Unsupervised Anomaly Based Botnet Detection in IoT Networks

Sven Nõmm
Department of Software Science
School of Information Technology
Tallinn University of Technology
Tallinn, Estonia
sven.nomm@ttu.ee

Hayretdin Bahşi
Department of Software Science
School of Information Technology
Tallinn University of Technology
Tallinn, Estonia
hayretdin.bahsi@ttu.ee

***Abstract*—Anomaly-based detection of the IoT botnets with emphasis on feature selection is elaborated in this paper. Due to the rapid growth of the Internet of Things technology, the number of vulnerable devices that become a part of a botnet has grown significantly. The detection of such malicious traffic is essential for taking timely countermeasures. While the idea of anomaly-based attack detection is not new and has been extensively studied, much less attention has been paid to dimensionality reduction in learning models induced for IoT networks. In this paper, we showed that it is possible to induce high accurate unsupervised learning models with reduced feature set sizes, which enables to decrease the required computational resources. Training one common model for all IoT devices, instead of dedicated model for each device, is another design option that is evaluated for resource optimization.**

***Index Terms*—Anomaly detection, feature selection, dimensionality reduction, botnet attack, internet of things.**

I. Introduction

The present paper is devoted to the anomaly based detection of bots in the Internet of Things (IoT) networks by using unsupervised machine learning methods. Particular attention is paid to the feature selection process. The idea, the application of anomaly detection to the internet security, is not new and was proposed more than thirty years ago [1]. Since that time, the amount of traffic over the Internet has dramatically increased, and the number of devices connected to this enormous network has ramped up. Nowadays, IoT technology is another significant enabler that greatly extends the scale of the Internet. However, the malicious actors can employ various physical, network and application layer attacks against IoT networks [2]. These attacks do not only pose an essential threat to these networks themselves, but they threaten other networks as well. In 2016, massive denial of service attacks, exceeding 1Tbps bandwidth sizes, originated from IoT bots and targeted some companies such as a hosting company, OVH [3], an internet performance management company, Dyn DNS [4]. In order to detect these and similar threats, the research community has recently addressed the intrusion detection problem in IoT networks [5].

Machine learning techniques constitute an important solution approach for creating a detection mechanism as they promise to identify the new types of attacks [6]. These techniques also have a prominent role in providing anomaly-based intrusion detection function in IoT networks. However, they should be optimised to be deployed in various locations of IoT networks depending on the placement strategy of intrusion detection system (distributed, centralized or hybrid) [5]. The distributed strategy may require to run tasks on resource-constraint IoT devices or fog nodes whereas centralized one should be able to process a vast amount of network data generated by a high number of IoT devices. In both cases, the minimization of resource consumption is essential.

Typical dataset describing the system activity may have many features (for example the dataset used for the present research has 115 features derived from network traffic [7]). First, high dimensional data requires more computational power which in turn will make the results unattractive from the viewpoint of practical deployment. Second, such data may harm the accuracy of machine learning method due to the curse of dimensionality problem. This problem requires to remove features possessing no discriminating power. Third, the ability to get a traceable decision is an important factor. All these factors justify the necessity of dimensionality reduction in this application area of machine learning.

In [8], it was demonstrated that, based on just selected three features, it is possible to construct a decision tree classifier, a supervised one, with higher accuracy rates. This led us to the idea of applying the feature selection in an unsupervised learning model for anomaly detection purpose. In real life cases, a benign (normal) traffic from each particular IoT device or entire network is usually available or relatively easy to obtain. In contrary, the attack traffic are not easily labelled as this effort requires human resources that most organizations cannot afford. This motivates us to train an anomaly detection model with only benign traffic and use the attack traffic (in addition to benign one) for model verification. The comparison with the outcome of a supervised learning model will be included to justify the accuracy of the achieved results. Strictly speaking, training with benign may not be accepted as pure unsupervised learning, more like a particular case of supervised learning where only one class is available for training. Nevertheless, techniques such as one class support vector machines (SVM) or local outlier factors are frequently

978-1-5386-6805-4/18/$31.00 ©2018 IEEE
DOI 10.1109/ICMLA.2018.00171

referred as unsupervised anomaly detection.

