An Enhanced Efficient Approach For Spam Detection In IOT Devices Using Machine Learning

S.Preethi
Student, Dept. of Computer science and engineering,
Vasireddy Venkatadri Institute of Technology, India

Sk.Neelofar
Student, Dept. of Computer science and engineering,
Vasireddy Venkatadri Institute of Technology, India

T.Chandralekha
Student, Dept. of Computer science and engineering,
Vasireddy Venkatadri Institute of Technology, India

S.Rishikanth
Student, Dept. of Computer science and engineering,
Vasireddy Venkatadri Institute of Technology, India

Prof.Dr.Shaik.Khaja Mohiddin
Asso.Prof, *Dept. of Computer science and engineering,*
Vasireddy Venkatadri Institute of Technology, India

Abstract- The number of Internet of Things (IoT) devices is growing at a quick pace in smart homes, producing large amounts of knowledge, which are mostly transferred over wireless communication channels. The volume of data released from these devices also increased. In addition to an increased volume, the IoT device produces a large amount of data with several different modalities having varying data quality defined by its speed in terms of time and position dependency. However, various IoT devices are susceptible to different threats, like cyber-attacks, fluctuating network connections, leakage of data, etc. However, the unique characteristics of IoT nodes render the prevailing solutions insufficient to encompass the whole security spectrum of the IoT networks. In such an environment, machine learning algorithms can play an important role in detecting anomalies in the data, which enhances the security of IoT systems. Our methods target the data anomalies present in general smart Internet of Things (IoT) devices, allowing for easy detection of anomalous events based on stored data. The proposed algorithm is employed to detect the spamicity score of the connected IoT devices within the network. The obtained results illustrate the efficiency of the proposed algorithm to analyze the time-series data from the IoT devices for spam detection.

Keywords –IoT devices, IoT security, Machine Learning, Smart Home, Spamicity Score

I. INTRODUCTION

IoT is taken into account as an interconnected and distributed network of embedded systems communicating through wired or wireless communication technologies. Massive growth and rapid development in the field of the Internet of Things (IoT), makes the presence of IoT devices prevalent in smart homes and smart cities. It is also defined because the network of physical objects or things empowered with limited computation, storage, and communication capabilities is also embedded with electronics (such as sensors and actuators), software, and network connectivity that permits these objects to gather, sometimes process, and exchange data. The things in IoT ask the objects from our lifestyle starting from smart household devices like a smart bulb, smart adapter, smart meter, smart refrigerator, smart oven, AC, temperature sensor, smoke detector, IP camera, to more sophisticated devices like frequency Identification (RFID) devices, heartbeat detectors, accelerometers, sensors in the parking zone, and a variety of other sensors in automobiles, etc.

There are various large amounts of applications and services offered by the IoT ranging from critical infrastructure to agriculture, military, home appliances, and personal health care. As the usage of IoT devices increases the anomalies generated by these devices also grow beyond the count. IoT applications need to ensure information protection to fix security issues like interruptions, spoofing attacks, Dos attacks, jamming, eavesdropping, spam, and malware. The maximum care to be taken is with web-based devices as the maximum number of IoT devices are web-dependent. It is common in the work environment that the IoT devices introduced in an association can be utilized to execute security and protection includes proficiently. For example, wearable devices that collect and send user's health data to a connected smartphone should prevent leakage of data to ensure privacy. It has been found in the market that 25-30% of working employees connect their Personal IoT devices with the organizational network. The expanding nature of IoT attracts both the audience, i.e., the users and therefore the attackers.

However, with the emergence of ML in various attack scenarios, IoT devices choose a defensive strategy and decide the key parameters in the security protocols for a trade-off between security, privacy, and computation. This work enhances the algorithm to affect the time-series regression model rather than a classification model and may also execute ML models in parallel. This proposed paper focuses on determining the trustworthiness of the IoT device within the smart home network. The algorithm scores an IoT device with a spamicity score to secure smart devices by calculating spam scores using different machine learning models.

1.1 Contributions

Based upon the above discussions, the following contributions are presented in this paper.

- The proposed scheme of spam detection is tested against four different machine learning models
- An algorithm is designed to calculate the spamicity score of each Machine Learning model.
- Based on this spamicity score the trustworthiness of IoT devices is analyzed.

