Group 13: Cyber Attack Detection in IoT Systems

Ayushi Mishra Shubham Sinha Rahul Kumar
21111262, 21111409, 21111069
ayushim21@iitk.ac.in, ssinha21@iitk.ac.in, rahulkumar21@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

With the advancement of the Internet of Things (IoT), the connection of different devices has increased significantly. Applications like smart home, smart city, and smart healthcare are helping people in their everyday lives. Although these intelligent applications provide aid in today's world, they are prone to several cyber-attacks and vulnerabilities. So, it is essential to build a robust system that can help detect these cyber-attacks using various Machine Learning (ML) and Deep Learning(DL) techniques. The project discusses detecting different kinds of cyber-attacks observed in a virtual smart home environment using various ML and DL methods. For creating a virtual environment, Distributed Smart Space Orchestration System (DS2OS) is used to deploy sensors and services like smartphones, light control, thermostat, door lock service, and washing service in a home. We used various ML approaches like Logistic Regression, Decision Trees, Random Forest, XGBoost, etc. to detect seven different attacks: denial of service (DoS), probing, spying, wrong setup, scan, malicious operation, and malicious control. Apart from ML methods, we used different DL methods like Dense Neural Network, CNN, LSTM, GRU, RNN, and different ensemble methods consisting of CNN-LSTM, and stacked ML model to detect the attacks.

1 Introduction

IoT models are becoming increasingly complicated as demand and growth in the IoT automated network system grow. People are becoming more accustomed to data-driven infrastructure, causing researchers to focus more on Machine Learning-based applications in conjunction with IoT. At this time, IoT and ML-based approaches are applied in almost every aspect of human existence.

The increasing complexity of IoT infrastructures is exposing their systems to unwelcome vulnerabilities. Security breaches and anomalies in IoT devices have become frequent in recent years. Therefore, a secure IoT infrastructure is required to protect against cybercrime. With the susceptibility of IoT devices, the security mechanisms in-place have become weak. For specific stakeholders and entrepreneurs, data is a form of money. This data is sensitive to some of the government agencies and private companies. An attacker can use a backdoor in IoT nodes to obtain confidential data from any key organization.

The project's primary purpose is to create an innovative, safe, and dependable IoT-based infrastructure that can detect vulnerabilities, have a secure firewall against any cyber threats, and automatically recover. In this project, a ML and DL-based solution is given for detecting and protecting the system in an abnormal state. Several ML classifiers were used. Another essential part of this project is that apart from just using the traditional ML algorithms, several DL techniques were introduced which gave better performance as compared to the ML approaches. We then introduced a new concept of ensemble methods, which is combining different algorithms together to show their performance in detecting cyber-attacks. Surprisingly, these algorithms gave us some decent results. We have covered all aspect of detecting assaults and cyber-attacks during this project.

In this report, Section 2 includes the related work which is already done for detecting cyber-attacks. We discussed how our approach is novel from the past work. Section 3 introduces to our proposed

framework used in this project. All the steps which we have performed in our project are mentioned here. In Section 4, we thoroughly explained all the steps discussed in the previous section. All the results generated from our methods were finally mentioned in Section 5. Various challenges that we faced during working on our project and what plans we have for future implementation is described in Section 6.

2 Related Work

As the internet-connected devices are increasing, a large amount of data gets generated. This data contains some crucial information that is confidential and needs to get secured. As a result, they are susceptible to cyber-attacks. Many Intrusion Detection System (IDS) and cyber-attack systems are proposed [1][2][3]. It is important to understand the assault patterns and behavior. Hence, ML plays a vital role in detecting and forecasting future intrusions and attacks. Several ML models like Naive Bayes, Decision Trees, Random Forest, SVM, Logistic Regression were built as supervised intrusion detection system[1] to detect various classes of attacks such as DoS, Man in the Middle attack, spoofing, and Reconnaissance like quick scan.

These ML approaches, though, provide good results in detecting the attacks, but during the case of zero-day attacks and detection of mutations of cyber-attack, these algorithms do not perform with good predictions[2]. Here, DL techniques play an essential role as they help in automatic feature extraction and fast detection rate. Moreover, the DL approaches can help in distributed IoT environments like fog computing which solves various challenges like bandwidth, latency, and communication. Thus, DL methods in fog computing will result in faster cyber-attack detection. Different DL techniques are proposed for the apprehension of network anomalies[4]. Traditional methods of packet inspection and intrusion detection increase the compute-power[5]. Thus, ML as an alternative will help in defend malware, botnet, and other attacks.

