Machine Learning and Deep Learning Methods for Better Anomaly Detection in IoT-23 Dataset Cybersecurity

Yue Liang[1] and Nikhil Vankayalapati[2]

[1]Department of Computer Science, Lakehead University, ON, Canada
[2]Department of Computer Science, Lakehead University, ON, Canada

***Abstract*—As smart devices and the Internet develop, the Internet of Things (IoT) technologies have become an important factor in our life. IoT helps manufactory companies to monitor the status of every machine in real time, the quality of products and the environment variables within the factory. This not only allows managers to reduce the risk of damages and losses, also help to make decision from a higher overall standpoint. In addition, IoT has changed people's life and behavior. People are now relied on IoT devices and services more than ever. However, anomalies can caused security and safety issues for an IoT network. It is important to detect anomalies and alarm user to prevent damages or losses. In this paper, we proposed using the Machine Learning and Deep Learning methods to detect anomalies in a network. The experiments were performed on the IoT-23 dataset. The performance and time cost for these models are compared to give us the best algorithm with high performance in less time.**

***Index Terms*—Internet of Things, security, malicious node, anomaly detection, Machine Learning, Deep Learning.**

I. Introduction

Internet of Things (IoT) is a revolution to the global information industry after the Internet. The IoT is a smart network that allows devices to exchange information and communicate with each other through internet. With IoT, human can achieve the purpose of tracking, monitoring, locating, identifying and managing things [1]. Since the revolution of the Internet and mobile devices, IoT has become an evolving and hot research topic within the computer science industry. The number of IoT devices on the Internet is increasing every year and in every sector such as: Smart Healthcare, Smart Transportation, Smart Governance, Smart Agriculture, Smart Grid, Smart Home, Smart Supply chain etc. [2].

Because of the convenience brought by IoT, the behavior of humans has also changed. People of younger generations are more used to use services from IoT devices such as smart bulbs, smart oven, smart refrigerator,AC,temperature sensor, smoke detector etc. [3] However, as IoT develops, the concerns of the privacy and security issues has increased among the users. As all the devices are connected to the internet and each other, this leads to more number of ways for the attacker to access the information possible. The connected devices collect data with personal information and stores it. Most of the users do not have knowledge about IoT technology, and the hackers can steal information from the users or even control the smart devices of the users.[4] This not only reduces the advancement of IoT technology but also slows down the development of IoT infrastructure. Therefore, providing security and privacy of these constantly and heavily connected devices has became a major challenge. Another key issue for providing security and privacy to these devices is the managing the huge amount of data generated by them, which is quite difficult using general data collection, storage and processing techniques[18].

With the development of Machine Learning (ML) and Deep Learning (DL), learning algorithms can learn from the results of trained data and adapt in order to increase the performance to make informed and intelligent decisions. A learning algorithm that has been trained by the data is able to establish the difference between regular benign traffic in the model with the malicious traffic. In other words, it can detect when there is an abnormal behaviour in the network thereby preventing unauthorized access. Learning algorithms are basically classified into two categories which are Supervised Learning and Unsupervised Learning. We try to use the light weighted machine learning methods and neural networks for accuracy improvement on detecting malicious node. The Central unit in the model captures IoT traffic data and sends the data to a selected trained Machine Learning or Deep Learning model. Multiple trained Machine Learning and Deep Learning models are tested. The reason for choosing multiple models is to fit the individual needs for different users or groups. In other words, it is important to find the efficient model for different type of user.

This large data in the IoT network and the heterogeneity of the data makes it to difficult to improve the security and to meet all the requirements such as cost effectiveness, reliability, performance etc. In some cases, if one of the feature is improved then it may effect performance of other features[16]. For example, an increase in the number of security checks and protocols in all data transfer then it may result in the increase in cost and latency of that particular application making it unsuitable for certain users. Also the increase in number of devices connected increases the chance for attacker to gain access the network by accessing the node or device that has a weak link for example a device like smart bulb. Most of the devices that are available in the market as of now do not have the security features like firewalls, anti-virus etc. As the IoT devices are resource constrained it is important for these

