Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

A NETWORK SECURITY FRAMEWORK FOR HYBRID BOTNET DETECTION IN CRITICAL INFRASTRUCTURE BY USING MACHINE LEARNING ALGORITHMS

D JYOTHI
JNTUH Research Student, MLR Institute of Technology, Hyderabad, India.
Email: damallajyothii@gmail.com

Dr. M.A.H FARQUAD
Associate Professor, Woxen University, Hyderabad, India. Email: farquadonline@gmail.com

Dr. G. NARSIMHA
Professor, JNTUH, Sulthanpur, India. Email: narsimha06@gmail.com

Abstract

Botnet attacks can carry out a variety of criminal activities besides aim of causing harm and collecting data from vulnerable machines, they have always been a severe issue for Critical Infrastructure and business organizations. In this research, we used Software Defined Networks, which is capable of recognizing botnet behavior by utilizing a machine learning approach and detection of related botnet attacks. We have detected the botnets by creating a monitoring frame work in the SDN environment to identify Botnet in the network flow.

Index Terms: Botnets, Machine Leaning, Software Defined Networks

1. INTRODUCTION

Number of people using the internet services increasing providing botnet attacker's scope for more attacks on the Critical Infrastructure. Botnet have spread large DDoS attacks on a number of targets in vital infrastructure. Extreme traffic was generated from the bot master using the infected bots for various reasons, particularly effective spreading, have contributed to making this possible. It may spread across a large number of heterogeneous devices through employing patterns and frequently changing the pattern. Following similar infection strategies, many botnet differences have been developed, making them the most common and efficient way to conduct botnet attacks.

A botnet is a network of vulnerable computers under the control of malware code (bot). The botnet is command and control by the bot master and used as a resource or platform for distributed denial of service (DDoS) attacks [1]. Flow based detection method helps in detecting the encrypted botnets and their botnet families because it identifies the network traffic patterns. Our aim is to detect the botnet when it is communicating instead of attack already happened. Botnets are trying to move towards encryption for hiding their identity and to increase the difficulty level of detection. Deep Packet Inspection (DPI) [2], which examines packet payloads to find any potential data included therein. But when the packets are encrypted, the tool loses its usefulness because we can no longer inspect the payload.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

The primary objective of the system built on this framework is to identify malicious network activity using a strategy using supervised machine learning that extracts general features connected to the traffic between networked devices using SDN to instruct a classifier and anticipate future connections.

1.1 Botnet life cycle

1) Infection Spreading,
2) C & C Contact
3) Report and Await Command
4) Evade Detection

Monitoring of network traffic to assure that it is optimal for the compliance requirements of the particular application. Detecting of attacks behavior in hand or during the occurrence Interference by taking suitable deed against the sensed threat

1.2 Software Defined Network

Application Layer, Control Layer, Infrastructure. SDN created a great potential having high bandwidth, dynamic applications and it is having rapidly dynamic architecture and it is cost effective. Decoupling from the forward function makes it directly programmable in its network control, Administrators adjust network traffic flow according to the changing needs of traffic flow Manageable from central control

Programmability: SDN [3] gives managers of networks a programmable infrastructure which they can use to dynamically configure and control the network using software based policies. The rapid implementation of new services and applications is made possible by this flexibility.

Centralized Control: Administrators can have a comprehensive overview of the entire network and apply uniform policies to all devices by centralizing network control in an SDN controller. The operation of the network is made simple and efficient thanks to this centralized control.

Network Automation: Automation of the network is made possible by SDN because to programmable interfaces and APIs, which may be used to automate network configuration, provisioning, and administration operations. Through automation, manual labor is reduced, network processes are speed up, and the likelihood of human error is reduced.

Traffic Optimization: Using SDN, network traffic may be intelligently routed and optimized based on the needs of the application and the current network conditions. Dynamically modifying traffic engineering and load balancing can assure optimal performance as well as efficient resource use.

Network Virtualization: SDN supports network virtualization, which enables the construction of numerous virtual networks (also known as network slices) on top of a

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

single physical infrastructure. Better resource utilization, security, and isolation are all benefits of this virtualization.

1.3 Machine Learning

Computers may learn how to complete undertakings, making forecasts, or categories collections of data without having those skills explicitly coded into them [4]. This is possible because machine learning algorithms can be trained to learn from previously acquired data that is pertinent to the issue at hand to more precisely distinguish between sets of data sets that we want to examine and categories, these previously learnt data might be represented by values called features.

Recurring patterns in traffic use classification algorithms to determine malicious traffic. Modern botnets, like Command & Control, attempt to be discrete, but examining covert botnets can help identify anomalous network behavior. Learning algorithms can help capture covert communication and identify potential issues.

2. LITERATURE REVIEW

	Contribution	Protocol	Methods	Dataset	Result	Comments
[5]	For hybrid botnet identity, network traffic analysis and host traffic analysis is connected.	IRC, HTTP, P2P	NB, DT	ISCX, CTU-13	Accuracy: 99.6%	The suggested method uses an offline mode to implement the classification algorithm.
[6]	DL was used to examine network traffic behavior from packets in order to find botnets.	IRC, HTTP	LSTM-RNN, MCFP	online data collection	Accuracy: 99.36%	The suggested strategy is capable of spotting various types of significant botnets and adapting to the environment when those botnets change how they operate or launch attacks.
[7]	Machine learning is used to detect P2P botnets using features.	P2P	J48 DT	(ISOT, ISCX 2012	Accuracy: 99.94%	The model's complexity and a lengthier processing run-time are the key constraints.
[8]	Deep Learning-based P2P botnet detection.	P2P	GNN	CAIDA	Accuracy: 99.5%	Identifying attack nodes, not specific attack flows.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

