each input variable is independent. It is easy to build and useful for large data sets.

**3.2.4 K-Nearest Neighbors (KNN)**

KNN [11] is a supervised machine learning algorithm that can be used for both classification and regression problems. The classifiers determine the class of a data point by the majority voting principle. For example, if K = 10, the classes of 10 closest points are checked. The KNN algorithm we picked for our research is below.

• **IBk** [12] **-** A machine learning algorithm that can select a suitable value of K based on cross-validation and can weight the distance.

To check the accuracy of the above seven classifiers, we loaded the training datasets in WEKA and ran the classifiers by supplying each testing dataset of the intrusion detection datasets that we chose.

**4. Results and Discussions**

This section has two parts: part one talks about intrusion detection datasets that we used, and part two presents the overall results of classifiers accuracy.

**4.1 Intrusion Detection Datasets**

By researching a number of data from online databases, we decided to use two IDS datasets, NSL-KDD and UNSW-NB15 for our research.

**NSL-KDD Dataset**

NSL-KDD [13] is a public IDS dataset that has been developed by Tavallaee et al. It comprises 22 training intrusion attacks and 41 attributes. The purpose of developing this dataset is to solve the problems of another IDS dataset KDD '99. The main problem in the KDD’99 dataset is the huge number of duplicate packets; approximately 78% and 75% of the network packets are duplicated in both the training and testing dataset (Tavallaee et al., 2009). This could lead machine learning methods to be biased towards normal instances and prevent them from learning irregular instances that are usually more damaging to the computer system. Thus, the NSL-KDD dataset does not have duplicate records; the training dataset consists of 125,973 records and the testing dataset contains 22,544 records. The number of records in the NSL-KDD dataset is much lower than the KDD’99 dataset, however, the number of records in the train and test sets are reasonable; it makes the dataset more practical to use the whole dataset without the necessity to sample randomly. The dataset contains four different types of attacks, DoS, probe, R2L, and U2R. Our research used two files, KDDTrain+ and KDDTest+. The KDD train+

|  |  |  |  |
| --- | --- | --- | --- |
| Name of Classifier | Accuracy | | Rank |
| Correctly Classified  Instances | Incorrectly Classified  Instances |
| J48 | 81.5339 % | 18.4661 % | 1 |
| Random Forest | 80.4516 % | 19.5484 % | 4 |
| Random Tree | 81.3565 % | 18.6435 % | 3 |
| REP Tree | 81.5073 % | 18.4927 % | 2 |
| SMO | 75.3948 % | 24.6052 % | 7 |
| Naïve Bayes | 76.1222 % | 23.8778 % | 6 |
| IBk | 79.3559 % | 20.6441 % | 5 |

Table 1: Results of the NSL-KDD Dataset

Table 1 presents the results of classifier accuracy when we run the classifiers on the NSL-KDD dataset. For this dataset, J48 achieved the highest correctly classified rates, 81.5339 %, followed by REP Tree, 81.5073 %, Random Tree, 81.3565 %, Random Forest, 80.4516 %, and others. Thus, every tree-based algorithm performed better than other algorithms on the NSL-KDD dataset.

**UNSW-NB15 Dataset**

UNSW-NB15 [14] [15] [16] [17] [18] is a public dataset that encompasses normal and malicious records. The packet of the dataset was created by using the IXIA Perfect Storm tool in the Cyber Range Lab of the University of New South Wales Canberra. The dataset contains nine types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. Our research used the training dataset which has 175,341 records and the testing dataset which has 82,332 records from different types, malicious and normal, to achieve our research goal.

|  |  |  |  |
| --- | --- | --- | --- |
| Name of Classifier | Accuracy | | Rank |
| Correctly Classified  Instances | Incorrectly Classified  Instances |
| J48 | 70.9894 % | 29.0106 % | 7 |
| Random Forest | 83.722 % | 16.278 % | 2 |
| Random Tree | 84.6512 % | 15.3488 % | 1 |
| REP Tree | 80.8556 % | 19.1444 % | 5 |
| SMO | 80.9418 % | 19.0582 % | 4 |
| Naïve Bayes | 76.6931 % | 23.3069 % | 6 |
| IBk | 83.0941 % | 16.9059 % | 3 |

Table 2: Results of the UNSW-NB15 Dataset

Table 2 presents the results of classifier accuracy when we run the classifiers on the UNSW-NB15 dataset. Random Tree achieved the highest correctly classified rates, 84.6512 %, followed by Random Forest, 83.722 %, IBk, 83.0941 %, SMO, 80.9418 %, and others.

