Neural Networks for DDoS Attack Detection using an Enhanced Urban IoT Dataset

Arvin Hekmati
Dept. of Computer Science
University of Southern California
Los Angeles, California, USA
hekmati@usc.edu

Eugenio Grippo
Dept. of Electrical and Computer Engineering
University of Southern California
Los Angeles, California, USA
egrippo@usc.edu

Bhaskar Krishnamachari
Dept. of Computer Science
University of Southern California
Los Angeles, California, USA
bkrishna@usc.edu

2022 International Conference on Computer Communications and Networks (ICCCN) | 978-1-6654-9726-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCCN54977.2022.9868942

***Abstract*— We investigate the application of artificial intelligence to cybersecurity, to contribute to the safe and secure growth of the internet of things (IoT). Specifically, we train and evaluate different neural networks models to detect distributed denial of service (DDoS) attacks in a large-scale IoT system. We consider futuristic attacks launched by sophisticated malicious entities that take over multiple distributed IoT nodes and are able to disguise their intrusion by closely mimicking the benign traffic of the network. Using data from prior work, we find that a truncated Cauchy distribution is a suitable fit for benign traffic volume from IoT devices, and we model the attack traffic volume as following the same distribution but with different parameters for location and scale. We emulate both benign and attack traffic by overlaying these traffic volume distributions on top of an activity status data trace from a real urban IoT deployment consisting of about 4000 nodes. Using our enhanced dataset, we compare four neural network models: multi-layer perceptron (MLP), convolutional neural network (CNN), long short-term memory (LSTM), and autoencoder (AEN), analyzing their performance as a function of a parameter that measures the deviation of the attacks from the benign data. We observe that all four models are sensitive to the distance between benign and attack traffic. We further observe that LSTM gives the best overall performance in terms of both high accuracy and high recall.**

***Index Terms*—IoT DDoS Attacks, datasets, neural networks, machine learning, botnet, Cauchy distribution**

I. Introduction

The Internet of things (IoT) has dramatically grown propelled by the impetuous development of technology ([1], [2]). It might be said, however, that its vulnerability has been growing almost at the same pace. As a consequence, cybersecurity of the IoT ought to be developed at an ever-faster rhythm, allowing/accompanying the safe and secure growth of networks ([3], [4]). With this goal in mind, this work addresses one of the most dangerous types of attacks involving IoT systems, namely distributed denial of sevice attacks (DDoS) ([5],[6],[7]).

As can be learned from the famous IoT-based DDoS Mirai incident ([8],[9]) in 2016, botnet attacks can hijack thousands of IoT nodes at the same time, dramatically increasing the network traffic (on the order of Tbps) and shutting down servers, affecting millions of end-users. Given this dangerous and dramatically growing threat, we explore the use of machine learning techniques ([10]) as the main tool to prevent such attacks ([11]), specifically training neural networks to respond as rapidly as possible to avoid propagation of undesired attacks.

As we know, the success of a neural network model directly relates to its training stage, creating the necessity to feed models with rich trusted datasets. Although real/synthetic DDoS datasets have been provided or generated for many years (e.g., [12], [13], [14], [15], [16], [17]), there is a remarkable paucity of large datasets specifically focused on IoT in the literature with few exceptions ([18], [19], [20]). Table I presents an overview of datasets in this field.

Recently, we have released an anonymized dataset containing real-trace data from an urban deployment of 4060 IoT devices [21]. In this work, we further enhance that dataset with a python open-source script that allows the user to emulate DDoS network attacks in different network locations and with different intensity. The basis for this emulation is grounded in our finding that a real urban IoT benign traffic can be well-modeled as a (truncated) Cauchy distribution (matching the observation of prior researchers that Ethernet traffic is well modeled by such a distribution [22]). Our proposed "dataset+script" allows the injection of Cauchy distributed attacks, and it also lets the user to parametrize the difference between the benign and attack traffic volume. In this way, a diverse training scenario can be effectively generated to improve the training of neural network models. All these generations are tuned by a parameter that we call "k" that determines both the location and scale parameter of the attack traffic volume distribution.

