Int. Jrl. of Computational Learning & Intelligence
Vol. 02, Issue No. 01, 2023

ORIGINAL RESEARCH

Machine Learning for Net Flow Based Anomaly Intrusion Detection System Using Neural Network Stages

S Swarna Keerthi . Konampeta Sai Srija . Peddigari Sai Pavan . Karre Prakash

Information Technology, Institute of Aeronautical Engineering, Hyderabad 500043, India

Received: 03 November 2022 / Revised: 30 November 2022 / Accepted: 06 January 2023

Abstract – Computer systems and networks suffer due to rapid increase of attacks, and in order to keep them safe from malicious activities or policy violations, there is need for effective security monitoring systems, such as Intrusion Detection Systems (IDS). Many researchers concentrate their efforts on this area using different approaches to build reliable intrusion detection systems. Flow-based intrusion detection systems are one of these approaches that rely on aggregated flow statistics of network traffic. Their main advantages are host independence and usability on high speed networks, since the metrics may be collected by network device hardware or standalone probes. In this paper, an intrusion detection system using two neural network stages based on flow-data is proposed for detecting and classifying attacks in network traffic. The first stage detects significant changes in the traffic that could be a potential attack, while the second stage defines if there is a known attack and in that case classifies the type of attack. The first stage is crucial for selecting time windows where attacks, known or unknown, are more probable. Two different neural network structures have been used, multilayer and radial basis function networks, with the objective to compare performance, memory consumption and the time required for network training. The experimental results demonstrate that the designed models are promising in terms of accuracy and computational time, with low probability

Index Terms – Intrusion Detection system, Anomaly Detection System, Neural network

I. INTRODUCTION

With the rapid growth of the Internet and due to increase in number of attacks, computer security has Additionally a flow includes aggregated information become a crucial issue for computer systems. Intrusion Detection Systems (IDS) technology is an effective approach in dealing with the problems of network security. Intrusion Detection Systems (IDS) have become increasingly important in recent years to reveal the growing number of attacks. They need to be able to adapt to the rise in the amount of traffic

A flow is defined as a unidirectional stream of packets that share common characteristics, such as source and destination addresses, ports and protocol type. about the number of packets and bytes belonging to the stream, as well as its duration. Flows data are often used for network monitoring, allowing us to obtain a real time overview of the network traffic as well as the increase in line speed. However, researchers assess the payload-based IDSs processing capability to lie between 100 Mbps and 200 Mbps when commodity hardware is used [2, 3], and exhibit high resource consumption when confronted with the overwhelming amount of data found in high-speed networks.

In addition, the spread of encrypted protocols poses a new challenge to payload-based systems. In addition to that, the constant increase in network traffic and the fast introduction of high speed (tens of Gbps) network equipment make it hard to preserve traditional packet based intrusion detection systems. Such systems rely on deep packet inspection, which does not scale well. Having this in mind, flow-based approaches seem to be a promising candidate for research in the area of IDS.

II. LITERATURE SURVEY

Intrusion detection is an important area of research. Traditionally, the approach taken to find attacks is to inspect the contents of every packet. However, packet inspection cannot easily be performed at high-speeds. Therefore, researchers and operators started investigating alternative approaches, such as flow- based intrusion detection. In that approach the flow of data through the network is analyzed, instead of the contents of each individual packet. The goal of this paper is to provide a survey of current research in the area of flow-based intrusion detection. The survey starts with a motivation why flow-based intrusion detection is needed. The concept of flows is explained, and relevant standards are identified. The paper provides a classification of attacks and defense techniques and shows how flow-based techniques can be used to detect scans, worms, Botnets and (DoS) attacks.

The process speed of network-based intrusion detection systems (NIDSs) is still low compared with the speed of networks. As a result, few NIDS is applicable in a high-speed network. A parallel NIDS for high-speed networks is presented in this paper. By dividing the overall traffic into small slices, several sensors can analyze the traffic concurrently and significantly increase the process speed. For most attacks, our partition algorithm ensures that a single slice contains all the evidence necessary to detect a specific attack, making sensor-to-sensor interaction unnecessary. Meanwhile, by making use of the character of the network traffic, the algorithm can also dynamically balance all sensors' loads. To keep the system as simple as possible, a specific sensor is used to detect the scan and the DoS attack. Although only one sensor is used for this kind of attacks, we argue that our system can still provide high processability.