3. BOTNET FRAME WORK

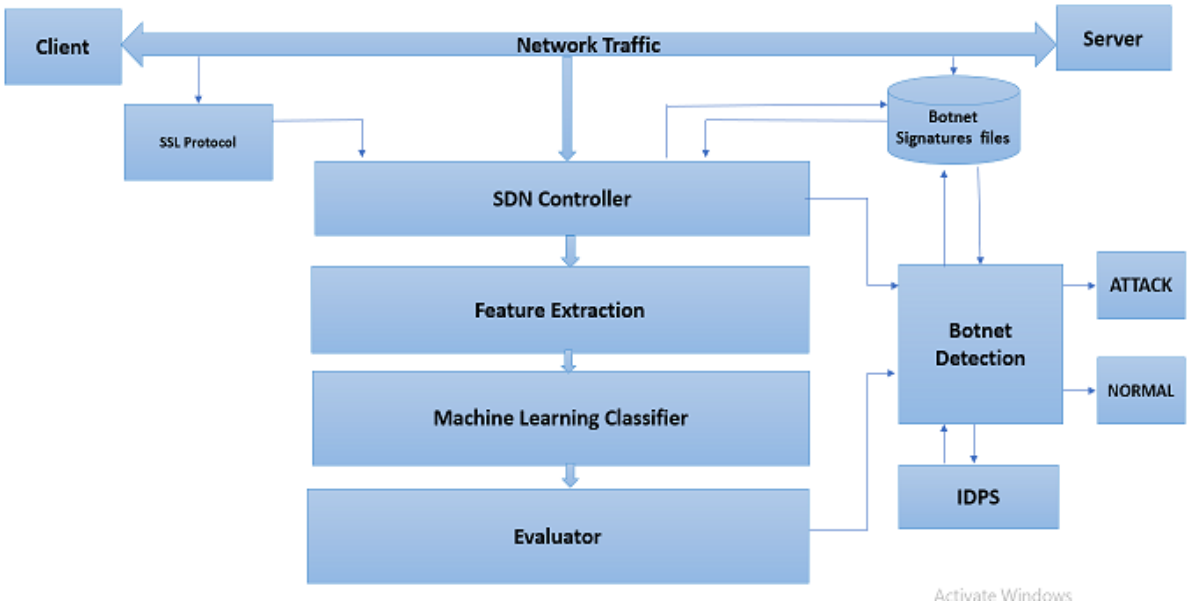

Fig 1: Architecture Implementation Diagram

3.1 Feature Selection

Few features cannot adequately capture the attack characteristics. It is important to pick the appropriate features because the choice of features will directly impact how well a model is learned. We have selected 11 subset of features extracted from Traffic volume, Communication ports, Communication patterns, Protocol Anomalies, Signature based Detection, IP Blacklisting, Network Flow Analysis, Behavioral Analysis.

S No	Feature Description
1	Length of the Connection
2	Protocol type
3	Maximum Flow Expire Time
4	Flow permanence time
5	Packets in bidirectional flow
6	Data bytes in bidirectional flow
7	From source to destination, data bytes.
8	From destination to source, data bytes
9	Certificate X.509 Validity
10	Certificate X.509time validity
11	Certificate request and Certificate Validity

Fig 2: Features

The Source IP, Destination IP and other socket information are all included in the InSDN dataset [9][10], Certificate X.509 time validity , Certificate request and certificate validity [11] to prevent the over fitting issue, where these properties can vary from network to network, all socket features are eliminated. 11 distinct features, excluding the traffic

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

category, are included in the final dataset. The ranges of the characteristics must be standardized in order to limit the scale of values between 0 and 1.

4. EVALUATION METHODOLOGY

4.1 Classifier Comparison

To find out if the system works well in situations with unexpected threats, we tested the entire dataset. False Positives (FP), False Negatives (FN), True Positives (TP), and True Negatives (TN) were all defined. False Negatives are the opposite of True Negatives, True Positives are when a positive outcome is accurately expected, and True Negatives are when a negative result is correctly predicted. Accuracy, Precision, Recall and F1-Score are used to compare algorithms Select optimal machine learning algorithms for the system. We used only one botnet: Rbot [12], Neris [13] Virut [14]

4.2 Analyzing a botnet at a time

As previously said, we tested the system in a first phase by restricting the dataset flows to only one botnet at a time. More specifically, we'll employ three separate botnets (Neris, Virut, and RBot) along with normal connections in the training dataset. We will only leave flows from one unique botnet plus regular connections in the test dataset. Here, we show the outcomes for all botnets on the training dataset as well as tests for two unidentified botnets that weren't included in the classifier's training phase. In this series of experiments, we used precisely 60008 normal connections to test the classifiers' performance in terms of flow sizes. Examination of the botnets found in the training dataset demonstrates the outcomes in merely identifying the botnet Neris, which makes use of IRC connections to talk to the Command & Control.

