International Journal on Advanced Science Engineering Information Technology

Vol.10 (2020) No. 4
ISSN: 2088-5334

A Development of Embedded Anomaly Behavior Packet Detection System for IoT Environment using Machine Learning Techniques

Youngchan Lim[a,1], Gicheol Choi[a,2], Kwangjae Lee[a,3]

[a]*Dept. of Information Security Engineering, Sangmyung University, Dongnam-gu, Cheonan, 31066, Korea*
E-mail: [1]youngchanlim@naver.com; [2]chlrlcjf123@naver.com; [3]begleam@smu.ac.kr

***Abstract*— Despite the growth of IoT technology and related markets, aspect of the IoT security in the IoT field is not handled correctly due to several factors such as indiscreet participation in the market, poor optimization for the various specifications. In this paper, an embedded anomaly packet detection system using machine learning technology for an IoT environment is proposed and evaluated. The suggesting system is composed of two main devices—the packet collection device and the packet analysis device. The packet collection device collects network packets from the IoT devices that are connected to the system. The packet analysis device detects anomalies from the packet data by using the machine learning model. Detected anomalies, which are mostly considered as intrusions such as new or bypassing HTTP attacks as well as existing attacks, are responded in real-time. For conformity assessment in a real-time environment, TPR, FPR, accuracy, and detection speed was measured, and the measured values of the target embedded board are 100%, 0.56%, 99.5, and 2.4 to 13.4 seconds, respectively. The results of TPR, FPR, and accuracy indicate the model itself has an excellent ability to discriminate between anomalies, but it is challenging to apply it to an embedded system in terms of detection speed. Future studies need to apply anomaly detection models that are more suitable for embedded devices and unique hardware accelerators for computing artificial neural networks.**

***Keywords*— anomaly detection; HTTP request; behavior-based; embedded IoT security system.**

I. INTRODUCTION

The rapid expansion of the internet of things (IoT) field leads to the rapid growth of the market, which occurred massive produce of IoT products [1]. With the increase of production in the market, the importance of security of IoT products grew together. However, a comparison with the growth rate of IoT production, the security of the IoT devices is way behind. Several issues have been pointed out that are causing IoT security problems. First, the IoT device has its security recommendations, but some manufacturers are trying to participate in the market without security considerations [2], [3]. Most produced products in this circumstance are vulnerable to multiple cyberattacks. Secondly, IoT devices have various kinds of specifications, which occur differently by their purpose. For example, the difference between an IoT sensor and a smartphone that has a significant gap. Since the devices have various specifications, the application of security should differ by the device [4]-[10]. To cover the security vulnerabilities in IoT devices, it is necessary to continuously application of software updates, firmware updates, and security patches from the manufacture. However, existing solutions are mostly outdated or unsuitable. Even if there is a suited solution for the device, the person who uses the device may have difficulties to apply the security system alone that it requires certain knowledge about the device. Therefore, a security system that functions regardless of the IoT device's specification is required to solve the security vulnerability problem of the IoT device. Also, which has an easiness to apply for the average person. Conventional security systems for IoT devices followed the form of an intrusion detection system (IDS) for existing network security [11]-[12]. These security systems solved the various specification problem and the applying problem partially. However, most of the existing system offers rule-based or signature-based attack detection that has difficulty in countering the new attack techniques that frequently occur [13]-[15].

Therefore, this paper proposes an embedded anomaly packet detection system for IoT environments using machine learning techniques. By applying the machine learning technique and applying it to an embedded frame, the proposed system satisfies the demand, which was not treated in past systems. The embedded anomaly detection model trains continuously with a dataset that is produced from the applied network environment. This process learns network packets in a benign status, which is the majority of the network. The anomaly detection model is adapted to the usual network and detects an anomaly, which is considered as malicious events. Additionally, since the system has a

form of embedded system, compared with the existing security system, the proposed system has advantages on the application process that provides easiness for people who have difficulty or non-experts.