Our research objective is to induce classifiers that utilize the minimal feature set with higher detection rates and thus, provide more interprerable results and require low computational complexity in an one-class learning setting where only the benign traffic is used for training. Moreover, we compared the detection performance of one common learning model (created for all IoT devices) with an approach that constructs a separate model for each IoT device as these modelling approaches may have a significant impact on computational complexity.

This paper demonstrates that it is possible to induce a high accurate unsupervised learning model for IoT botnet detection with reduced feature set. The main contribution of this study is the detailed analysis of discriminatory capabilities of features and the comparison of detection performance of one common model with separate one.

This paper is organized as follows: In Section II, some background information is presented, and the review of literature is given. The dataset utilized in this study is introduced, and the problem statement is stated in Section III. The methodology of the proposed solution and the main results are presented in Section IV. Section V provides a discussion about the achieved results. The last section is devoted to concluding remarks.

II. Background Information and Literature Review

The botnet life-cycle can be divided into four phases, formation, command and control (C&C), attack and post-attack [9]. The malicious activities which correspond to the delivery of the malicious payload to the target and exploitation of the vulnerabilities belong to the formation phase. C&C phase covers the communication of compromised devices (bots) with C&C structure. The attacks launched by the bots, which constitute the primary objective of the botnet, are carried out at the attack phase. The efforts of attackers for extending the botnet with new members are performed at the post-attack stage. The dataset that we used in this study includes network traffic belonging to the attack phase (i.e., spam and denial of service attacks) and partly post-attack phase (i.e., scanning attacks).

Botnet detection approaches can be classified into two categories, signature- and anomaly-based detection [10]. The signature-based approach relies on the rules created by security experts. It does not enable to detect unknown types, imposing a significant limitation in the rapidly evolving cyber threat landscape. The update of the detection systems with the recently generated signatures is strictly required. However, in a real-world setting, such updates may not be completed due to many procedural misconducts. On the other side, anomaly-based systems, which deduce the profiles of normal usage and then use them for detecting the deviations, may also cause false positives as they may report any normal but rare behavior as suspicious activity. Another disadvantage is that they may require intensive computation resources for profiling the normal system usage [5].

Machine learning methods have provided promising results in intrusion detection [6]. These methods have been also specifically applied to botnet detection in classical networks. A group of bots are detected by X-means, an unsupervised clustering method [11]. This study correlates the outcomes of two separate clustering processes, one for detecting the communication at C&C phase and one for identifying the malicious behaviour of bots at the attack stage. In another study, a clustering algorithm is applied to the network traffic generated by the bots installed in the controlled environment [12]. Unsupervised methods, Self-Organizing Map (SOM), Local Outlier Factor (LOF) and k-NN outlier were applied to botnet detection problem [13]. This study creates a distinct classifier for each network service (e.g., HTTP, HTTPS, DNS) and trains the model with only normal data.

There are relatively few studies that focus on the adaptation of machine learning methods to the botnet detection problem in IoT networks. An unsupervised anomaly detection model is formed for each IoT device by using deep autoencoders, a deep learning method (note that our study uses the dataset created in this work) [7]. Despite the fact that this study achieves high accurate results, the method requires considerable computational resources and the outputs are not interpretable without the utilization of additional constructs. k- Nearest neighbors (kNN), support vector machines (SVM), decision trees, random forests and neural networks are applied to an IoT dataset [14]. Although this study only focuses on feature selection in a supervised classifier, it just compares the discriminatory power of stateless features against other features without giving more detailed analysis. A model with dense random neural networks is constructed for detecting denial of service attacks [15]. Principal Component Analysis (PCA) is utilized for reducing the dimension of the dataset to detect intrusions on a dataset that does not include IoT networks [16].

