Network Flow based IoT Botnet Attack Detection using Deep Learning

Sriram S*, Vinayakumar R†*, Mamoun Alazab‡, Soman KP*
*Amrita School of Engineering, Amrita Vishwa Vidyapeetham Coimbatore, Tamil Nadu, India.
sri27395ram@gmail.com
†Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center, Cincinnati, OH, United States
Vinayakumar.Ravi@cchmc.org, vinayakumarr77@gmail.com
‡College of Engineering, IT and Environment, Charles Darwin University, Casuarina, NT 0810, Australia
alazab.m@ieee.org

***Abstract*—Governments around the globe are promoting smart city applications to enhance the quality of daily-life activities in urban areas. Smart cities include internet-enabled devices that are used by applications like health care, power grid, water treatment, traffic control, etc to enhance its effectiveness. The expansion in the quantity of Internet-of-things (IoT) based botnet attacks is due to the growing trend of Internet-enabled devices. To provide advanced cyber security solutions to IoT devices and smart city applications, this paper proposes a deep learning (DL) based botnet detection system that works on network traffic flows. The botnet detection framework collects the network traffic flows, converts them into connection records and uses a DL model to detect attacks emanating from the compromised IoT devices. To determine an optimal DL model, many experiments are conducted on well-known and recently released benchmark data sets. Further, the datasets are visualized to understand its characteristics. The proposed DL model outperformed the conventional machine learning (ML) models.**

***Index Terms*—Cyber Security, Botnet, Smart Cities, Internet of Things, Big Data, Machine Learning, Deep Learning.**

I. Introduction

In recent times, the distributed denial of service (DDoS) attack is a standout amongst the most fascinating and broadly observed cyberattacks[1]. A DDoS attack is a cyberattack in which a hacker briefly subjugates various compromised systems to attack a given target i.e., make simultaneous requests to a server for a particular service, which overpowers the server and influences it to disregard real requests from end-users. Attackers often use a network of compromised devices called botnet to perform various cyber attacks [1]. The compromised devices could be desktops, laptops, smartphones or Internet-of-things (IoT) devices. The botmaster can instruct each bot in the botnet to perform a coordinated attack. In [2], Akamai has observed that there is a 138% increase in DDoS attacks (in comparison to 2015) where traffic is more than 100 Gbps which is attributed to increased utilization of IoT devices in DDoS attacks. The number of platforms that gives DDoS attacks as service, has also been increased. For as meager as $6, one can secretly arrange a 5-6 Gbps DDoS attack, which can last for 10 minutes or more.

As the quantity of IoT devices is moving into several billion, a considerable lot of which have no genuine security protections and they face a developing risk from botnet attacks. In the U.S.A, the quantity of assaults on infrastructure has increased from 200 in 2012 to 300 in 2015 [2]. All smart cities should secure their utility center and power infrastructures as they are most vulnerable and most likely to be attacked by cybercriminals. In 2016, the attackers knocked out 20% of the city's power by intruding into the systems that belong to an electrical substation in Kiev [3]. so, guaranteeing security for critical facilities must be given priority. In smart cities, botnet attacks could take control of IoT devices and weaponize them so that the network of infected devices can be employed to conduct coordinated attacks like DDoS attack which is capable of sending huge traffic to the target to bring down the availability of service to the users. Hence, it is essential to create a robust botnet attack detection framework for safeguarding smart cities.

Currently, the commercial security products are mostly based on the threshold, signature, heuristics based methods or characteristics based on statistical measures [3]. These methods are effective at identifying known attacks. but, they completely fail at the detection of variants of existing attacks or new attacks. Also, these methods require domain expert knowledge and continual updates. Many researchers have proposed machine learning (ML) based solutions to deal with the drawbacks of classical methods [4]. On the other hand, deep learning (DL), which is an enhanced version of classical neural networks, does not necessarily need feature engineering as they can obtain optimal features automatically from raw data. In recent days, DL architecture has performed well in solving many cyber security problems such as botnet detection [5], [6], intrusion detection [3], malware detection [7], [8], etc. Additionally, the DL architectures have been employed to improve the performance of IoT botnet detection [9], [4]. In [9], Meidan et al. proposed an empirically evaluated network-based approach for identifying attacks launched from IoT botnets. In this work, a framework for botnet attack detection and categorization based on network flow data is proposed. The framework employs the DL model to extract optimal features from network traffic data to learn to differentiate

[1] https://securelist.com/ddos-report-q1-2019/90792/

botnet attack traffic from normal legitimate ones. Further, the attacks are categorized into their respective families.