devices to detect an intrusion with less complexity and time. So, the use of Machine learning(ML) and Deep Learning(DL) techniques helps to reduce this complexity as these models learn from the trained data. It is important for the central unit to classify the message's integrity. The privacy and security issues of IoT motivates researches for developing framework of automatic IoT sensors attack and anomaly detection[14]. In this paper, We proposed to use ML/DL algorithms such as Support Vector Machines, Decision Trees, Naive Bayes and Convolutional Neural Networks for anomaly detection and based on their accuracy and time cost the better algorithm to use can be concluded. And we used the IoT-23 dataset for the implementation of ML/DL methods. The paper goes as follows, in Section II literature review is discussed, in Section III methodology is explained, in Section IV results are discussed with evaluation metrics and comparison. In sections V we concluded the paper with a few suggestion of future work. At last, references for this study is included.

II. Literature Review

In this section, all the different anomaly detection algorithms and methodologies are briefly discussed. There are a number of different mechanisms to improve the safety and privacy of IoT devices. For example, in [12], chaos based encryption technique is used to generate symmetric keys to provide secured data transmission between server and the IoT device which guarantees the data integrity and authenticity. According to [13], a mechanism with low computational complexity has been proposed by using, random hopping sequence and random permutations to hide valuable information. Moreover, in [14], Doshi presented a method to detect DDoS attacks in the network layer with low-cost machine learning approach, including KNN, LSVM, NN, Decision Tree, and Random Forest. This method can detect which node is attacking the central unit with IP address. This method was reported to achieve high testing accuracy for all five machine learning algorithms. In [21] detection of anomaly is done using the fog computing, which clusters the different types of anomalies present in the sensor layer or edge nodes without performing computation on both the cloud and sensor layer but in the fog layer of the network. By using the fog computing method it has become more easy to detect an anomaly. In [17], the author tries to implement malware detection system by using different classifiers of k-NN and random forest to build the model. The device filters TCP packets and selects important features such as frame numbers, length, labels etc. The k-NN algorithm assigns traffic to the class while the random forest classifier builds decision trees to detect the malware. The authors have proposed a new methodology in [22] which uses game theory and nash equilibrium to help the resource constrained IoT devices to detect an anomaly using Intrusion Detection System(IDS), activating it only when needed. When an attack occurs the attack pattern (signature) is stored and then model is trained and whenever pattern repeats it is identified by the signature detection technique and anomaly is detected. Using IDS all the time can be resource consuming, so the game theory and nash equilibrium come into place to determine when to activate the IDS to detect an anomaly and to add a new rule to signature pattern and build the model. Machine learning or Deep Learning methods have been discussed in [16]. The various types of attacks at different levels of IoT infrastructure are clearly explained and the possible solutions to these attacks using Machine learning are also clearly explained, that which are caused due to the lack of proper security data available, the low quality data available and performance of the learning algorithms could be the key in providing and improving the security and privacy of IoT devices. In this paper we would like to calculate the accuracy and time cost for the models, thereby comparing them to get the model that gives highest accuracy with less amount of time to detect and prevent the malware attacks in resource constrained IoT devices.

III. Methodology

A. Proposed Model

This study proposes an anomaly detection system model for IoT security. Fig.1 is the diagram of the proposed anomaly detection system model. In our proposed model, a traffic capture unit captures traffic flow from sensors to the central unit. The captured traffic flow will be send to a compute unit, which can be a cloud or local computer. Then the compute unit will run multiple Machine Learning (ML) and Deep Learning (DL) models in order to get the performance and cost of each individual model. Also, the compute unit will store the traffic flow to its database for future studies or model re-calibration. After getting the performance and cost of the ML/DL models, the user or system will select the model that is going to be used for anomaly detection. When detecting anomalies, the compute unit will send message or commands back to the central unit such as dropping packets, malware scan, physical inspection, marking IP address and alarming user. With our proposed model, users can choose the ML/DL model based on the performance and cost, such as accuracy and time cost. Since every user has different situation and usage of a anomaly detection system for IoT security, it is important to offer the best fit for different users. Moreover, since our proposed model captures traffic flow and store them into the database, the new dataset can be generated and be used for future re-calibration in the existing ML/DL models to further improve performance.