We train, evaluate and compare the performance of four different neural network models, namely multi-layered perceptron (MLP), convolutional neural networks (CNN), long short-term memory (LSTM) and autoencoders (AEN). Prior work has explored the training of various neural network models for detecting anomalous DDoS traffic. For example, Meidan *et al.* [23] trained an autoencoder and found high performance for a setting where the attack traffic volume is quite substantially (orders of magnitude) different from benign traffic. We add to this literature in two ways: (1) by considering more sophisticated futuristic attacks where the attack traffic volume is much closer to the benign traffic, as determined by the parameter "k", and (2) by providing a quantitative

978-1-6654-9726-8/22/$31.00 ©2022 IEEE

comparison of the performance of four different NN models as a function of that parameter, helping to identify the best choice of model under different levels of similarity between benign and attack traffic. Through extensive simulation experiments, we conclude that:

- Both the accuracy and recall of all four NN models are sensitive to the "k" parameter.
- LSTM model provides generally the highest accuracy and recall for different values of "k".

We make the dataset and the attack emulation script along with our illustrative NN model available as an open-source repository online at https://github.com/ANRGUSC/IoT_DDoS_NN.

This work is organized as follows: in section II, we present the raw urban IoT dataset and its benign traffic characteristics; this section also introduces the modeling of the IoT benign traffic (defining the (truncated) Cauchy distribution) and builds the synthetic dataset defining the parameter that will regulate the relation/distance between benign and attack traffic. In section III we use the dataset to train, validate and compare different NN models to detect DDoS attacks deployed in the customized dataset. Lastly, section V summarizes this work and proposes future research steps.

II. Original and Benign Activity Datasets

The original data has been collected from the activity status of real event-driven IoT nodes deployed in an urban area. The source of this data, originally presented in [21], has been anonymized for privacy and security reasons. The original dataset contains three main features, the node ID, the location of the node in Latitude and Longitude, and a timestamp of the activity status of the IoT node. A record has been added to the original dataset whenever the activity status of a node changes. The raw dataset has 4060 nodes with one month worth of data with no missing data points.

Having a record of each node whenever the status of that node changes provides a bias towards the information of nodes that have more activity changes during the day. In order to overcome this issue, we also provide a script that takes the original dataset and generates a new benign activity dataset showing the activity status for each node every t_s seconds. In this way, all nodes in the benign activity dataset will have the same number of records. The script can generate a customized benign dataset by providing the beginning and ending date, the number of IoT nodes, and the time step, i.e., t_s.

In addition to the activity status of the nodes, going beyond what was presented in [21], we add the number of packets transmitted in each time step for each node to the benign dataset. Meidan et al. [23] presented a dataset containing the benign network traffic of 9 IoT nodes. We used the benign packet volume in a time window of 10 seconds of a security camera with ID XCS7_1003 as presented in [23]. We use this real network traffic data to generate traffic data for the benign dataset. Various distributions have been used in the literature for estimating the network traffic [31]. We analyzed 80 different distributions to fit to the packet volume in the real IoT security camera node. The Cauchy distribution fitted the best to the real network traffic by having the minimum mean square error. Due to the unusual nature of the full Cauchy distribution – unbounded in value and not having a defined mean, we use instead a truncated Cauchy distribution with a low of 0 and high of the maximum packet volume observed in the real data for generating the packet volumes in the benign dataset. The truncated Cauchy distribution for the benign traffic network is also compatible with prior work on modeling network traffic [22]. In order to generate the packet volume in the benign dataset, we will set the packet volume to zero whenever the node is inactive. On the other hand, when the node becomes active, packet volume will be generated from the fitted truncated Cauchy distribution. Table II presents a few sample data points in the original dataset.

In order to generate the packet volume for the attack dataset, we define a new truncated Cauchy distribution with the following parameters:

$$x_a = (1+k) \cdot x_b \tag{1}$$

$$\gamma_a = (1+k) \cdot \gamma_b \tag{2}$$

$$m_a = (1+k) \cdot m_b \tag{3}$$

where, x_b, γ_b, m_b refer to the location, scale, and maximum packet volume, respectively, of the truncated Cauchy distribution of the benign traffic; while, x_a, γ_a and m_a are the location, scale, and maximum packet volume, respectively, of the generated truncated Cauchy distribution of the attack traffic. k, is the tunable parameter for generating packets with higher location, scale, and maximum packet volume. While one could potentially use three different parameters for creating new x_a, γ_a, and m_a, we use only one parameter k just for simplicity.