The major contributions of this study are the following.

- This work proposes a DL based botnet attack detection framework that relies on network flow.
- This work comparatively analyses the performance of ML and the deep neural network (DNN) models for botnet detection with various datasets.
- To understand the characteristics of the datasets used in this work, the t-distributed stochastic neighbor embedding (t-SNE) visualization technique is employed.

II. Related work

Against DDoS in IoT, Ozcelik et al. [10] has introduced an edge-oriented detection and mitigation approach using software-defined networking (SDN) and Fog approaches. SDN was utilized as an adaptable approach to set out new flow rules and to revise them dynamically whenever needed. This proposed approach demonstrated that using SDN, the mitigation of DDoS attacks for IoT devices, can be done close to the devices itself using edge computing. Summerville et al. [11] built a lightweight, deep-packet anomaly detection technique that works on low powered IoT devices with limited computational ability. In this work, the n-gram bit-patterns are used and the n-gram size is allowed to vary by dimension. Pa et al. [12] proposed an IoT honeypot to gather information and sandbox to analyze Telnet-based attacks that are originated from many IoT devices running on different CPUs. The analysis shows that the captured DDOS malware files are from at least five unique families. A game theory based lightweight anomaly detection system is proposed in [13] and it is capable of running on devices that have limited computational abilities. Bostani et al. [14] proposed an anomaly and specification based intrusion detection scheme that is used for detecting 2 routing attacks called sinkhole and selective forwarding attacks in IoT.

Kalis was presented by Midi et al. [15] which is a knowledge-driven intrusion detection framework that can adapt itself. It can detect attacks continuously over a wide scope of IoT frameworks in real-time without impacting the IoT applications. Furthermore, it enables collaborative security scenarios. In [16], a new intrusion detection framework for the IoT called SVELTE was proposed. It was implemented in Contiki OS and thoroughly evaluated. Essentially focus in this work was on routing attacks such as altered or spoofed information, sinkhole, and selective forwarding. In [17], a signature based intrusion detection was proposed for handling two variations of DoS attacks. The simulation outcomes demonstrated that these attacks may affect the reachability of specific IoT devices and their energy consumption. In [9], an empirically evaluated network-based approach for identifying attacks originated from infected IoT devices is proposed. DL strategies are used by this approach to perform anomaly detection. A deep autoencoder is trained for each IoT device to gain proficiency with the IoT's typical behavior utilizing behavioral snapshots of the IoT traffic. The deep autoencoder endeavors to compress snapshots. The IoT device is said to be compromised when an autoencoder fails to recreate a snapshot.

The feature selection technique is applied in [18] to reduce the number of features involved in identifying the IoT bots. It was observed that the models, that used fewer features, accomplished extremely high accuracy rates. Furthermore, it was also shown that the outcomes are affordable and interpretable with a decision tree classifier. Nõmm et al. [19] proposes an unsupervised learning based anomaly detection approach which emphasis on feature selection. The proposed method achieves high accuracy even though the number of features is reduced. Chawathe et al. [20] proposes a scalable bot detection approach that monitors network traffic of IoT devices without the need for any special access. The system uses a data set that is obtained from a real IoT network. Koroniotis et al. [4] proposed a new well-structured and representative data set, Bot-IoT, for training and validating the reliability of the system. This data set consolidates legitimate and simulated IoT network traffic, alongside different sorts of attacks.