1.2 Organization

The rest of the paper is organized as follows. Section2 discussed the related work. Section 3 illustrated the proposed scheme. Results are discussed and analyzed in Section 4. Finally, the paper is concluded in Section 5.

II.RELATED WORK

Several Machine learning and soft computing approaches have been applied in the analysis of spam detection in IoT devices(Eg.Smart homes).[1]Fatima Hussain, Rasheed Hussain, Syed Ali Hassan, Ekran Hossain, extensive efforts have been made to address the security and privacy issues in IoT networks.[2]Choi J, Jeong, HKim J, Jung W; Kim H, Kim J, this paper, Sihai Tang took it as a challenge to discover hidden information from the massive amount of stored data in the cloud.[3]Aaisha Makkar, Dr. Neeraj Kumarin this paper, they had analyzed to identify the best and efficient Spam Detection in IoT Devices .Extreme Gradient Boosting, Decision Trees, Gradient Boosted regression, Bagged Model, Bayesian Generalized Linear model, Generalized Linear Model with Stepwise Feature Selection models are used.[4]Ameema Zainab, ShadyS. Refaat, OthmaneBouhali, in this paper, they presented a new approach for Spam Detection in Smart homes. Extreme Gradient Boosting, Decision Trees, Random Forest, Gradient Boosted regression models are used.

III. REVIEW OF LITERATURE

IoT systems are susceptible to network, physical, and application attacks similar to privacy leakage, comprising objects, services, and networks.

3.1 Denial of service (DDoS) attacks

The attackers can flood the target database with unwanted requests to forestall IoT devices from having access to varied services. These malicious requests produced by a network of IoT devices are commonly referred to as bots. This type of attack can exhaust all the resources provided by the service provider. It can block authentic users and should make the network resource unavailable.

3.2 Radio-frequency Identification (RFID) attacks

These attacks are mostly seen in the physical layer of IoT devices. This attack ends up in losing the integrity of the device. Attackers try to modify the information either at the node storage or while it's within the transmission within the network. The common attacks possible at the sensor node are attacks on availability, attacks on authenticity, attacks on confidentiality, Cryptography keys brute-forcing. The countermeasures to make sure prevention of such attacks include password protection, encoding, and restricted access control.

3.3 Internet attacks

The IoT device can stay connected with the Internet to access various resources available over the internet. The intruders who want to steal other systems' information or want their target website to be visited continuously, use spamming techniques. Ad fraud is one of the common techniques that is used. It generates artificial clicks at a targeted website for profit related to money. Such a practicing team is known as cybercriminals.

3.4 Near-field Communication (NFC) attacks

These attacks are mainly concerned with electronic payment frauds. The possible attacks are unencrypted traffic, Eavesdropping, and Tag modification. The solution for this problem is conditional privacy protection. So, the attacker fails to form the identical profile with the assistance of the user's public key this model is predicated on random public keys by a trusted service manager.

Machine Learning is the study of computer algorithms that improve automatically through experience by executing different tasks. Machine learning is the subset of computer science. Various machine learning techniques like supervised learning, unsupervised learning, and reinforcement learning have been widely accustomed to improve network security. The existing ML technique, which helps in the detection of the above-mentioned attacks

Supervised machine learning techniques

The models like support vector machines (SVMs), random forest, Naive Bayes, K-nearest neighbor (K-NN), and neural networks (NNs) are used for labeling the network for detection of attacks. In IoT devices, these models successfully detected the DoS, DDoS, intrusion, and malware attacks

Unsupervised machine learning techniques

These techniques outperform their counterpart's techniques within the absence of labels. It works by forming clusters. In IoT devices to detect Dos attacks, multivariate correlation analysis is used.

Reinforcement machine learning techniques

These models enable an IoT system to pick security protocols and key parameters by trial and error against different attacks. Q-learning has been used to improve the performance of authentication and will help in malware detection also.

Semi-supervised learning

Semi-supervised learning falls between unsupervised learning (don't contain any labeled training data) and supervised learning (with completely labeled training data) machine learning techniques. A number of the training examples are missing training labels, yet many machine-learning researchers have found that unlabelled data, when utilized in conjunction with a little amount of labeled data, can produce a substantial improvement in learning accuracy. In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to induce, resulting in larger effective training sets. Machine learning techniques help to create protocols for lightweight access control to save a lot of energy and extend the IoT systems lifetime.