**4.2 Overall Results**

To determine the overall results, we calculated the mean of each classifier’s results that we obtained by running the classifiers on the NSL-KDD and UNSW-NB15 datasets.

|  |  |  |  |
| --- | --- | --- | --- |
| Name of Classifier | Accuracy | | Rank |
| Mean of  Correctly Classified  Instances | Mean of  Incorrectly Classified  Instances |
| J48 | 76.26165 % | 23.73835 % | 7 |
| Random Forest | 82.0868 % | 17.9132 % | 2 |
| Random Tree | 83.00385 % | 16.99615 % | 1 |
| REP Tree | 81.18145 % | 18.81855 % | 4 |
| SMO | 78.1683 % | 21.8317 % | 5 |
| Naïve Bayes | 76.40765 % | 23.59235 % | 6 |
| IBk | 81.225 % | 18.775 % | 3 |

Table 3: Means of the Two Results (Table 1 and Table 2)

Table 3 presents the means of correctly classified rates and incorrectly classified rates from the results in table 1 and table 2. As we can see from table 3, an algorithm that achieved the highest accuracy overall was Random Tree, 82.0868 %, followed by Random Forest, 82.0868 %, IBk, 81.225 %, and others.

**5. Conclusion**

In this paper, we tested seven machine learning algorithms, J48, Random Forest, Random Tree, REP Tree, SMO, Naïve Bayes, and IBk, in WEKA by using two datasets, NSL-KDD, and UNSW-NB15 datasets. From the results, we found that Random Tree achieved the highest accuracy in all seven algorithms we tested. Thus, to create the most accurate IDSs by using one from the seven algorithms, we recommend using Random Tree.

Currently, our experiments are limited to tree-based algorithms, SVM algorithm, Naïve Bayes algorithm, and KNN algorithm. In the future, we should continue researching more types of algorithms and their performance, such as processing time. By doing this, we are hoping to develop an efficient machine learning algorithm that integrates threat intelligence into intrusion detection systems to increase their efficiency through automation.

**References**

[1] He, Y. (2021). Research on the Key Technology of Network Security Based on Machine Learning. *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 972–975. <https://doi.org/10.1109/ICSP51882.2021.9408756>.

[2] Pietraszek, T., (2004). Using adaptive alert classification to reduce false positives in intrusion detection. *International Workshop on Recent Advances in Intrusion Detection*, 102-124. <https://link.springer.com/chapter/10.1007/978-3-540-30143-1_6>.

[3] Fang, W., Tan, X., Wilbur, D., Elhoseny, M., & Yuan, X. (2020). Research on machine learning method and its application technology in intrusion information security detection. *Journal of Intelligent & Fuzzy Systems*, 38(2), 1549–1558. <https://doi.org/10.3233/JIFS-179518>.

[4] Tungjaturasopon, P., & Piromsopa, K. (2018). Performance Analysis of Machine Learning Techniques in Intrusion Detection. *Proceedings of the 2018 VII International Conference on Network, Communication and Computing*, 6-10. <https://dl.acm.org/doi/10.1145/3301326.3301335>.

[5] Quinlan, R., (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc.

[6] Breiman, L., (2001). Random Forests. *Machine Learning*, 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324>.

[7] Asha Kiranmai, S., & Jaya Laxmi, A., (2018). Data mining for classification of power quality problems using WEKA and the effect of attributes on classification accuracy. *Prot Control Mod Power Syst 3*, 29. <https://doi.org/10.1186/s41601-018-0103-3>.

[8] Cortes, C., & Vapnik, V., (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>.

[9] Platt, J. (1998). Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. *Microsoft*. <https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/>.

[10] John, G. H., & Langley P., (1995). Estimating continuous distributions in Bayesian classifiers. In P. Besnard, & S. Hanks (Eds.), *Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 338-345). Morgan Kaufmann Publishers Inc.

[11] Altman, N. S., (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46 (3), 175–185. <https://doi.org/10.2307/2685209>.

[12] Aha, D., & Kibler, D., (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37-66. <https://doi.org/10.1007/BF00153759>.

[13] Tavallaee, M., et al. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 1-6. <https://doi.org/10.1109/CISDA.2009.5356528>.

[14] Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *Military Communications and Information Systems Conference (MilCIS)*, 1-6. <https://doi.org/10.1109/MilCIS.2015.7348942>.

[15] Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset. *Information Security Journal: A Global Perspective*, 1-14. <https://doi.org/10.1080/19393555.2015.1125974>.

[16] Moustafa, N., et al. (2017). Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. *IEEE Transactions on Big Data*, 5(4), 481-494. <https://doi.org/10.1109/TBDATA.2017.2715166>.

[17] Moustafa, N., et al. (2017). Big data analytics for intrusion detection system: statistical decision-making using finite dirichlet mixture models. *Data Analytics and Decision Support for Cybersecurity*, 127-156. <https://doi.org/10.1007/978-3-319-59439-2_5>.

[18] Sarhan, M., et al. (2020). NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. *Big Data Technologies and Applications*, 117-135. <https://doi.org/10.1007/978-3-030-72802-1_9>.