Because of the proliferation of IoT devices, cyberspace is no longer limited to the network level, and these devices are also vulnerable to cyber threats[6]. ML is one of the options for promptly responding to cyber-attacks. Spammers target the cellphones that are connected to these IoT devices as well. ML approaches are critical in detecting and identifying spam on mobile devices, such as spam in SMS, calls, email apps, mobile data, photos, and videos. Trustworthy ML application in cyberspace is required to provide high-level correctness assurances rather than model speed and accuracy.

In our proposed system, we have considered both ML and DL techniques for the detection of cyber-attacks. Also, we included some novel ways which were not proposed in previous research[7][8][9][10]. We have introduced ensemble methods in our project, including combining different ML and DL models to detect cyber-attacks. For building a sound intrusion detection system, not only was the accuracy considered, but we also looked at the training time for all the models for real-world usage. Our approach covers all the space for detecting any anomaly, some part of which was missing in models introduced in previous research work.

3 Proposed Idea

For our project, we have used DS2OS traces collected from a virtual smart home environment. The orchestration system acts like a middleware where services like light controller, motion sensor, washing service, battery service, door lock, smartphone, and thermostat communicate. These services are then deployed in different locations at home where they perform read or write operations to get the traces of simulation. Fig 1 shows our proposed method of the project. During the communication phase, several attacks can happen inside a house. Some of the scenarios can be a smart speaker, that can be used by hackers to issue their voice commands. Items like a smart refrigerator and an intelligent coffee machine might cause severe problems in the kitchen if successfully hacked. Hackers can program an intelligent refrigerator to enter incorrect expiration dates or place massive online grocery orders. Intelligent robot vacuum cleaners, which can move about the house, can give hackers information on the property's layout. The hackers might use this information to plan future operations and moves.

The dataset collected includes 357,592 samples consisting of 13 features and eight kinds of attacks. The data is then pre-processed using various steps like Data cleaning, which handles missing values.

We also did feature extraction by converting each feature into vectors for more straightforward computation. Besides data pre-processing, exploratory data analysis (EDA) was performed to get better

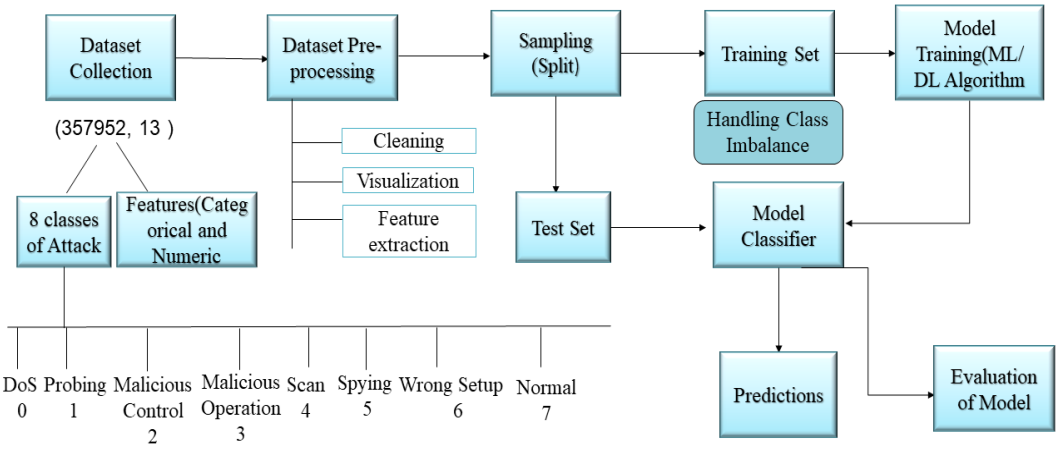


Figure 1: Proposed Framework

data insights. After this phase, data was split in the ratio of 70:30, where the training set contains 70 % of the data and test set contains 30 % of the data. During the training phase, we have handled the class imbalance problem as it will lead to overfitting. So. to avoid this problem, a smote algorithm was used, which over-sampled the minority class, resulting in balancing each class label with equal proportion.

