Energy Audition based Cyber-Physical Attack Detection System in IoT

Yang Shi
Center for Cyber-Physical Systems,
University of Georgia
Athens, Georgia
yang.atrue@uga.edu

Fangyu Li
Center for Cyber-Physical Systems,
University of Georgia
Athens, Georgia
fangyu.li@uga.edu

WenZhan Song
Center for Cyber-Physical Systems,
University of Georgia
Athens, Georgia
wsong@uga.edu

Xiang-Yang Li
University of Science and Technology
of China
Hefei, China
xiangyangli@ustc.edu.cn

Jin Ye
Center for Cyber-Physical Systems,
University of Georgia
Athens, Georgia
jin.ye@uga.edu

ABSTRACT

In this paper, we propose an attack detection framework in the Internet of Things (IoT) devices. The framework applies a data-centric method to process the energy consumption data and classify the attack status of the monitored device. We implement the framework in real hardware, and emulate common types of attacks to evaluate the performance of the attack detection framework. Due to the characteristic of the energy data, not only cyber attacks but also physical attacks such as heating are also emulated and tested. To shorten the detection time, a two-stage strategy is also proposed to first apply a short time window for a rough detection, then a long time window to the fine detection of anomalies. The accuracy of short-term detection is 90%, while in the long-term detections the accuracy reaches 99.5%. Due to the nature of information from energy consumption data, the framework is more secure in cases the kernel of the device is already compromised.

CCS CONCEPTS

• Security and privacy → Intrusion/anomaly detection and malware mitigation; **• Networks → Network security**; **Sensor networks**; **• Computing methodologies →** *Machine learning*.

KEYWORDS

Cyber-physical attack detection,Internet of Things, Energy consumption, Machine learning

ACM Reference Format:
Yang Shi, Fangyu Li, WenZhan Song, Xiang-Yang Li, and Jin Ye. 2019. Energy Audition based Cyber-Physical Attack Detection System in IoT. In *ACM Turing Celebration Conference - China (ACM TURC 2019) (ACM TURC 2019), May 17–19, 2019, Chengdu, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3321408.3321588

ACM TURC 2019, May 17–19, 2019, Chengdu, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7158-2/19/05…$15.00
https://doi.org/10.1145/3321408.3321588

1 INTRODUCTION

Computer systems use energy, and energy provides information about the system [29]. This information can be utilized in for security purposes. As the computer systems are more connected than before, the chance of a device getting affected by attacks are higher today. When human beings involve in the computational tasks a lot for computers and laptops, it is easier for the operators to find the attacks as their operations are affected by the attacks directly. However, for Internet of Things (IoT) systems, attacks are usually hard to be detected as the impacts can be delayed [25]. The security issues in IoT has become more important in recent days as IoT is more attainable to our daily life recently.

There are a lot of attack detection technologies are specifically developed for IoT systems to address the IoT security issue [14, 21, 25, 28]. Among the works solving the issues of IoT security, the detection methodologies are categorized as software-based methods and data-driven methods [13, 24]. While the software-based methods like blacklisting [32], firewalls [15] are not versatile enough for the changes in attacks, the data-driven methods are capable to adjust to changing attacks. The data-driven detection results are generated from the observations of the system usages and statistics, and thus is more secure [34]. However, since the system usage and statistics data are reported from the kernel, once the computer is compromised, the data integrity cannot be guaranteed. Moreover, the traditional data-driven methods are usually designed for detections of cyber attacks, but they rarely cover physical attacks. Some more recent works focus on using energy to detect anomalies [6, 11, 18, 19], but none of them dedicated to classify the category of attacks.

Energy consumption information from devices are good resources for monitoring the system security. This information is monitored outside from the operating system kernel, and thus cannot be compromised by hackers in the kernel of the monitored devices. Given the monitoring device is not participating on the computation and information sharing tasks of IoT, we use local networks for the monitoring devices to secure the energy consumption data integrity. Thus, we propose using energy consumption data as a more secure data driven solution for IoT attack detection. Moreover, previous works mainly focus on the cyber attacks, while for the physical attacks, our framework is a better fit naturally because of the data source. For example, in the attacks of Virus, the energy consumption is mainly from the computation of CPU, while in the physical attack of Heating, the actual usage of each component is not changed, but as the temperature increases, less heat cannot be radiated out from the devices, and thus the total energy usage is increased. In most existing methods, the data source is the system statistics or usages, which is the internal information of the system [24]. Energy consumption data is more capable as a physical data source for the physical attacks.