Figure 1 shows the benign packet complementary cumulative distribution function (CCDF) in blue color besides the truncated Cauchy distribution with different k values. As we can see, the original truncated Cauchy distribution is well fitted to the real packet CCDF. By increasing k, we are basically increasing the location and scale of the truncated Cauchy distribution, which means that we will have higher probabilities for larger packet volumes, which is suited for the case of a DDoS attack where the attacker transfers a huge amount of packets during the attack. However, intuitively, a larger k may also make it easier to detect the attack.

Figure 2 presents the mean number of active nodes in the benign dataset versus the time of the day on one particular day of the dataset. As we can see, up to 65% of the nodes get activated around the middle of the day, but by midnight only about 20% of the nodes are active.

III. Attack and Defense Mechanism

This section presents how synthetic DDoS attacks are generated on the IoT nodes. Furthermore, here we define the training dataset features and also the detection mechanism.

TABLE I: Related Papers with IoT datasets

DATASET	Date	Number of Nodes	IoT specific/General	Binary activity or Traffic Volume	Benign/Attack traffic
DARPA 2000[24]	2000	60	general	traffic volume	both
CAIDA UCSD DDoS Attack 2007 [25]	2007	unclear	general	traffic volume	attack
Shiravi et. al. [26]	2012	24	general	traffic volume	both
CICIDS2017 [27]	2017	25	general	traffic volume	both
Meidan et. al. [19]	2018	9	IoT specific	traffic volume	both
CSE-CIC-IDS2018 on AWS [28]	2018	450	general	traffic volume	attack
Meidan et. al. [23]	2018	9	IoT specific	traffic volume	both
CICDDoS2019 [29]	2019	25	general	traffic volume	attack
The Bot-IoT Dataset (Univ. of NSW) [20]	2019	unclear	IoT specific	traffic volume	both
Ullah et. al [18]	2020	42	IoT specific	traffic volume	both
Erhan et. al. [30].	2020	4000	general	traffic volume	both
Hekmati et. al. [21].	2021	4060	IoT specific	binary activity	both
Hekmati et. al. (to be published)	2021	4060	IoT specific	traffic volume	both

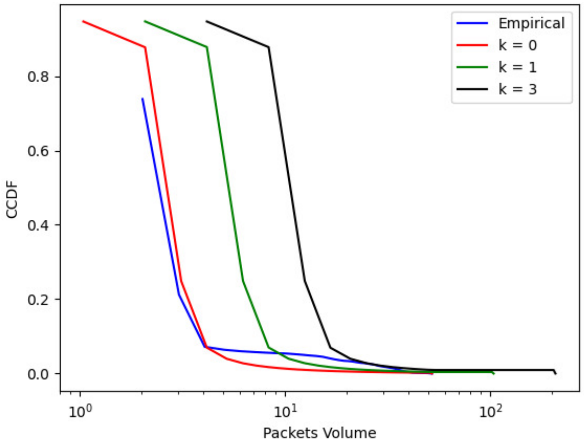


Fig. 1: Packets volume CCDF vs truncated Cauchy distribution

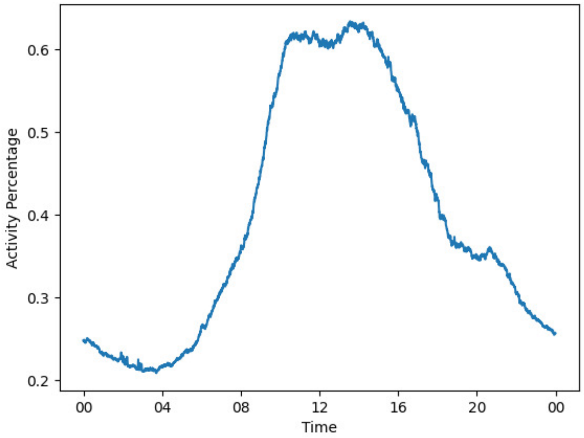


Fig. 2: Active Nodes Percentage vs Time

TABLE II: Sample Data Points in Benign Dataset

NODE	LAT	LNG	TIME	ACTIVE	PACKET
5276	33.962	40.087	2021-01-01 23:00:00	0	0
5276	33.962	40.087	2021-01-01 23:00:30	0	0
5276	33.962	40.087	2021-01-01 23:01:00	1	9
5276	33.962	40.087	2021-01-01 23:01:30	1	11

