**2018 15th International Conference on
Control, Automation, Robotics and Vision (ICARCV)
Singapore, November 18-21, 2018**

Dimensionality Reduction for Machine Learning Based IoT Botnet Detection

Hayretdin Bahşi, Sven Nõmm, Fabio Benedetto La Torre

Abstract— **The rapid development of the internet of things caused severe security problems such as the cyber attacks launched by extremely huge botnets comprised of IoT devices. The detection of these devices is essential for protecting the networks. Recently, some of the studies have demonstrated the high accuracy of machine learning methods, including deep learning, in detecting IoT botnets. However, the minimizing of the required features for classification is highly needed for overcoming scalability and computation resource problems in IoT environments. Having results which can be readily interpretable by cyber security analysts and producing signatures for the contemporary intrusion detection or network monitoring systems are other significant factors in this area in which quick and widespread security adaption is highly required. In this study, we applied feature selection to minimize the number of features in detecting the IoT bots. It is shown that fewer features can achieve very high accuracy rates and afford interpretable results with a multi-class classifier based on a shallow method, decision tree.**

I. INTRODUCTION

Internet of Things (IoT) technology has been utilized in many areas including health monitoring, energy management, transportation, home automation or manufacturing. However, IoT devices are prone to various physical, network and application layer attacks [1] which may lead to business interruptions, privacy violations or even physical injuries. The implications of these attacks are not limited to the users of these systems; they create significant problems for all other information systems as the compromised devices enormously increase the damage capacity of botnets especially in denial of service attacks.

Recently, machine learning techniques have been utilized for solving many problems in the area of intrusion detection including the botnet detection [2]. On the one hand, a large number of dimensions makes the problem attractive from the deep-learning perspective [3]. On the other hand, the problem may be tackled by proper feature selection and dimensionality reduction methods. In [3], it was demonstrated that the application of deep learning methods may lead to very accurate models. At the same time, deep learning techniques require higher computational power, and results are not as easily interpretable as compared to some shallow learning cases.

In this study, we employed feature selection methods to explore the discriminatory powers of feature categories and identify the optimum number of features that provide higher accuracy rates in detecting the IoT botnets. We used publicly available IoT dataset created by [3] for training and testing our classifier models. We applied standard machine learning work-flow consisting of data preparation, feature selection, model training and validation, result interpretation. Proper application and analysis of the feature selection process is the main contribution of this study. The reduction of the feature set allows minimizing the computational complexity of the classifiers. In our study, we used decision tree classifiers which generated outputs that can be easily interpreted by security operators and utilized in the current intrusion detection and network monitoring systems. Another characteristic property of our work is that a single model is trained, which makes it more easy from the practical deployment and performance viewpoint.

The present paper is organized as follows: background information regarding some IoT-based botnet attacks is given, and literature about the application of machine learning to intrusion detection is reviewed in Section II. Section III explains the dataset used in this study and presents the applied methodology. The main results are presented in Section IV. Section V is devoted to the interpretation of the results. The final section concludes our study.

II. BACKGROUND INFORMATION

A series of massive distributed denial of service attacks were launched against several high profile targets such as a hosting company, OVH [4], an internet performance management company, Dyn DNS [5] and a journalist web page, Krebs on Security [6], in 2016. The main issue that distinguishes them from the other similar attacks is the huge traffic (exceeding 1Tbps) generated by a botnet, namely Mirai, comprised of IoT devices, mainly security cameras [7]. Mirai was an improved form of Bashlite and based on its source code [8].