The outer detection scheme as developed, for example, applies K-NNs to address the issue of unregulated outer detection in WSNs. The literature survey demonstrates the applications of Machine learning in enhancing network security. Therefore, in this paper, the given problem of web spam is detected with the implementation of various machine learning techniques.

IV. PROPOSED SCHEME

4.1 Machine Learning Models

The use of machine learning models within the IoT has shown promising results for identifying malicious internet traffic using anomaly detection research. Moreover, either detection of anomalies or the employment of a spamicity score to trace the safety of the network components are motivated to possess a safe and secure network infrastructure.

Several ML models are utilized for supervised machine learning; however, this paper uses ensemble methods, a group of ML techniques supported by decision trees. The machine learning models utilized within the paper are described as follows.

4.1.1 Support Vector Machines (SVM)

Support vector machines, also referred to as support vector networks, are a group of related supervised learning methods used for classification and regression. However, it's mostly utilized in classification problems. Within the SVM algorithm, we plot each data item as a degree in n-dimensional space (where n is the number of features you have) with the worth of every feature being the worth of a specific coordinate. Then, we can classify by finding the hyper-plane that differentiates both classes. Hence, we can say that the main objective of SVM is to find a hyperplane in an N- dimensional space that distinctly classifies the data points.

SVM can classify both linear and non-linear data. To classify non-linear data it uses a method called the kernel trick to rework your data so it supports these transformations and it also finds an optimal boundary between the possible outputs. A kernel is a function which maps a lower dimensional data into higher dimensional data. Simply put, it does some extremely complex data transformations, then figures out a way to separate your data supporting the labels or outputs you've defined. Given a group of coaching examples, each marked as belonging to 1 of two categories, an SVM training algorithm builds a model that predicts whether a replacement example falls into one category or the opposite.

An SVM training algorithm could be a non-probabilistic, binary, linear classifier, although methods like Platt scaling exist to use SVM in a very probabilistic classification setting. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what's called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. SVM also uses another method called Soft Margin which allows SVM to make certain number of mistakes and keep the margin as wide as possible so that other points can be classified correctly.

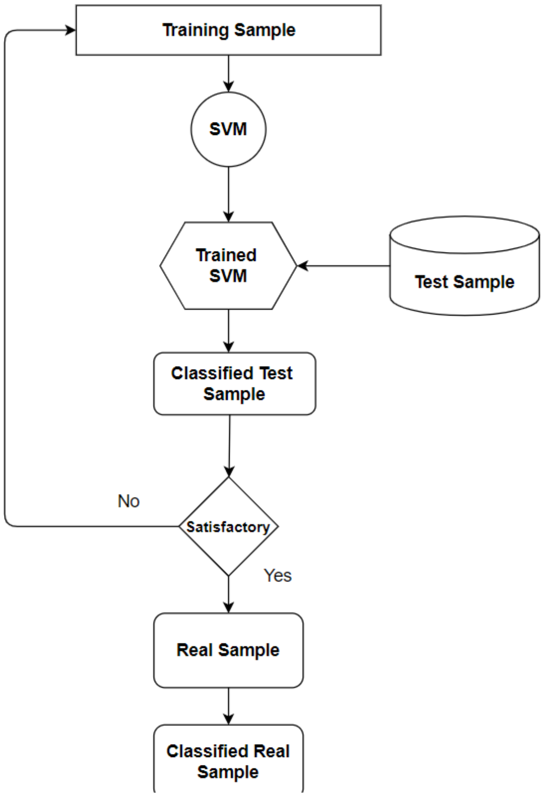


Figure 1.SVM

4.1.2 Random Forest

Random forests (RF) are a bag containing n Decision Trees (DT) having a special set of hyper-parameters and trained on different subsets of information. In machine learning language, Random Forests also are called an ensemble or bagging method. Random forest is one amongst the foremost used algorithms due to its simplicity and stability. Random forests are more stable and reliable than simply a choice tree.