A. Generating Attacked Dataset

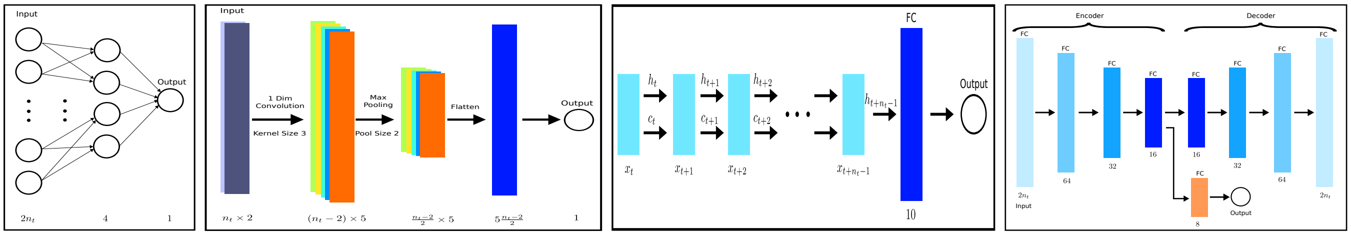
In this paper, we synthetically generate a DDoS attack on the IoT nodes by setting all attacked nodes to active status for the duration of the attack. Additionally, we set the packet volume distribution parameter, k and use equations (1), (2), and (3) to define the distribution and sample in i.i.d fashion from that distribution to generate the packets volume transmitted in each time step. Four parameters can be set in generating the attacks: the start time of the attack, the duration of the attack, the percentage of the nodes under attack, and the attack packet distribution parameter (k).

B. Generating Training Dataset

Given the attacked dataset, a labeled training dataset will be generated. The attacked dataset has two features of active status and packet volume with time-stamps. The attacked dataset could be considered as a time-series dataset. Therefore, in the training dataset, we stack the past n_t entries of each time-stamp to predict the attack status of the sample.

C. Defense Mechanism

For the defense mechanisms, we use four different deep learning models to do binary classification on each sample of the training dataset. For each deep learning model, we train a model for each IoT node by using its data alone. Note that this is a simple approach that will not take into account any correlations in the data across different nodes, so there is scope for further improvement by developing more complex models that integrate the inputs from multiple IoT devices. We defer the investigation of such more complex models to



(a) MLP

(b) CNN

(c) LSTM

(d) AEN

Fig. 3: Neural Network Models

future work. For each neural network model, we can choose n_t which presents the number of entries prior to the sample used for predicting the attack status. The input layer either used a two dimension mode ([n_t, 2]) or one dimension mode ([$2n_t$]), based on the neural network model, given that 2 refers to two features of the dataset. We used the standard scaler for scaling the training, validation, and testing dataset in all models. Furthermore, due to having an unbalanced training dataset with the minor class of attacked samples, we also analyzed the case of using upsampling in the training process for each model and the case without upsampling. Figure 3 presents the architecture for the neural network models. We tried different number of nodes and layers for each model and selected the one which performed the best. The details of each model are the following:

- **Multilayer Perceptron (MLP)**: In this model, we have an input layer with $2n_t$ neurons. The input layer is followed by one dense layer with 4 neurons and ReLU activation. A dropout of 20% and batch normalization are also used at the end of the hidden layer. The output is a single neuron with the Sigmoid activation function.
- **Convolutional Neural Network (CNN)**: In this model, we have an input layer in the shape of $[n_t, 2]$, The input layer is followed by one 1-dimensional convolution layer with 5 filters, kernel size of 3, and ReLu activation. This layer is followed by a 1-dimensional max-pooling layer with a pool size of 2, and then a flattened layer. Finally, the output is a single neuron with the Sigmoid activation function.
- **Long Short-Term Memory (LSTM)**: In this model, we have an input layer in the shape of $[n_t, 2]$. The input layer is followed by one LSTM layer with 10 units. Finally, the output is a single neuron with the Sigmoid activation function.
- **Autoencoder (AEN)**: In this model, we have an input layer with $2n_t$ neurons. The input layer is followed by an encoder that consists of three dense layers with 64, 32, 16 neurons, each followed by batch normalization and Tanh activation function. The decoder consists of three dense layers with 16, 32, 64 neurons, each followed by batch normalization and Tanh activation function. Finally, we have a dense layer with $2n_t$ neurons and Tanh activation function in the output layer. By using this model, we essentially train the encoder to encode our input dataset into a latent space. Then, we design a classification model that gets this latent space as input. This input layer is followed by one dense layer with 8 neurons and Tanh activation function. The output layer consists of one neuron with a Sigmoid activation function.