Although various signature-based intrusion detection systems have been deployed and used for identification of attacks, machine learning methods promised to solve some problems of those systems such as dependence on signatures and false negative rates due to new types of attacks. These methods have been applied to botnet detection in many studies [9][10][11]. However, it is required to adapt them to the IoT environment in which additional system requirements

H. Bahşi is with Department of Software Science, School of Information Technology, Tallinn University of Technology, 12618 Tallinn, Estonia, hayretdin.bahsi@ttu.ee

S. Nõmm is with The Department of Software Science, School of Information Technology, Tallinn University of Technology, 12618 Tallinn, Estonia, sven.nomm@ttu.ee

F. La Torre, Department of Electronics, Information and Bioengineering, School of Industrial and Information Engineering, Politecnico di Milano, 20133, Milan, Italy, fabiobenedetto.latorre@mail.polimi.it

978-1-5386-9582-1/18/$31.00 ©2018 IEEE

should be fulfilled depending on the detection scope and locations of sensors, in the core or edge networks. The components at edge networks usually work with limited resources regarding the network bandwidth, computation power, battery capacity or storage size. Various real-world applications themselves demand low latency and high quality of service, imposing an additional burden on these resource-constraint edge devices. Although fog computing paradigm [12] helps to overcome these obstacles by locating fog nodes, which have balanced resources, and assigning security functions to them [13], resource utilization problem persists as lightweight devices, in contrast to, for example, considerably well-resourced smart-phones, are used in many application areas due to the design restrictions [14]. If the detection is performed at the core networks of organizations, then the solutions should scale to the enormous sizes of network traffic, encompass various attack types and discriminate malicious behavior from different types of normal traffic patterns.

The other main issue that has not drawn much attention in this domain is the interpretability of machine learning outputs. The detection decision is an input to a process, not a standalone action, in which the human experts who conduct incident analysis or system monitoring in security operating centers, are involved. A high accurate but low interpretable classifier does not suit the requirements of these operational environments. Another point is that it is always better if the results of learning outcomes are adapted to the current system monitoring and intrusion detection infrastructure without additional investments and major upgrades.

Machine learning has been applied to botnet detection in many studies which have not focused on specifically IoT networks [9][10][11]. Feature selection methods based on computationally expensive wrapper methods were proposed for optimizing the detection in those networks [15][16]. Deep autoencoders are utilized as an anomaly detection method for IoT networks (our study uses the dataset created in this work) [3]. In another study, classifiers such as k- nearest neighbors (kNN), support vector machines (SVM), decision trees, random forests and neural networks are applied to an IoT dataset[17]. Although this study uses Gini Score for feature selection, it evaluates the performance of one set of features (stateless features), does not elaborate on the minimum set and discuss the interpretability of the results. [18] formed a classifier, by using dense random neural networks, to detect denial of service attacks and gave a proof-of-concept demonstration of detection without systematic validation. Deep Eigenspace Learning is applied to detect malware by using the operational code (OpCode) sequences extracted from IoT devices [19]. Principal Component Analysis (PCA) is utilized for reducing the dimension of the dataset to detect intrusions [20]. PCA transforms the original features into a form in which the output of the classifier cannot be easily interpreted by cyber security analysts and readily utilized by signature-based intrusion detection systems. This study uses KDD Cup 99 which is not derived from IoT networks. A deep learning schema composed of Stacked Autoencoder and Softmax classification was proposed in [21]. However, the results are validated by using NSL-KDD dataset which does not either include IoT networks.

The studies which apply machine learning methods into the intrusion detection problem have mostly focused on improving the accuracy scores, paid less attention to the optimization of the system performance and had almost no intention to address the interpretability and adaptability issues. However, placing the sensors, collecting the training data, establishing the classifier, utilizing it for the detection and evaluating the result in a decision process should be reconsidered for the successful real-world implementation of these methods. For this purpose, collecting and processing minimal data that accomplishes high accuracy scores with better interpretable results are essential in the IoT context.

III. Methods

A. Dataset

The dataset utilized in our study includes the statistics of network traffic captured in a lab environment in which the typical normal behaviour and attack cases are simulated [3]. The network contains nine IoT devices belonging to different application categories such as security camera, webcam, baby monitor, thermostat, and door-bell. The malicious traffic includes the attacks launched by IoT devices compromised by Bashlite and Mirai malware. Each data record has 115 numeric features which reflect statistics of aggregated streams extracted from raw network traffic in most recent five time windows, 100ms, 500ms, 1.5 sec, 10 sec and 1 min. The traffic aggregation is performed in five major feature categories, host-IP, host-MAC&IP, channel, network-jitter, and socket. These categories and features (with the statistical methods used for producing the relevant feature) are shown in Table I. Statistics of the outgoing network traffic originated from the same IP are given in Host-IP category. Host-MAC&IP covers the traffic having the same MAC and IP addresses. Channel category includes the network statistics determined by source and destination hosts whereas socket also covers source and destination ports beside their IP addresses. Network-jitter category covers the time intervals between packet arrivals of channel type communication.