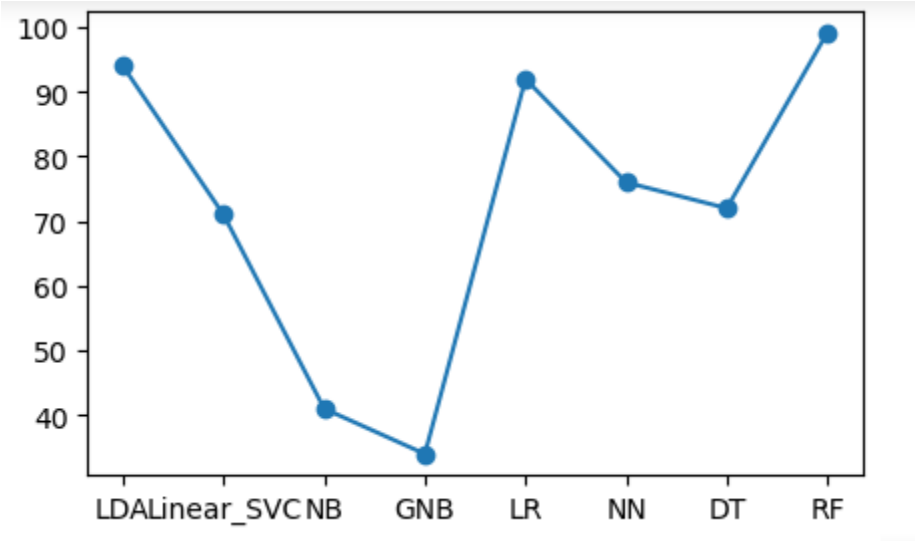

Fig 3: Result of different algorithm detecting Virut Botnet

We have the outcomes for Virut, a different botnet that is present in the training dataset.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

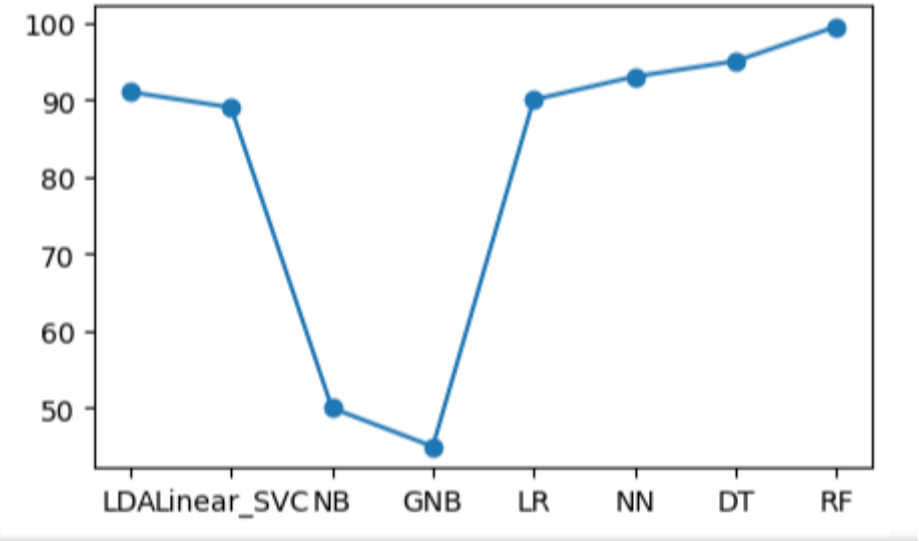

Fig 4: Result of different algorithm detecting R Bot

We have the outcomes for R Bot, a different botnet that is present in the training dataset.

4.3 Confusion matrices for these results

Logistic Regression and Linear Discriminate Analysis, both of which were more accurate than 90%. With the exception of the Naive Bayes based methods, practically all classifiers in the case of Virut worked well. The large number of connections observed in the testing dataset, which produced excellent metrics values, may be the cause of Virut's excellent performance. RBot was the last known botnet to have been tested. When using Random forest and Decision Trees, for example, both the accuracy and recall are excellent, but the precision is terrible. This is due to the fact that, despite the fact that almost all malicious flows were successfully detected, as demonstrated by the Recall values, only a very small number of connections were discovered in the testing dataset for RBot. We can see this detail more clearly in the next graphics since they display the Confusion matrices for these outcomes. For the purpose of simplicity, we will only provide here the matrix related to the best classifier of each algorithm.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