An IDS, that employs random forest classifier and a neural network for identification and categorization of attacks respectively is proposed in [21]. IDS for the IoT was proposed in [22]. It employs ML techniques. It is also capable of successfully identifying simple forms of DoS attacks and network scanning probing. Taheri et al. [23] proposes a botnet detection using DL and image processing based method. In this method, the network traffic flows are transformed into images, application of CNN is used for feature extraction and support vector machine (SVM) for classification.

III. Proposed Architecture for IoT Security Environment

The proposed architecture for botnet attack detection collects the network traffic snapshots from the connected devices in the packet capture (PCAP) format. Further, it converts the traffic information into connection records. During training, these connection records are labeled manually and passed into several classical ML algorithms and DNNs. In the experiments conducted, the performance obtained by all algorithms is closer. Since the DNNs can automatically learn the best features unlike the classical ML algorithms, DNNs can be used. The framework initially classifies the connection record into either legitimate or attack. Then, it classifies the attacks into their corresponding attack categories and as well as to corresponding botnet families. This further helps to understand the botnet family characteristics more accurately. Many DNN with the different number of hidden layers and neurons are trained and grid search is used for selected optimal hyperparameters. The proposed DNN model has 4 hidden layers with 115, 256, 128, 64 neurons at each layer. Each hidden layer is followed by the ReLU activation function and the dropout regularization method is also employed. The batch size and learning rate are set to 64 and 0.001 respectively. The adam optimizer is used.

IV. Description of Data set

In this work, 2 datasets are used. The first data set that is used for IoT botnet detection is N-BaIoT [9] and it is referred to as $DS1$. The data set is developed by collecting network traffic flow from IoT devices that are infected with BASHLITE and Mirai botnets. BASHLITE attacks include Scan, Junk, UDP, TCP, and COMBO and Mirai attacks include Scan, Ack, Syn, UDP, and UDPplain. The $DS1$ data set is transformed into 3 sets for running 3 different types of experiments. The first set $DS1-V1$ aids to classify the connection records into normal or attack. The second set $DS1-V2$ aids to classify the connection records into BASHLITE or Mirai. The final set $DS1-V3$ aids to classify the attack connection records of BASHLITE and Mirai botnet into their categories. The $DS1-V1$ set contains 172,641 normal and 240,000 attack samples and the $DS1-V2$ contains 135,000 and 104,000 samples that belong to BASHLITE and Mirai class respectively. The detailed statistics of $DS1-V3$ is reported in Table I.

TABLE I: Statistics of $DS1-V3$.

Botnet	Attack	#Samples
BASHLITE	combo	30,000
	junk	24,000
	scan	27,000
	tcp	26,999
	udp	27,000
Mirai	ack	21,000
	scan	21,000
	syn	21,000
	udp	21,000
	udpplain	21,000

The second data set that is used in this work is BoT-IoT [4] and it is referred to as $DS2$. This data set was obtained from a real-time network lab setup at the Research Cyber Range lab of UNSW Canberra. This dataset is transformed into 2 sets. The first set $DS2-V1$ aids to classify the connection record into normal or attack and the second set $DS2-V2$ aids to classify the attack connection records to their corresponding categories. The $DS2-V1$ set has 370 and 107 normal samples and 2,934,447 and 733,598 attack samples in train and test sets respectively. The second set $DS2-V2$ has four attack categories. The first one is DDoS and it has 1,541,315 and 385,309 samples in train and test sets respectively. The second one is DoS and it has 1,320,148 and 30,112 samples. Reconnaissance is the third category and it contains 72,919 and 18,163 samples. Lastly, the fourth category, Theft contains 65 and 14 samples in train and test sets respectively.

V. Experiments, Results and Observations

All ML algorithms are implemented using Scikit-learn[2] and DL models are implemented using TensorFlow[3] with Keras[4]. All the ML models takes works on default parameters set by Scikit-learn. GPU enabled machines were used in all experiments. All the source codes are publically made available for further research[5].

The main aim of this work is to classify the given connection records as either normal or attack and also, to classify the attacks into their corresponding categories. To learn the characteristics of botnets, the information of attacks and attack categories are used to identify the botnet family. This further helps to detect and block the botnet communication in a network.