III. Data Description and Formal Problem Statement

The dataset contains 115 numeric features which are the statistics of the benign (normal) or malicious (attack) network traffic generated by 9 IoT devices such as security camera, webcam, baby monitor, thermostat, and door-bell [7]. The malicious traffic includes denial of service and spam type attacks launched by the devices compromised by Bashlite or Mirai malware. The features are classified into five categories, host-IP, host-MAC&IP, channel, network jitter and socket (see Table I). The statistics belonging to the most recent five different time windows (100ms, 500ms, 1.5 sec, 10 sec and 1 min) constitute the feature set. Host-IP category consists of features related to packet counts, mean and variance of packet sizes of traffic originated from the same IP. Host-MAC&IP category gives the same statistics of the traffic having the same MAC and IP addresses. Channel category contains the network statistics determined by the source and destination hosts whereas socket category adds the source and destination port information as an aggregation criterion. More detailed statistics such as magnitude, radius, covariance and correlation coefficient of packet sizes are given for channel and socket categories. Network-jitter category covers the time intervals between packet arrivals of channel type communication.

In this paper, we represent a feature 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. The source dataset has 502,605 normal, 2,835,317 bashlite and 2,935,131 mirai records, meaning that the label distributions are 8%, 45%, and 47%, respectively. Our model treats bashlite and mirai records as one category (as attack). Therefore, the classifier model learns from benign data and dissects the malicious one from the benign traffic at the verification phase.

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)

IV. Proposed Solution and Main Results

As the original dataset has a much larger proportion of the attack data than a normal one, it may cause higher detection accuracy scores. In order to avoid this problem and show the validity of our results, we also verified our models with a dataset which is sampled with equal proportions of normal and attack records. Feature selection has a well-established procedure for supervised and unsupervised learning. In supervised learning, the discriminating power of the feature is evaluated concerning the labels of the training dataset. In the case of unsupervised learning, the features leading better clustering tendency are selected.

Three measurement approaches which require considerably less computational complexity are used for selecting the candidate feature sets. The first approach identifies the features with higher entropy. The second one selects the features demonstrating higher variance as they can properly identify isolation boundaries. The third one uses Hopkins statistics that measures the similarity of the corresponding feature values with the uniform distribution. As Hopkins statistics is a less popular notion, let us briefly explain it for the sake of self-sufficiency. For each feature of the original dataset D, let R be the r -points, a sample randomly drawn from this dataset, and S be a synthetic sample of r uniformly distributed data points from the same interval with the given feature of the original dataset. Let $\alpha_1, \ldots, \alpha_r$ be the distances between points of R to their nearest neighbour in D and $\beta_1, \ldots, \beta_r$ the distances between points of S to their nearest neighbour in D. Then [17] defines Hopkins statistic as follows:

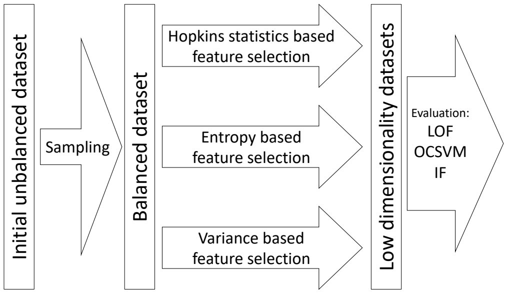
Initial unbalanced dataset
Sampling
Balanced dataset
Hopkins statistics based feature selection
Entropy based feature selection
Variance based feature selection
Low dimensionality datasets
Evaluation:
LOF
OCSVM
IF

Fig. 1. Data processing work-flow.

$$H = \frac{\sum_{i=1}^{r} \alpha_i}{\sum_{i=1}^{r} \alpha_i + \beta_i}. \tag{1}$$

The data with chosen feature sets are then evaluated with using the three most popular techniques for anomaly detection: local outlier factor (LOF), one class SVM and isolation forest (IF). Corresponding average accuracy scores and recall score values are computed by 10-fold cross-validation procedure. As the obtained accuracy rates for LOF were very low, those results are not reported in this paper.

