DNN-BASED INTELLIGENT INTRUSION DETECTION SYSTEM

Ms. Usha Maheshwari [1], A. Bhargavi [2], M. Vaswitha [3], P. Manovya [4],

[1] Assistant Professor, Malla Reddy Engineering College for Women (Autonomous Institution) Hyderabad.

[2] [3] [4] Student, Malla Reddy Engineering College for Women (Autonomous Institution) Hyderabad.

ABSTRACT: *In order to automatically and quickly identify and categorize cyber-attacks at the host and network levels, machine learning techniques are being utilized extensively in the development of intrusion detection systems (IDS). But no prior research has demonstrated a comprehensive evaluation of the efficacy of different ML algorithms on a variety of open-source datasets. This research delves into the exploration of deep neural networks (DNNs), a subset of deep learning models, with the goal of creating adaptable and efficient intrusion detection systems (IDS) capable of detecting and categorizing previously unseen cyber-attacks. Due to the ever-evolving nature of both network activity and assaults, it is essential to assess different datasets produced throughout time using both static and dynamic methods. Research of this kind helps to find the most effective algorithm for spotting future cyberattacks. On several publicly accessible benchmark malware datasets, a thorough evaluation of trials including DNNs and other traditional machine learning classifiers is demonstrated. Incorporating the IDS data into our DNN model's numerous hidden layers allows it to learn the features' abstract and high-dimensional representation. It has been proven via extensive experimental testing that DNNs outperform standard machine learning classifiers. Last but not least, we provide Scale-Hybrid-IDS-AlertNet (SHIA), a framework for hybrid DNNs that can be utilized in real-time to successfully monitor host-level events and network traffic in order to proactively notify potential cyber-attacks.*

INTRODUCTION

Many different types of assaults, both from inside and outside the organization, can compromise the sensitive user data that is handled by information and communications technology (ICT) systems and networks. Data breaches go undiscovered because these assaults, which can be either human or computer-generated, are varied, and are getting better at obfuscation. Take

Yahoo as an example; their data breach cost $350 million, while the Bitcoin hack cost an estimated $70 million. As computing power, operating systems, and network architectures like the Internet of Things (IoT) continue to grow, so do the algorithms used in these types of assaults. A new intrusion detection system (IDS) that is more adaptable, dependable, and innovative is necessary due to the significant security risks posed by malicious cyber-attacks. Network- and host-level intrusions, assaults, and policy violations may be quickly and automatically identified with the help of an intrusion detection system (IDS). Intrusion detection systems are categorized into two main types: host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS). Network intrusion detection systems (NIDS) are a type of IDS that employ network behavior. To detect assaults and potential dangers hidden in network traffic, networking devices like switches, routers, and network taps duplicate network behaviors and analyze them.

A host intrusion detection system (HIDS) is an intrusion detection system (IDS) that detects assaults by analyzing system activity recorded in log files generated by the local host computer. Local sensors are used to gather the log data. In contrast to NIDS, which examines the contents of each packet in network traffic flows, HIDS makes use of data contained in log files. These files contain information on sensors, systems, software, file systems, disk resources, users' accounts, and more. A combination of NIDS and HIDS is used by many organizations. Misuse detection, anomaly detection, and stateful protocol analysis are utilized for the purpose of analyzing network traffic flows. To identify threats, misuse detection makes use of filters and signatures that have already been set. The signature database is continuously updated by human contributions. Finding known assaults is where this strategy really shines, however it falls flat when it comes to unknown strikes. In order to uncover undiscovered harmful actions, anomaly detection employs heuristic algorithms. A significant false positive rate is produced by anomaly detection in the majority of circumstances. Most companies' commercial solution solutions employ a mix of misuse and anomaly detection to tackle this issue. Since stateful protocol analysis operates on the network, application, and transport layers, it outperforms the previously listed detection approaches. To identify inappropriate protocol and application variations, this makes advantage of the specified vendor standard settings. There has been little research to compare these machine learning algorithms with publically available datasets, even though deep learning methods are being evaluated as a means to improve the intelligence of intrusion detection systems.