II. Materials and Method

A. Anomaly Detection

The anomaly detection is a technique for identifying unexpected items or events in data sets, which differ from the norm. In network security, the technique is used to identify unexpected events such as malicious actions that have a difference from the benign. It has been extensively applied to a network intrusion detection system to recognize malicious traffic [16]-[18]. The technique is used to apply in two main methods of data analysis, which are machine learning and statistics. Ourmon is open-source network management and anomaly detection system. The system statistically analyzes the traffic flow of network protocols, including internet control message protocol (ICMP), internet protocol (IP), transmission control protocol (TCP) and user datagram protocol (UDP). The system's analyzing process detects network attacks such as synchronize (SYN) flood and Bot-net traffic attacks [19]. The machine learning method in network security, the anomaly detection technique is frequently used to analyze two types of datasets, network traffic, and network packet. McPAD used a payload of the network packet as a dataset for anomaly detection. The system was designed to detect malicious shell-code and hypertext transfer protocol (HTTP) attacks, which are contained in the network packet payload. McPAD is evaluated with the DARPA 1999 dataset and the real HTTP traffic dataset they collected [20].

B. IoT Security

The network traffics of the IoT showed different behavior compared to other existing types of network traffics. Notably, various hardware specifications and applications caused the demand for various security systems, which corresponds to each specification. According to the circumstances, several papers applied the type identification method. One research proposes a method to detect suspicious IoT devices connected to a network automatically. In this research, random forest, a supervised machine learning algorithm, was applied to features extracted from network traffic data to identify IoT device types [21] accurately. IoT SENTINEL is distinguished using device-type fingerprint, which is generated from IoT device traffic behavior in the setup process. The fingerprint was mapped by random forest as a classifier and Damerau-Levenshtein distance for comparison [22].

As another method, after the device identification method, research on an anomaly detection method of a specific attack has been proposed. DÏoT is a detection system that detects compromised IoT devices by autonomous self-learning. The system classifies connected IoT devices in device types by modeling the network packet sequence of the IoT device using a K-nearest neighbor (KNN) algorithm. The modeling procedure proceeds when the new connection occurs, which builds a model from corresponding benign communication for subsequent anomalous behavior detection using gated recurrent units (GRUs) technique. DÏoT used 33 IoT devices for evaluation and used Mirai malware in four different stages as pre-infection, infection, scanning, and denial of service (DoS) attacks. The system obtained 94% of true positive rate and 0% of false positives and rated 2.26 seconds for average detection time [23]. DDoS detection framework using machine learning for the IoT device was proposed. The proposed framework presents a four-step process to detect an anomaly; traffic capture, packet grouping by device and time, generation of feature vectors of network packets, binary classification. The binary classification used several classifiers such as KNN, random forests, decision trees, support vector machines, and deep neural networks. The suggested framework detected DoS attacks with higher than 99.9% accuracy [24].

C. Background

Since the IoT device aims flexibility of connection to an existing system, the manufacturing companies try to develop the product that could be connected easily. Thus, most of the IoT devices and controlling devices for IoT sensors use Wi-Fi for communicating with the internet. Especially, malicious actions coming through HTTP have been a significant issue in network security for a long time. As the produced IoT devices use the existing network environment, also the security vulnerability inherits. Therefore, the experiment applied three kinds of HTTP based attacks, which was a problem in the past network and remained a problem in the present.

1) SQL Injection: A structured query language (SQL) injection is one of the attack methods that mostly aim data-driven applications in which process flow is governed by data. The attack proceeds by inserting malicious SQL statements into an entry field for the execution of the application. The SQL injection exploits the security vulnerability of the application's software, such as embedding incorrectly filtered string escape characters in SQL statements. Most SQL databases are attackable [25].

2) Cross-Site Scripting: A cross-site scripting (XSS) is an attack technique in which an unauthorized user inserts a script into a website. XSS attacks are also included in the open web application security project (OWASP) Top 10. Most of the case, it happens in bulletin boards where users can write and read, but it also happens in places where the user's input values are displayed on webpages. To redirect to the command and control (C&C) server, a malicious user injects a redirection script and uses it as an intermediate waypoint or takes away the user's cookies and carries out session hijacking attacks [26].

3) Path Traversal / Directory Traversal: A path traversal attack or also defined as a directory traversal attack, allows unauthorized users to access data stored outside the web-root folder. By manipulating variables that reference files with " ../ " sequences and its variations or by using absolute file paths. The directories that are not allowed in public have a high possibility to contain crucial information such as the structure of the corresponding server or data of the database. In addition, it is possible to access system configuring files and directories, including application source code, which could occur serious problems.