For training the model, different ML and DL techniques were used employed with different optimization techniques. ML algorithms used for training the model are Logistic Regression, Decision Trees, Random Forest, XGBoost, LightGBM, Catboost, and, Histgradientboost. The DL methods we included are Recurrent Neural Network (RNN), Dense Neural Network (DNN), 1D Convolutional Neural Network (CNN), Long-Short-Term Memory (LSTM), CNN-LSTM. We also used one stacked ML method to improve the model performance. After the model classifier was built, the test set held aside was used to make final predictions. The model was then evaluated using different metrics like accuracy, precision, recall, and F1 score. A confusion matrix was also made for algorithms to consider the difference between actual and predicted labels.

4 Methodology

The purpose of the project is to build an innovative and reliable system that can detect any vulnerabilities. ML and DL techniques will help learn the past patterns of attack and detect future assaults. Various steps taken while building the projects are:

4.1 Data Collection

The open source dataset was collected from kaggle[11].In the dataset, there are 357,952 samples and 13 features. The dataset has 347,935 Normal data and 10,017 anomalous data and contains eight classes which were classified. Table 1 gives a detailed picture of the distribution of different attacks. Description of 13 features are given in Table 2. There were seven classes of attacks:

- Denial of Service (Dos): It occurs due to congestion of traffic in single source and receiver.
- Data Type Probing: In this attack, a malicious node can write a different datatype than an intended datatype.
- Malicious Control: an attacker gain a session key and can control the entire system.
- Malicious Operation: This attack is generally caused by the malware which affects the device's performance.
- Scan: when data is acquired by hardware while scanning, the data may get corrupted sometimes.

- Spying: In this attack, the attacker uses a backdoor system to cause vulnerabilities in the system and steal confidential information.
- Wrong Setup: At the time of setting up the system, then also data can be disrupted.

Table 1: Classes of Attack

Attacks	Frequency Count 2
Denial of Service	5780
Data Type Probing	342
Malicious Control	889
Malicious Operation	805
Scan	1547
Spying	532
Wrong Setup	122

Table 2: Features Description

Features	Data Type
Source ID	Nominal
Source Address	Nominal
Source Type	Nominal
Source Location	Nominal
Destination Service Address	Nominal
Destination Service Type	Nominal
Destination Location	Nominal
Accessed Node Address	Nominal
Accessed Node Type	Nominal
Operation	Nominal
Value	Continuous
Timestamp	Discrete
Normality	Nominal

4.2 Data Analysis

It is important to get some insights of the data. So, we perform exploratory data analysis of some of the features present in the dataset. Fig 2 shows the analysis of features described in Table 2.

4.3 Data Preprocessing

After collecting the data, it was then preprocessed using different approches:

- Data Cleaning:
  - Handling Missing Values: Features like Accessed Node Type and Value contain some empty fields. The empty values for Accessed Node Type are dropped and the empty fields in Value are filled with mean values.
  - Features like Timestamp was dropped as it does not have any significant impact on the data analysis.
- Feature Engineering:
  - It is vital to look at the datatype of all the features.
  - The categorical features are converted into vectors. We have used Label Encoding to convert it into feature vectors.

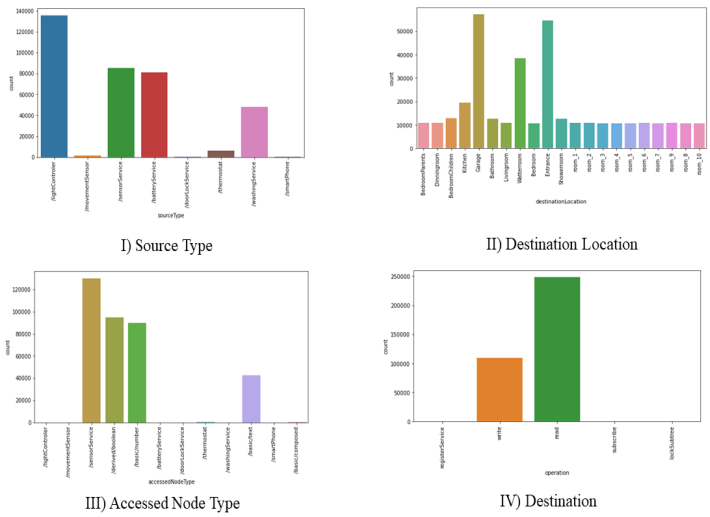


Figure 2: Data Analysis of Features : I) Source Type, II) Destination Location, III) Accessed Node Type, IV) Operation

  - By using Label Encoding, it will not increase the dimensionality of the data and processing time will be less.
- Data Normalization: The training data is normalized to make it in a same scale.
- Class Imbalance Problem: It was observed that the collected dataset was highly imbalanced and it leads to the problem of overfitting. To overcome this issue, we have used Smote algorithm which is used to create virtual records. The minority classes which consists of all the attacks classes were oversampled. This makes all the classes with equal number of samples.