The current state of machine learning-based solutions has several common problems. Firstly, they are prone to false positives from a variety of attacks. Secondly, most studies have only used one dataset to evaluate the models' performance. Lastly, the models that have been studied so far do not account for the massive network traffic that exists today. Lastly, solutions are needed to keep up with the ever-increasing size, speed, and dynamics of today's high-speed networks. These difficulties are the driving force behind this study, which aims to assess the performance of several deep neural networks (DNNs) and standard machine learning classifiers when used to NIDS and HIDS. The following is assumed in this work: Attackers try to fool intrusion detection systems by seeming to be legitimate users. Intruding behavior patterns, however, vary in one way or another. This is because each hacker has their own unique goal, such as gaining unauthorized access to systems and networks. While it is possible to record how network resources are being used, the current approaches produce a large number of false positives. Intruder patterns with a low profile for extended periods of time are seen in regular traffic.

RELATED WORK

An exhaustive examination and evaluation of intrusion detection using machine learning techniques.

Nowadays, intrusion detection is a major concern when it comes to cyber security. Many methods have been created that rely on machine learning strategies. Having said that, they do a poor job of detecting every kind of intrusion. This study delves deeply into the topic of machine learning approaches in order to uncover the root causes of their shortcomings when it comes to identifying invasive behaviors. Each assault is accompanied with a mapping of its characteristics and a classification of the attack. We also address and provide workable solutions to problems associated with identifying low-frequency assaults in network attack datasets. There has been research and comparison of machine learning methods for attack detection capabilities across different types of assaults. Additionally, we cover the limitations that come with each grouping of them. There are a number of data mining tools for ML covered in the article as well. Finally, some suggestions for where the field of attack detection using ML may go from here are included.

Botnet Identification Traffic Utilizing Machine Learning

With the expansion of the Internet, more and more sophisticated, publicly available, and easy-to-use toolkits for computer assaults and intrusions have emerged, such as the Zeus botnet toolkit. Spam, DDoS, identity theft, and phishing are just a few cyber-attacks that botnets are responsible for. The vast majority of botnet toolkits out there regularly issue updates that include new features, enhancements, and support. The detection and prevention of bots are made more difficult by this. The topology of botnets can vary from centralized (like IRC or HTTP) to distributed (like P2P) and encryption-resistant, rendering current botnet detection methods mainly useless. Preliminary study on anticipating new bots before they begin their attack is presented in this paper based on real world data sets. Based on the Classification of Network Information Flow Analysis (CONIFA) framework, we present a comprehensive collection of network traffic characteristics that may identify patterns in C&C communication channels and malicious traffic. A prominent botnet toolset, Zeus, is the subject of our case study on how to utilize the method. Based on the results of the experimental evaluation, it appears that botnets can be successfully detected before they launch attacks by analyzing traffic behaviors and by constructing a classifier using machine learning from an earlier version of the Zeus botnet toolkit. This detection can be done during the botnet C&C communication. It should be demonstrated that different versions of the Botnet toolkit have comparable C&C structures and that the network characteristics of botnet C&C traffic vary from regular network traffic. These techniques have the potential to lessen the burden on various resources that are now used to detect malicious traffic and C&C communication channels.

Applying ML and DL Techniques to Cyber Defense

The cyber security landscape is dark and evolving at a rate comparable to the rate at which the Internet is expanding. With a concise instructional overview of each ML/DL approach, this survey report outlines important literature surveys on ML/DL methods for network analysis of intrusion detection. We indexed, reviewed, and summarized papers from each technique according to their thermal or temporal correlations. Given the centrality of data to ML/DL methodologies, we outline many popular network datasets for ML/DL, talk about the difficulties of using ML/DL to cybersecurity, and offer some recommendations for further study.

