An Efficient Spam Detection Technique for IoT Devices using Machine Learning

Dr. Aaisha Makkar*, *Student Member, IEEE,* Dr. Sahil (GE) Garg[†] , *Senior Member, IEEE*
Dr. Neeraj Kumar [‡], *Senior Member, IEEE* Prof. M. Shamim Hossain [§], *Senior Member, IEEE*

Prof. Ahmed Ghoneim [¶], *Senior Member, IEEE* Dr. Mubarak Alrashoud [‖], *Senior Member, IEEE*
*Computer Science and Engineering Department, Chandigarh University (Punjab), India (aaisha.e8847@cumail.in).
[†] Ecole de technologie superieure, Universite du Quebec, Montreal, Canada (sahil.garg@ieee.org)
[‡] King Abdulaziz University (Saudi Jeddah), Asia University (Taiwan) and Thapar University (Patiala) (nehra04@yahoo.co.in)
[§] College of Computer and Information Sciences,King Saud University, Riyadh (mshossain@ksu.edu.sa)
[¶] College of Computer and Information Sciences, King Saud University, Riyadh (ghoneim@ksu.edu.sa)
[‖] College of Computer and Information Sciences, King Saud University, Riyadh (malrashoud@ksu.edu.sa)
.

***Abstract*—The Internet of Things (IoT) is a group of millions of devices having sensors and actuators linked over wired or wireless channel for data transmission. IoT has grown rapidly over the past decade with more than 25 billion devices are expected to be connected by 2020. The volume of data released from these devices will increase many-fold in the years to come. In addition to an increased volume, the IoT devices produces a large amount of data with a number of different modalities having varying data quality defined by its speed in terms of time and position dependency. In such an environment, machine learning algorithms can play an important role in ensuring security and authorization based on biotechnology, anomalous detection to improve the usability and security of IoT systems. On the other hand, attackers often view learning algorithms to exploit the vulnerabilities in smart IoT-based systems. Motivated from these, in this paper, we propose the security of the IoT devices by detecting spam using machine learning. To achieve this objective, *Spam Detection in IoT using Machine Learning* framework is proposed. In this framework, five machine learning models are evaluated using various metrics with a large collection of inputs features sets. Each model computes a spam score by considering the refined input features. This score depicts the trustworthiness of IoT device under various parameters. REFIT Smart Home dataset is used for the validation of proposed technique. The results obtained proves the effectiveness of the proposed scheme in comparison to the other existing schemes.**

I. Introduction

Internet of Things (IoT) enables convergence and implementations between the real-world objects irrespective of their geographical locations. Implementation of such network management and control make privacy and protection strategies utmost important and challenging in such an environment. IoT applications need to protect data privacy to fix security issues such as intrusions, spoofing attacks, DoS attacks, DoS attacks, jamming, eavesdropping, spam, and malware.

The safety measures of IoT devices depends upon the size and type of organization in which it is imposed. The behavior of users forces the security gateways to cooperate. In other words, we can say that the location, nature, application of IoT devices decides the security measures [1]. For instance, the smart IoT security cameras in the smart organization can capture the different parameters for analysis and intelligent decision making [2]. The maximum care to be taken is with web based devices as maximum number of IoT devices are web dependent. It is common at the workplace that the IoT devices installed in an organization can be used to implement security and privacy features efficiently. For example, wearable devices collect and send user's health data to a connected smartphone should prevent leakage of information 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 the attackers.

However, with the emergence of ML in various attacks scenarios, IoT devices choose a defensive strategy and decide the key parameters in the security protocols for trade-off between security, privacy and computation. This job is challenging as it is usually difficult for an IoT system with limited resources to estimate the current network and timely attack status.

A. Contributions

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

1) The proposed scheme of spam detection is validated using five different machine learning models.
2) An algorithm is proposed to compute the spamicity score of each model which is then used for detection and intelligent decision making.
3) Based upon the spamicity score computed in previous step, the reliability of IoT devices is analyzed using different evaluation metrics.

B. Organization

Rest of the paper is structured as follows. Section II discussed the related work. Section III illustrated the proposed scheme. Results are discussed and analyzed in Section IV. Finally, the paper is concluded in Section V.

II. Literature Review

IoT systems are vulnerable to network, physical, and application attacks as well as privacy leakage, comprising objects, services, and networks. These attacks are presented in Fig. 1. Let's have a look at some of the attack scenarios launched by the attackers.