Previous research works that uses $DS1$ data set have modeled the problem as anomaly detection and some of them reported zero false positive rate. However, in real-time, the performance might not be the same [3]. Some works have reported that the performance of the anomaly detection system is very poor and it produces a high false-positive rate [3]. Some existing works that uses $DS2$ data set have reported that DL architectures like recurrent networks peformed better when compared to classical ML algorithms. However, since the connection records are extracted features, the application of feed forward network with $ReLU$ typically called the DNN can be employed. This is because the DNN method is computationally inexpensive and it can perform well in real-time applications. In this work, the botnet detection problem is modelled as intrusion detection and various classical ML algorithms and DNN is applied.

Since the performance reported by [9], [4] on $DS1$ and $DS2$ data sets respectively is very high, the visualization method is used to understand the main reason why the ML algorithms achieve good performance. The t-SNE technique is used for the connection records visualization for $DS1$ and the histogram plot is used for visualizing $DS2$ data set.

The t-SNE is a non-linear dimensionality reduction method that can be used to map high-dimensional data into a lower dimension, particularly two or more dimensions [3]. The t-SNE visualization is done for all three sets of $DS1$ dataset and its configuration details reported in Table II. Fig. 1 represents the visualization of connection records that belong to normal and attack classes. Fig. 2 represents the connection records that belong to BASHLITE and Mirai classes. Fig. 3 and Fig. 4 represents the visualization of connection records that belong to subcategories of BASHLITE and Mirai attack respectively. All four t-SNE visualization figures show that the almost all records appear in separate clusters. Some overlap can be found in Fig. 3 and Fig. 4 and DNN may be used to model it properly. Eventhough the data is non-linear, the data distributions are simple and almost separable. This might be one of the main reasons why the existing methods show good performance. However, the existing anomaly detection based methods may not perform well in the real world scenario as the pattern of attacks changes with time due to the changes in behaviors of the botnet attack. Thus to avoid this issue, the botnet detection problem is modeled as intrusion detection in this work.

To understand the representation of data distribution, a histogram plot is constructed for data set $DS2$ and shown

[2] https://scikit-learn.org/stable/

[3] https://www.tensorflow.org/

[4] https://keras.io/

[5] https://github.com/vinayakumarr/IoT-Botnet

TABLE II: Detailed parameter information involved in feature visualization using t-SNE

Data set	Iteration	Error	Time in Seconds
Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000			
$DS1-V1$	1000	19.28236435	431.81
$DS1-V2$	1000	18.85542265	431.93
$DS1-V3$ (BASHLITE)	1000	17.9763077	893.50
$DS1-V3$ (Mirai)	1000	19.53498875	636.63

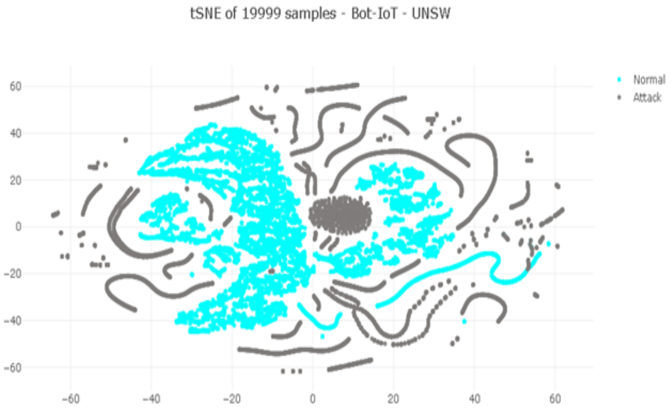

Fig. 1: t-SNE visualization of normal and attack connection records - $DS1-V1$.