4) XML External Entity Injection: An extensible mark-up language (XML) external entity injection is a type of attack against an application that parses XML input by using the vulnerability of the server settings of an external entity. This technique attacks when XML input containing a reference to an external entity is processed by a weakly constructed XML parser. This attack could lead to serious problems such as the disclosure of confidential data, denial of service, server-side request forgery, scanning of the internal system, port scan.

D. System Structure

This paper aimed to develop a comprehensive embedded IoT network security system that detects anomaly based on packets and responds to intrusions of newly occurring and bypassing HTTP attacks as well as existing attacks. Fig. 1 shows the overall flow chart of the proposed system. The system consists of two primary embedded devices: the packet collection device and the packet analysis device. The packet collection device collects and copies all generated packet from the network and transmits to the packet analysis device. Then, the packet collection device receives the result of anomaly detection from the packet analysis device. Based on the received result, the device blocks the connection of the IoT device, which was determined as an anomaly by using its firewall.

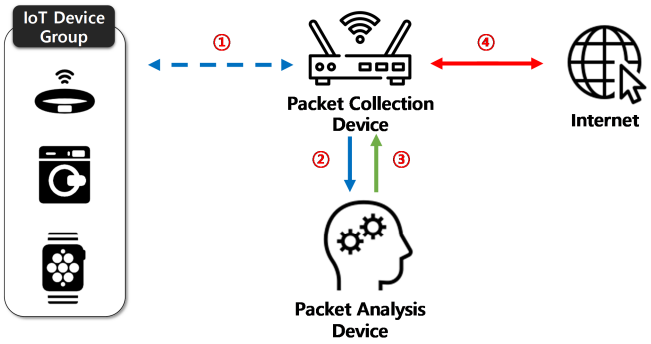


Fig. 1 An overall flow chart of the proposed security system. 1) A packet collection device collects packets from connected IoT devices. 2) The packet collection device transmits the collected packets to the packet analysis device. 3) A packet analysis device detects anomaly from received data and transmits the result to the packet collection device. 4) The packet collection device manages the connection with the internet and the IoT devices.

1) Packet Collection Device: A packet collection device proceeds two main functions. First, in the security system, this device functions as an access point (AP) and a router of the wireless local area network (Wireless LAN) that includes all IoT devices. During the collection, the collection program filters HTTP packets from collected network packets. Second, the packet collection device manages the connection of linked IoT devices, which could block the link of IoT devices that is determined as an anomaly. The block procedure is processed by the firewall of the packet collection device itself. Fig. 2 shows the entire flow chart of the packet collection device. The hardware specification of the packet collection device with these functions does not require high performance. Even the performance of an embedded system is enough. In the past experiments, the Raspberry Pi 3 B 1.2v showed enough performance [27]. Thus, we defined the hardware performance used in our experiment as a few conditions with relatively low performance: Ethernet faster than 100Mbps, wireless LAN function, and CPU that least outperforms 1.2GHz.

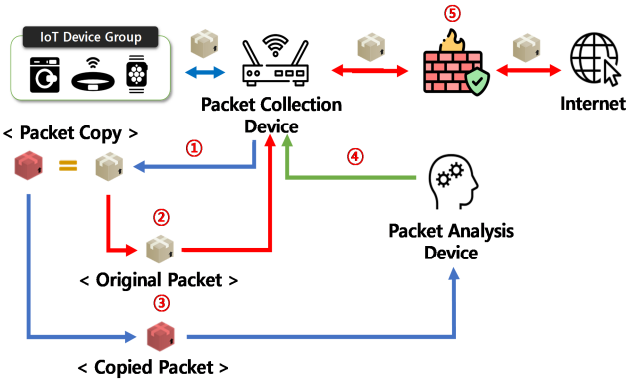


Fig. 2 A flow chart of the packet collection device. 1) The device functions as an Access Point (AP) and collects every generated packet from the network. 2) Collected packets are copied and send the original packets to the designated destination. 3) The collected packets are transmitted to the packet analysis device. 4) The packet collection device receives the detection results from the packet analysis device. 5) Based on the result, the collection device uses its firewall to block the connection of the IoT device that is found as an anomaly.

2) Packet Analysis Device: A packet analysis device contains three main functions, as Fig. 3 shows in the flow chart. First, after receiving the packet data from the packet collection device, a pre-processing function executes to transform the data to input to the anomaly detection model. Second, the anomaly detection model determines the anomaly status of the packet.