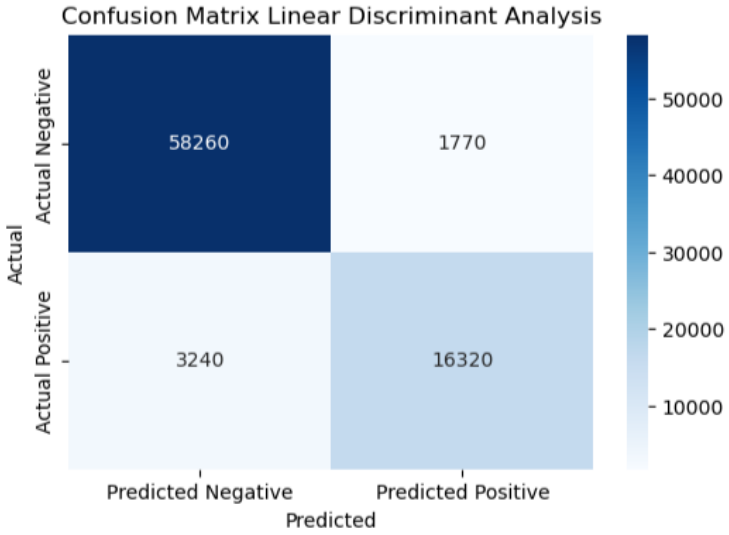

Fig 5: Confusion Matrix for Neris

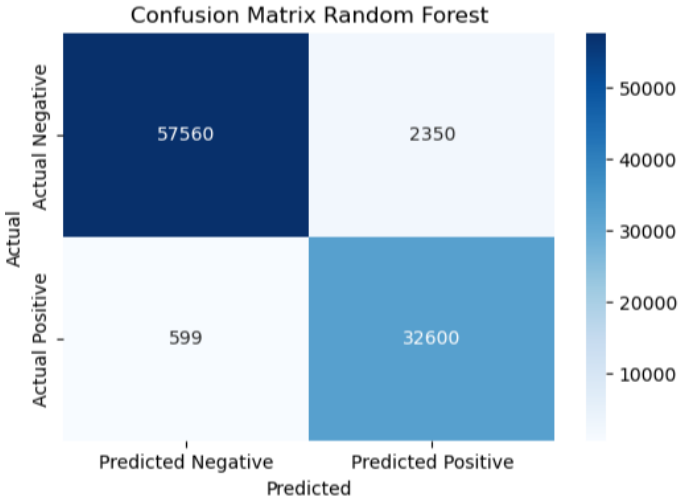

Fig 6: Confusion Matrix for Rbot

All of the classifiers successfully identify nearly all of the negative (or normal) connections that can be located in the matrix's top left corner. But we can see the True Positives value in the bottom right corner. The remaining two squares stand in for the False Positives (top right corner) and False Negatives (bottom left corner), respectively. Confusion matrices are beneficial because they neatly summarize the values of TP, TN, FP, and FN in a straight forward picture.

Two botnets, Trickbot [13] and QakBot, were evaluated in relation to unidentified threats. The reason Trickbot was our choice was that it appeared in only 485 connections, which was a lower proportion compared to normal flows. 4539 flows were used in the QakBot botnet test. Display the classifier's findings in identifying these two dangers

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

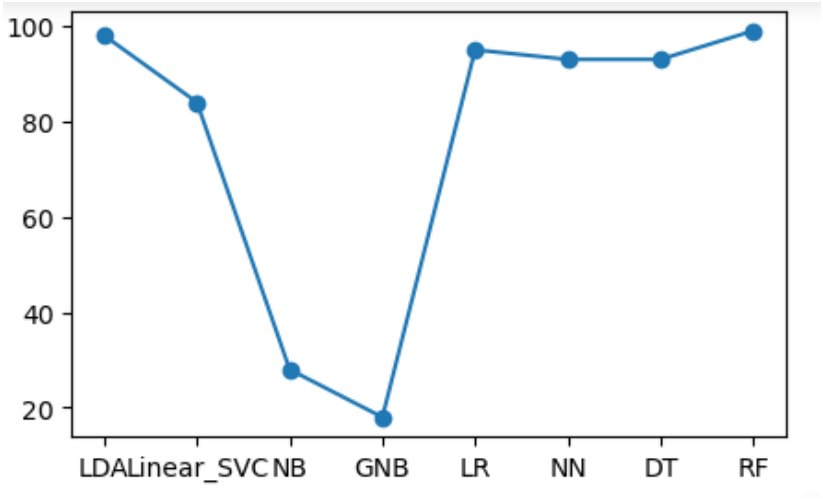

Fig 7: Results for the different algorithms in detecting the Trickbot botnet

The results from the three tables and the results above are slightly different. We may observe a large decrease in the Precision metric from the Trickbot connections. This is as a result of the lack of distinguishable links. For instance, the best classifier (LDA) [14] only accurately identified 212 out of the 495 links; nonetheless, the issue is with the False Positives. The classifier identified a total of 1783 false positives, which belong to 3 percent of the flows are classified. Since nearly every regular link was correctly predicted, the accuracy remained high. The QakBot botnet is in a similar predicament, although with marginally greater Precision and Recall. Overall, the detection of a single botnet at a time was successful thanks to the classification mechanism. Hitting high values in the Accuracy and Recall metric, the detector did perform better when the experiment contained a botnet present in the training dataset.

4.4 Related Metrics

In order to better analyze the outcomes, we present below two additional statistical visualizations related to the classification. The True Positive Rate is plotted on the Y axis and the False Positive Rate is plotted on the X axis on the ROC curve, for example. Maximizing the True Positive rate while lowering the False Positive rate is ideal in this situation.

Examining the whole dataset for the second experiment, we put the system to the test throughout the entire dataset to see how it would react to a variety of concurrent threats.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

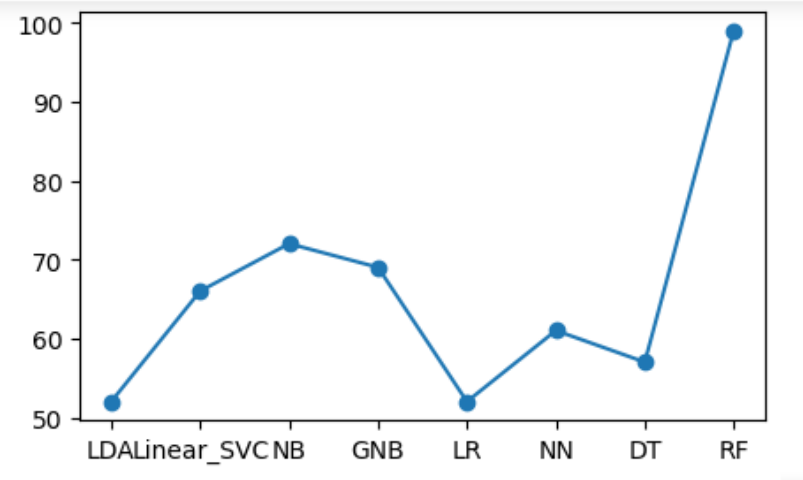

Fig 8: Results of evaluating every dataset with several methods

Cross Validation

A statistical technique known as cross validation is frequently used in machine learning to assess how well an algorithm would work and, as a result, help us select the best classifier [15]. The objective behind cross validation is to divide the dataset into training and validation samples on many occasions in order to estimate the risk associated with each technique. Theoretically, this tactic works well since it prevents over fitting.