A data-centric framework for detecting attacks in IoT devices is proposed in this paper. In the framework, we attach a low-cost energy meter to monitor and collect the energy consumption data. The collected data is sent to a centralized server. Six major cyber or physical attacks including Virus, Intrusion, Deny of Service (DoS), Heating, Trojan, Power Line Cut for IoT devices are then emulated in the devices. The collected energy consumption data is labeled accordingly. We train and store statistical models based on the labeled data on the server. Once new data is collected from the devices, the models perform two-stage classifications. Within a short time, the short-term model detects whether the device is with anomalies, and within a longer time, the long-term model performs a finer classification of the attack type in the device.

The main contribution of this work is summarized as below:

- A data-driven framework of energy data based fast attack detection is designed, implemented and evaluated.
- The proposed framework is also able to detect not only cyber attacks and anomalies but also physical attacks.
- Six common cyber and physical attacks are emulated in IoT devices, device energy consumption data are collected.
- A two-stage detection strategy is designed and a statistical model is trained on the collected data, through which the security status of IoT devices are detected. Both cyber and physical types of attacks are classified with the model.

The rest of this paper is organized as follows. Section 2 describes the proposed system design, the architecture, and attacks we designed. The data processing is also introduced in this section. The analysis of our proposed framework, and its comparison with existing related schemes are presented in Section 3. Finally, we present our concluding remarks and future research directions in Section 4.

2 ANOMALY DETECTION SYSTEM DESIGN

The system of the framework is designed as shown in Figure 1. In this section, we first introduce the hardware design of the proposed IoT security system, then we introduce the data processing and classification of the framework.

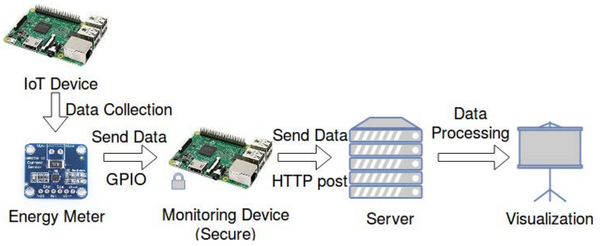

Figure 1: The proposed framework.

2.1 Hardware design

In the emulation of the energy monitoring and attack detections, we use the Raspberry Pi 3 Model B [3] as the IoT devices. This model is powered by USB cables and has GPIO pins for external inputs and outputs. The Raspberry Pis are installed with Raspbian [4] as the operation system. Raspbian is a Linux based system specifically designed for Raspberry Pis. Inside the Raspbian kernel, we install a background program to emulate the normal behaviors of IoT devices. The background programs by default periodically collect data from sensors, and store the data locally, then send the data to the processing server by batch. To achieve this, the Raspberry Pis are mounted with sensors for background program data collection. In our experiment, we also use the energy meters as the sensors.

We use INA 219 [1] as the energy meter in our experiment settings. The INA 219 energy meter is a lightweight and low-cost energy consumption sensor. It can be easily attached on the side of IoT devices. The USB cables are cut open and the positive power conductors are series connected with the energy meters. Every device is monitored by one energy meter.

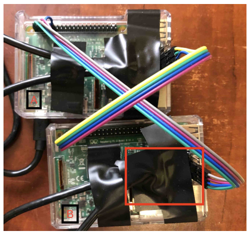

Figure 2: Hardware design of the proposed framework.

Shown in Figure 2 is our hardware of the framework. The IoT device B in the figure is the device to be monitored, and the device A is the monitoring device. As introduced, the INA 219 energy meter is shown in the area in the red frame, covered by the electrical tape. In the cover of the tape of B, as the USB offers energy for device B, the energy consumption is collected by the energy meter and sent to the device A through the GPIO pins. The power of the energy meter is also supplied by device A from the GPIO pins. The energy consumption data sampled at 1 Hz in device A is sent to a centralized InfluxDB [2] server, then processed for attack detection.