Machine Learning algorithms such as Support Vector Machine(SVM), Random forest, Naive Bayes, Nearest Neighbours etc. and Deep learning methods such as Convolutional Neural Networks(CNN) are trained with the data and then computation is done to detect the anomaly in the system which can be done on a local machine or on cloud. The dataset is divided into training and testing data and then based on the algorithm trained, conclusions can be drawn from the obtained results. If an anomaly is detected then certain possible actions can be taken based on the result such as: Dropping packets, Blacklist sender's IP address, Alarm user, Physical inspection and more. The system can then scanned to detect any malware present and also physical inspection can be done on the marked devices.

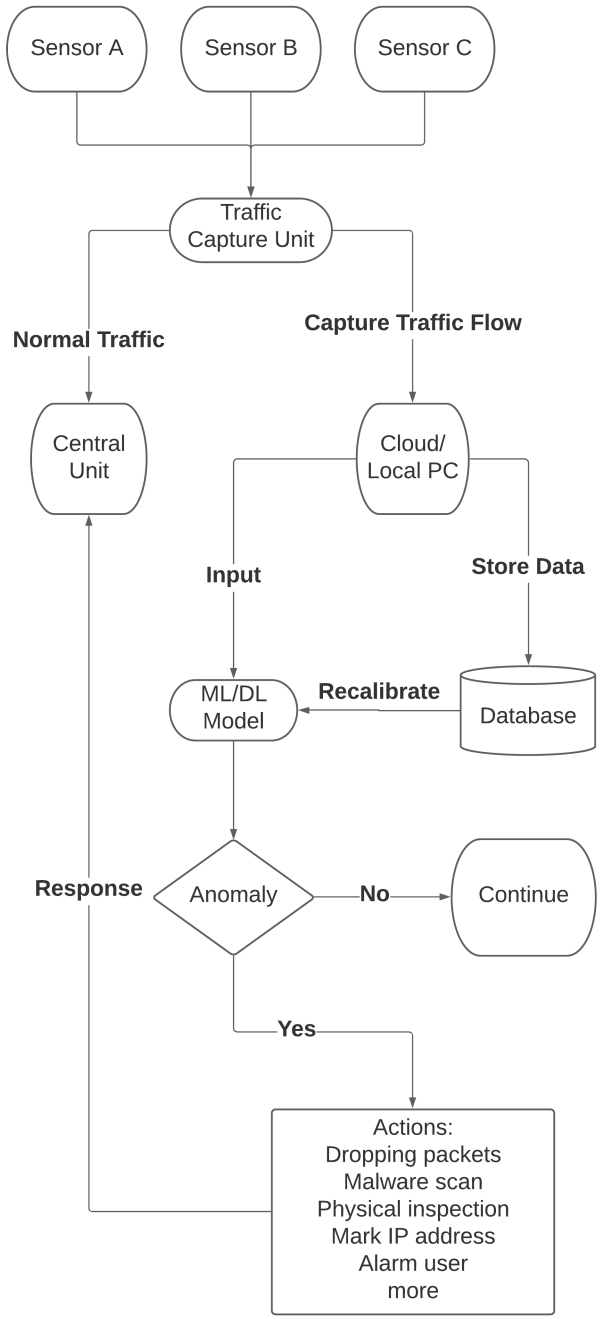

Fig. 1. The proposed anomaly detection system model

The results obtained are compared with each other in order to define the efficient method that can be used for the real time data. The factors taken into consideration are ”accuracy” and ”time cost” taken for the algorithm. For example, even if a model gives 100 percent accuracy and takes a lot of time it isn't suitable for IoT network because the devices are resource constrained. Therefore, our proposed model is to offer an optimal solution for different type of users, such as a big company with lots of resources that aiming for the highest accuracy or a small company that worries about cost efficiency.

TABLE I
VARIABLES AND DEFINITION FOR ZEEK FILES

ts	This is the time of the first packet
uid	A unique identifier of the connection
id	The connection's 4-tuple of endpoint addresses/ports
proto	The transport layer protocol of the connection
service	An identification of an application protocol
duration	How long the connection lasted
orig_bytes	The number of payload bytes the originator sent
resp_bytes	The number of payload bytes the responder sent
conn_state	Possible connection state values
local_orig	If the connection is originated locally, this will be T
local_resp	If the connection is responded locally, this will be T
missed_bytes	Indicates the number of bytes missed in content gaps
history	Records the state history of connections as a string
orig_pkts	Number of packets that the originator sent
orig_ip_bytes	Number of IP level bytes that the originator sent
resp_pkts	Number of packets that the responder sent
resp_ip_bytes	Number of IP level bytes that the responder sent
tunnel_parents	uid values for any encapsulating parent connections
orig_l2_addr	Link-layer address of the originator