With the rapid expansion of computer networks during the past few years, security has become a crucial issue for modern computer systems. A good way to detect illegitimate use is through monitoring unusual user activity. Methods of intrusion detection based on hand-coded rule sets or predicting commands on-line are laborious to build or not very reliable. This paper proposes a new way of applying neural networks to detect intrusions. We believe that a user leaves a 'print' when using the system; a neural network can be used to learn this print and identify each user much like detectives use thumbprints to place people at crime scenes. If a user's behavior does not match his/her print, the system administrator can be alerted of a possible security breech. A back propagation neural network called NNID (Neural Network Intrusion Detector) was trained in the identification task and tested experimentally on a system of 10 users. The system was 96% accurate in detecting unusual activity, with 7% false alarm rate. These results suggest that learning user profiles is an effective way for detecting intrusions.

Intrusion detection (ID) is an interesting approach that could be used to improve the security of network systems. IDS detects suspected patterns of network traffic on the remaining open parts through monitoring user activities (runtime gathering of data from system operations), and the subsequent analysis of these activities. The purpose of this work is to contribute ideas of finding a solution to detect attacks (intrusion) through building artificial detection system using feed forward neural networks to detect attacks with low false negative rate (which is the most important point), and low false positive rate. To do so, two feed forward neural networks architectures (one for non fuzzified data, the other for fuzzified data) are suggested, and their behaviors in detecting the attacks are studied. In this research, the suggested IDS not only has the ability to distinguish if the access is normal or attack, but also capable of distinguishing the attack type.

Global-scale attacks like viruses and worms are increasing in frequency, severity and sophistication, making it critical to detect outbursts at routers/gateways instead of end hosts. In this paper we leverage data streaming techniques such as the reversible sketch to obtain HiFIND, a High- speed Flow-level Intrusion Detection system. In contrast to existing intrusion detection systems, HiFIND is scalable to flow-level detection on high-speed networks; zs DoS resilient; can distinguish SYN flooding and various port scans (mostly for worm propagation) for effective mitigation; enables aggregate detection over multiple routers/gateways; and separates anomalies to limit false positives in detection. Both theoretical analysis and evaluation with several router traces show that HiFIND achieves these properties. To the best of our knowledge, HiFIND is the first online DoS resilient flow-level intrusion detection system for high-speed networks (approximately 10s of Gigabit/second), even for the worst case trafic of 40-byte-packet streams with each packet forming a flow.

III. PROPOSED MODEL

In this paper, an intrusion detection system using two neural network stages based on flow-data is proposed for detecting and classifying attacks in network traffic. The first stage detects significant changes in the traffic that could be a potential attack, while the second stage defines if there is a known attack and in that case classifies the type of attack. The first stage is crucial for selecting time windows where attacks, known or unknown, are more probable. Two different neural network structures have been used, multilayer and radial basis function networks, with the objective to compare performance, memory consumption and the time required for network training.

Merits of proposed system

The experimental results demonstrate that the designed models are promising in terms of accuracy and computational time, with low probability of false alarms. From the below diagram, the collector will gather the information from different types of exporters and then undergoes preparation of features. The features of network are given to neural network stage1 ifthere is any anomaly it is detected and passed to neural network stage 2. The stage 2 will detect the type of attack and passes it to the alert module to notify the user. The labeled training data set is directly given to the stage 2 to detect type of attack.