Packet counts, mean and variance of packet sizes, are included in all categories. Additionally, more detailed statistics such as magnitude, radius, covariance and correlation coefficient of packet sizes are given for channel and socket categories. In this paper, we referred to a feature with the representation as ”Feature Category Type-Time Window-Statistic Type”. For instance ”Host_IP-100ms- Pkt Count” corresponds to the feature that is computed by the packet count of the host-IP category at interval 100ms. Our classifiers are based on three labels, normal, Bashlite (gafgyt) or Mirai.

The dataset has 502,605 normal, 2,835,317 bashlite and 2,935,131 mirai records, meaning that the label distributions are approximately 8%, 45%, and 47%.

TABLE I: Feature Categories

Feature Categories	Features
Host-IP	Packet count, mean and variance (outbound)
Host-MAC&IP	Packet count, mean and variance (outbound)
Channel	Packet count, mean and variance (outbound) Magnitude, Radius, Covariance, Correlation Coef. (inbound and outbound)
Network Jitter	Count, mean and variance of packet jitter in channel
Socket	Packet count, mean and variance (outbound) Magnitude, Radius, Covariance Correlation Coefficient (inbound and outbound)

B. Machine learning Work-flow

While [3] provides a dataset which is relatively large and clean, it is unbalanced so that the ratio of normal data is very less compared to attack data. In the experiments in Section IV except Subsection IV-D, we did not change this distribution property but induced classifier with randomly selected around 1% of the records in each class. However, we also conducted additional experiments with the altered two datasets, which are balanced and unbalanced (skewed to normal data), in order to illustrate the validity of our results. Since all the features are numeric, Fishers score [22] values were computed for each feature concerning the three classes (normal, Gafgyt and Mirai).

$$F = \frac{\sum_{i=1}^{N} p_i(\mu - \mu_i)^2}{\sum_{i=1}^{N} p_i \sigma_i^2} \qquad (1)$$

where N is the number of different classes (in this case $N = 3$), μ is the mean value for the given feature computed over all classes, μ_i and σ_i are the mean value and standard deviation of the particular feature computed over the points belonging to the class i. Finally p_i is the proportion of the points belonging to the class i. Fisher's score allows ordering the features concerning their discriminating power. Large values of the Fishers score correspond to the higher discriminating power.

For classification purpose, we utilized decision tree and k-NN methods.

IV. Results

A. Choosing Classification Method and Feature Selection Parameter

Firstly, we induced classifiers with varying sizes of top features selected by Fisher's score. As the accuracy values of classifiers are shown in Table II, we selected 2, 3 and 10 features with the best scores and applied decision tree and k-NN. Decision tree classifier with three features provided a reasonable trade-off between feature numbers and accuracy values. Therefore, we used decision tree in the remaining experiments.

While an optimal number of features were selected with respect to the classification technique, it is worth to mention here that results are somewhat surprising. Just three features have allowed achieving very high accuracy scores. As scatter plot illustrating data distribution between normal traffic and two attack types in Figure 1, the classes can be readily identified when best three features, Host_MAC&IP-1min-Mean, Host_IP-1min-Mean, Host_MAC&IP-10sec-Mean, are utilized in the classifier.