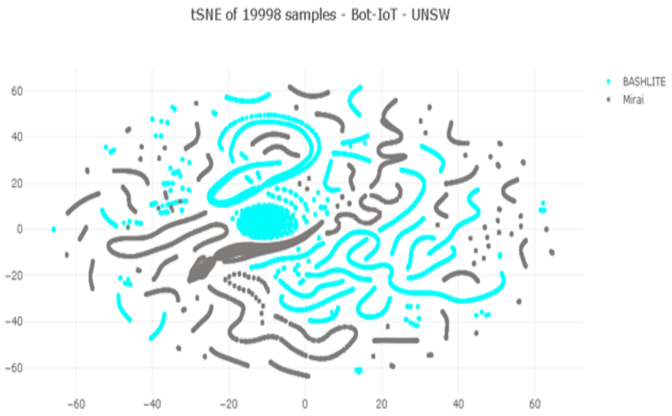

Fig. 2: t-SNE visualization of BASHLITE and Mirai attack connection records - $DS1-V2$.

in Fig. 5 and Fig. 6 for normal and attack types respectively. This groups the range of values of all features into a certain number of bins. From Fig. 5 and Fig. 6, it can be observed that the normal and attack connection records can be easily distinguishable using these features. This might be one of the main reasons why the existing methods have achieved zero false-positive rates in anomaly detection. However, the feature ’drate’ looks similar in both the normal and attack connection records which indicates that the contribution of this feature may be very less for classification compared to other features and it can be discarded to check if it enhance the performance.

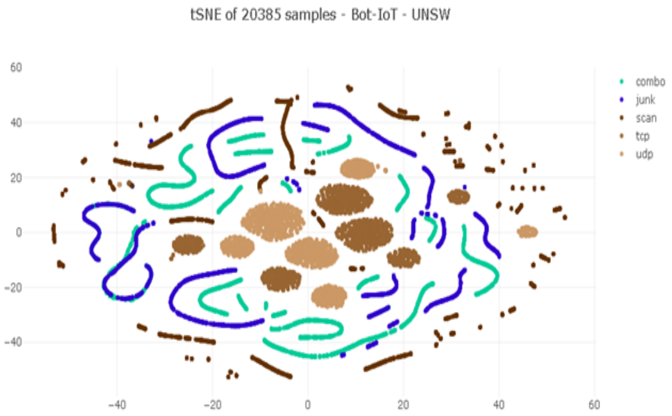

Fig. 3: t-SNE visualization of subcategories of BASHLITE attack connection records from $DS1-V3$ set.

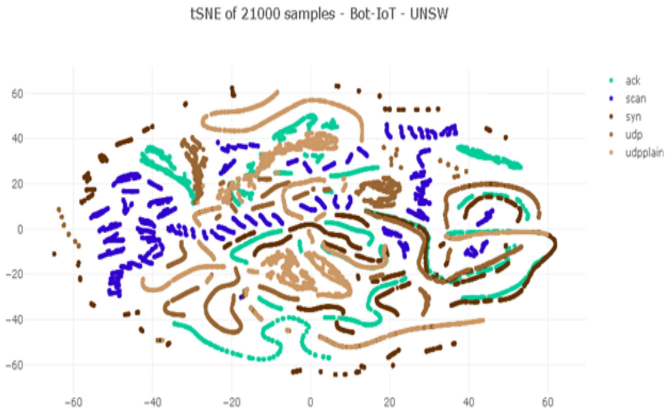

Fig. 4: t-SNE visualization of subcategories of Mirai attack connection records from $DS1-V3$ set.

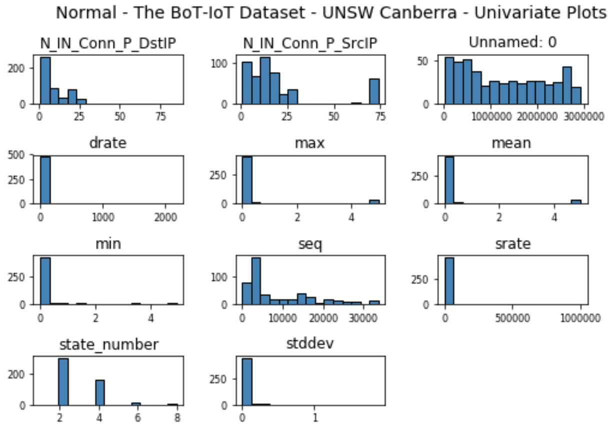

Fig. 5: Normal connection records of $DS2-V1$ data set feature visualization using univariate histogram plots.