IV. Neural Network Models Evaluations

In this section, we evaluate the proposed neural network models by doing extensive simulations to analyze each model's performance in different scenarios.

A. *Experiment Setup*

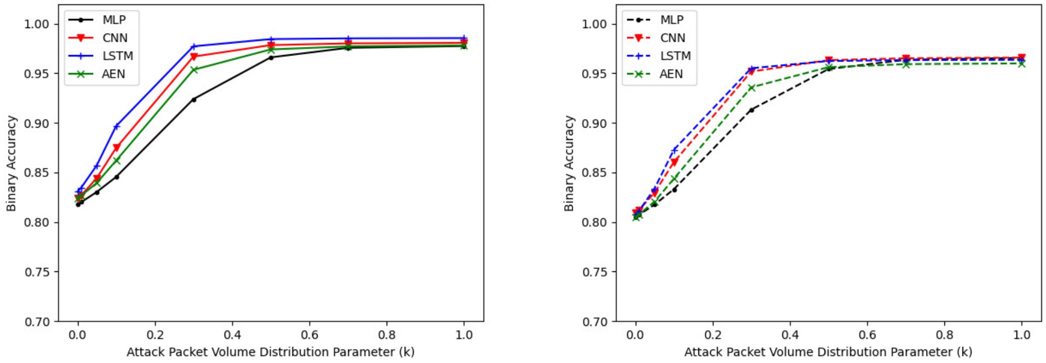
In these experiments, we randomly selected 20 IoT devices out of 4060 nodes, and a time step of 30 seconds was used for generating the benign dataset. We used three different days of the dataset for the training, validation, and testing. The attacks are started at 2 AM on all of the nodes with durations of 1, 2, 4, 8, and 16 hours. Note that one could randomize the starting time of the attacks. In that case, we can also use the timestamp as an input feature to the neural network models for better prediction. A list of 8 different values for k is used for generating attacks: 0, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, and 1. Note that this approach helps generate attacks that have very similar packet volume to the benign traffic ($k = 0$) and generate attacks that are very different from the benign traffic ($k = 1$). In order to generate the training dataset, we considered a time window of 10, i.e. $n_t = 10$, which means the neural network models will make predictions based on the information on the past 10 time slots of each sample. After generating the datasets, it has been shuffled to do not have prediction bias towards the time of the day. All proposed neural network models have been trained for 50 epochs using 32 batches.

B. *Mean Accuracy and Recall*

Figures 4, 5, present the binary accuracy and recall value of the neural network models versus the attack packet volume distribution parameter (k), in the case of not upsampling the training dataset. Furthermore, figures 6, 7, present the binary accuracy and recall value of the neural network models versus the attack packet volume distribution parameter (k), in the case of upsampling the training dataset. Finally, table III presents the mean recall and accuracy of the 20 models trained for detecting DDoS attacks, for k equal to 0 and 1 in the training

TABLE III: Mean testing dataset accuracy and recall for the neural network detection models

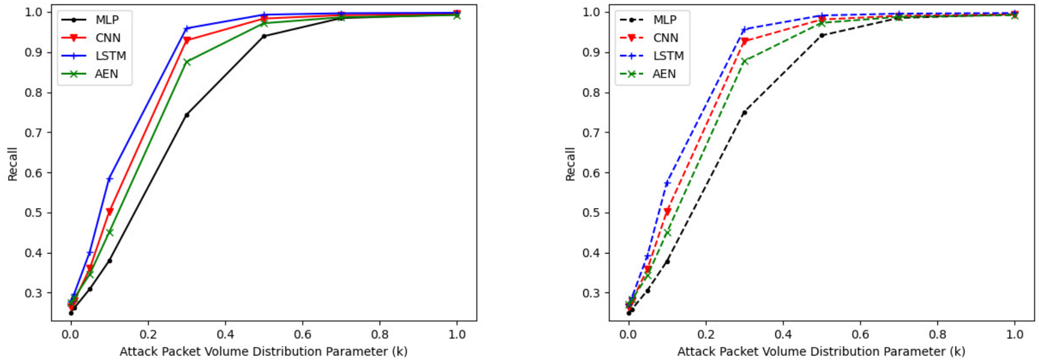
	Mean Accuracy Without Upsampling		**Mean Accuracy With Upsampling**		**Mean Recall Without Upsampling**		**Mean Recall With Upsampling**	
k	0	1	0	1	0	1	0	1
MLP	0.81	0.96	0.77	0.83	0.25	0.99	0.72	1.0
CNN	0.81	0.96	0.78	0.86	0.26	0.99	0.66	1.0
LSTM	0.81	0.97	0.79	0.89	0.27	1.0	0.56	1.0
AEN	0.81	0.96	0.78	0.86	0.26	0.99	0.65	1.0