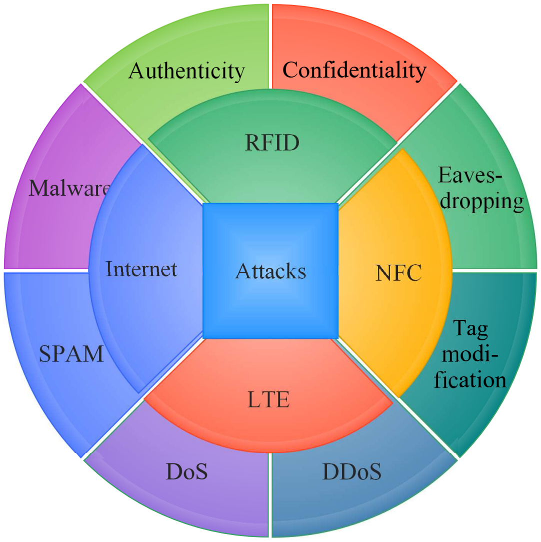

Fig. 1: Protocols with possible attacks

- Denial of service (DDoS) attacks: The attackers can flood the target database with unwanted requests to stop IoT devices from having access to various services. These malicious requests produced by a network of IoT devices are commonly known as bots [3]. DDoS can exhaust all the resources provided by the service provider. It can block authentic users and can make the network resource unavailable.
- RFID attacks: These are the attacks imposed at the physical layer of IoT device. This attack leads to loose the integrity of the device. Attackers attempt to modify the data either at the node storage or while it is in the transmission within network. The common attacks possible at the sensor node are attacks on availability, attacks on authenticity, attacks on confidentiality, Cryptography keys brute-forcing [4]. The countermeasures to ensure prevention of such attacks includes password protection, data encryption and restricted access control.
- Internet attacks: The IoT device can stay connected with Internet to access various resources. The spammers who want to steal other systems information or want their target website to be visited continuously, use spamming techniques [5]. The common technique used for the same is *Ad fraud*. It generates the artificial clicks at a targeted website for monetary profit. Such practicing team is known as cyber criminals.
- 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 the conditional privacy protection. So, the attacker fails to create the same profile with the help of user's public key [6]. This model is based on random public keys by trusted service manager.

Various machine learning techniques such as supervised learning, unsupervised learning, and reinforcement learning have been widely used to improve network security. The existing ML technique, which help in detection of above mentioned attacks is discussed in Table I. Each machine learning technique according to its type and role in detection of attacks is described as below.

- Supervised machine learning techniques: The models such as 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 [7] [8] [9] [10].
- Unsupervised machine learning techniques: These techniques outperform their counterparts techniques in the absence of labels [9]. It works by forming the clusters. In IoT devices, multivariate correlation analysis is used to detect DoS attacks [11].
- Reinforcement machine learning techniques: These models Enable an IoT system to select security protocols and key parameters by trial and error against different attacks. Q-learning has been used to improve the performance of authentication and can help in malware detection as well [12] [9] [13].

Machine learning techniques help to build protocols for lightweight access control to save 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 [14]. The literature survey demonstrates the applications of Machine learning in enhancing the network security. Therefore, in this paper, the given problem of web spam is detected with an implementation of various machine learning techniques.

III. Proposed Scheme

A. System model

The digital world is completely dependent upon the smart devices. The information retrieved from these devices should be spam free. The information retrieval from various IoT devices is a big challenge because it is collected from various domains. As there are multiple devices involved in IoT, so a large volume of data is generated having heterogeneity and variety. We can call this data as IoT data. IoT data has various features such as real-time, multi-source, rich and sparse.