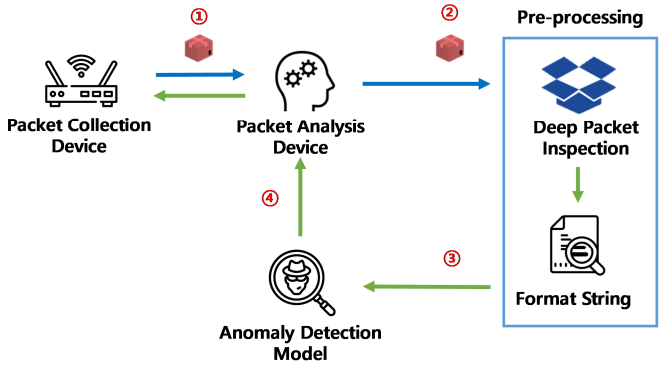


Fig. 3 A flow chart of the packet analysis device. 1) When the packet analysis device receives the copied network packets from the packet collection device. 2) Received network packets go throughs a pre-processing that has Deep Packet Inspection and Formatting string data procedures. 3) Input the pre-processed data to the anomaly detection model. 4) Detection results are transmitted to the packet collection device.

Fig. 4 shows the process of pre-processing two procedures: deep packet inspection (DPI) and the formatting of the extracted string data. DPI is a type of data process that inspects in detail includes the header and the payload of the packet, and it outputs string data form from the raw hex packet data. The DPI usually applied to IDS or Network IDS (NIDS) for analyzing the network packets from network traffic for security. A DPI procedure in the system followed a partial function of the entire existing process that extracts only HTTP packet payload. After extraction, the string data

is formatted in a specific string format. The string format is optimized for the anomaly detection model.

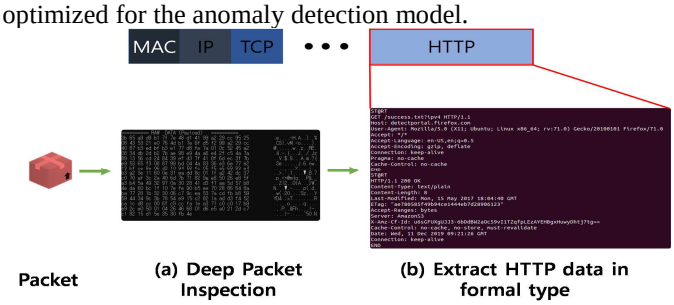


Fig. 4 A Pre-processing procedure. (a) A Deep packet inspection procedure that extracts string form data from raw hex packet data. (b) Formatting the extracted string data to specific form for the anomaly detection model.

Fig. 5 shows the structure of the anomaly detection model that is applied to the proposed system. The anomaly detection model is based on Seq2Seq Autoencoder. This model aims to train end-to-end sequences of input HTTP packet payload data. The encoding procedure is a multi-layer Long Short-Term Memory (LSTM) that maps the input sequence of HTTP packet payload data to a fixed-dimensional vector. The decoding procedure is also a multi-layer LSTM that decodes the encoder vector into the sequence input in the encoding procedure [28]. Therefore, the anomaly detection model sets its target values equal to its input value. The system trains the model with network packets, which is occurred at the applying network, and thus, the model learns to re-create network packets that have seen. The trained model could detect an anomalous by re-creating the input packet payload data with a high degree of error, which never trained previously.

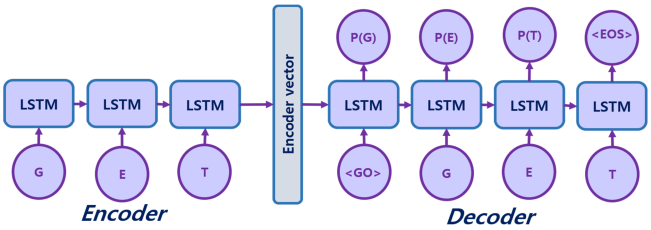


Fig. 5 A structure of an anomaly detection model is based on Seq2Seq Autoencoder. Each encoding procedure and the decoding procedure has a form of multi-layered Long Short-Term Memory (LSTM).