(a) Training dataset

(b) Testing dataset

Fig. 4: Binary accuracy of training and testing dataset vs k without upsampling



(a) Training dataset

(b) Testing dataset

Fig. 5: Recall value of training and testing dataset vs k without upsampling

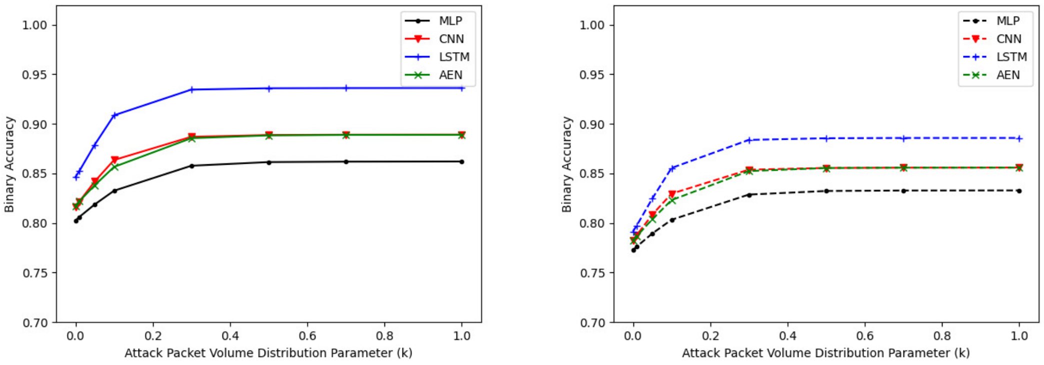
and testing dataset with/without upsampling. As we can see, in general, for all models, the upsampling method in both training and testing dataset, improves recall performance at low values of k but hurts accuracy for large values of k. With and without upsampling, the LSTM model generally has the highest binary accuracy and recall. The only exception is the recall value for low values of k in the case of upsampling, where other models are performing better than the LSTM, with MLP offering the highest recall in this case.

C. Attack Prediction versus Time

Figures 8 and 9 show the ground truth for attacks (True), attack predictions true positive (TP) and false positives (FP) mean overall nodes versus time, for both training and testing dataset, with k value equals to 0 and 1. We used the attack duration of 16 hours in these figures and the LSTM model with upsampling for our predictions. As we can see, in the case of $k = 0$, the model has a hard time detecting the attacks because the attacker is blending its behavior with the real behavior of the IoT nodes. On the other hand, in the case of $k = 1$, since the attackers are using higher packet volumes for attacking, the LSTM model is doing a very good job of detecting the attacks.

V. Conclusion

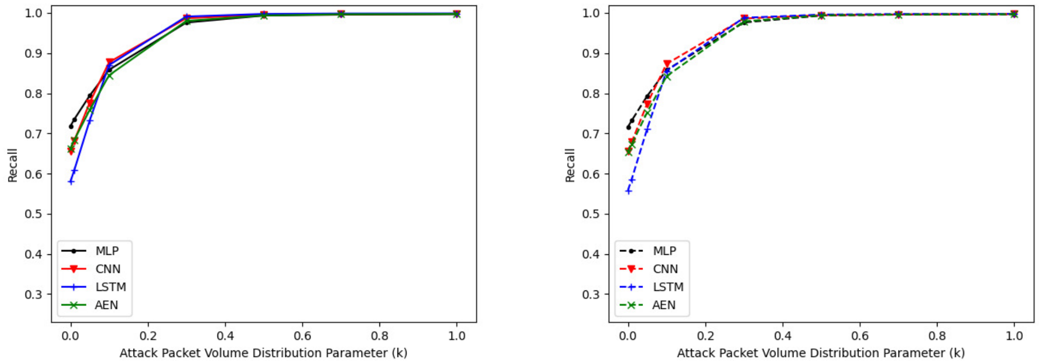
After modeling the benign traffic of a city IoT based system as a (truncated) Cauchy distribution, we developed a 4060 synthetic dataset based on a real urban IoT data, enhanced