The overall work-flow of data processing is depicted in Figure 1. First, this workflow is applied to the sampled data to evaluate the performance of one model created regardless of the IoT device type. Second, a separate model for each IoT device is created and trained with selected feature sets. In each workflow iteration, the experiments are conducted for two datasets having different class label distributions. The first dataset is created by randomly selecting 20,292 records from attack records (bashlite or mirai) and 2747 from normal records. This dataset is called ”unbalanced” as the ratio of the normal data is around 11% which is similar to the original dataset. The second dataset, which is called ”balanced”, is comprised of 6406 attack and 6407 normal records, meaning that ratio of normal data is around 50%.

Table II presents the accuracy and recall values obtained from the experiments in which different feature-set sizes, feature selection and unsupervised learning methods are applied for unbalanced and balanced datasets (i.e., one common model is created for all IoT devices). The first column describes the details of the induced model. For instance, the value, ”3-Entropy-SVM”, indicates that the model uses entropy for feature selection, applies one class SVM for unsupervised learning and selects the top three features according to the scores obtained by the selection method. It is important to note that in our previous study [8], in which supervised learning method is applied and feature selection is based on class labels, just three features were sufficient to discriminate the normal and attack traffic and visualize the separations of decision boundaries. In those experiments, we obtained high accuracy rates above 0.99 for unbalanced data although the model is also able to identify the malware type. It can be easily derived from Table II that the unsupervised models provide different results, and they are lower when compared to the results of our previous study. Entropy and isolation forests give relatively higher results in both unbalanced and balanced datasets whereas entropy and SVM achieved better results only in the unbalanced dataset. The best accuracy and precision values, which are above 90%, are obtained by entropy and isolation forests with five features. Additionally, variance and isolation forests with ten features provide a similar result. As a feature selection method, entropy, and as an unsupervised learning method, isolation forests are superior to the others.

TABLE II
EVALUATION OF THE DIFFERENT FEATURE SUBSETS WITH RESPECT TO THE CLASSIFIER ACCURACY / RECALL FOR THE CASE OF BALANCED DATASET

	Unbalanced Distr.		Balanced Distr.	
	Accuracy	Precision	Accuracy	Precision
3-Entropy-SVM	0.9315	0.9627	0.8337	0.7686
5-Entropy-SVM	0.9233	0.9199	0.6765	0.6072
10-Entropy-SVM	0.8827	0.8825	0.5050	0.5025
3-Entropy-Iso. For.	0.8385	0.9839	0.8585	0.8883
5-Entropy-Iso. For.	0.9561	0.9860	0.9234	0.9052
10-Entropy-Iso. For.	0.6527	0.9780	0.7752	0.8604
3-Variance-SVM	0.6403	0.8546	0.3604	0.3645
5-Variance-SVM	0.6904	0.8595	0.5302	0.4991
10-Variance-SVM	0.9132	0.9103	0.6364	0.5790
3-Variance-Iso. For.	0.4191	0.9628	0.6229	0.7773
5-Variance-Iso. For.	0.3922	0.9600	0.6055	0.7557
10-Variance-Iso. For.	0.9357	0.9860	0.9220	0.9068
3-Hopkins-SVM	0.7901	0.8753	0.3523	0.4112
5-Hopkins-SVM	0.8565	0.8777	0.4567	0.4738
10-Hopkins-SVM	0.8826	0.8824	0.5069	0.5035
3-Hopkins-Iso. For.	0.5743	0.9753	0.6316	0.7818
5-Hopkins-Iso. For.	0.5727	0.9744	0.6483	0.8057
10-Hopkins-Iso. For.	0.5768	0.9752	0.7809	0.8672

The confusion matrices of the model that give the best result (the case which uses entropy and isolation forests with five features) for unbalanced and balanced datasets are presented in Tables III and IV respectively. Although the model misclassified some of the normal records (277 out of 2747) in the unbalanced dataset, it was able to identify the great portion of the attack cases (19520 out of 20,292). The balanced dataset resulted in a similar classification rate for normal records. This is expected, because, both models learn from normal records regardless of the number of attack records. However, the ratio of misclassified attack records is higher in balanced data.