Random Forest is a supervised machine learning technique used for both classification and regression. But we'll discuss its use for classification because it’s more intuitive and straightforward to grasp. It's an ensemble of decision trees that helps in reducing the variance in decision trees. It fulfills a balance between high variance and high bias by sampling with each tree fitted and a sample of features at each split, respectively.

The performance of random forest relies on the appropriate selection of the number of trees, N. As within the case of bagging, a greater value of N doesn't necessarily overfit the info, and hence, a sufficiently large value of N is chosen.

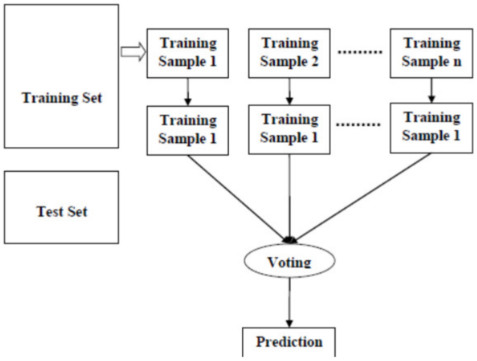


Figure 2. Random Forest

4.2.3 Decision Tree

A decision tree is a "set of rules" created by learning on a dataset that can be used to make predictions on future data. It employs a top-down approach, by utilizing variance reduction to partition the info into subsets of homogeneous values. It incorporates mixtures of categorical and numerical predictor variables with an integral a part of the procedure to perform internal feature selection.

These are the explanations why decision trees have emerged together of the foremost popular data processing learning methods. Decision trees can create an over-complex tree, which doesn't tend to generalize the information well and might lead to overfitting, while the choice tree doesn't perform as neural networks for nonlinear networks, it's usually prone to noisy data. Decision trees expect visible trends within the data and also perform well on sequential patterns; if this is often not the case, then decision trees must be avoided for statistical applications.

Decision trees are used for regression and classification problems. Most often used in classification problems. Decision trees finds the relation between target data and input features with simple decision rules. If the rule is met then the classification of particular node gets break and called terminated leaf. The parent node in decision tree consists of total samples and gets further classified upon decision rules. Edges in decision trees are rules or values that helps to classify the data.

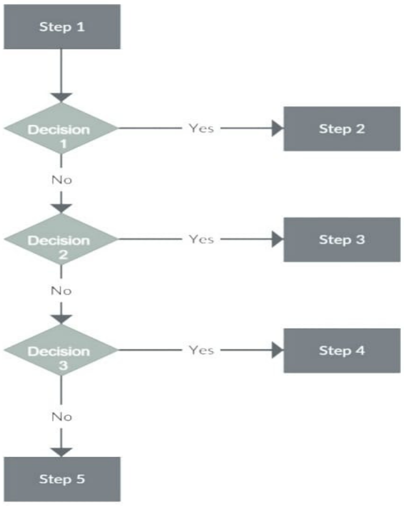


Figure 3. Decision Tree

4.2.4. Extreme Gradient Boosting (XGBOOST)

Extreme Gradient Boosting is a well-liked supervised machine learning model with characteristics of distributed and out-of-core computation, efficiency, and parallelization. The parallelization occurs for multiple nodes in a very single tree and not across trees. It's a gradient boosting system that is efficient and scalable. The package includes a good linear model solver and an algorithm for tree learning. It supports various objective functions like regression, grouping, and ranking. It works with numeric vectors. It’s ten times quicker than existing gradient boosting algorithms. The strategy of gradient boosting uses more accurate approximations to seek out the most effective tree model. It uses a variety of clever tricks that make it particularly competitive with structured data normally. The poor learner is made up in each training round and its predictions are matched with the correct outcome.

The gap from prediction to reality is our model’s error rate. We can use these errors to calculate the gradient. The gradient is nothing special, but it's simply the loss function partial so it defines the steepness of the error function. The gradient may be accustomed to find them thanks to adjusting the parameters of the system so the errors within the next round of learning are often minimized (maximum) by ”downgradient". The biggest advantage of XGBoost is its scalability and quick speed, and it always outperforms the alternative ML models.