TABLE II: Accuracy scores for different feature sets and different classifiers

Feature Set Size	Decision Tree	k-NN
2	0.9843	0.9805
3	0.9851	0.9724
10	0.9897	0.9497

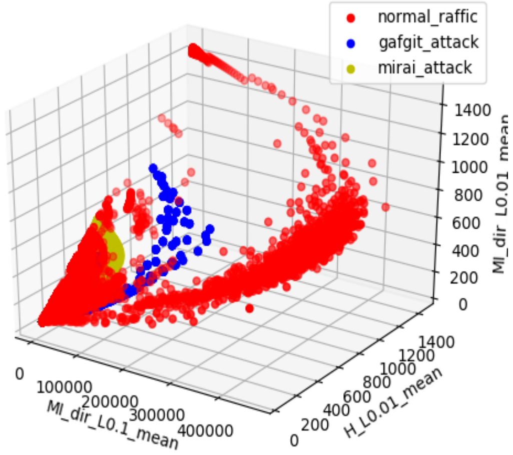

Fig. 1: Scatter plot describing the distribution of data points between different classes.

B. Discriminatory Power Analysis of Features

We computed the discriminatory power of feature categories and identified that which one contributes best to the classification. We used only the corresponding features of that category for training and testing the classifier (see Table I for the relevant categories). Table III presents the average accuracy values of 10-fold cross-validation results for each feature category. The obtained results demonstrate that host-based features, Host-IP or Host-MAC&IP, and network jitter have higher discriminatory powers as compared to channel and socket related features. We obtained accuracy value, 0.9996 via only host-based features, meaning that these features are enough for inducing high accurate classifiers. Also, the network jitter category provides very high value such as 0.9981.

TABLE III: Classification Results for Each Feature Category

Feature Category	**Accuracy**
Host-MAC&IP	0.9996
Host-IP	0.9996
Channel	0.7856
Network Jitter	0.9981
Socket	0.7821

Recall that the dataset includes statistics based on five different time intervals, 100ms, 500ms, 1.5 sec, 10 sec and 1 min. We investigated which time windows have the most capability to categorize the network traffic. As host-based features are sufficient for the classification with high accuracy, we analyzed the results of only these features for the given intervals. The average of accuracy values obtained by 10-fold cross-validation for all features (six features covering number, mean and variance of IP and MAC-IP categories) are shown in Table IV. The utilization of all features provides high accuracy values over 0.996 in all cases. The wider time windows have slightly better discriminatory powers.

TABLE IV: Classification Results Based on Time Windows Features

Time Intervals	Accuracy
100 ms	0.9961
500 ms	0.9969
1.5 sec	0.9986
10 sec	0.9994
1 min	0.9995

Table V gives the top ten features that have the highest Fisher's score values. It is important to note that the top features identified by the feature selection method are host-based features which are in line with the findings given in Table III. Additional observation is that all features are variance or mean of outbound packets. The time windows are 1 min, 10 sec or 1.5 sec, meaning that larger time windows have higher discriminating power (which is consistent with the results given in Table IV).

TABLE V: Best Features Based on Fisher's Scores

Feature	Fisher's Score
Host_MAC&IP-10sec-Variance	1.7035
Host_IP-10sec-Variance	1.7035
Host_IP-1.5sec-Variance	1.7051
Host_MAC&IP-1.5sec-Variance	1.7051
Host_MAC&IP-1min-Variance	1.7073
Host_IP-1min-Variance	1.7073
Host_IP-10sec-Mean	1.7858
Host_MAC&IP-10sec-Mean	1.7858
Host_MAC&IP-1min-Mean	1.8229
Host_IP-1min-Mean	1.8229

C. Device-Based Modeling

In the previous subsections, the classifier is applied to the whole dataset without dealing with data of each device seperately. In this subsection, we analyze such distinct modeling perspective by creating a distinct classifier for each IoT device.

Table VI provides the list of best three features, their Fisher's scores for four IoT devices (one security camera, one doorbell, one thermostat and one baby monitor). The scores are computed concerning the records of that specific device. Additionally, the accuracy value obtained by inducing a decision tree classifier is given in this table. All values are above 0.9890, indicating very high accuracy levels. In contrast to the features given in Table V, packet count values of host-based features are among the ones which have highest discriminatory power. All features belong to the time periods, 1min or 10sec. These observations indicate that despite commonalities in the feature selection based on the whole data, device-type modeling identifies variations regarding the selected features, reflecting the possible differences in device behaviour.