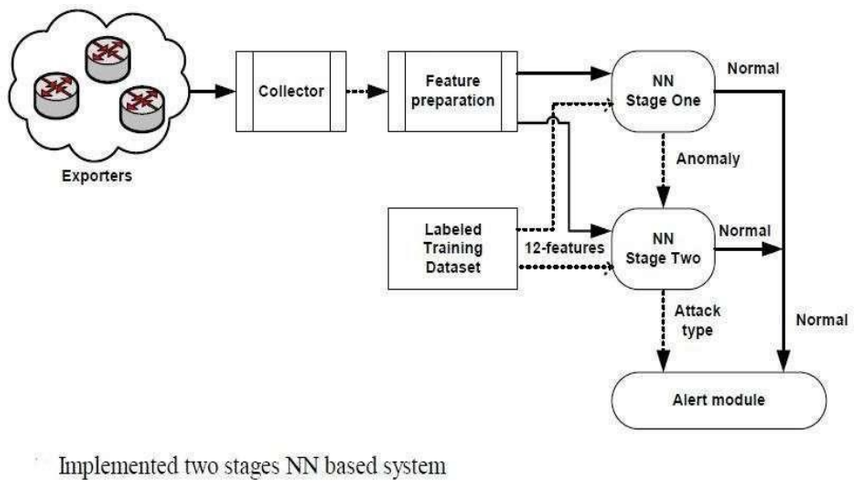

Fig 1: System Design

Algorithm

In this model, we are using neural networks. Neural network algorithm works as follows: A neural network is made up of neurons connected to each other; at the same time, each connection of our neural network is associated with a weight that dictates the importance of this relationship in the neuron when multiplied by the input value.

Each neuron has an activation function that defines the output of the neuron. Training our neural network, that is, learning the values of our parameters (weights wij and bj biases) is the most genuine part of Deep Learning and we can see this learning process in a neural network as an iterative process of "going and return" by the layers of neurons. The "going" is a forward propagation of the information and the "return" is a back propagation of the information.

IV. EXPERIMENTAL RESULTS & DISCUSSION

Input design

Input design is the part of overall system design. The main objective during input design is: To produce the cost effective input. To achieve the highest level of accuracy and to ensure that input is understandable by the user.

Output design

Output from computer systems are required primarily to communicate the processing results to users. They are also used to produce permanent copy of results for future consultation. The various forms of outputs are: External outputs: whose destination is outside the organization. Internal outputs: whose destination is within the organization and they are User's main interface with the computer. Operational outputs: whose use is purely within the computer department. Interface outputs: which involves user in communicating directly.

Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre- driven process links and integration points.

White Box Testing is a testing in which in which the software tester has knowledge of Inner workings, structure and language of the software, or atleast its purpose. It is purpose It is used to test areas that cannot be reached from a black box level. Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

It is a testing in which the software under test is treated, as a black box .You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works. Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g.components in a software system or – one step up – software applications at the company level –interact without error. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

V. EXPERIMENTAL OUTCOMES

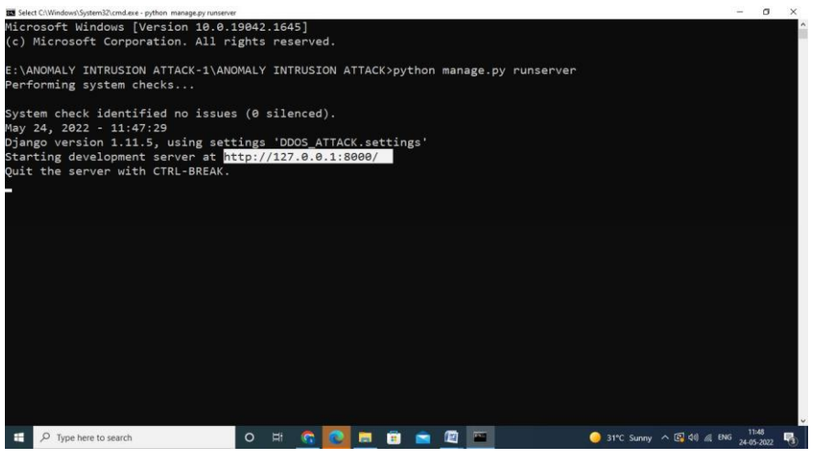

Fig 2: MySQL console

To see the output, first start the WAMP (Windows- Apache-MySQL-PHP) server. There are there indications for status of the server:
RED- The server is in offline mode. ORANGE- The server started running. GREEN- The sever is running.
Now, enter the drive in the file explorer where all the related files are located. Enter into the path where the key file for implementing is located. Here, the key file is manage.py. then, you the MySQL command prompt will be opened as shown in above figure. Enter the key command “python manage.py runserver”. Paste the URL in any favourite browser Such as google chrome Mozilla firefox. Now, the output window will be opened as shown in figure below.The credentials are mentioned in views.py python file and can be changed.