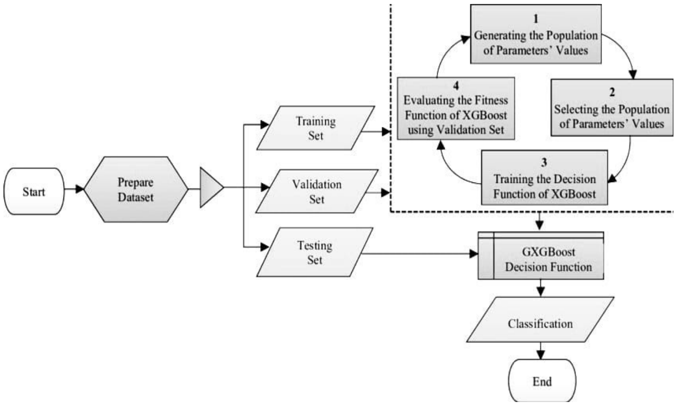

1
Generating the Population of Parameters' Values
4
Evaluating the Fitness Function of XGBoost using Validation Set
2
Selecting the Population of Parameters' Values
Training Set
3
Training the Decision Function of XGBoost
Start
Prepare Dataset
Validation Set
Testing Set
GXGBoost Decision Function
Classification
End

Figure 4. Xgboost

Model no.	Model	Module	Package	Tuning parameters
Model1	Support Vector Classifier	SVC	sklearn	None
Model2	Random forest	Random-rest Classifier	sklearn	None
Model3	Decision tree	Decision Tree Classifier	sklearn	None
Model4	eXtreme Gradient Boosting	XGB Classifier	Xgboost	nrounds,lambda,alpha

Table 1. Machine Learning Models

4.2.5 DATA SET

The Data Set used in this paper is REFIT Smart Home Data Set with some modifications. Firstly, the feature reduction is finished. In the IoT dataset used in this proposal, we have 10 features as shown in the above table. After the feature extraction, the feature selection is performed [5]. The features along with their importance score computed by the entropy-based filter are presented in Table I. For better understanding of dataset refer to [6].

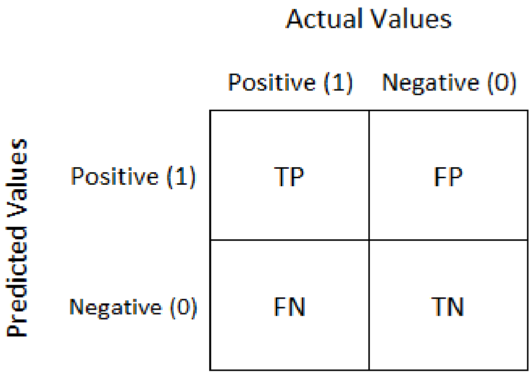
Table -1: Results of entropy based filters

Feature	**attr_importance**
Use _KW_	3.116543
Plug id	3.1165543
Space id	1.970967
Model	1.474572
Barn _KW_	1.338013
Gen _KW_	0.915861
Condtn type	0.915861
Wine cellar _KW_	0.847213
metertype	0.574533
windowtype	0.376330

V. PERFORMANCE METRICS

5.1 Confusion Matrix

Confusion matrix could be a very important measure used while solving classification problems. It is often applied to binary classification similarly as for multiclass classification problems.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

- The target variable has 2 values: Positive or Negative
- The columns represent the particular values of the target variable
- The rows represent the expected values of the target variable

5.2 Spamicity Score

```
--------------------------------------------------------------------------------
             Algorithm for calculating Spamicity Score
--------------------------------------------------------------------------------

Input: actual value, predicted value
Output: Spamicity Score

Procedure spam_score (pred, actual):
   lmse <- []
   c <- 1                              ➢where c is  count
   spam_score <- 0
   for i in n do:
      temp <- (y_test[i]-pred[i])**2       ➢where n  length of testlist
      lmse.append(temp/c)
      c <- c+1                          ➢ where le length of
   end for                             meansquare error values
   for i in fl do:            ➢where fl is length of feature list with
     for j in le do:            desired entropies
      spam_score<- spam_score+(feature_list[i]*le[j])
     end for
   end for
   spam_score <- spam_score/c
   return spam_score
end procedure
```

Complexity analysis:

Time complexity:

Step4to8: O (n)

Step9 to 13: O (n)

T(c):O(1)+O(1)+O(1)+O(n)+O(n)

T(c): O (n)

Where n refers to size of input

Space complexity: as n is fixed input with linear matrix

The space complexity would be

S(c): O (n)

5.3 Accuracy

Accuracy is one metric for evaluating classification models. Machine learning model accuracy is that the measurement accustomed to determine which model is best at identifying relationships and patterns between variables in an exceedingly dataset supports the input, or training, data.