2.2 Data flow and visualization

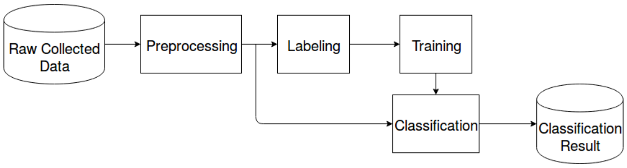

Figure 3: The data flow in the processing server.

In the processing server, the data is processed as shown in Figure 3. The data is first preprocessed and features are extracted for machine learning. We match the anomaly or normal situation of the collected data, and label the data in categories of Normal, Virus, Intrusion, DoS, Heating, Trojan, Power Line Cut. The labeled data is then trained with machine learning models, and the models are stored locally on the server. Once new data is collected from the devices and sent to the server, same preprocessing and feature extraction operations are applied and classifications are then performed by the trained model. The results of the classifications are stored in InfluxDB for visualization.

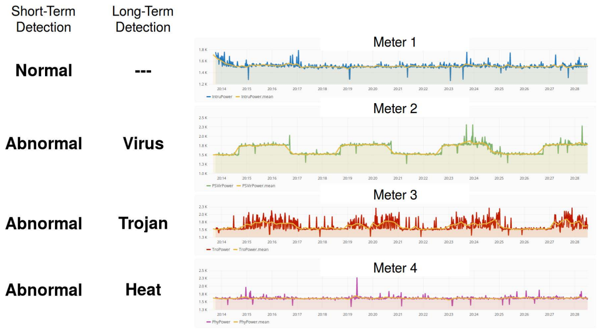

Figure 4: The visualization of proposed framework.

After the data is collected to the server, visualization is available in Grafana [12]. The visualization is shown in Figure 4, where we have four IoT devices monitored, and energy consumption data are plotted in separate panels. The detection results are in the first two panels of every row. The first panel of each row is the short term detection result, and detects whether the monitored device is under abnormal situation. The second panel of each row is the long term detection results, and shows the fine-grained detection result of the attack category that the anomaly belongs to.

2.3 Attack Design

We design the attack prototypes to perform the concept of attacks while avoid making malignant impacts on the systems. Recently, different attacks have been emulated in works from industrial IoT [27], smart home [10], smart phone [30] application scenarios. According to these works, the length of attacks can be long or transient. We decided to focus on the long attacks because of the limitation of the sampling rate of the energy consumption data. The length of a single attack prototype is set as 2 minutes. In total, we have 6 types of attacks, the cyber attacks are designed as below:

- `Virus` [23]: A program that continuously consume computational resource is created. The program performs prime factorization continuously for 2 minutes, stores the results locally, and then sleep for another 2 minutes.
- `Intrusion` [7]: The attacking device continuously guess the password of the monitored device, and try to use `ssh` command to log in the kernel.
- `DoS` [33]: Considering the similarity of DoS attack and port scanning, we use port scanning in the simulation of DoS attack. Another device continuously scan the port of the monitored device.
- `Trojan` [16]: One major characteristic and threat of trojans is that it will send critical informations to the controller of the trojans. We create a program that sends a trained deep learning model to the server every 2 minutes to simulate the process of sending data through networks.

The physical attacks are designed as below:

- `Heating` [9]: We place a 12 watt light bulb within 5 cm of the monitored device to heat the device.
- `Power Line Cut` [22]: The power line is directly pulled off in this attack type to simulate the situation that power lines are cut.

With the settings of emulated attacks, energy consumption data are collected and used to classify anomalies in the IoT devices.

2.4 Preprocessing and feature extraction

The collected signal is noisy and unstable. To get more stable readings, in the preporcessing part, a Savitzky-Golay filter [26] is applied to smooth the signals in a sliding time window. Since the normal and abnormal status are continuous and no transient attacks are emulated, we assume the short and intense value spikes in the collected data are noises. Median filters has been applied to eliminate these noisy spikes.