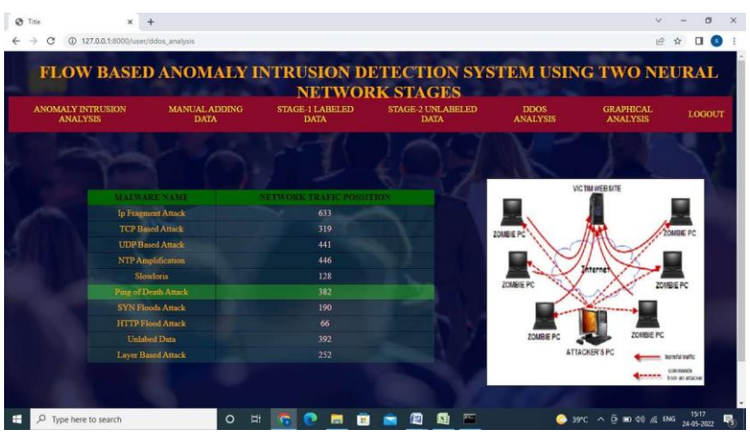

Fig 3: Classification of Attacks

The URL to be tested for the network attack is entered in manual added data tab in the above figure. If we observe, there is change in the number in the network traffic position column of NTP amplification attack and unlabeled data column. So, we can say that the entered URL contains NTP amplification attack.

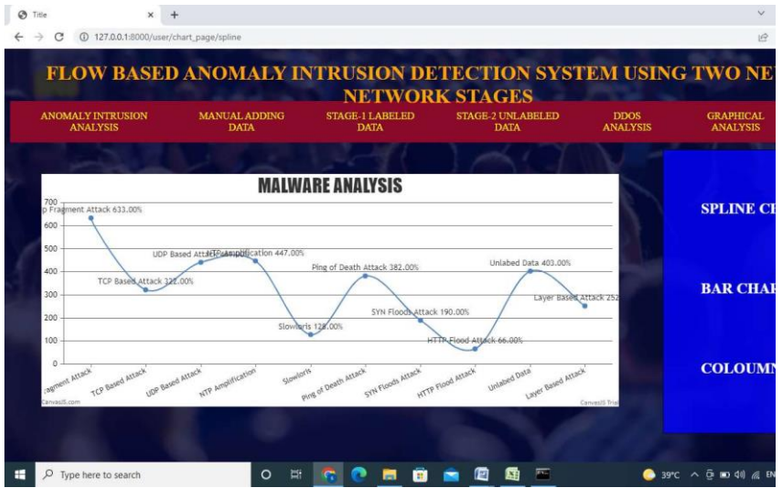

Fig 4: Output in graphical format in graphical analysis tab.

VI. CONCLUSION

In this paper we presented flow based intrusion detection and classification method using two neural networks for separate tasks. One neural network detects traffic anomalies that can be attacks and the other one classifies attacks if they exist. This system can easily be extended, configured, and/or modified by replacing some features or adding new features for new types of attacks. The training of the NNs modules requires a very large amount of NetFlow data with known types of attacks and considerable time to ensure that the results from the NNs are accurate. The changes in patterns of usage of the network should not be undetected, but at the same time, these changes are isolated to NN1.