In this work, the minimized version of the $DS1$ data set [9] is taken. The followings are the different type of experiment conducted using $DS1$ data set:

- The classification of the connection records as either normal or attack.

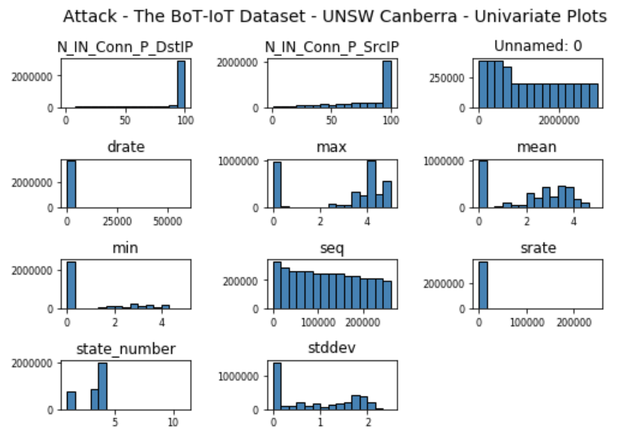

Fig. 6: Attack connection records of $DS2-V1$ data set feature visualization using univariate histogram plots.

- The classification of the connection records into different attack categories.
- The classification of the connection records into different botnet families.

The performance of the models is determined using standard metrics such as accuracy (Acc), precision (Pre), recall (Rec) and F1-score (F1). Different classical ML algorithms are initially trained and evaluated using different sets of $DS1$ data set. The detailed statistics are reported in Tables III, IV and V. The train and test data sets of $DS1$ are disjoint to each other. The performances obtained by the logistic regression (LR) and navie bayes (NB) classifiers were considerably less when compared to other classifiers such as k-nearest neighbors (KNN), decision tree (DT), adaboost (AB), random forest (RF), Linear SVM (LSVM) and radial basis function SVM (RSVM) all three different sets of $DS1$ data set. Particularly, the DT classifier showed the best overall performance in all three experiments when compared with other classifiers in terms of accuracy, F1-score and as well as in terms of testing time. These ML classifiers have achieved zero false positive rates in the first two experiments. However, in the third experiment, DNN [3] achieved the best performance with zero false positive rates.

TABLE III: Test results for identifying whether connection record is normal or attack - $DS1-V1$.

Model	Acc	Pre	Rec	F1	Time (seconds)	
					Training	Testing
LR	80.9	80.4	88.9	84.4	3.222	0.016
NB	45.6	88.4	7.6	14.0	0.360	0.160
KNN	99.8	99.8	99.8	99.8	2.290	46.870
DT	100	100	100	100	8.110	0.0240
AB	100	100	100	100	176.240	1.670
RF	100	100	100	100	59.320	0.740
RSVM	100	100	100	100	518,400.540	251.040
LSVM	100	100	100	100	451,215.210	185.020

TABLE IV: Test results for identifying whether connection record is BASHLITE or Mirai - $DS1-V2$.

Model	Acc	Pre	Rec	F1	Time (seconds)	
					Training	Testing
LR	56.0	0	0	0	2.794	0.030
NB	51.3	47.5	99.7	64.3	0.269	0.104
KNN	100	100	99.9	99.9	14.614	37.181
DT	100	100	100	100	0.749	0.016
AB	100	100	100	100	0.809	0.027
RF	100	100	100	100	14.933	0.332
RSVM	100	100	100	100	501,541.460	221.800
LSVM	100	100	100	100	478,541.410	169.700

TABLE V: Test results for classifying the connection records into the subcategories of BASHLITE or Mirai - $DS1-V3$.