One botnet at a time analysis

Since we had to display the visuals for each of the eight classifiers for each botnet in the experiment that involved analyzing one botnet at a time, we will only select one botnet to represent the findings of the cross validation test here. We choose to highlight the Virut botnet since it produced excellent results with only a few flows taught in the past.

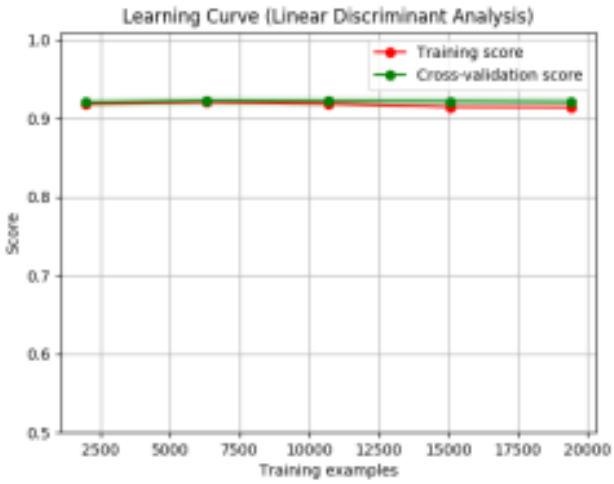

Fig 9: LDA - Virut's learning curve

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

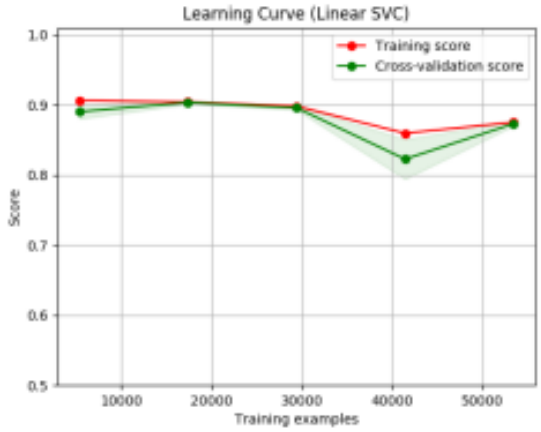

Fig 10: Linear SVC - Virut learning curve

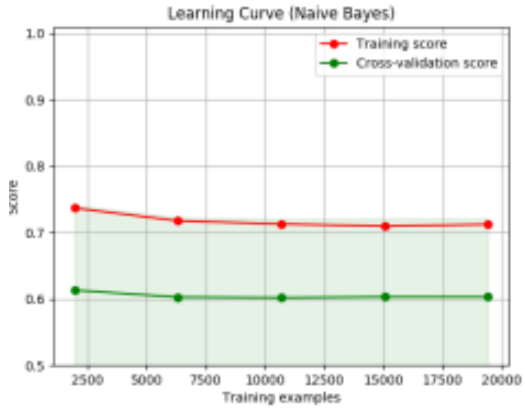

Fig 11: Learning curve for the Virut Naive Bayes model

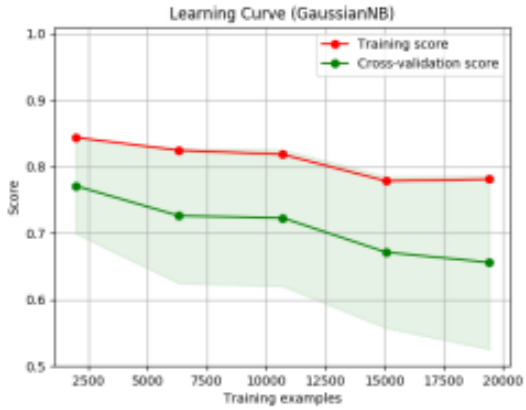

Fig 12: Gaussian NB - Virut learning curve

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

We can infer that the results are excellent and reassuring from the cross validation [16] experiment's output figures. All methods outperformed Naive Bayes and the Gaussian Naive Bayes classifier, as shown by the closeness of the two lines in the pictures. We can claim that we have achieved the best outcome for the classifier when the test score and training score of each cross validation iteration (each point in the graph) almost completely overlap. For instance