TABLE VI: Discriminatory Power of Features and Accuracy Values for Device Type

Device	Features	F. Score	Accuracy
Security Camera PT-838	Host_MAC&IP-10sec-mean	1.24	0.9930
	Host_MAC&IP-1min-mean	1.25	
	Host_IP-1min-mean	1.25	
Baby Monitor (B120N10)	Host_IP-10sec-mean	1.32	0.9994
	Host_MAC&IP-1min-Pkt Count	1.57	
	Host_IP-1min-Pkt Count	1.57	
Thermostat (Ecobee)	Host MAC&IP-10sec-Mean	3.33	0.9929
	Host MAC&IP-1min-Mean	3.38	
	Host_IP-1min-Mean	3.38	
Doorbell (Danmini)	Host MAC&IP-10sec-Variance	1.47	0.9896
	Host MAC&IP-1min-Variance	1.48	
	Host IP-1min-Variance	1.48	

Also, the variation may be observed by the scatter plots describing the distribution of the data points for different devices. For example, Figures 2 and 3 depict the classification results obtained for the doorbell and baby monitor devices, respectively. While in both cases class boundaries can be easily recognizable, distributions of the normal points in both devices differ. It can be derived that the normal traffic in door bell shows a very similar pattern so that all of the points are clustered in very small regions whereas the normal records of baby monitor are scattered to a wider area, meaning that behaviour of the baby monitor includes different patterns.

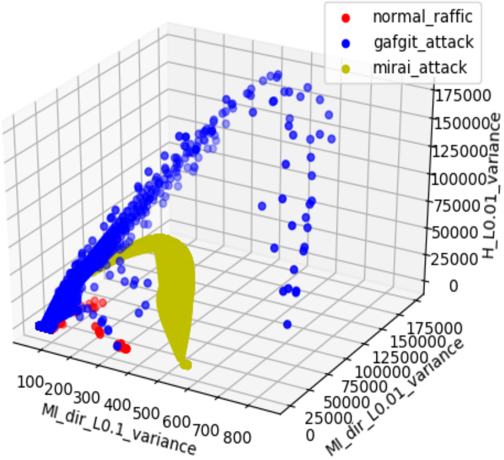

Fig. 2: Scatter plot describing the distribution of the data for door bell traffic.

D. Impact of Dataset Balancing on the Classifier Accuracy

The dataset is skewed to attack data (the ratio is around 92%) as it includes the data of denial of service attacks originated from compromised IoT devices in a controlled

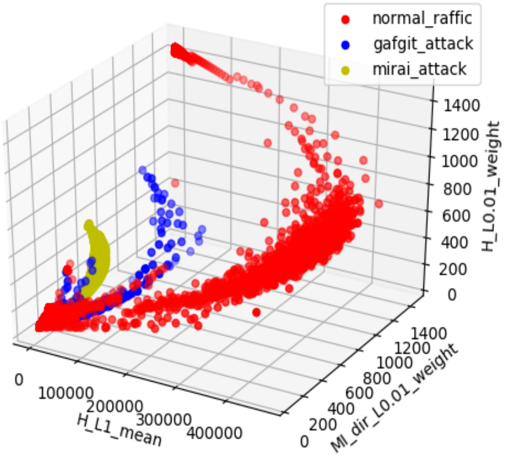

Fig. 3: Scatter plot describing distribution of the data traffic produced by the baby monitor.

environment. In a real-world setting, the balance of the dataset may differ as labeling requires human resources and simulating the attacks in an operational environment is not preferable. We conducted experiments with datasets having different balancing ratios in order to compare the detection rates of induced classifiers. We created additional two datasets out of the original data, balanced one that has equal number of records from each three classes and unbalanced dataset that is skewed to normal data. Table VII shows the distribution of the records in each dataset and gives the accuracy result when the best three features are used in the classifier. The obtained detection rates indicate that the balance of the dataset do not have a major impact on the detection rates as all the accuracy values are so close to each other.