Model	Acc	Pre	Rec	F1	Time (seconds)	
					Training	Testing
LR	11.1	10.3	11.1	2.3	199.484	0.096
NB	7.2	10.7	7.2	8.6	0.426	0.681
KNN	93.9	95.2	93.9	93.7	11.620	37.498
DT	98.5	98.6	98.5	98.5	58.227	1.236
AB	38.3	28.3	38.3	26.2	121.012	3.936
RF	93.6	94.9	93.6	93.4	7.695	0.026
RSVM	95.8	94.1	94.8	95.1	645,847.280	284.100
LSVM	96.8	95.7	95.1	95.9	517,114.250	210.700
DNN	100	100	100	100	412, 111.150	1.310

To accurately identify a generalizable algorithm, the same set of ML classifiers are trained with the same set of parameters on the $DS2$ data set. This data set contains two sets, one is to classify the samples into either normal or attack and the other is to classify the attack into their corresponding attack families. The detailed statistics are reported in Tables VI and VII. In the first experiment, all the models achieved very similar performance. However, in the second experiment, the LR, NB and AB classifiers performed poorly when compared to the rest of classifiers. Moreover, most of these classifiers have achieved a zero false positive rate. Most importantly, the DT classifier performed well in both the experiments in terms of standard metrics as well as training and testing time. It is computationally inexpensive compared to other classifiers. DNN is not trained for the $DS2$ data set as some of the ML models achieved perfect performance. From the experiments conducted using both data sets, it can be observed that the DT is an optimal algorithm and it is generalizable across different types of data sets. It is noticed that DNN performed better than the DT classifier in the case of the $DS1$ dataset. Thus, both the DT and DNN models can be used for botnet detection and can be deployed in real-time IoT systems.

TABLE VI: Test results for identifying whether connection record is normal or attack - $DS2-V1$.

Cl	Acc	Pre	Rec	F1	Time (seconds)	
					Training	Testing
LR	100	100	100	100	7.263	0.011
NB	99.4	100	99.4	99.7	0.413	0.090
KNN	100	100	100	100	7925.903	13.838
DT	100	100	100	100	42.048	0.052
AB	100	100	100	100	283.057	6.739
RF	100	100	100	100	884.666	5.681
RSVM [4]	100	100	100	100	1,672,675.840	14.790
LSVM [4]	100	100	100	100	905,893.870	5.553

Apart from the DT and DNN models, the RF, KNN and

TABLE VII: Test results for categorizing attacks into their corresponding categories - $DS2-V2$.

Cl	Acc	Pre	Rec	F1	Time (seconds)	
					Training	Testing
LR	52.6	33.4	26.4	19.8	53.707	0.070
NB	52.9	47.1	53.4	24.4	0.561	0.199
KNN	99.8	99.6	99.7	99.6	8488.109	13.189
DT	99.9	98.2	98.2	98.2	36.960	0.083
AB	12.2	61.2	35.3	20.5	296.122	10.937
RF	99.1	99.5	97.8	98.6	881.833	9.363
RSVM [4]	100	100	100	100	864,000.140	351.150
LSVM [4]	100	100	100	100	777,600.180	264.500

SVM models also showed considerable good performance. However, when considering the training and testing time, they are at a disadvantage at critical real-time security applications. It can be observed that the SVM models took a long time for learning all the different patterns that exist in data sets during training and it is computationally expensive compared to all other algorithms.

VI. Conclusion

This paper presents a DL based IoT botnet attack detection framework that relies on network flows and it can be used to secure smart city applications like health care, power grid, water treatment, traffic control, etc. The proposed system can be employed to handle a very large amount of data by incorporating more resources to the existing big data environment. The proposed DL based method outperformed the classical ML classifiers. This work uses features that are extracted from the network traffic flows. Instead, the entire payload data can be analyzed to distinguish between the legitimate and botnet attacks activity. Additionally, the network flow can be represented in the form of images and application CNN can be utilized for analysis and multi-modal learning can also be employed to enhance the performance of the existing works.

Acknowledgment

This work was supported by the Department of Corporate and Information Services, Northern Territory Government of Australia and in part by Paramount Computer Systems and Lakhshya Cyber Security Labs. We are grateful to NVIDIA India, for the GPU hardware support to the research grant. We are also grateful to Centre for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore, for encouraging this research.