Accuracy is defined because the percentage of correct predictions for the test data. It is often calculated easily by dividing the quantity of correct predictions by the amount of total predictions.

$$accuracy = \frac{correct\ predictions}{all\ predictions}$$

Table-2: Spam Scores of Analyzed Models

ML Model	Spam Score	Accuracy
SVM	0.07	83.4
Random Forest	0.04	87.7
Decision Tree	0.06	79.8
Xgboost	0.03	87.9

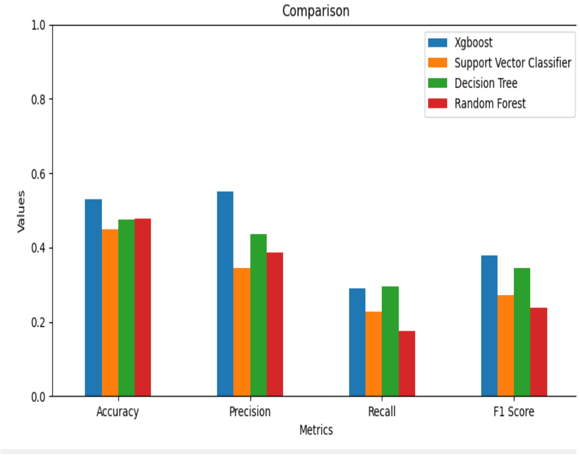


Figure 5.Performance metrics comparison

5.4 Precision

Precision is defined as the fraction of relevant instances among the retrieved instances. In simple words, it is the ratio between actuality positives and each one positives. Precision helps when the prices of false positives are high.

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

5.5 Recall

It's the amount of correct positive results divided by the quantity of all relevant samples. High recall means an algorithm returned most of the relevant results.

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Table -3: performance metrics table

ML Model	precision	recall	f1-score
SVM	70	83	76
Random Forest	80	81	81
Decision Tree	81	80	80
Xgboost	85	86	85

VI. CONCLUSION

The proposed system identifies the spam boundaries of IoT devices utilizing ML models. The IoT dataset utilized for tests is pre-prepared by utilizing highlight designing methodology. This paper decides the utilization of the spamicity score to comprehend the dependability of IoT gadgets in the smart home organization. Through thorough tests and analyses, different ML models were used to examine the time-arrangement information produced from keen meters. Different commitment levels of the IoT gadgets were resolved with the assistance of group techniques for ML by granting a spam score to the IoT gadgets in a smart home. The outcomes show that the spamicity score of the devices helps in refining the conditions to be taken for the successful working of IoT devices in the smart home.

REFERENCES

[1] Fatima Hussain,Rasheed Hussain,Syed Ali HassanHossain. Machine Learning in IoT Security: Current Solutions and Future Challenges

[2] Choi, J.; Jeoung, H.; Kim, J.; Ko, Y.; Jung, W.; Kim, H.; Kim, J. Detecting and identifying faulty IoT devices in smart homes with context extraction. In Proceedings of

The 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2018, Luxembourg, 25–28 June 2018; pp. 610–621.

[3] Tang, S.; Gu, Z.; Yang, Q.; Fu, S. Smart Home IoT Anomaly Detection based on Ensemble Model Learning from Heterogeneous Data. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 4185–4190.

[4] Makkar A.; Garg S.; Kumar, N.; Hossain, M.S.; Ghoneim, A.; Alrashoud, M. An Efficient Spam Detection Technique for IoT Devices using Machine Learning. IEEE Trans. Ind. Inform. 2020.

[5] Ameema Zainab, Shady S. Refaat and Othmane Bouhali;Ensemble-Based Spam Detection in Smart Home IoT Devices Time Series Data Using Machine Learning Techniques

[6] L. University, “Refit smart home dataset,” https://repository.lboro.ac.uk/ articles/REFIT Smart Home dataset/2070091, 2019 (accessed April 26, 2019