B. Dataset

The dataset in this study was obtained from [20], the IoT-23 dataset, which is a very recent one that was published in January 2020 consisting of network traffic from 3 different smart home IoT devices. The devices used were Amazon Echo, Philips HUE and Somfy Door Lock. It is a large dataset of real and labeled IoT malware infections and benign traffic especially made to develop Machine learning algorithms. It consists of 23 captures(also called scenarios), in the 23 captures, there are 20 malicious captures and 3 benign captures. Captures from infected devices will have the possible name of the malware sample executed on each scenario.

The malware labels for IoT-23 dataset are: Attack, C&C, C&C-FileDownload, C&C-HeartBeat, C&C-HeartBeat-Attack, C&C-HeartBeat-FileDownload, C&C-Mirai, C&C-Torii, DDoS, FileDownload, Okiru, Okiru-Attack, PartOfA-HorizontalPortScan.

In addition, Zeek is a software that perform network analysing. The IoT-23 dataset we used is in the format of conn.log.labeled, which is the Zeek conn.log file that was generated from the Zeek network analyser using the original pcap file. The variable types and definition for IoT-23 dataset are as shown in Table I.

Since the dataset is huge, we have decided to capture part of records from each individual dataset, then combine them to a new dataset. By doing this, our computer can handle the workload for the new dataset, and the new dataset remains most of the attack types of IoT-23 dataset.

C. Data Preprocessing

First, we used the Python library Pandas to load all 23 datasets separately of the IoT-23 Dataset into data frames with a condition of skipping the first 10 rows and reading the one hundred thousand rows after. Then we combined all 23 data frames into a new data frame. Next, we dropped the variables that have no impact to the results. These variables are: ts, uid, id.orig_h, id.orig_p, id.resp_h, id.resp_p, service, local_orig, local_resp, history. Furthermore, we gave dummy values to the proto and conn_state variables and replaced all the missing

TABLE II
COUNTS OF ATTACK TYPES FOR FILE IOT23_COMBINED.CSV

Label	count
PartOfAHorizontalPortScan	825939
Okiru	262690
Benign	197809
DDoS	138777
Attack	3915
C&C-HeartBeat	349
C&C-FileDownload	43
C&C-Torii	30
FileDownload	13
C&C-HeartBeat-FileDownload	8
C&C-Mirai	1

values with 0. Last, the combined dataset is generated and saved as the iot23_combined.csv file.

The iot23_combined.csv file contains a total of 1,444,674 records. Moreover, as shown in Table II, the combined file has 10 types of attack, including PartOfAHorizontalPortScan, Okiru, DDoS, Attack, C&C-HeartBeat, C&C-FileDownload, C&C-Torii, FileDownload, C&C-HeartBeat-FileDownload, and C&C-Mirai.

For validation, we splited the combined dataset into a training dataset with a size of 0.8 and a testing dataset with a size of 0.2.

IV. PERFORMANCE EVALUATION AND ANALYSIS

The results of the algorithms are discussed in this section. It includes the confusion matrix of each algorithm along with the time it has taken to calculate the anomaly.

A. Hardware and Environment Settings

The experiments were run on a personal computer with an Intel Core 7700k CPU @ 4.50 GHz, 24 GB of RAM @ 3200 MHz, and MSI GeForce RTX 2080. In addition, the experiments were performed on Windows 10, Anaconda Jupyter Notebook, Python 3.8 and Tensorflow 2.4 environments.

B. Evaluation of Metrics

To evaluate the results of the model certain metrics are used which are described below.

1) Time: The amount of time taken for an algorithm to run a particular ML/DL model is taken into consideration. As mentioned earlier, an algorithm which takes heavy amount of time may not be suitable for IoT environment.

2) True Positives: The outcome where the model correctly predicts the positive class.

3) False Positives: The outcome where the model incorrectly predicts the positive class.