After getting the denoised and smoothed data, statistical and spectral features are extracted. As investigated on the related works [8, 31], features listed below are applicable for anomaly detections:

- `Mean,Standard deviation, Skewness, Kurtosis`: The first, second, third and forth moment of the windowed signal.
- `Min, Max`: The minimum and maximum value of the windowed signal.
- `Interquartile range (IQR)`: The distance of the first and third quartile of the windowed signal.
- `cumulative sum (CUSUM)`: The cumulative sum of the windowed signal.
- `Fast Fourier transform (FFT)`: The real part of the FFT on the windowed signal.

The extracted features are concatenated with the smoothed signal for the classification model training and evaluation. Due to the high dimensionality of the data, we apply principle component analysis (PCA) on the concatenated data for a faster classification. In the PCA process, we select the eigenvectors with the 10 highest eigenvalues, which contains 99.56% information from the original selected features, which are originally with a dimension of 548.

2.5 Two-stage detection

Due to the design of the system, the classification is based on the newest data available, and thus the detection speed is dependent on the sliding window of the detections. Since the system requires a fast detection, we separated our detection tasks into two stages, namely short-term and long-term classification.

For the short-term classification, we only use the smoothed signal without the features extracted in a short sliding time window, and for the long-term classification, we use the concatenated data containing the extracted features. The classification process requires a longer sliding time window to perform a higher performance.

2.6 Classification algorithms

In the framework, models are leveraged in the anomaly classification. As indicated, features of different attacks are different according to the attack characteristics. To distinguish the different distributions of the features, we apply k-nearest neighbor (KNN) [20] algorithm and neural networks (NN) [17] with different parameters, and also provide a performance comparison for evaluation.

In KNN algorithm, the neighbors within the k distance are assigned with a weight of $1/k$. The classification precess is described in the equation below:

$$\hat{y} = arg \min_{c \in C} \sum_{x_i \in \mathcal{N}} w_i ||x - x_i||_2, \tag{1}$$

where, the neighboring samples in the neighborhood $\mathcal{N}$ labeled as c are denoted as x_i, and the corresponding weight is denoted as w_i. In our situation, w_i is assigned as $1/k$. The input is classified as the label of samples that have a minimal weighted distance.

NN is a statistical model that use multi-layer perceptrons to build the classifier. In our comparison, we use Adam [5] as the optimizer and rectified linear unit (ReLU), denoted as $R(\cdot)$, as the activation functions. In each layer, the NN perform a linear transformation from the previous layer:

$$h_{i+1} = R(w_i h_i + b_i), \tag{2}$$

where the i-th layer's input is $h_i \in \mathbb{R}^{m_i \times n}$, the weight and bias on the layer are respectively denoted as $w_i \in \mathbb{R}^{m_{i+1} \times m_i}$ and $b_i \in \mathbb{R}^{m_{i+1} \times n}$, giving the sample size is n. The classifier employs a Softmax function to calculate the final possibilities of the label that the input sample is classified as. The Softmax function is described as:

$$\hat{y} = arg \max_{c \in C} \frac{h_L(c)}{\sum_{c \in C} h_L(c)}, \tag{3}$$

where the total layer of the model is L, and $h_L \in \mathbb{R}^{C \times n}$. The total number of classes is denoted as C.

3 EVALUATION

We evaluate the framework in two parts. After introducing the experiment settings, we first evaluate the classification accuracy, and then due to the fast requirement, we also evaluate on the detection speed of the framework.

3.1 Experiment settings

In the experiments, we launch the proposed attacks in four Raspberry Pi 3 models, and collect the energy consumption data in the attacked devices. We also collect the energy consumption data from the devices under the normal status as comparisons. It should also be noticed that according to the attack prototype design, we do not consider transient attacks in the evaluations.

In the classification period, the evaluation is based on 10-fold cross-validations to make sure the evaluation result is validated. We compare KNN and NN as the classification algorithms. The NN algorithms in training phase has the maximum iteration as 200, learning rate as 0.001. For 1 layer classifications, the layer has 100 feature nodes, and for 2 layer classifications, the features nodes are 100, 20 respectively. In short-term classifications, the length of time window is 10 seconds, while for long-term classifications, the time window length is 180 seconds.