An method for detecting intrusions on networks based on Adaboost.

Differentiating between malicious and benign Internet activity is the primary goal of network intrusion detection systems. Its role in the information security framework is critical. Quick, accurate, and machine-learning-based intrusion detection algorithms are required because of the wide range of network behaviors and the ever-changing nature of attack styles. In this letter, we provide an AdaBoost–based intrusion detection method. Weak classifiers are decision stumps in the algorithm. There are decision rules for both continuous and categorical features. Without the need for forced conversions between continuous and categorical characteristics, the links between these two types of features may be handled naturally by merging the weak classifiers for continuous and categorical features into a strong classifier. To enhance the algorithm's effectiveness, we use starting weights that may be adjusted and a straightforward method to prevent overfitting. Using the benchmark sample data, we were able to demonstrate that our approach outperforms algorithms with higher computational complexity in terms of both error rates and computational complexity.

Intrusion detection in computer networks using ensemble learning.

One of the most important aspects of contemporary computer systems is the security of their networks. There are a lot of software technologies in development right now that aim to impose high levels of security against attacks. If an attacker manages to bypass the "first line" of defense, an intrusion detection system will be able to catch them. Network intrusion detection using pattern recognition and ensemble learning methods is presented in this study. We highlight the possibilities of this strategy for data fusion and discuss some unresolved concerns.

METHODOLOGY

Classifying network intrusion threats utilizing a Deep Neural Network (DNN), Random Forest, and Support Vector Machines (SVM) is the main emphasis of this study. Data preparation, model training, and assessment are all part of the process. The methods, laid out in detail below:

1. Upload NSL KDD Dataset

The first step involves uploading the NSL KDD dataset, which contains various network intrusion records. This dataset is crucial for training and testing the models.

Steps:

- The user double-clicks on the run.bat file to launch the application.
- The user clicks on the "Upload NSL KDD Dataset" button.
- The application loads the dataset, and the user can view the uploaded data on the interface.

2. Preprocess Dataset

Numeric data is necessary for machine learning algorithms. In order to prepare the data for model training, this module gives numerical values to categorical characteristics such attack names.

Steps:

- The user clicks on the "Preprocess Dataset" button.
- The system processes the dataset, converting all string-based attack names into numeric IDs.
- The preprocessed data is displayed, showing the newly assigned numeric IDs for each attack type.

3. Generate Training Model

By dividing the dataset into a training set and a testing set, this module gets the dataset ready for training. The dataset has a structure that works well with the methods used for machine learning.

Steps:

- The user clicks on the "Generate Training Model" button.
- The system organizes the dataset into training and testing sets.
- The formatted dataset is displayed, indicating that it is ready for model training and testing.

4. Run SVM Algorithm

Applying the Support Vector Machine (SVM) algorithm on the prepped dataset is the next stage. After training the SVM model, the accuracy of its predictions is computed and shown.

Steps:

- The user clicks on the "Run SVM Algorithm" button.
- The SVM model is trained using the training dataset.
- The application displays the prediction accuracy of the SVM model, which in this case is 52%.

5. Run Random Forest Algorithm

The dataset is subjected to the Random Forest method, which is similar to the SVM technique. After training, the model's accuracy is computed and shown.

Steps:

- The user clicks on the "Run Random Forest Algorithm" button.
- The Random Forest model is trained using the same dataset.
- The application displays the prediction accuracy of the Random Forest model, which is also 52%.

6. Run DNN Algorithm

The Deep Neural Network (DNN) algorithm is executed, which typically performs better due to its ability to capture complex patterns in the data through multiple hidden layers.

Steps:

- The user clicks on the "Run DNN Algorithm" button.
- The DNN model is trained using the dataset. The number of hidden layers, specified in the code, is set to 8.
- The application displays the DNN model's prediction accuracy, which is higher than the other algorithms. The accuracy may vary across different runs due to the random selection of hidden layers.