III. Results and Discussion

A. Experimental Environment

The experiments in this paper are designed to assess the suitability of a real-time environment comprehensively. The subject of the experiment is a typical computer environment and a testbed. Each experimental subject has applied the anomaly detection model that is trained in the same dataset, and each experimental subject receives the same test dataset. Every condition is equal except for the hardware specification. After each subject receives the test dataset, the anomaly detection model was proceed. The dataset for the experiment is from the positive research journal, which also has been introduced by the DEFCON AI village article [29]. Table 1 shows the specification of the experiment computer that functions the applied anomaly detection model.

TABLE I
Specification of an Experiment Computer

System	Specification
CPU	AMD CPU Ryzen 7 2700 @ 3.2 GHz - 8 Core / 16 Thread - 7-zip MIPS: 66,483 - RAM: DDR4 8GB
GPU	NVIDIA GeForce RTX 2060 - NVIDIA Cuda Cores: 1920 - Built-in memory: 6GB GDDR6 - Memory speed: 14Gbps
OS	Windows 10 Pro (64bits)

Table 2 shows the specification of the target embedded board, the NVIDIA Tegra X2, that functions as the packet analysis device of the system.

TABLE II
Specification of a Target Embedded Board

System	Specification
CPU	HMP Dual Denver 2/2 MB L2 + Quad ARM A57/2 MB L2 - RAM: LPDDR4 8GB - Memory speed: 58.3 GB/s
GPU	NVIDIA Pascal - NVIDIA Cuda Cores: 256
OS	Ubuntu 16.04 LTS

Table 3 shows the amount of the test dataset. As the paper mentioned earlier, the dataset divided into nine to one ratio. The dataset consists of 21991 benign and 1097 anomalous HTTP requests from a banking application. The dataset is divided into nine to one ratio, and each portion is used to train and test the anomaly detection model.

TABLE III
Dataset

Dataset	Purpose	Amount
Benign	Train	19791
	Test	2200
Anomaly	Test	1097

Fig. 6 shows the flow chart of the experiment. The test dataset, the packet data are inputted to the experiment subjects. After the experimental computer and the Tegra X2 which refers the packet analysis device receives the dataset, each subject process applied anomaly detection model.

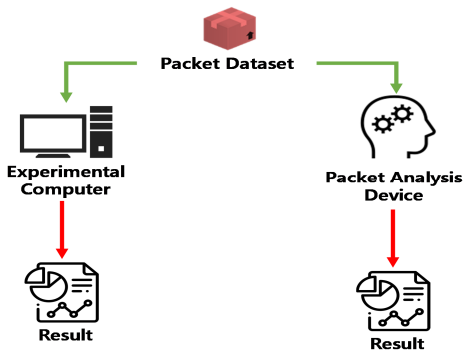


Fig. 6 A flow chart of a proposed experiment

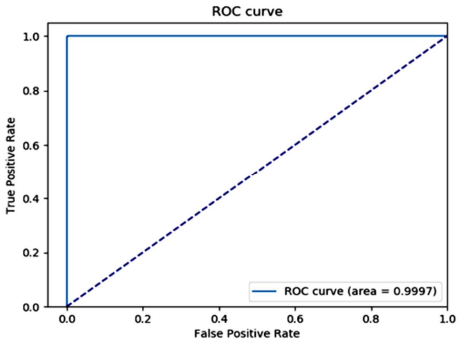
B. *Experimental Results*

During the experiment, several values were labeled for evaluation of suitability in a real-time environment. Table 4 provides the labeled measurements of each experimental subject, such as the True Positive Rate (TPR), the False Positive Rate (FPR), accuracy, and the detection speed. The TPR is a percentage of an actual anomaly packet from packets that are correctly detected as an anomaly. The FPR is a percentage of a benign packet from packets that are detected as an anomaly. Accuracy is a rate of detected anomaly packet from an actual anomaly. The detection speed is a consumed time for one packet to get determined as a benign or anomaly. In the experiment, the true positive rate and the false positive rate shows the correctness prediction ratio of detection results of the anomaly and the benign. The accuracy measured 99.6% and 99.5% for each experimental computer and the testbed. The following three results represent the model itself has a high performance of ability to distinguish the anomaly.