The dataset consists of recordings from 4 IoT devices. Different real attacks are performed and emulated, and each recording is of 2 hours length. 3 recordings are collected from each device. In total, 85666 samples from 12 recordings are collected, among which half are the normal data, and another half are different attack data, from different categories of attacks.

3.2 Anomaly detection performance

The long-term anomaly detection performance is shown in Table 2, and the short-term anomaly detection performance are shown in Table 1. In the confusion matrices, every row is the ground truth, and the every column is the classification results. The positive is defined as the anomalies. The metrics that we use are accuracy, false positive rate (FPR), and sensitivity. The accuracy evaluates the general performance of the classifiers, while FPR and sensitivity evaluate the false alarms and missing detections respectively.

Table 1: Short-term anomaly detection model performances

Model	Accuracy	FPR	Sensitivity
1-layer NN	61.7%	12.7%	36.9%
2-layer NN	85.5%	8.7%	79.9%
2-nearest neighbor	85.3%	21.1%	91.5%
7-nearest neighbor	90.0%	6.9%	87.1%
12-nearest neighbor	89.9%	7.4%	87.2%

In the short-term classifications, the detection time is satisfying, which is only 5 seconds. For performance, the NN based models cannot classify the anomalies correctly, especially in 1 layer model. The KNN model performs better than the NN model in this situation, and achieves around 90% accuracy in $k = 7$ and $k = 12$ models. The accuracy is not satisfying in short-term situation, we further evaluate on the long-term version for a comparison.

Table 2: Long-term anomaly detection model performances

Model	Accuracy	FPR	Sensitivity
1-layer NN	97.2%	0.3%	94.8%
2-layer NN	97.8%	0.2%	95.8%
2-nearest neighbor	99.4%	0	98.8%
7-nearest neighbor	99.4%	0	98.9%
12-nearest neighbor	99.4%	0	98.8%

In the long-term classifications, the accuracies are satisfying. For NN based classifiers, the accuracy is around 97% to 98%, while for the KNN based algorithms, the accuracy is above 99%. The only misclassifications happen in the classification of Trojan and Virus, since both of them are periodical, and the periods of them are set as same in our experiment. and The classification accuracy is comparable with the software based anomaly detection in most state-of-art works. This result also shows that our proposed framework provides flexibility in the model that is chosen. However, the long-term classification requires long time window, which means that the classification is not stable until collecting the full time window of data. This is the reason we keep the short-term model available in the framework for a faster detection.

3.3 Anomaly detection speed

Since the detection speed is also important factor in real-time frameworks, we also compare the detection speed of different classification algorithms. The averaged classification time of each of the 80000 sample of the inputs are shown in Table 3. The experiment is performed on a PC with Intel Core i5-2500 CPU 3.30GHz, and the memory size is 32 GB.

Table 3: Detection time of anomaly detection models

Model	Short-term(s)	Long-term(s)
1-layer NN	1.78×10^{-6}	1.49×10^{-6}
2-layer NN	7.01×10^{-6}	1.86×10^{-6}
2-nearest neighbor	5.57×10^{-5}	2.04×10^{-5}
7-nearest neighbor	8.29×10^{-5}	4.98×10^{-5}
12-nearest neighbor	1.02×10^{-4}	8.82×10^{-5}

The detection time of a single anomaly classification depends on the time window length in our framework. The comparisons show that with a detection of less than 0.1 microseconds, the implementation of the framework satisfies with the requirement of our attack detection system.

4 CONCLUSION

In this paper, a framework for IoT cyber-physical attack detection is introduced, implemented and evaluated. The energy consumption based method is a data centric method, and more secure than kernel data based methods because of the nature of side channel analysis. The framework is not only able to detect some common types of the cyber attacks, but also physical attacks such as power line cut and heating. The system currently is able to detect the anomalies in a time range of around 5 seconds. In order to provide a more accurate classification, we use the long-term classification to detect the anomalies with an accuracy of more than 99% in 3 minutes.

ACKNOWLEDGEMENT

Our research is partially supported by NSF-1663709 and Southern Company.