4.4 Machine Learning Models

Initially, we tried with traditional ML models to interpret model performance on them. The first ML approach used was logistic regression. The model performs very poorly with just 41% accuracy. After that, we trained a decision tree classifier which gave an accuracy of 74.5%. Both the standard ML models did not give good results. To improve the model performance, we then employed stacked ML method.

- Stacked ML Method: In this method, decision trees, random forest and XGBoost algorithm were together combined to form a model stack. A stacked classifier was used to give this model stack to logistic regression classifier which gave the final results with am accuracy of 98%.

Various tree based ensemble classifiers were used to fit the data like Random Forests, Hist Gradient Boost, XGBoost, LightGBM, Catboost.

- Model Selection:
  - Reason why tree based ensemble classifiers were used is that they are the only type of classic ML models that are non-linear, independent of kernel function approximations as in Support Vector Machines(SVMs) and storing training data as in k-Nearest Neighbours(KNNs). So, they are more flexible to train.
  - Neural Networks based models can approximate any function with just one hidden layer. But training time is much higher. Also packet data is structured so auto feature selection property is not required.
- Model Introduction:
  - Random Forest and HistGradientBoost are classic forest and boosting algorithms provided by sci-kit learn library. Forest takes subset of training data and features for each tree while boosting builds upon successively from previous trees.

  - XGBoost (eXtreme Gradient Boosting) is an advanced implementation of gradient boosting algorithm. It has built in support for regularization, missing values, cross-validation and uses parallel processing.
  - Light GBM is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm, used for ranking, classification and many other machine learning tasks.It splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth wise or level wise rather than leaf-wise.
  - CatBoost is similar to XGBoost and LightGBM and has the flexibility of giving indices of categorical columns so that it can be encoded as one-hot encoding.
- Parameter Tuning:
  - Two types of hyper-parameter tuning were done - to control over-fitting and to control training speed. To control parameters related to learning rate, max depth, regularization, number of leaves were trained. For training speed, parameters like percentage features per split, number of iterators were trained.
  - Optimal values for these parameters were found using HalvingGridSearch method provided in scikit-learn. It is a variation of grid search which uses tournament method to filter out best candidates in each iteration. Scripts were run in Google Colab.

4.5 Deep Learning Techniques

With the advancement in DL techniques, it can be applied to various other domains like network intrusion detection, image processing, speech recognition and many others. Since our datasets are more alike to sequential data types so, we considered some sequential data modelling based algorithm that contains some sort of memory and can store for longer amount of time during training. Out of all other models, RNN(recurrent neural network) and LSTM(long short-term memory) based models are quite powerful, which can learn dynamic temporal behaviors in arbitrary length of large-scale network traffic data.

4.5.1 Simple RNN

We first tried out simple RNN model to detect the anomalies in the network. To leverage this, we model network traffic as time series, and implemented this simple RNN with a supervised learning method, using millions of known good and bad network connections. It is also called short-term memory based model because it stores the information for shorter time. About the model architecture, we implemented two stacks of RNN of having 8 nodes and dropout layer at rate of 0.1 and followed by the dense layer with 8 nodes. With this lightweight model we're able to achieve good performance on the datasets. Since the datasets size is huge so dropout layer is required to overcome the over-fitting problem while training the model. Table 3 shows the other hyperparameters used in the training phase. With this model, we're able to achieve 96.03%.

4.6 1D CNN

Convolutional Neural Network, generally known to be used for image data can also be used with continuous data. For our 1D CNN model, first two layers of convolution is added with 64 filters and kernel size as 1. In these layers, relu activation function is used. After that a dropout layer is added to avoid overfitting problem. To get better features, max-pooling layer is added with pool-size of 2. At last, the layers are flattened to get one column vector which is then passed to a dense layer with 16 hidden units to get the final output. The modle gave the accuracy of 98.8 % which was good. Different hyperparameters used in this model are described in Table 3

4.6.1 LSTM

LSTM has introduced memory block instead of conventional simple RNN unit, which handle vanishing and exploding gradient problem that actually helps to retain the memory for longer time and performed accordingly based on temporal behaviors in the network traffic. LSTM model is exactly same as in RNN model architecture except we used LSTM layer here instead of RNN. With the property of storing

memory for longer time, this model is giving good performance around 98.54% as compared to RNN. Table 3 shows the other hyperparameters used in the training phase.