4) Precision: Precision is described as a measure of calculating the correctly identified positives in a model and is given by:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

5) Recall: It is a measure of actual number of positives that are correctly identified and is given by:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

TABLE III
NAIVE BAYES RESULTS

metrics	precision	recall	f1score	support
accuracy	–	–	0.30	288935
macro avg	0.45	0.50	0.28	288935
weighted avg	0.85	0.30	0.21	288935
time cost				6 seconds

TABLE IV
SVM RESULTS

metrics	precision	recall	f1score	support
accuracy	–	–	0.69	80000
macro avg	0.33	0.26	0.25	80000
weighted avg	0.60	0.69	0.57	80000
time cost				5849 seconds

6) F1 score: Taking into account both false positives and false negatives, f1 score is a metric that calculates the harmonic mean of precision and recall and is considered to be a better measure. It is given by

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

7) Support score: The support score is a measuring metrics of the python library scikit-learn, which indicates the number of occurrences of each label where it is true.

C. Test Results for ML and DL Methods

1) Naive Bayes: The supervised learning algorithm is based on Bayes theorem and is generally used for classification problems which predicts based on the probability. It is known to be simple and effective algorithm for building ML models.

As shown in Table III, the overall accuracy for the Naive Bayes algorithm is only 30 percent and time taken to execute is 6 seconds. The Naive Bayes obtained the lowest accuracy in our results.

2) Support Vector Machine: The support vector machine (SVM) algorithm tries to find the hyperplane which is dependant on the number of features that classifies the data points. Hyperplane is a decision boundary between the data points, to classify them based on either side of the hyperplane. Using the extreme data points as support vectors the margin of classifier can be maximised leading to better classification.

As shown in Table IV, it shows that the overall accuracy for SVM is only 69 percent while explaining the precision for each attack. The time taken to execute is almost around 2 hours. The SVM obtained a similar accuracy compared to Decision Trees and CNN, but it has the highest time cost out of all the results.

TABLE V
DECISION TREES RESULTS

metrics	precision	recall	f1score	support
accuracy	–	–	0.73	722337
macro avg	0.63	0.50	0.50	722337
weighted avg	0.77	0.73	0.65	722337
time cost				3 seconds

TABLE VI
CNN MODEL SUMMARY

Layer (type)	Output Shape	Number of Parameters
Input (Dense)	(None, 2000)	50000
dense_1 (Dense)	(None, 1500)	3001500
dropout_1 (Dropout)	(None, 1500)	0
dense_2 (Dense)	(None, 800)	1200800
dropout_2 (Dropout)	(None, 800)	0
dense_3 (Dense)	(None, 400)	320400
dropout_3 (Dropout)	(None, 400)	0
dense_4 (Dense)	(None, 150)	60150
dropout_4 (Dropout)	(None, 150)	0
Output (Dense)	(None, 12)	1812
Total parameters: 4,634,662		
Trainable parameters: 4,634,662		
Non-trainable parameters: 0		

TABLE VII
CNN RESULTS

training accuracy	training loss	testing accuracy	testing loss
0.6937	0.8583	0.6935	0.8602
time cost			242 seconds

3) Decision Trees: Supervised Machine Learning classifier that is generally used for classification problems consisting of nodes and leave connected by branches. Where the nodes represents features of the dataset, leaf node represents the outcome and the branches are the decision rules of the classification.

As shown in Table V, it shows that the overall accuracy was able to achieve 73 percent while the time cost is only around 3 seconds. The Decision Trees achieved the highest accuracy and the lowest time cost in our result, which makes the Decision Trees as the best solution method in our study.

4) Convolutional Neural Networks: Convolutional neural networks (CNN) is a deep learning model with minimal pre-processing required with an architecture mimicking pattern of neurons of human brain. It has many different layers convolutional layers, pooling layers, fully connected layers and normalisation layers. The convolutional layer has several attributes named hyper parameters such as the number of input and output channels, padding size, kernels with particular width and height etc. Pooling layers reduce the dimension of data by combining output of previous neuron cluster to a single neuron in the next layer. While in convolutional layer the input of neuron is specific to certain neuron, while in the fully connected layer every neuron receives input from all the neurons in previous layer.