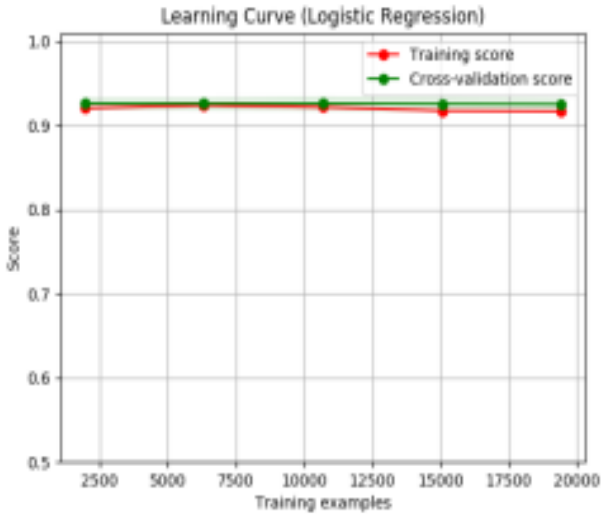

Fig 13: LR - Virut's learning curve

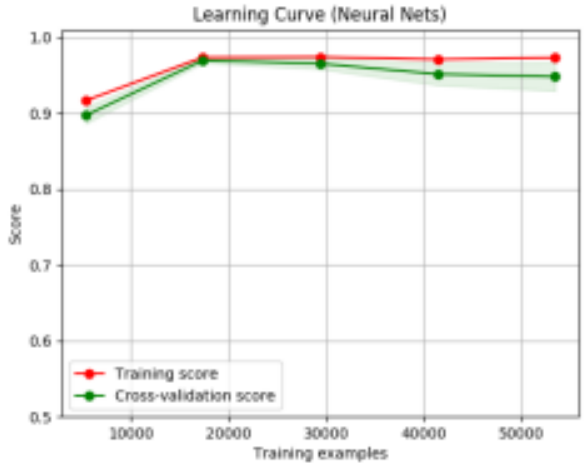

Fig 14: Learning curve for Neural Net Virut's

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

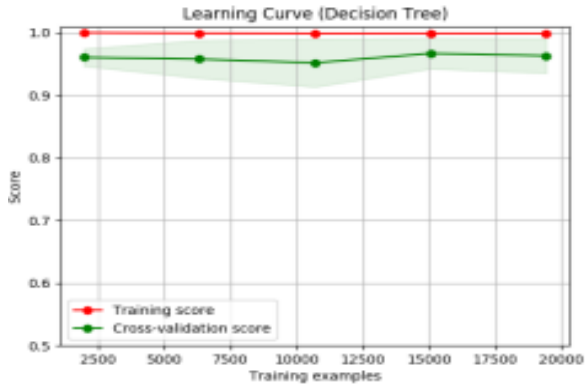

Fig 15: Decision Tree - Virut learning curve

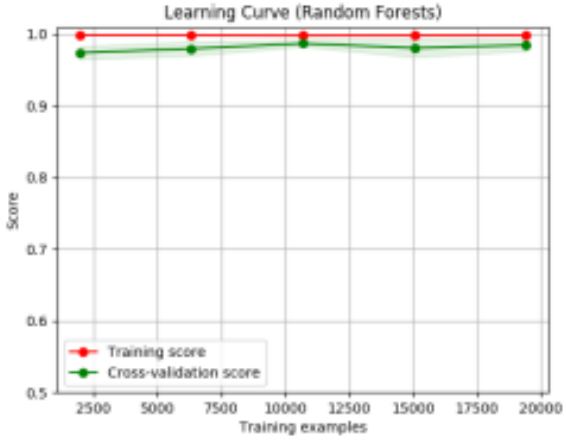

Fig 16: Learning curve for Virut's Random forest

The two lines in the case of Random forest [17] are fairly close, and Accuracy is higher than 95%. These findings only support our view that the system performs admirably in more realistic circumstances because it is so effective at identifying individual botnets. Given the least desirable results displayed in the corresponding figures, we may infer that neither Naive Bayes nor Gaussian Naive Bayes are likely suitable for this type of classification.

Examining the whole dataset

The learning curves for each technique are depicted in the following pictures, where the dataset consists of all testing botnets that were discovered in it, mixed in with connections from regular activity.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