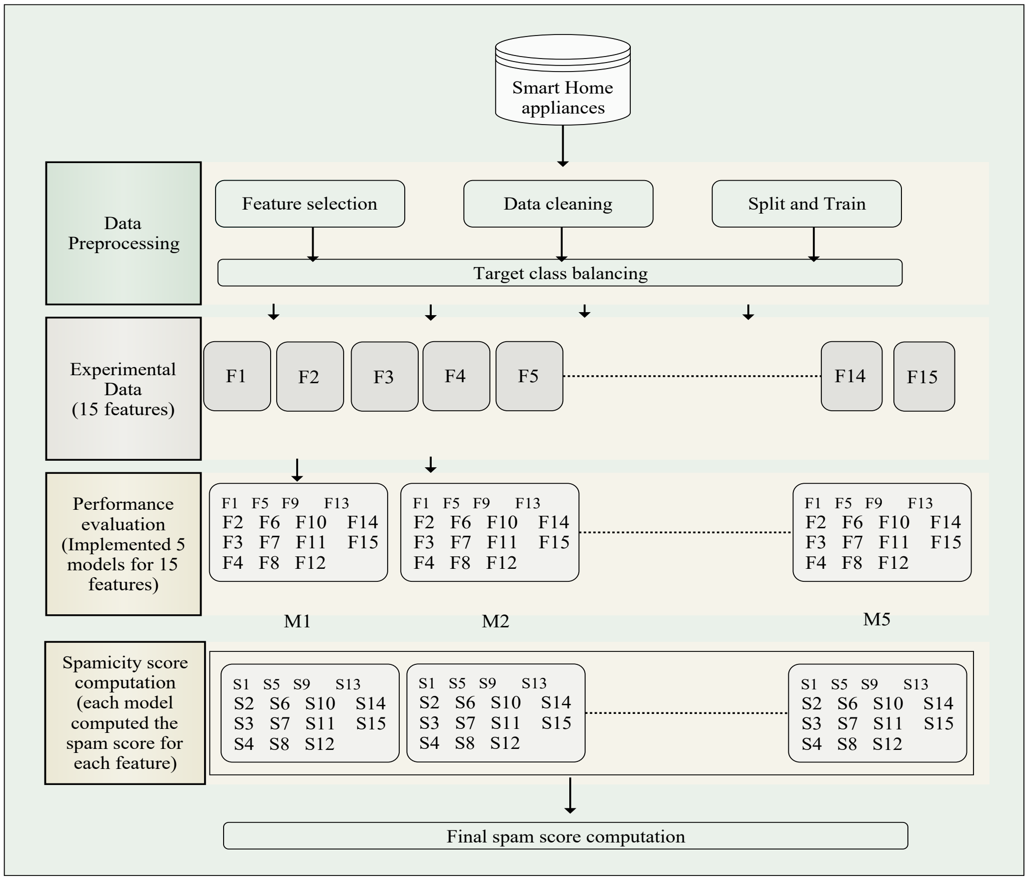

Fig. 2: Approach followed in the proposed scheme

TABLE I: Machine learning techniques used for the detection of different attacks

Author	Machine learning technique	Target attack	Performance
Kulkarni et al. , 2009 [7]	Neural Network	DOS	Improved the performance of system
Tan et al. , 2013 [11]	Multivariate correlation analysis	DOS	Improved accuracy
Li et al. ,2016 [12]	Q-Learning	DOS	Solved the associated optimality equations
Alsheikh et al., 2014 [8]	SVM, Naive Bayes	Intrusion	Detected the WSN attacks successfully
Buczak et al., 2015 [9]	Machine learning techniques	Cyber attacks	survey of ML techniques for detection of cyber attacks
Xiao et al.,2017 [13]	Q-Learning	Malware	Improve the detection accuracy
Narudin et al., 2016 [10]	Random forest, K-NN	Malware	99.97% true-positive rate (TPR)

The efficiency IoT data increases, if stored, processed and retrieved in an efficient manner. This proposal aims to reduce the occurrence of spam from these devices as defined by Eq. 1.

$$\min P(s) = \aleph - \vec{s} \tag{1}$$

In Eq. 1, $\aleph$ refers to the collection of information. $\vec{s}$ is the vector of spam related information, which is subtracted from $\aleph$ to decrease the probability of getting spam information from IoT devices.

B. Proposed methodology

To protect the IoT devices from producing the malicious information, the web spam detection is targeted in this proposal. We have considered various machine learning algorithms for the detection of spam from the IoT devices. The target is to resolve the issues in the IoT devices deployed within home. But, the proposed methodology considers all the parameters of data engineering before validating it with machine learning models. The procedure used to accomplish the target is presented in Fig. 2 and discussed in various steps as follows.