In our proposed CNN model, as shown in Table VI, it has 1 input layer, 4 dense layers, 4 dropout layers with 0.2 dropout rate, and 1 output layer. The activation function for dense layers is Relu, which is a linear function that will output the input directly if the result is positive or output zero if the result is not positive. The activation function for output layer is Softmax, which is a logistic function to normalize the output into a probability distribution. The optimizer for the CNN model is Adam, which is a gradient descent searching algorithm. There are a total of 4,634,662 parameters for our proposed CNN model, and all the parameters are trainable.

TABLE VIII
EXPERIMENT RESULTS

Method	Testing Accuracy	Time Cost
Naive Bayes	0.30	6 seconds
SVM	0.69	5849 seconds
Decision Tree	0.73	3 seconds
CNN	0.6935	242 seconds

TABLE IX
RESULTS COMPARISON WITH PAPER[19]

Method	Testing Accuracy
Naive Bayes (ours)	0.30
Naive Bayes (paper[19])	0.23
SVM (ours)	0.69
SVM (paper[19])	0.67

As shown in Table VII that the testing accuracy for CNN model is 69 percent and the execution time is around 4 minutes. Although CNN has lower accuracy and higher time cost than Decision Trees, CNN can have a better performance when dealing with a more complex dataset.

D. Results Comparison

The experiment results are shown in Table VIII. The obtained results for each of the algorithm are compared with each other, then the comparison will be done on the basis of accuracy and the cost of time for each algorithm to execute. For Naive Bayes, while it results in an accuracy of 0.30, other ML/DL methods result in accuracy around 0.70. For SVM, it results in an accuracy of 0.69, which is about the same compared to the CNN model and about 6% lower accuracy compared to Decision Trees. However, the time cost for SVM is about 2 hours, which is 1,950 times slower than Decision Trees and 24 times slower than the CNN model. For CNN, it results in an accuracy of 0.694, which is lower than Decision Trees and higher than SVM. The time cost for CNN is about 4 minutes, which is 80 times slower than Decision Trees. For Decision Trees, it results in an accuracy of 0.73 and cost about 3 seconds, which are the best accuracy and the lowest time cost among all the tested ML/DL algorithms in this study.

In addition, paper [19] also tested multiple Machine Learning algorithms on the IoT-23 dataset. Unlike our study, paper [19] implemented Random Forest (RF), Naive Bayes, Support Vector Machine, Artificial Neural Network (ANN) and AdaBoost. As shown in Table IX, the results of paper [19] shows that the Naive Bayes algorithm has 23 percent accuracy and the SVM algorithm has 67 percent accuracy. Compared to our study, although our results has higher accuracy with the Naive Bayes and SVM algorithms, both results show that Naive Bayes algorithm has the lowest accuracy among all algorithms. However, since the combined dataset in [19] is a much larger dataset compared to our combined dataset, this might have impacts to the results. In short, the result comparison with paper [19] shows that our result is accurate.

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented an anomaly detection system for IoT security with the performance comparison

of different learning algorithms and methods. Based on our results, Naive Bayes has the worst performance of all learning algorithms and methods, and Decision Trees has shown the highest accuracy with least cost of time among all the ML/DL methods. In the future, more datasets from different environment should be tested in the ML/DL methods used in this study. This can help to further clarify the performance, time cost and comparison between the methods.

REFERENCES

[1] S. Chen, H. Xu, D. Liu, B. Hu and H. Wang, "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective," in IEEE Internet of Things Journal, vol. 1, no. 4, pp. 349-359, Aug. 2014, doi: 10.1109/JIOT.2014.2337336.

[2] G. Shen and B. Liu, "The visions, technologies, applications and security issues of Internet of Things," 2011 International Conference on E-Business and E-Government (ICEE), Shanghai, China, 2011, pp. 1-4, doi: 10.1109/ICEBEG.2011.5881892.