References

[1] Vinayakumar, R., Alazab, M., Srinivasan, S., Pham, Q. V., Padannayil, S. K., & Simran, K. (2020). A Visualized Botnet Detection System based Deep Learning for the Internet of Things Networks of Smart Cities. IEEE Transactions on Industry Applications.

[2] Akamai, "State of the Internet Security Q3 2016", 2016. Source: https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q3-2016-state-of-the-internet-security-executive-summary.pdf.

[3] Vinayakumar R, Mamoun Alazab, Soman KP, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman. "Deep Learning Approach for Intelligent Intrusion Detection System." IEEEAccess.

[4] Koroniotis, Nickolaos, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset." arXiv preprint arXiv:1811.00701(2018).

[5] Vinayakumar R., Soman K.P., Poornachandran P., Alazab M., Jolfaei A. (2019) DBD: Deep Learning DGA-Based Botnet Detection. In: Alazab M., Tang M. (eds) Deep Learning Applications for Cyber Security. Advanced Sciences and Technologies for Security Applications. Springer, Cham

[6] S. Akarsh, S. Sriram, P. Poornachandran, V. K. Menon and K. P. Soman, "Deep Learning Framework for Domain Generation Algorithms Prediction Using Long Short-term Memory," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 666-671.

[7] V. R., M. Alazab, A. Jolfaei, S. K.P. and P. Poornachandran, "Ransomware Triage Using Deep Learning: Twitter as a Case Study," 2019 Cybersecurity and Cyberforensics Conference (CCC), Melbourne, Australia, 2019, pp. 67-73.

[8] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran and S. Venkatraman, "Robust Intelligent Malware Detection Using Deep Learning," in IEEE Access, vol. 7, pp. 46717-46738, 2019.

[9] Meidan, Yair, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders." IEEE Pervasive Computing 17, no. 3 (2018): 12-22.

[10] M. Ozcelik, N. Chalabianloo, and G. Gur, "Software-Defined Edge Defense Against IoT-Based DDoS," in 2017 IEEE International Conference on Computer and Information Technology (CIT). IEEE, 8 2017, pp. 308–313.

[11] D. H. Summerville, K. M. Zach, and Y. Chen, "Ultra-lightweight deep packet anomaly detection for Internet of Things devices," in 2015 IEEE 34th International Performance Computing and Communications Conference, IPCCC 2015, 2016.

[12] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: A Novel Honeypot for Revealing Current IoT Threats," Journal of Information Processing, vol. 24, no. 3, pp. 522–533, 2016.

[13] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, "A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology," in 2016 IEEE International Conference on Communications (ICC). IEEE, 5 2016, pp. 1–6.

[14] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach," Computer Communications, 2017.

[15] D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino, "Kalis A System for Knowledge-Driven Adaptable Intrusion Detection for the Internet of Things," in 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 6 2017, pp. 656–666.

[16] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," Ad Hoc Networks, vol. 11, no. 8, 2013

[17] Ioulianou, Philokypros, et al. "A Signature-based Intrusion Detection System for the Internet of Things." Information and Communication Technology Form (2018).

[18] Bahşi, Hayretdin, Sven Nõmm, and Fabio Benedetto La Torre. "Dimensionality Reduction for Machine Learning Based IoT Botnet Detection." 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). IEEE, 2018.

[19] Nõmm, Sven, and Hayretdin Bahşi. "Unsupervised Anomaly Based Botnet Detection in IoT Networks." 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2018.

[20] Chawathe, S. S. (2018, November). Monitoring IoT Networks for Botnet Activity. In 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA) (pp. 1-8). IEEE.

[21] Mohamed, T., Otsuka, T., & Ito, T. (2018, June). Towards Machine Learning Based IoT Intrusion Detection Service. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 580-585). Springer, Cham.

[22] Anthi, Eirini, Lowri Williams, and Pete Burnap. "Pulse: An adaptive intrusion detection for the Internet of Things." (2018): 35-4.

[23] Taheri, S., Salem, M., & Yuan, J. S. (2018). Leveraging Image Representation of Network Traffic Data and Transfer Learning in Botnet Detection. Big Data and Cognitive Computing, 2(4), 37.