4.7 CNN-LSTM

In the front end, CNN layers are added, followed by LSTM layers with a dense layer on the output to create a CNN-LSTM model. This architecture defines two sub-models: the CNN model for feature extraction and the LSTM model for feature interpretation. One of the benefits of using CNN with LSTM is that the LSTM layers can save all the model summary from previous timestamp generated from the CNN model. This helps the model to take any random CNN layer and make the predictions. The CNN-LSTM model we proposed contains two convolution layer followed by max-pooling layer and afterwards, LSTM layer is added which was then passed to dense layer to give the final output. The accuracy achieved with this model was 98.70%. Table 3 shows hyperparameters used in the training phase. Another advantage of using CNN-LSTM model together is that instead of using a large number of CNN layers again, the LSTM can store them as a memory and use them as and when required.

4.7.1 GRU

Gated recurrent unit (GRU) is a modified version of LSTM in which only two gates in internal structure of GRU units are used which makes it faster than LSTM in model training speed. It also has less number of parameters to train so it converges quickly than LSTM. We have tried early stopping criteria to stop the experiments if there's no major changes in the performance. So, we observed that GRU converged at epoch=6 with batch size 512 while RNN and LSTM took more time to converge. About the model architecture, four stacks of layers are used in which GRU layer of having 32 nodes and dropout at an rate 0.1 and other hyperparameters are same as other models. Table 3 shows the other hyperparameters used in the training phase.

4.7.2 Deep Neural Network

Other than these memory based techniques, we have also tried deep neural network to detect the cyber-attacks in the network traffic. This is a bit heavy model as compared to others. There are basically four dense layers with varying number of hidden units i.e 1024, 768, 512, 256 with a combination of dropout layer having rate=1. GRU and this deep neural network performed quitetab similar with an accuracy of around 98.74%. Table 3 shows the other hyperparameters used in the training phase.

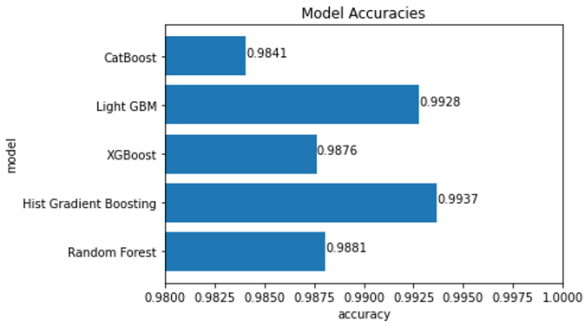
Hyperparameter	
Epoch	10
Batch Size	512
Loss	Categorical Crossentropy
Activation Function	ReLU & Softmax
Optimizer	Adam

Table 3: Hyperparameter

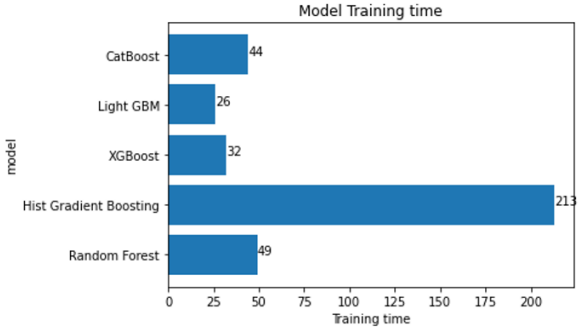
5 Results

In this project, we looked at not just the model accuracy, but also which features have more importance in final prediction. Also, the training time of all the algorithms was considered. Fig 4(a) shows the importance of each feature that have more impact in final results. As some of the m For different ML models, the accuracies of all models and the training time are mentioned in Fig 3(a) and Fig 3(b). The HistGradientBoosting model gave the best accuracy but has the highest training time. The ML algortithm that proved to be good for the attack detection was LightGBM as it takes the least time with an accuracy of 99.28%.