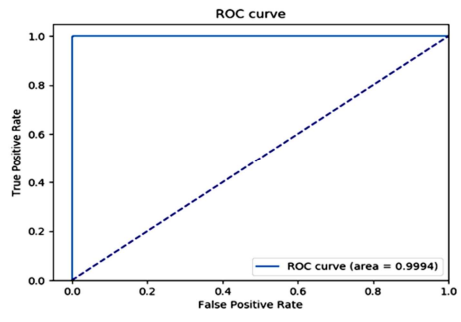
TABLE IV
EXPERIMENT MEASUREMENT

	Experimental Computer	**Target Embedded Board (NVIDIA Tegra X2)**
True Positive Rate (%)	100	100
False Positive Rate (%)	0.55	0.56
Accuracy (%)	99.6	99.5
Detection speed (sec)	0.1 ~ 0.2	2.4 ~ 13.4

Fig.7 presents the receiver operating characteristic (ROC) curve of the general computer and the Tegra X2. The ROC curve plots TPR against FPR of the anomaly detection model.



(a)



(b)

Fig. 7 An ROC curve. (a) A curve of an experiment computer case. (b) A curve of a target embedded board.

The plotting illustrates the cumulative distribution function of the anomaly detection probability in the y-axis versus the cumulative distribution function of the false positive probability on the x-axis. However, since this proposed system aimed at an anomaly packet detection for IoT embedded environments, the system requires a certain performance for the real-time process. The detection speed of each experiment objects in Table 4 shows negative results. Depending on the type of the application network or network device such as IoT devices, minimum network speed is required to perform its functions. Recently, 100Mbps came very commonly, and this rate is the minimum speed required to perform the device. The max size of the packet is 1,538 bytes, and the 100Mbps network handles 12,500,000 bytes per second, which roughly 8,000 packets could communicate. The results of Table 4 show the anomaly detection model on the experimental computer could function normally, but the embedded device shows the negative result, which could not function the linked device to the embedded system. Therefore, to improve the proposed system, it is necessary to apply anomaly detection models that are more suitable for embedded devices, and unique hardware accelerators for artificial neural network computation are required.

IV. CONCLUSION

In this paper, we suggested an embedded anomaly packet detection system for IoT environments using machine learning techniques. The development of the system has aimed at a comprehensive embedded IoT network security system that detects anomaly based on packets and responds to intrusions of newly occurring and bypassing HTTP attacks as well as existing attacks. The processed experiment was designed to evaluate the suitability in a real-time environment and two objects, the general computer and the embedded test board, were selected and applied the anomaly detection machine-learning model. The result showed unqualified performance at the embedded board. Since the anomaly detection is processed by each packet, the system should show a certain process speed of each packet. However, the performance of the embedded board represented that the processing speed was unsatisfying to apply on an existing network. Therefore, for further enhancement of the suggested system, first, application of more suitable anomaly detection model for embedded device is required. Second, specialized hardware accelerator for artificial neural networks is required.

ACKNOWLEDGEMENT

This research was funded by a 2020 research Grant from Sangmyung University, South Korea.

REFERENCES

[1] Columbus, Louis. (2018) IoT market predicted to double by 2021, reaching $520b. [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2018/08/16/iot-market-predicted-to-double-by-2021-reaching-520b

[2] Khan, M. A. and Salah, K., "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Comput. Syst.*, vol. 82, 2018, pp. 395-411.

[3] Sharma, Pradip Kumar, and Jong Hyuk Park, "Blockchain based hybrid network architecture for the smart city," *Future Generation Comput. Syst.*, vol. 86, pp. 650-655, 2018.

[4] Hadar, N., Siboni, S., and Elovici, Y, "A Lightweight Vulnerability Mitigation Framework for IoT Devices," in *Proc. 2017 Workshop on Internet of Things Secur. Privacy*, 2017, pp. 71-75.

[5] Ammar, Mahmoud, Giovanni Russello, and Bruno Crispo, "Internet of Things: A survey on the security of IoT frameworks," *J. Inf. Secur. Appl.*, vol. 38, pp. 8-27, 2018.

[6] T. W. Tseng, C. T. Wu, and F. Lai, "Threat Analysis for Wearable Health Devices and Environment Monitoring Internet of Things Integration System," *IEEE Access*, vol. 7, pp. 144983-144994, 2019.

[7] T. A. Ahanger and A. Aljumah, "Internet of Things: A Comprehensive Study of Security Issues and Defense Mechanisms," *IEEE Access*, vol. 7, pp. 11020-11028, 2019.

[8] Miloslavskaya, N. and Tolstoy, A., "Internet of Things: information security challenges and solutions," *Cluster Comput.*, vol. 22, no. 1, pp. 103–119, 2019.