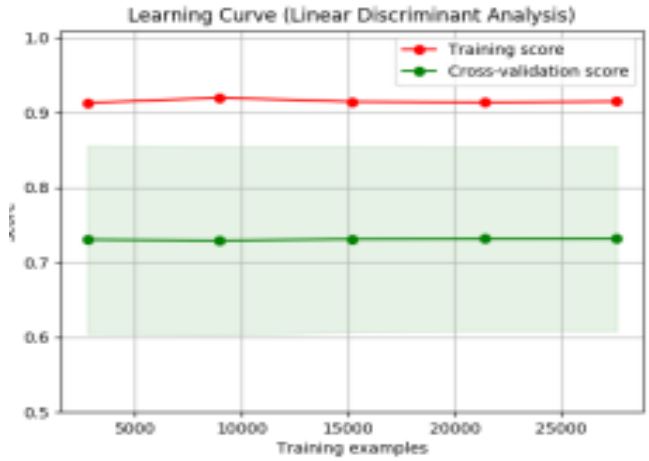

Fig 17: LDA learning curve

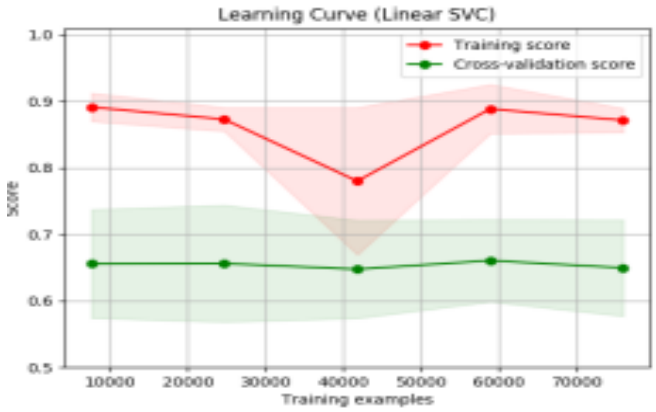

Fig 18: Linear SVC learning curve

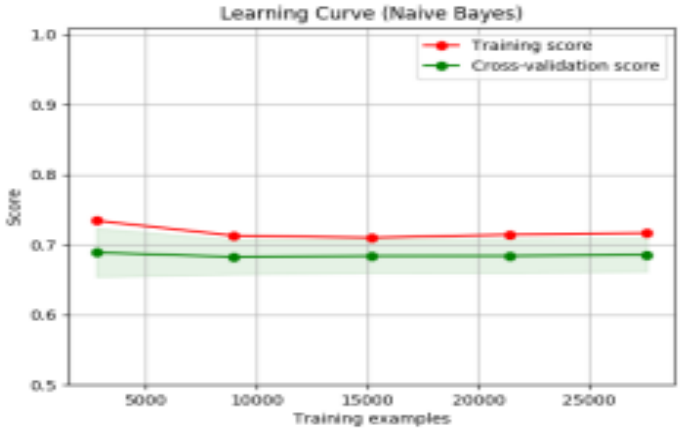

Fig 19: Learnability of Naive Bayes

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

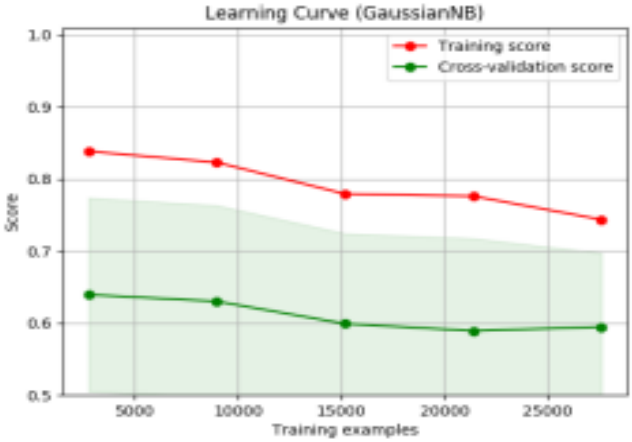

Fig 20: Gaussian NB learning curve

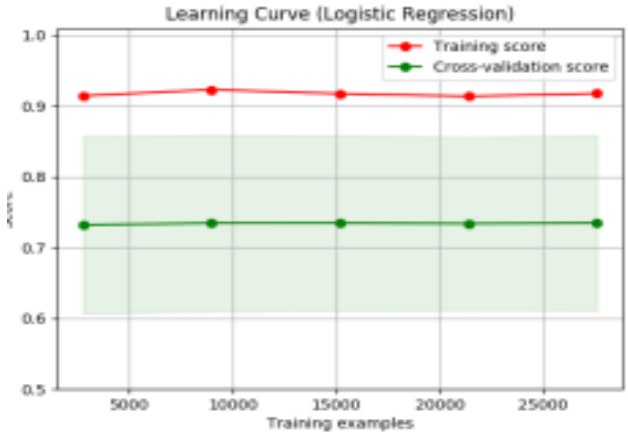

Fig 21: Curve of Learning for Logistic Regression

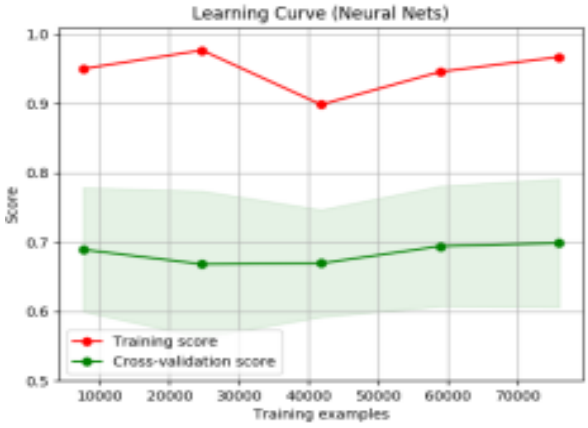

Fig 22: Neural network learning curve

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

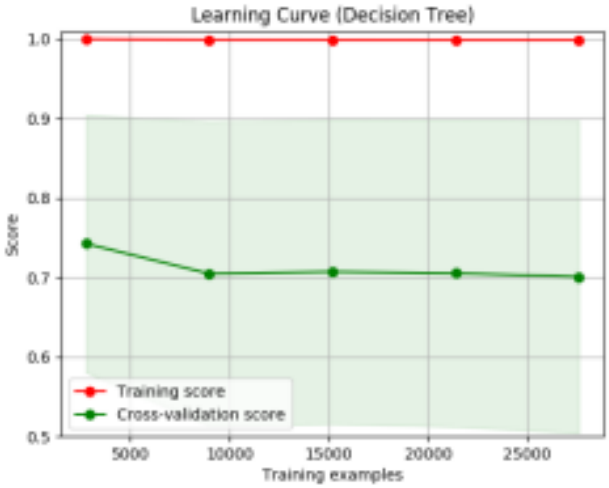

Fig 23: Decision Tree's learning curve

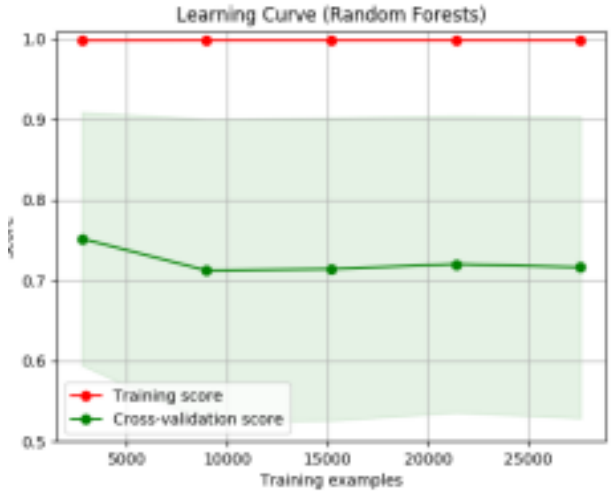

Fig 24: Random forest ' learning curve

All algorithms, with the exception of Gaussian Naive Bayes and Naive Bayes, as we saw on the preceding cross validation study. The figures depict this potential scenario when the training score is significantly higher compared to the testing score and it isn't dropping, thus if we could carry out more iterations of the findings would stabilize and slightly improve with more training connections during cross validation. However, the green region on the numbers suggests that the results may be improving or declining. As was previously said, the green region shows the range of scores from the cross validation. So, the ideal scores might be there, in principle.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