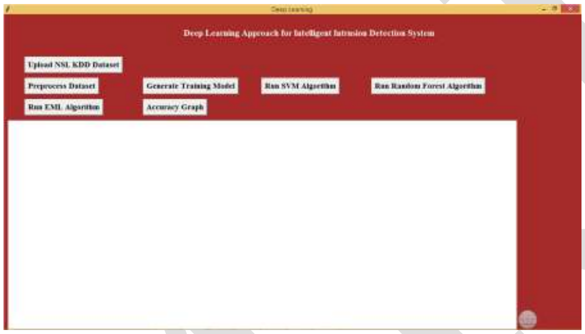
7. Display Accuracy Graph

Finally, the accuracy of all the algorithms (SVM, Random Forest, DNN) is compared visually using a bar graph. This helps to clearly demonstrate the superior performance of the DNN algorithm.

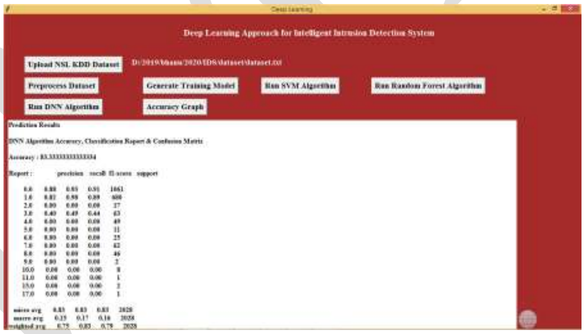
Steps:

- The user clicks on the "Accuracy Graph" button.
- The graph is displayed, with the X-axis representing the algorithm names and the Y-axis representing their respective accuracies.
- The DNN algorithm, being the proposed technique, shows a higher accuracy compared to SVM and Random Forest.

RESULT AND DISCUSSION



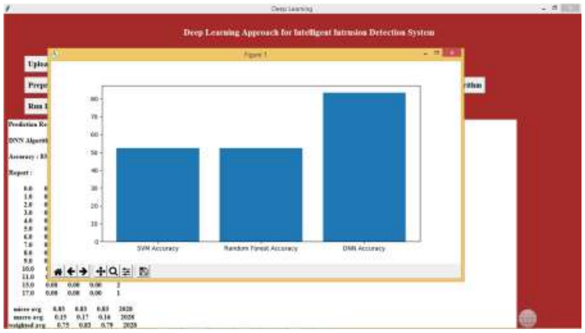
In above screen click on ‘Upload NSL KDD Dataset’ button to upload dataset



D:/2019/bhanu/2020/IDS/dataset/dataset.txt

```
Prediction Results

DNN Algorithm Accuracy, Classification Report & Confusion Matrix

Accuracy : 83.33333333333334

Report :          precision    recall  f1-score   support

         0.0       0.88      0.95      0.91      1061
         1.0       0.81      0.98      0.89       600
         2.0       0.00      0.00      0.00        37
         3.0       0.40      0.49      0.44        43
         4.0       0.00      0.00      0.00        49
         5.0       0.00      0.00      0.00        11
         6.0       0.00      0.00      0.00        25
         7.0       0.00      0.00      0.00        42
         8.0       0.00      0.00      0.00        46
         9.0       0.00      0.00      0.00         2
        10.0       0.00      0.00      0.00         8
        11.0       0.00      0.00      0.00         1
        15.0       0.00      0.00      0.00         2
        17.0       0.00      0.00      0.00         1

   micro avg       0.83      0.83      0.83      2028
   macro avg       0.15      0.17      0.16      2028
weighted avg       0.75      0.83      0.79      2028
```

In above screen we can see DNN accuracy is better than other two algorithms. DNN algorithm accuracy may be vary different times as it hidden layer will be chosen randomly from dataset. Now click on ‘Accuracy Graph’ button to get below graph