(a) Training dataset

(b) Testing dataset

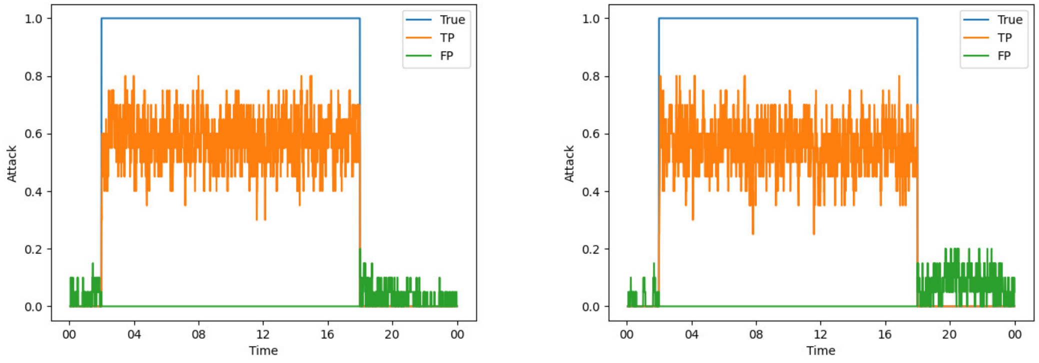
Fig. 6: Binary accuracy of training and testing dataset vs k with upsampling



(a) Training dataset

(b) Testing dataset

Fig. 7: Recall value of training and testing dataset vs k with upsampling

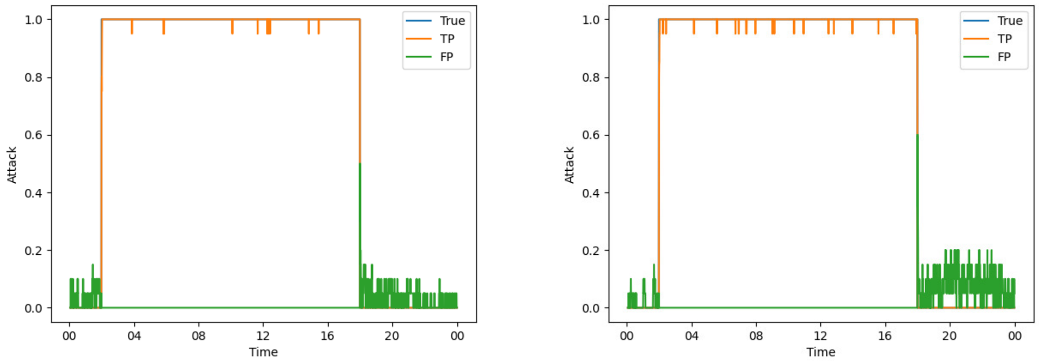


(a) Training dataset

(b) Testing dataset

Fig. 8: Attack prediction vs time for $k = 0$

by a script to inject attack traffic following a parameterized truncated Cauchy distribution. The proposed program allows the user to modify the dataset by modifying a single parameter, called “k”, that regulates the distance (in the location and scale sense) between the benign and attack traffic; this feature generates different dataset scenarios where benign and attack distributions scale and location are varied as desired, bringing multiple dataset choices for proper training and analysis of



(a) Training dataset

(b) Testing dataset

Fig. 9: Attack prediction vs time for $k = 1$

NN models performance. We particularly observed that the DDoS attack detection performance of four different NN models (MLP, CNN, LSTM and autoencoders) trained with this dataset are sensitive to the mentioned distance parameter.

In future work, we plan to consider more complex detection models that can take into account spatial correlations in the benign traffic. It may also be of interest to develop a real packet generator based on our dataset to emulate benign and attack traffic for use in cybersecurity testbeds.

VI. Acknowledgments

This material is based upon work supported by Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0160 for the Open, Programmable, Secure 5G (OPS-5G) program. Any views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] S. H. Shah and I. Yaqoob, "A survey: Internet of things (IoT) technologies, applications and challenges," in *2016 IEEE Smart Energy Grid Engineering (SEGE)*. IEEE, 2016, pp. 381–385.

[2] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-khah, and P. Siano, "IoT-based smart cities: A survey," in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, 2016, pp. 1–6.