TABLE III
CONFUSION MATRIX FOR ISOLATION FOREST MODEL WITH 5 FEATURES SELECTED BY ENTROPY FOR UNBALANCED DATA

	Predicted Normal	Predicted Attack
Actual Normal	2470	277
Actual Attack	772	19520

Table V lists the best ten features selected by entropy method. The main observation that can be derived from this list is that the host-based features belonging to Host_IP and Host_MAC&IP categories constitute the most discriminating features (except one channel-based feature). These results are in line with the features selected by Fisher's Score in [8], as mean and variance statistics obtained from the host-based category constituted the best ten discriminating features in that study. Additionally, the best six features identified in supervised model are also on the list obtained in this work by the entropy method. The most of the features are obtained from larger time-frames (1 minute or 10 seconds) in both cases.

TABLE IV
CONFUSION MATRIX FOR ISOLATION FOREST MODEL WITH 5 FEATURES SELECTED BY ENTROPY FOR BALANCED DATA

	Predicted Normal	Predicted Attack
Actual Normal	2469	278
Actual Attack	117	2631

TABLE V
BEST 10 FEATURES BASED ON ENTROPY

Feature
Host_IP-1.5sec-Mean
Host_MAC&IP-1min-Variance
Host_IP-1min-Variance
Channel-1min-Mean
Host_MAC&IP-1min-Pkt. Count
Host_IP-1min-Pkt. Count
Host_IP-10sec-Mean
Host_MAC&IP-10sec-Mean
Host_IP-1min-Mean
Host_MAC&IP-1min-Mean

Table VI provides the category distribution of the best ten features selected by the corresponding feature selection method. Entropy and Fisher's score select nearly all features from host-based categories such as Host-IP and Host-MAC&IP. However, variance-based measurement method identifies attributes from the only network-jitter category, and Hopkins chooses from mostly socket and network jitter categories (one feature from channel category). In our previous study [8], although we found that the host-based categories have the highest discriminatory power, we also identified that network jitter category has similar power (slightly less) with host-based features. Therefore, it is apparent that the variance-based method was able to determine one of the highest discriminatory categories. On the other side, half of the features selected by Hopkins belong to Channel and Socket categories which have less discriminatory power.

These results helped us better describe some patterns in Table II. The models using entropy reach the highest accuracy and precision values with three features in SVM and five features in Isolation Forests. Even increasing further the feature size lowers the accuracy in those models as the results with ten features give less accurate results. The models benefiting from variance method reach their highest values when the size of the feature set is increased to 10, which may justify the choice of slightly less discriminatory features. On the other side, increasing the number of features do not make such difference in the models utilizing Hopkins. The reason behind

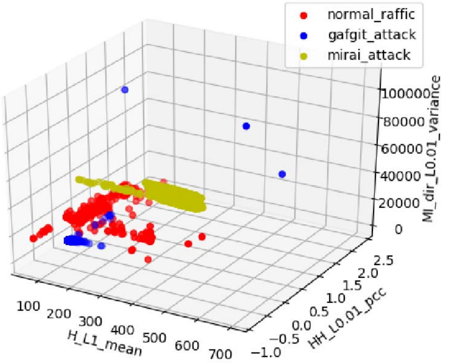

Fig. 2. The scatter plot depicting distribution of the normal- and attack-traffic.