5. CONCLUSION

In a software defined network, the proposed method detects botnets. The approach makes advantage of a comprehensive feature set that was extracted from SDN to improve detection rate and decrease false positives. The research yields encouraging findings for both known and undiscovered bots. In comparison to other methodologies with the same degree of testing data diversity, the detection rate is significantly higher. The suggested system has good accuracy. Separating statistics from the control plan allows for centralized statistics visibility while also lessening the calculation load on the controller. The classifier's results were optimistic, with a accuracy rate of roughly 99.8%, cross validation technique had a lower detection rate and 75% accuracy. In Future we want to expand this work to design a lightweight architecture for detecting botnets by using stack ensemble machine learning in the network environment are potential extensions of this study.

References

1) Kumar, A., Shridhar, M., Swaminathan, S., & Lim, T. J. (2022, June). Machine learning-based early detection of IoT botnets using network-edge traffic. Computers & Security, 117, 102693. https://doi.org/10.1016/j.cose.2022.102693

2) Yang, J. H., & Han, S. J. (2009, August 31). Implementation and Performance Analysis of Efficient Packet Processing Method for DPI (Deep Packet Inspection) System using Dual-Processors. *The KIPS Transactions: PartC*, *16C* (4), 417–422. https://doi.org/10.3745/kipstc.2009.16-c.4.417

3) Shinan, K., Alsubhi, K., Alzahrani, A., & Ashraf, M. U. (2021, May 12). Machine Learning-Based Botnet Detection in Software-Defined Network: A Systematic Review. *Symmetry*, *13*(5), 866. https://doi.org/10.3390/sym13050866

4) Kumar, D. S., & Raja, P. (2022, November 27). Implementation of Artificial Intelligence in a Software-Defined Wireless Sensor Network. *Journal of Artificial Intelligence, Machine Learning and Neural Network*, *26*, 32–42. https://doi.org/10.55529/jaimlnn.26.32.42

5) Zorkadis, V., Karras, D., & Panayotou, M. (2005, July). Efficient information theoretic strategies for classifier combination, feature extraction and performance evaluation in improving false positives and false negatives for spam e-mail filtering. *Neural Networks*, *18*(5–6), 799–807. https://doi.org/10.1016/j.neunet.2005.06.045

6) Matharaarachchi, S., Domaratzki, M., & Muthukumarana, S. (2021, December). Assessing feature selection method performance with class imbalance data. Machine Learning With Applications, 6, 100170. https://doi.org/10.1016/j.mlwa.2021.100170

7) Shi, W.C.; Sun, H.M. DeepBot: A time-based botnet detection with deep learning. Soft Comput. 2020, 24, 16605–16616.

8) Almutairi, S.; Mahfoudh, S.; Almutairi, S.; Alowibdi, J.S. Hybrid Botnet Detection Based on Host and Network Analysis. J. Comput. Netw. Commun. 2020, 2020

9) Yue, M., Wang, H., Liu, L., & Wu, Z. (2020). Detecting DoS Attacks Based on Multi-Features in SDN. *IEEE Access*, *8*, 104688–104700. https://doi.org/10.1109/access.2020.2999668

10) Elsayed, M. S., Le-Khac, N. A., & Jurcut, A. D. (2020). InSDN: A Novel SDN Intrusion Dataset. IEEE Access, 8, 165263–165284. https://doi.org/10.1109/access.2020.3022633

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 01:2023
DOI: 10.5281/zenodo.8285603

11) Malik, M., Kamaldeep, Dutta, M., & Granjal, J. (2023). L-ECQV: Lightweight ECQV Implicit Certificates for Authentication in the Internet of Things. *IEEE Access*, *11*, 35517–35540. https://doi.org/10.1109/access.2023.3261666

12) Kumar, p. A. (2020, May 15). Botnet Detection with Machine Learning Classifiers. Journal of Research on the Lepidoptera, 51(2), 329–335. https://doi.org/10.36872/lepi/v51i2/301100

13) Hong, Y., Li, Q., Yang, Y., & Shen, M. (2023, June). Graph based Encrypted Malicious Traffic Detection with Hybrid Analysis of Multi-view Features. *Information Sciences*, 119229. https://doi.org/10.1016/j.ins.2023.119229

14) Xiang, C., Lihua, Y., Shuyuan, J., Zhiyu, H., & Shuhao, L. (2013, February 19). Botnet spoofing: fighting botnet with itself. *Security and Communication Networks*, *8*(1), 80–89. https://doi.org/10.1002/sec.749

15) Manickam, S. (2020). Botnet Monitoring Mechanisms on Peer-to-Peer (P2p) Botnet. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3713662

16) Gadde Ramesh and Suresh Pabboju, "Machine Learning Techniques for Sensing IoT Botnets" in Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition) @ 2022, ISSN: 1671-5497, Vol: 41, Issue: 12:2022, CNKI Number: CN-22-1341/T, pp: 843-859.

17) Gadde Ramesh and Suresh Pabboju "A Host–Based P2P Host Identification Approach with Flow–Based in Detection of P2P Bots" in Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban) Journal of Tianjin University Science and Technology @ 2022, ISSN: 0493-2137, Vol: 55, Issue: 01:2022, pp: 1-17, https://www.scopus.com/sourceid/75573