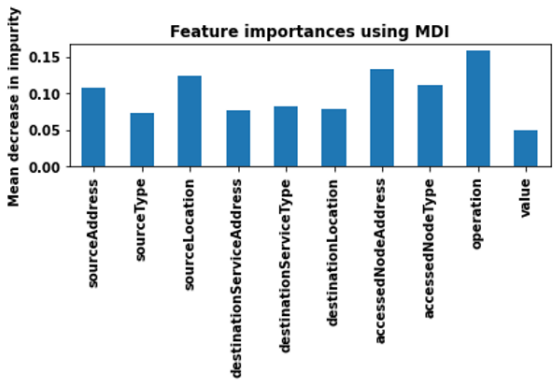
The DL approaches used also gave some decent results. GRU and 1D CNN gave the best results with an accuracy of 98% The Model accuracies and training time are given in Fig 4(b).



((a)) Accuracies of ML models



((b)) Training Time of ML models



((a)) Feature Importance

Algorithms	Accuracy	Model Training Time
Logistic Regression	41%	0.32
Decision Trees	74.5%	0.062
Random Forest	98.80%	49 sec
XGBoost	98.76	32 sec
LightGBM	99.28%	26 sec
Hist Gradient Boosting	99.37%	213 sec
CNN	98.8%	1200.5 sec
RNN	96.03%	1519 sec
LSTM	98.54%	1795 sec
GRU	98.80%	2701.Sec
Deep NN	98.74%	601.88sec
CNN + LSTM	98.70%	1582.07sec
Stacked ML	98%	2400sec

((b)) Accuracy and Training time of all algorithms

6 Discussion and Future Work

During the project, there were many challenges which we faced. Some of them are:

- As there was no experimental setup for the traces collected from different sensors and services, we cannot deploy any kind of attacks. So, in future, different classes of attacks can be deployed from laptop, which consists of all the data packets captured while communicating with different devices.
- To make a trustworthy ML or DL models, it is required to make models robust and smart in any kind of situations. Presently, the various ML and DL techniques used can't guarantee their success in some of the critical conditions like in healthcare, where if a cyber-attack occurs then it can also leads to death of a patient.

In future, we can solve all these issues by adopting some ideas and further studies in this area. We can deploy the attacks after collection of data. Also, during an ongoing criminal activity, a notification can be send to the user in the smartphone. For an investigation of cyber-attack, we can use digital forensics to investigate the data and collect evidence. This will help in gathering information like which part of the system got affected and what kind of data was corrupted, whether it was a file artifact or emails, audio and video files. So, an end-to-end system can be generated which will contain detection, prevention and investigation resulting in much reliable systems.

7 Conclusion

Based on the full study, it was found that both ML and DL algorithms performed extremely well. Also, the ensemble methods proved to be quite useful and gave better results for the algorithms which previously did not perform well. Besides, this work is based on virtual environment data. In the case of real-time data, there may raise different problems. A more empirical study is needed on this problem focusing on real-time data. In the IoT network, micro-services behave differently at different times which causes deviations in normal behavior in IoT services thus creating an anomaly. Further study is needed to interpret these problems in a more in-depth way.

8 Individual Contributions

Each member in the group have contributed equally while working in the project. Table 4 shows individual contribution of group members.

Table 4: Individual Contribution in project

Name of the Member	Contribution in project
Ayushi Mishra	Literature Survey, Data Collection, Data Preprocessing, Data Analysis, Trained few ML and DL models
Shubham Sinha	Literature Survey, trained all tree based ML models
Rahul Kumar	Literature Survey, trained all DL models including 1 ensemble DL approach

References

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

[2] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042–9053, 2019.

[3] A. Bandekar and A. Y. Javaid, "Cyber-attack mitigation and impact analysis for low-power iot devices," in *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 1631–1636, 2017.

[4] M. Roopak, G. Yun Tian, and J. Chambers, "Deep learning models for cyber security in iot networks," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0452–0457, 2019.

[5] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for iot: Toward universal and resilient systems," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018.

[6] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[7] C. Braghin, S. Cimato, E. Damiani, F. Frati, L. Mauri, and E. Riccobene, *A Model Driven Approach for Cyber Security Scenarios Deployment*, pp. 107–122. 02 2020.

[8] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home iot devices," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 07 2019.

[9] H. Haddadpajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," *Future Generation Computer Systems*, vol. 85, 03 2018.

[10] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. Hashem, "Attack and anomaly detection in iot sensors in iot sites using machine learning approaches," *Internet of Things*, vol. 7, p. 100059, 2019.

[11] FrancoisXA, "Ds2os traffic traces," 2017.