REFERENCES

[1] [n. d.]. Adafruit INA219 Current Sensor Breakout. https://learn.adafruit.com/adafruit-ina219-current-sensor-breakout/overview

[2] [n. d.]. InfluxDB. https://www.influxdata.com/

[3] [n. d.]. Raspberry Pi 3 Model B. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[4] [n. d.]. Raspbian. https://www.raspberrypi.org/downloads/raspbian/

[5] 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 http://arxiv.org/abs/1412.6980

[6] Robert Bridges, Jarilyn Hernández Jiménez, Jeffrey Nichols, Katerina Goseva-Popstojanova, and Stacy Prowell. 2018. Towards malware detection via cpu power consumption: Data collection design and analytics. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 1680–1684.

[7] Ismail Butun, Salvatore D. Morgera, and Ravi Sankar. 2014. A Survey of Intrusion Detection Systems in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials* 16, 1 (FebJan 2014), 266–282. https://doi.org/10.1109/surv.2013.050113.00191

[8] Shane S. Clark, Benjamin Ransford, Amir Rahmati, Shane Guineau, Jacob Sorber, Kevin Fu, and Wenyuan Xu. 2013. WattsUpDoc: Power Side Channels to Nonintrusively Discover Untargeted Malware on Embedded Medical Devices. In *2013 USENIX Workshop on Health Information Technologies*. Washington, D.C. https://www.usenix.org/conference/healthtech13/workshop-program/presentation/Clark

[9] Luigi Coppolino, Valerio DAlessandro, Salvatore DAntonio, Leonid Levy, and Luigi Romano. 2015. My Smart Home is Under Attack. In *2015 IEEE 18th International Conference on Computational Science and Engineering*. IEEE, 145–151. https://doi.org/10.1109/cse.2015.28

[10] Ali Dorri, Salil S. Kanhere, Raja Jurdak, and Praveen Gauravaram. 2017. Blockchain for IoT security and privacy: The case study of a smart home. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* January (2017), 618–623. https://doi.org/10.1109/PERCOMW.2017.7917634

[11] Carlos Aguayo Gonzalez and Alan Hinton. 2014. Detecting malicious software execution in programmable logic controllers using power fingerprinting. In *International Conference on Critical Infrastructure Protection*. Springer, 15–27.

[12] Grafana Labs. 2018. Grafana. https://grafana.com/

[13] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Pierre-Louis Dubouilh, Ephraim Iorkyase, Christos Tachtatzis, and Robert Atkinson. 2016. Threat analysis of IoT networks using artificial neural network intrusion detection system. In *2016 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 1–6. https://doi.org/10.1109/isncc.2016.7746067

[14] Prabhakaran Kasinathan, Claudio Pastrone, Maurizio A. Spirito, and Mark Vinkovits. 2013. Denial-of-Service detection in 6LoWPAN based Internet of Things. In *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 600–607. https://doi.org/10.1109/wimob.2013.6673419

[15] Sylvain Kubler, Kary Främling, and Andrea Buda. 2015. A standardized approach to deal with firewall and mobility policies in the IoT. *Pervasive and Mobile Computing* 20 (July 2015), 100–114. https://doi.org/10.1016/j.pmcj.2014.09.005

[16] Amey Kulkarni, Youngok Pino, and Tinoosh Mohsenin. 2016. Adaptive real-time Trojan detection framework through machine learning. In *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 120–123. https://doi.org/10.1109/hst.2016.7495568

[17] S. C. Lee and D. V. Heinbuch. 2001. Training a neural-network based intrusion detector to recognize novel attacks. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 31, 4 (July 2001), 294–299. https://doi.org/10.1109/3468.935046

[18] Fangyu Li, Yang Shi, Aditya Shinde, Jin Ye, and Wen Zhan Song. 2019. Enhanced Cyber-Physical Security in Internet of Things through Energy Auditing. *IEEE Internet of Things Journal* (2019).

[19] Fangyu Li, Aditya Shinde, Yang Shi, Jin Ye, Xiang-Yang Li, and Wen Zhan Song. 2019. System Statistics Learning-Based IoT Security: Feasibility and Suitability. *IEEE Internet of Things Journal* (2019).