In above graph x-axis represents algorithm name and y-axis represents accuracy and DNN is the propose technique. In below code screen you can see i specify DNN hidden layer as 8

CONCLUSION

In this study, we provide a hybrid IDS/IPS system that can monitor both host and network activity, built on a scalable architecture that runs on commodity hardware servers. To manage and analyze massive amounts of data in real-time, the system used a distributed deep learning model using DNNs. A thorough evaluation of the DNN model's performance in contrast to traditional machine learning classifiers on several benchmark IDS datasets was conducted in order to choose it. And to use the suggested DNN model for intrusion and attack detection, we also used real-time feature collection based on hosts and networks. We found that DNNs outperformed traditional machine learning classifiers in every single instance. In both HIDS and NIDS, our suggested design outperforms traditional ML classifiers that were previously in use. We have not found any other framework that can distributely gather host-level and network-level actions using DNNs for more accurate attack detection, so far as we are aware. Including a module to track DNS and BGP events in the networks will significantly improve the performance of the suggested framework. The execution time of the suggested system may be increased by adding more nodes to the existing cluster. Not to mention that the suggested system is vague when it comes to describing the malware's structure and traits. Training complicated DNNs designs on modern hardware using a distributed way can considerably enhance performance. This study did not train DNNs using the benchmark IDS datasets because of the high computational cost of their complicated structures. Thought to be a major avenue for future research, this will be a crucial undertaking in a hostile ecosystem.

REFERENCES

[1] Mukherjee, B., Heberlein, L. T., & Levitt, K. N. (1994). Network intrusion detection. IEEE network, 8(3), 26-41.

[2] Larson, D. (2016). Distributed denial of service attacks-holding back the flood. Network Security, 2016(3), 5-7.

[3] Staudemeyer, R. C. (2015). Applying long short-term memory recurrent neural networks to intrusion detection. South African Computer Journal, 56(1), 136-154.

[4] Venkatraman, S., Alazab, M. "Use of Data Visualisation for Zero-Day Malware Detection," Security and Communication Networks, vol. 2018, Article ID 1728303, 13 pages, 2018. https://doi.org/10.1155/2018/1728303

[5] Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Communications Surveys & Tutorials.

[6] Azab, A., Alazab, M. & Aiash, M. (2016) "Machine Learning Based Botnet Identification Traffic" The 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (Trustcom 2016), Tianjin, China, 23-26 August, pp. 1788-1794.

[7] Vinayakumar R. (2019, January 19). vinayakumarr/Intrusion-detection v1 (Version v1). Zenodo. http://doi.org/10.5281/zenodo.2544036

[8] Tang, M., Alazab, M., Luo, Y., Donlon, M. (2018) Disclosure of cyber security vulnerabilities: time series modelling, International Journal of Electronic Security and Digital Forensics. Vol. 10, No.3, pp 255 - 275.

[9] V. Paxson. Bro: A system for detecting network intruders in realtime. Computer networks, vol. 31, no. 23, pp. 24352463, 1999. DOI http://dx.doi. org/10.1016/S1389-1286(99)00112-7

[10] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436.

[11] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... &Wang, C. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access.

[12] Hofmeyr, S. A., Forrest, S., & Somayaji, A. (1998). Intrusion detection using sequences of system calls. Journal of computer security, 6(3), 151180.

[13] Forrest, S., Hofmeyr, S. A., Somayaji, A., & Longstaff, T. A. (1996, May). A sense of self for unix processes. In Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on (pp. 120-128). IEEE.

[14] Hubballi, N., Biswas, S., & Nandi, S. (2011, January). Sequencegram: n-gram modeling of system calls for program based anomaly detection. In Communication Systems and Networks (COMSNETS), 2011 Third International Conference on (pp. 1-10). IEEE.

[15] Hubballi, N. (2012, January). Pairgram: Modeling frequency information of lookahead pairs for system call based anomaly detection. In Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on (pp. 1-10). IEEE.