Appearance of new patterns of attack affects only classification in NN2, which is the main reason to have two stage neural networks instead of one. Consequently, the events that require retraining for the two networks are completely independent. Experiments with different NNs were crucial to define the NN which yields the best classification and training speed results for both NN stages. The experimental results of the proposed method prove that the use of NetFlow dataset and extracting only features that significantly contribute to intrusion detection gives promising results. The obtained detection rate (94.2% for anomaly detection at stage one, and 99.4% for classification at stage two) is remarkably good compared to other approaches, which use larger training sets [20]. These results are comparable to the best researches that are based on a similar approach using the same type of training dataset. The multilayer Feedforward neural network has a better classification ability compared to RBFN, but memory and time consumption is 3-5 times greater. Otherwise, RBFN has a simple architecture and hybrid learning algorithm which leads to less time/memory consumption and it is better for working in real-time and for retraining with new data.

VII. FUTURE ENHANCEMENT

Our future research will be directed towards developing a more accurate model that can be used in real-time for detecting and classifying anomaly with minimum features and less time for training.

REFERENCES

1. Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., & Stiller, B. (2010). An overview of IP flow-based intrusion detection. *IEEE communications surveys & tutorials*, *12*(3), 343-356.
2. Lai, H., Cai, S., Huang, H., Xie, J., & Li, H. (2004, June). A parallel intrusion detection system for high-speed networks. In *International Conference on Applied Cryptography and Network Security* (pp. 439-451). Springer, Berlin, Heidelberg.

3. Gao, M., Zhang, K., & Lu, J. (2006, April). Efficient packet matching for gigabit network intrusion detection using TCAMs. In *20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06)* (Vol. 1, pp. 6-pp). IEEE.
4. Fathima, A. S., Prakesh, D., & Kumari, S. (2022). Defined Circle Friend Recommendation Policy for Growing Social Media. *International Journal of Human Computations & Intelligence*, *1*(1), 9-12.
5. De Bruijn, W., Slowinska, A., Van Reeuwijk, K., Hruby, T., Xu, L., & Bos, H. (2006). Safecard: a gigabit ips on the network card. In *International Workshop on Recent Advances in Intrusion Detection* (pp. 311-330). Springer, Berlin, Heidelberg.
6. Ahmed, S. T., Singh, D. K., Basha, S. M., Abouel Nasr, E., Kamrani, A. K., & Aboudaif, M. K. (2021). Neural network based mental depression identification and sentiments classification technique from speech signals: A COVID-19 Focused Pandemic Study. *Frontiers in public health*, *9*, 781827.
7. Sreedhar, S., Ahmed, S., Flora, P., Hemanth, L. S., Aishwarya, J., & Naik, R. (2021, January). An Improved Approach of Unstructured Text Document Classification Using Predetermined Text Model and Probability Technique. In *Proceedings of the First International Conference on Advanced Scientific Innovation in Science, Engineering and Technology, ICASISET 2020, 16-17 May 2020, Chennai, India*.
8. Vasiliadis, G., Antonatos, S., Polychronakis, M., Markatos, E. P., & Ioannidis, S. (2008, September). Gnort: High performance network intrusion detection using graphics processors. In *International workshop on recent advances in intrusion detection* (pp. 116-134). Springer, Berlin, Heidelberg.
9. Ahmed, S. T., Ashwini, S., Divya, C., Shetty, M., Anderi, P., & Singh, A. K. (2018). A hybrid and optimized resource scheduling technique using map reduce for larger instruction sets. *International Journal of Engineering & Technology*, *7*(2.33), 843-846.
10. Abuadlla, Y., Kvascev, G., Gajin, S., & Jovanovic, Z. (2014). Flow-based anomaly intrusion detection system using two neural network stages. *Computer Science and Information Systems*, *11*(2), 601-622.
11. Jaaz, Z. A., Oleiwi, S. S., Sahy, S. A., & Albarazanchi, I. (2020). Database techniques for resilient network monitoring and inspection. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *18*(5), 2412-2420.
12. Elzentani, H., & Center, I. Flow Based Intrusion Detection System Using Multistage Neural Network.
13. Taher, A. M. M., & Mehrotra, B. M. (2009). Intrusion Detection System: A design perspective.