[9] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating Critical Security Issues of the IoT World: Present and Future Challenges," *IEEE Internet of Things J.*, vol. 5, no. 4, pp. 2483-2495, Aug. 2018.

[10] Poonia A.S., Banerjee C., Banerjee A., and Sharma S.K, "Security Issues in Internet of Things (IoT)-Enabled Systems: Problem and Prospects," *Soft Comput.: Theories Appl.*, vol. 1053, pp.1419-1423, 2020.

[11] Raza, Shahid, Linus Wallgren, and Thiemo Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad hoc netw.*, vol. 11, no. 8, pp. 2661-2674, 2013.

[12] Adat, Vipindev, and B. B. Gupta, "Security in Internet of Things: issues, challenges, taxonomy, and architecture," *Telecommun. Syst.*, vol. 67, no.3, pp. 423-441, 2018.

[13] Amouri, A., Alaparthy, V. T., and Morgera, S. D., "Cross layer-based intrusion detection based on network behavior for IoT," in *WAMICON'18*, 2018, pp. 1-4.

[14] Amouri, Amar, Vishwa T. Alaparthy, and Salvatore D. Morgera. "A Machine Learning Based Intrusion Detection System for Mobile Internet of Things," *Sensors*, vol. 20, no.2, pp. 1-15, 2020.

[15] M. Ramadan, Y. Liao, F. Li, and S. Zhou, "Identity-Based Signature with Server-Aided Verification Scheme for 5G Mobile Systems," IEEE *Access*, vol. 8, pp. 51810-51820, 2020.

[16] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Commun. Surveys Tut.*, vol. 16, no. 1, pp. 303-336, 2013.

[17] Hamamoto, Anderson Hiroshi, *et al*., "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Syst. Appl.*, vol. 92, pp. 390-402, 2018.

[18] Zhang, Daokun, *et al*., "Network representation learning: A survey," IEEE *Trans. Big Data*, vol. 6, no. 1, pp. 3-28, 2020.

[19] J. R. Binkley and B. Massey, "Ourmon and Network Monitoring Performance," in *USENIX'05 Ann. Technical Conf.*, 2005, pp. 95-108.

[20] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, and W. Lee, "McPAD: A multiple classifier system for accurate payload-based anomaly detection," *J. Comput. Netw.*, vol. 53, no. 6, pp. 864-881, 2009.

[21] Meidan, Y., Bohadana, M., Shabtai, A., Ochoa, M., Tippenhauer, N. O., Guarnizo, J. D., and Elovici, Y., "Detection of Unauthorized IoT Devices Using Machine Learning Techniques," *arXiv:1709.04647 [cs.CR]*, Sep. 2017.

[22] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, and S. Tarkoma, "IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT," in *ICDCS'17*, 2017, pp. 2177-2184.

[23] T. D. Nguyen, S. Marchal, M. Miettinen, N. Asokan, and A.-R. Sadeghi, "DÏoT: A Federated Self-learning Anomaly Detection System for IoT," in *ICDCS'19*, 2019, pp. 756-767.

[24] Doshi, R., Apthorpe, N., and Feamster, N., "Machine Learning DDoS Detection for Consumer Internet of Things Devices," in *SPW'18*, 2018, pp. 29-35.

[25] Microsoft. (2012) SQL Injection. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms161953(v=sql.105)

[26] Symantec, "Symantec Internet Security Threat Report: Trends for July–December 2007 (Executive Summary)," *Symantec Corp.*, vol. 13, Apr. 2008.

[27] G. Choi, Y. Lim, and K. Lee, "A Development of Anomaly Behavior Detection System for IoT Environment using Machine Learning," in *ICICPE'19*, Dec. 2019, pp. 63-65.

[28] Chawla, A., Jacob, P., Lee, B., and Fallon, S., "Bidirectional LSTM Autoencoder for Sequence based Anomaly Detection in Cyber Security," *Int. J. Simul. Syst., Sci. & Technol.*, vol. 20, no. 5, pp. 7.1-7.6, 2019.

[29] Alexandra Murzina, Irina Stepanyuk, Fedor Sakharov, and Arseny Reutov. (2019) Detecting web attacks with a Seq2Seq autoencoder. [Online]. Available: http://blog.ptsecurity.com/2019/02/detecting-web-attacks-with-seq2seq.html