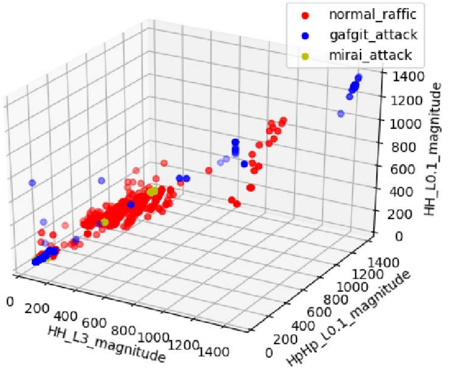

Fig. 3. Scatter plot depicting the distribution of the normal and attack traffic of Ecobee Thermostat

this relatively similar results could be that Hopkins identifies features with less discriminatory power.

TABLE VI
THE DISTRIBUTION OF BEST 10 FEATURES SELECTED BY FEATURE SELECTION METHODS ACCORDING TO THEIR CATEGORIES

Category	Fisher's	Entropy	Variance	Hopkins
Host-IP	5	5		
Host-MAC&IP	5	4		
Channel		1		1
Network Jitter			10	5
Socket				4

In Figure 2 the scatter plot provides visualisation of the classification results for the balanced dataset (i.e., three features with the highest entropy values are selected). Although there exist some overlaps between the regions of normal and attack records, the features achieve observable boundaries.

One of the computationally complex alternatives to the presented one-common-model solution is constructing a separate model for each IoT device to learn their normal behaviour more deeply. We repeated the work-flow given in Figure 1 for each IoT device with balanced and unbalanced datasets and obtained the results shown in Table VII. We presented the model that gives the best result for each device. The best model provides higher accuracy and precision values with at most ten attributes for some devices such as Danmini Doorbell, Ecobee Thermostat, SimpleHome XCS7-1002 and XCS7-1003. However, we obtained relatively low values for devices, Philips B120N/10, Provision PT-373E and PT-838. The study, which uses deep autoencoders as an unsupervised learning method, also gives relatively greater high false positive rates for aforementioned devices [7]. These results indicate that it could be hard for unsupervised methods to acquire the exact normal behavior of some devices. The best result of each device (except Philips B120N/10) is achieved by isolation forests. The other observation is that entropy is not the method that gives the best outcome for each device. Although entropy is the best alternative for three of the seven devices, variance provided better results in three devices, and the remaining one benefited from Hopkins. Similar to the results given in Table II, entropy method gives the best result when three or five features are selected whereas two of the three devices using variance reaches an optimum level with ten features.

TABLE VII
RESULTS OF DEVICE-BASED MODEL

		Unbalanced Distr.		Balanced Distr.	
Device	Model	Accur.	Prec.	Accur.	Prec.
Danmini Doorbell	5 Entropy Iso. For.	0.9954	0.9952	0.9500	0.9091
Ecobee Thermo.	3 Entropy Iso. For.	0.9981	0.9981	0.9494	0.9084
Philips B120N/10	10 Variance SVM	0.9537	0.9474	0.8618	0.7834
Provision PT-737E	10 Variance Iso. For.	0.8701	0.9914	0.8866	0.8962
Provision PT-838	5 Entropy Iso. For.	0.9561	0.9860	0.9234	0.9052
Simple Home XCS71002	10 Hopkins Iso. For.	0.9941	0.9938	0.9512	0.9116
Simple Home XCS71003	3 Variance Iso. For.	0.9976	0.9976	0.9565	0.9207

Figures 3 and 4 present the scatter plots that demonstrate the distribution of normal and traffic records of devices, Ecobee Thermostat and Philips B120N/10, respectively, according to the three best features selected by entropy method. In both figures, three features show observable boundaries between different classes. However, the normal records of Philips B120N/10 are more scattered, justifying the difficulty of learning in an anomaly-based approach.