[3] W. H. Hassan *et al.*, "Current research on internet of things (IoT) security: A survey," *Computer networks*, vol. 148, pp. 283–294, 2019.

[4] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[5] R. Hallman, J. Bryan, G. Palavicini, J. Divita, and J. Romero-Mariona, "IoDDoS-the internet of distributed denial of sevice attacks," in *2nd international conference on internet of things, big data and security. SCITEPRESS*, 2017, pp. 47–58.

[6] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[7] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer*, vol. 50, no. 2, pp. 76–79, 2017.

[8] H. Sinanović and S. Mrdovic, "Analysis of mirai malicious software," in *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2017, pp. 1–5.

[9] A. Marzano, D. Alexander, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, M. H. P. C. Chaves, I. Cunha, D. Guedes, and W. Meira, "The evolution of bashlite and mirai iot botnets," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 00 813–00 818.

[10] C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet detection in the internet of things using deep learning approaches," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[11] N. Sharma, A. Mahajan, and V. Malhotra, "Machine learning techniques used in detection of DoS attacks: a literature review," *International Journal of Advance Research in Computer Science and Software Engineering*, vol. 6, no. 3, pp. 100–105, 2016.

[12] "University of New Brunswick, Canadian Institute for Cybersecurity," https://www.unb.ca/cic/datasets/, accessed: 09-11-2021.

[13] "University of California Irvine, KDD Archive," https://www.kdd.org/kdd-cup/view/kdd-cup-1999/Data, accessed: 09-11-2021.

[14] "University of New South Wales, The UNSW-NB15 Dataset," https://research.unsw.edu.au/projects/unsw-nb15-dataset, accessed: 09-11-2021.

[15] D. Gümüşbaş, T. Yıldırım, A. Genovese, and F. Scotti, "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems," *IEEE Systems Journal*, vol. 15, no. 2, pp. 1717–1731, 2021.

[16] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "Flow-based benchmark data sets for intrusion detection," in *Proceedings of the 16th European Conference on Cyber Warfare and Security. ACPI*, 2017, pp. 361–369.

[17] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "Creation of flow-based data sets for intrusion detection," *Journal of Information Warfare*, vol. 16, pp. 40–53, 2017.

[18] "A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks," https://sites.google.com/view/iot-network-intrusion-dataset, accessed: 09-12-2021.

[19] "University of California Irvine, Machine Learning Repository," https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT, accessed: 09-12-2021.

[20] "University of New South Wales, The Bot-IoT Dataset," https://research.unsw.edu.au/projects/bot-iot-dataset, accessed: 09-12-2021.

[21] A. Hekmati, E. Grippo, and B. Krishnamachari, "Large-scale urban iot activity data for ddos attack emulation," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 560–564. [Online]. Available: https://doi.org/10.1145/3485730.3493695

[22] T. Field, U. Harder, and P. Harrison, "Network traffic behaviour in switched ethernet systems," in *Proceedings. 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*. IEEE, 2002, pp. 33–42.

[23] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—network-based detection of IoT botnet

attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[24] "DARPA 2000 Intrustion Detection Scenario Specific Data Sets," https://www.ll.mit.edu/r-d/datasets/, accessed: 10-08-2021.

[25] "The CAIDA UCSD "DDoS Attack 2007" Dataset, 2007," https://www.caida.org/catalog/datasets/ddos-20070804_dataset/, accessed: 10-08-2021.

[26] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

[27] "Canadian Institute for Cybersecurity, "CICIDS2017," unb.ca, 2017," https://www.unb.ca/cic/datasets/ids-2017.html, accessed: 09-13-2021.

[28] "University of New Brunswick, "CSE-CIC-IDS2018 on AWS", 2018," https://www.unb.ca/cic/datasets/ids-2018.html, accessed: 09-13-2021.

[29] "University of New Brunswick, "DDoS Evaluation Dataset (CICDDoS2019)",unb.ca, 2019," https://www.unb.ca/cic/datasets/ddos-2019.html, accessed: 09-13-2021.

[30] D. Erhan and E. Anarım, "Boğaziçi University distributed denial of service dataset," *Data in brief*, vol. 32, p. 106187, 2020.

[31] B. Chandrasekaran, "Survey of network traffic models," *Washington University in St. Louis CSE*, vol. 567, 2009.