[3] Huang, Y., Benford, S., Price, D., Patel, R., Li, B., Ivanov, A., and Blake, H. (2020). Using Internet of Things to Reduce Office Workers' Sedentary Behavior: Intervention Development Applying the Behavior Change Wheel and Human-Centered Design Approach. JMIR mHealth and uHealth, 8(7), e17914–. https://doi.org/10.2196/17914

[4] Almusaylim, Z., and Zaman, N. (2018). A review on smart home present state and challenges: linked to context-awareness internet of things (IoT). Wireless Networks, 25(6), 3193–3204. https://doi.org/10.1007/s11276-018-1712-5

[5] Singh, K., and Singh, N. (2020). An ensemble hyper-tuned model for IoT sensors attacks and anomaly detection. Journal of Information and Optimization Sciences, 1–25. https://doi.org/10.1080/02522667.2020.1799515

[6] Kumar, S., Vealey, T., and Srivastava, H. (2016). Security in Internet of Things: Challenges, Solutions and Future Directions. 5772–5781. https://doi.org/10.1109/HICSS.2016.714

[7] Tahsien, S., Karimipour, H., and Spachos, P. (2020). Machine learning based solutions for security of Internet of Things (IoT): A survey. Journal of Network and Computer Applications, 161, 102630–. https://doi.org/10.1016/j.jnca.2020.102630

[8] Tahsien, S., Karimipour, H., and Spachos, P. (2020). Machine learning based solutions for security of Internet of Things (IoT): A survey. Journal of Network and Computer Applications, 161, 102630–. https://doi.org/10.1016/j.jnca.2020.102630

[9] A. Mosenia and N. K. Jha, "A Comprehensive Study of Security of Internet-of-Things," in IEEE Transactions on Emerging Topics in Computing, vol. 5, no. 4, pp. 586-602, 1 Oct.-Dec. 2017, doi: 10.1109/TETC.2016.2606384.

[10] J. Deogirikar and A. Vidhate, "Security attacks in IoT: A survey," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, 2017, pp. 32-37, doi: 10.1109/I-SMAC.2017.8058363.

[11] M. Nawir, A. Amir, N. Yaakob and O. B. Lynn, "Internet of Things (IoT): Taxonomy of security attacks," 2016 3rd International Conference on Electronic Design (ICED), Phuket, 2016, pp. 321-326, doi: 10.1109/ICED.2016.7804660.

[12] T. Song, R. Li, B. Mei, J. Yu, X. Xing and X. Cheng, "A Privacy Preserving Communication Protocol for IoT Applications in Smart Homes," in IEEE Internet of Things Journal, vol. 4, no. 6, pp. 1844-1852, Dec. 2017, doi: 10.1109/JIOT.2017.2707489.

[13] M. N. Aman, B. Sikdar, K. C. Chua and A. Ali, "Low Power Data Integrity in IoT Systems," in IEEE Internet of Things Journal, vol. 5, no. 4, pp. 3102-3113, Aug. 2018, doi: 10.1109/JIOT.2018.2833206.

[14] Doshi, R., Apthorpe, N., and Feamster, N. (2018, April 11). Machine Learning DDoS Detection for Consumer Internet of Things Devices. https://doi.org/10.1109/SPW.2018.00013

[15] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," in IEEE Access, vol. 7, pp. 82721-82743, 2019, doi: 10.1109/ACCESS.2019.2924045.

[16] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," in IEEE Communications Surveys and Tutorials, vol. 22, no. 3, pp. 1646-1685, thirdquarter 2020, doi: 10.1109/COMST.2020.2988293.

[17] L. Xiao, X. Wan, X. Lu, Y. Zhang and D. Wu, "IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?," in IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 41-49, Sept. 2018, doi: 10.1109/MSP.2018.2825478.

[18] F. Hussain, R. Hussain, S. A. Hassan and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," in IEEE Communications Surveys and Tutorials, vol. 22, no. 3, pp. 1686-1721, thirdquarter 2020, doi: 10.1109/COMST.2020.2986444.

[19] N. A. Stoian, "Machine Learning for anomaly detection in IoT networks : Malware analysis on the IoT-23 data set," http://purl.utwente.nl/essays/81979

[20] IoT-23 Dataset "https://www.stratosphereips.org/datasets-iot23"

[21] L. Lyu, J. Jin, S. Rajasegarar, X. He and M. Palaniswami, "Fog-Empowered Anomaly Detection in IoT Using Hyperellipsoidal Clustering," in IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1174-1184, Oct. 2017, doi: 10.1109/JIOT.2017.2709942.

[22] H. Sedjelmaci, S. M. Senouci and M. Al-Bahri, "A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology," 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 2016, pp. 1-6, doi: 10.1109/ICC.2016.7510811.