V. DISCUSSION

The main result of this study is that achieving low computational complexity by reducing the feature set is possible for an IoT botnet detection system in which the learning model is

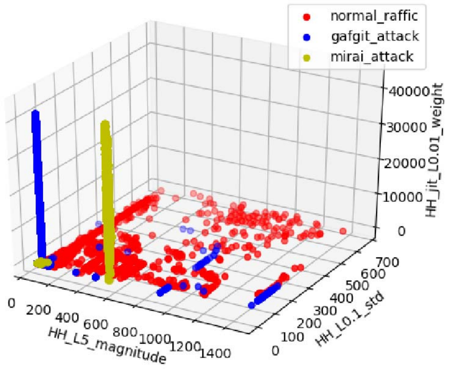

Fig. 4. Scatter plot depicting the distribution of the normal and attack traffic of Philips B120N/10

trained by only normal traffic. It is shown that such a trained model provides a reasonable accuracy and precision results.

The common approach in anomaly-based detection systems is the learning of normal behavior of each entity (it is IoT device in our context) by using separate model for each device. This study challenges this approach by evaluating the detection performance of one common model for all devices as model creation and maintenance may not be manageable in big IoT networks otherwise. Although we identified two learning model configurations that provide accuracy and precision values above 90% (in both balanced and unbalanced datasets), it can be concluded that separate model for each IoT device outperforms one common model.

In this study, we evaluated the performance of separate model (for each IoT device) that utilizes classical learning methods with selected features. Deep encoders, a deep learning method, was applied as an anomaly detection method to the same dataset in another study [7]. We achieved similar detection rates by using at most ten features with SVM and isolation forests (with separate model). As we used a limited number of features and learning methods requiring less computing resources, our models are more appropriate for IoT networks. Moreover, decreasing the feature set may provide better interpretable results for security analysts.

On the other side, we had considerably lower detection rates for some IoT devices, which are in line with the similar results in [7]. It can be derived that it is not easy to capture the normal behavior of some IoT devices fulfilling different functions (as depicted in Figure 4). However, it is important to note that, in general, profiling of IoT devices may be less complicated than the profiling of servers and user computers in classical networks which may include various services. It can be expected that research efforts regarding the anomaly detection in IoT networks could provide more promising results.

VI. Conclusions

The present paper has demonstrated that a feature selection procedure can reduce the required number of features in an unsupervised learning model that provides anomaly-based detection function in IoT networks. Reduced feature set enables to consume less computational resources and may lead to more interpretable results. It is shown that one model that uses classical learning methods (such as SVM or isolation forests) with less than ten features can achieve reasonable detection rates which is preferable from the scalability point of view. The other important finding is that although one common learning model for all IoT devices can achieve reasonable detection rates, inducing separate model for each device gives better detection rates.

References

[1] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, Feb 1987.

[2] 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.

[3] 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*.

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

[5] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[6] 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.

[7] 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.

[8] H. Bahsi, S. Nmm, and F. B. La Torre, "Dimensionality reduction for machine learning based iot botnet detection," in *15th International Conference on Control, Automation, Robotics Vision (ICARCV 2018)*. IEEE, 2018.

[9] J. Leonard, S. Xu, and R. Sandhu, "A framework for understanding botnets," in *Availability, Reliability and Security, 2009. ARES'09. International Conference on*. IEEE, 2009, pp. 917–922.

[10] S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey," *Computer Networks*, vol. 57, no. 2, pp. 378–403, 2013.

[11] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection," 2008.

[12] F. Tegeler, X. Fu, G. Vigna, and C. Kruegel, "Botfinder: Finding bots in network traffic without deep packet inspection," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 2012, pp. 349–360.

[13] W. Chen, X. Luo, and A. N. Zincir-Heywood, "Exploring a service-based normal behaviour profiling system for botnet detection," in *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*. IEEE, 2017, pp. 947–952.

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

[15] 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*, 2018.

[16] 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, 2017 IEEE 15th Intl*. IEEE, 2017, pp. 836–843.

[17] C. C. Aggarwal, *Data Mining: The Textbook*. Springer Publishing Company, Incorporated, 2015.