TABLE VII: Dataset Balancing and Classifier Accuracy

Dataset Type	Normal	Gafgyt	Mirai	Accuracy
Unbalanced (Skewed to Normal)	49683	2798	2870	0.9980
Unbalanced (Skewed to Attacks)	5025	28382	29351	0.9947
Balanced (Equal Distr.n)	4000	4000	4000	0.9991

V. Discussion

In section IV, by means of cross-validation technique, it was established that three features allow training a decision tree classifier resulting in an accuracy score, 0.99 . A tree that depicts one of the training attempts is given in Figure 4, and corresponding confusion matrix is presented in Table VIII (note that this result is obtained from a dataset skewed to normal data). For this particular tree, the accuracy is 0.9897. One may easily see that the largest part of wrongly classified points belongs to the confusions between the attack types, Bashlite and Mirai, whereas the proportion of confusions between normal traffic and attack traffic may be neglected. It is evident that training and prediction with decision tree classifiers described above requires lesser computational power and may be easily deployed to work in real time. Besides the decision tree classifier, performance of k- nearest neighbors (kNN) is evaluated. The accuracy scores of the (kNN) classifiers are less than those of decision trees as shown in Table II.

One of the factors motivating the present study is to provide interpretable classification results. In this context, decision tree classifier is the best choice since it may be depicted in the form of tree graph where each node is a condition (rule) to be verified. The decision tree given in Figure 4 preserves the original features and clearly illustrates how the classifier determines the labels. The height of the tree is reasonable for human interpretation. A cyber security analyst can easily understand the attack or normal behavior and use this knowledge in further interpretation of security event or incident. The tree composition can be simply transformed to a set of rules that can be utilized by the contemporary signature-based intrusion detection systems. Although the easiness level of this utilization highly depends on the overlap between analytic constructs of intrusion signatures and features of utilized datasets (i.e, the signatures should be able to compute the relevant statistics used in our dataset), decision tree, as a machine learning method, does not provide an additional burden on the signature creation. For example, Snort, a well-known intrusion detection system, can represent the outcomes of our classifiers within their signature context after some minor upgrades as it has constructs for analyzing the network packets (by default, it uses socket representation) in a determined time window.

Another technique, frequently used for dimension reduction, is Principal Component Analysis (PCA). PCA uses orthogonal transformation to convert the data set with possible linear correlations between the variables into the dataset of where the variables are linearly uncorrelated. There exist a variety of results demonstrating applicability of PCA for feature selection in supervised learning. Nevertheless, it should be noted that during the orthogonal transformation, features (variables) lose their meaning, making it more complicated to construct attack signatures by trained classifiers.

Forming distinct classifier for each device may detect some variations in the behaviour, but it does not change the accuracy scores as the classifiers in both options (classifier built on the whole data or device-based classifier) give very high accurate values. Device-based classifier may not scale well in big networks; it is impractical when the detection is done at core-networks such as gateways of organisational networks. Although device-based approach could be an instrument in anomaly-based detection (as performed in [3]), we showed that accurate detection is possible by one common classifier with less number of features in a multi-class supervised setting.

VI. Conclusions

Results reported in this paper describe the applicability of decision tree classifier for botnet attack detection. Appli-

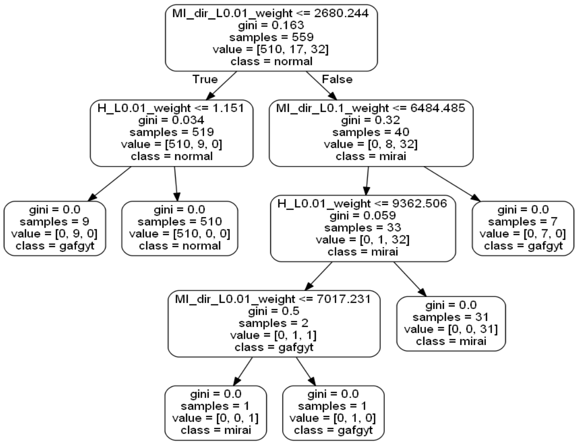

Fig. 4: Graph diagram of the trained decision tree classifier

TABLE VIII: Confusion Matrix - corresponding to the decision tree classifier depicted in Figure 4