[20] Yihua Liao and Vemuri. 2002. Use of K-Nearest Neighbor classifier for intrusion detection. *Computers & Security* 21, 5 (Oct. 2002), 439–448. https://doi.org/10.1016/s0167-4048(02)00514-x

[21] Caiming Liu, Jin Yang, Run Chen, Yan Zhang, and Jinquan Zeng. 2011. Research on immunity-based intrusion detection technology for the Internet of Things. In *2011 Seventh International Conference on Natural Computation*. IEEE, 212–216. https://doi.org/10.1109/icnc.2011.6022060

[22] Ren Liu, Ceeman Vellaithurai, Saugata S. Biswas, Thoshitha T. Gamage, and Anurag K. Srivastava. 2015. Analyzing the Cyber-Physical Impact of Cyber Events on the Power Grid. *IEEE Transactions on Smart Grid* 6, 5 (Sept. 2015), 2444–2453. https://doi.org/10.1109/tsg.2015.2432013

[23] Mark A. Ludwig. 2016. *The Giant black book of computer viruses*. http://www.worldcat.org/isbn/194811755

[24] Jesus Pacheco and Salim Hariri. 2016. IoT security framework for smart cyber infrastructures. In *Foundations and Applications of Self* Systems, IEEE International Workshops on*. IEEE, 242–247.

[25] Pavan Pongle and Gurunath Chavan. 2015. A survey: Attacks on RPL and 6LoWPAN in IoT. In *2015 International Conference on Pervasive Computing (ICPC)*. IEEE, 1–6. https://doi.org/10.1109/pervasive.2015.7087034

[26] William H. Press and Saul A. Teukolsky. 1990. Savitzky-Golay Smoothing Filters. *Computers in Physics* 4, 6 (1990), 669+. https://doi.org/10.1063/1.4822961

[27] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. 2015. Security and privacy challenges in industrial internet of things. In *Proceedings of the 52nd Annual Design Automation Conference on - DAC '15 (DAC '15)*. ACM Press, New York, NY, USA, 1–6. https://doi.org/10.1145/2744769.2747942

[28] Niki Tsitsiroudi, Panagiotis Sarigiannidis, Eirini Karapistoli, and Anastasios A. Economides. 2016. EyeSim: A mobile application for visual-assisted wormhole attack detection in IoT-enabled WSNs. In *2016 9th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, 103–109. https://doi.org/10.1109/wmnc.2016.7543976

[29] Tsuyoshi Ueno, Fuminori Sano, Osamu Saeki, and Kiichiro Tsuji. 2006. Effectiveness of an energy-consumption information system on energy savings in residential houses based on monitored data. *Applied Energy* 83, 2 (Feb. 2006), 166–183. https://doi.org/10.1016/j.apenergy.2005.02.002

[30] Dong Wang, Jiang Ming, Ting Chen, Xiaosong Zhang, and Chao Wang. 2018. Cracking IoT Device User Account via Brute-force Attack to SMS Authentication Code. In *Proceedings of the First Workshop on Radical and Experiential Security - RESEC '18*. ACM Press, 57–60. https://doi.org/10.1145/3203422.3203426

[31] Hongyu Yang and Ruiwen Tang. 2016. Power Consumption Based Android Malware Detection. *Journal of Electrical and Computer Engineering* 2016 (2016), 1–6. https://doi.org/10.1155/2016/6860217

[32] Asma Zahra and Munam A. Shah. 2017. IoT based ransomware growth rate evaluation and detection using command and control blacklisting. In *2017 23rd International Conference on Automation and Computing (ICAC)*. IEEE, 1–6. https://doi.org/10.23919/iconac.2017.8082013

[33] Heng Zhang, Peng Cheng, Ling Shi, and Jiming Chen. 2015. Optimal Denial-of-Service Attack Scheduling With Energy Constraint. *IEEE Trans. Automat. Control* 60, 11 (Nov. 2015), 3023–3028. https://doi.org/10.1109/tac.2015.2409905

[34] Minhui Zou, Chengliang Wang, Fangyu Li, and WenZhan Song. 2018. Network Phenotyping for Network Traffic Classification and Anomaly Detection, In 2018 IEEE International Symposium on Technologies for Homeland Security (HST). *2018 IEEE International Symposium on Technologies for Homeland Security (HST).*