	Actual normal	Actual Gafgyt	Actual Mirai
Predicted normal	49712	24	0
Predicted Gafgyt	13	2448	253
Predicted Mirai	14	266	2581

cation of the entire machine learning work-flow has allowed dramatically reduce the dimensions of the feature-set without any considerable loss on detection accuracy. The proposed model is based on one common classifier, instead of forming a classifier for each IoT devices, which makes it attractive from the deployment and online usage viewpoints in core networks. Finally, the results of decision tree classifier may be readily interpreted by the cyber security analysts and converted to the rules for signature-based intrusion detection systems which are used widely in many organisations.

References

[1] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of things: Security vulnerabilities and challenges," in *Computers and Communication (ISCC), 2015 IEEE Symposium on*. IEEE, 2015, pp. 180–187.

[2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[3] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici, "N-baiot: Network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 13, no. 9, 2018.

[4] P. Paganini, "Ovh hosting hit by 1tbps ddos attack, the largest one ever seen," *https://securityaffairs.co/wordpress/51640/cyber-crime/tbps-ddos-attack.html*.

[8] A. Gonzaga, D. Oliveira, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, M. Chaves, I. Cunha, D. Guedes, and W. Meira, "The evolution of bashlite and mirai iot botnets," in *IEEE Symposium on Computers and Communications*, 2018.

[5] S. Hilton, "Dyn analysis summary of friday october 21 attack (2016)," *URL https://dyn. com/blog/dyn-analysis-summary-of-fridayoctober-21-attack*.

[6] B. Krebs, "Krebsonsecurity hit with record ddos," *https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/*.

[7] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the mirai botnet," in *USENIX Security Symposium*, 2017.

[9] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, and D. Garant, "Botnet detection based on traffic behavior analysis and flow intervals," *Computers & Security*, vol. 39, pp. 2–16, 2013.

[10] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: detecting botnet command and control servers through large-scale netflow analysis," in *Proceedings of the 28th Annual Computer Security Applications Conference*. ACM, 2012, pp. 129–138.

[11] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*. IEEE, 2014, pp. 797–801.

[12] M. Iorga, L. Feldman, R. Barton, M. J. Martin, N. S. Goren, and C. Mahmoudi, "NIST Special Publication 500-325 Fog Computing Conceptual Model," Tech. Rep., 2018.

[13] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.

[14] W. Trappe, R. Howard, and R. S. Moore, "Low-energy security: Limits and opportunities in the internet of things," *IEEE Security & Privacy*, vol. 13, no. 1, pp. 14–21, 2015.

[15] E. B. Beigi, H. H. Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *Communications and Network Security (CNS), 2014 IEEE Conference on*. IEEE, 2014, pp. 247–255.

[16] F. V. Alejandre, N. C. Cortés, and E. A. Anaya, "Feature selection to detect botnets using machine learning algorithms," in *Electronics, Communications and Computers (CONIELECOMP), 2017 International Conference on*. IEEE, 2017, pp. 1–7.

[17] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," *arXiv preprint arXiv:1804.04159*, 2018.

[18] O. Brun, Y. Yin, E. Gelenbe, Y. M. Kadioglu, J. Augusto-Gonzalez, and M. Ramos, "Deep learning with dense random neural networks for detecting attacks against iot-connected home environments," in *Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London. Lecture Notes CCIS*, no. 821, 2018.

[19] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, "Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning," *IEEE Transactions on Sustainable Computing*, 2018.

[20] S. Zhao, W. Li, T. Zia, and A. Y. Zomaya, "A dimension reduction model and classifier for anomaly-based intrusion detection in internet of things," in *Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence & Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2017 IEEE 15th Intl*. IEEE, 2017, pp. 836–843.

[21] A. Abeshu and N. Chilamkurti, "Deep learning: the frontier for distributed attack detection in fog-to-things computing," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 169–175, 2018.

[22] C. Aggarwal, *Data Mining: The Textbook*. Springer International Publishing, 2015. [Online]. Available: https://books.google.ee/books?id=IbrirQEACAAJ