1) Feature Engineering: The machine learning algorithms works accurately with the appropriate instances and their attributes. We all know that the instances are the real data world value, gathered from the real world smart objects deployed across the globe. Feature extraction and feature selection are the core of feature engineering process.
   - Feature reduction: This methods is used to reduce the dimension of data. In other words, feature reduction

is the procedure to reduce the complexity of features. This technique reduces the issues like, over-fitting, large memory requirement, computation power. There are various feature extraction techniques. Among these, principal component analysis (PCA) is the most popular [15]. But, the method used in this proposal is PCA along with following IoT parameters.

  - Analysis time: The dataset used in the experiments, contains the data recorded for the span of eighteen months. For better results and accuracy, we have considered the data of one month. Considering the fact, the climate is the important parameter for the working of IoT device, the month with maximum variations has been taken into the consideration.
  - Web based appliances: Only those appliances are included, which stay connected with web for their working. The data collection includes the appliances: Television, Set top box, DVD player/recorder, HiFi, Electric heater, Fridge, Dishwasher, Toaster, Coffee maker, Kettle, Freezer, Washing machine, Tumble dryer, Electric heater, DAB radio, Desktop PC, PC monitor, Printer, Router, Electric heater, Electric heater, Shredder, Freezer, Lamp, Alarm radio, Lava lamp, CD player, Television, Video player, Set top box, Hub (network).

- Feature selection: It is the process of computing the most important subset of features. It works by computing the importance of each feature [16]. Entropy based filter is used as a feature selection technique in this proposal.
  - Entropy-based filter: This algorithm uses the correlation among the discrete attributes with continuous attributes to find out the weights of discrete attributes [17]. There are three functions using this entropy based filter namely, information.gain, gain.ratio, symmetrical.uncertainty. The syntax for these functions are:
    information.gain(formula, data, unit)
    gain.ratio(formula, data, unit)
    symmetrical.uncertainty(formula, data, unit)
    The arguments used in the function definition are described here.
    a) formula: It is the description of the working behind the algorithm.
    b) data: It is the set of training data with the defined attributes for which the selection is to be made.
    c) unit: It is the unit which is used for entropy computing. By default it takes the value "log".

TABLE II: Machine learning models

Model no.	**Model**	**Method**	**Package**	**Tuning parameters**
Model1	Bagged Model	Bag	Caret	Vars
Model2	Bayesian Generalized Linear Model	bayesglm	Arm	None
Model3	Boosted Linear Model	BstLm	bst, plyr	mstop, nu
Model4	eXtreme Gradient Boosting	xg-bLinear	Xgboost	nrounds, lambda, alpha
Model5	Generalized Linear Model with Stepwise Feature Selection	glm-StepAIC	MASS	None

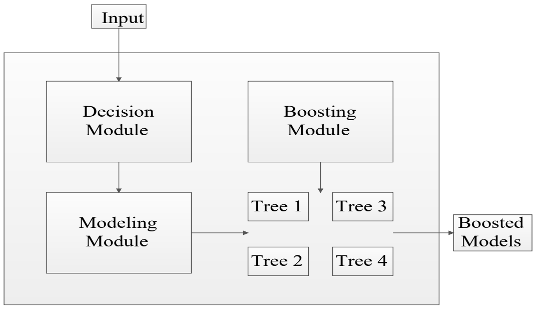

Fig. 3: Boosted linear model phases

C. Machine learning models

The proposed technique is validated by finding the spam parameters using machine learning technique. The machine learning models used for experiments are summarized in Table II.

1) Bayesian Generalized Linear Model (BGLM): It is a log likelihood uni-modal for exponential family forms, consistent, asymptotically efficient, and asymptotically normal. These essential elements are the real emphasis of Bayesian methods [18][19].

- First, prior information is incorporated. In general, prior information is quantitatively specified in the form of a distribution and represents a distribution of probability for a coefficient.
- Second, the prior is paired with a function of likelihood. The function of probability represents the results.
- Third, the combination of the prior and the probability function results in a subsequent distribution of coefficient values being formed.
- Fourthly, simulations are taken from the posterior distribution to construct an empirical distribution for the population parameter of probable values.
- Fifth, to sum up the statistical distribution of simulates from the posterior, simple statistics are used.

2) Boosted linear model: For the data elements, multiple decision trees are created, with the decision tree models by dividing the data series into a plurality of data classes.

Therefore, as a linear function, each of the data groups is modeled. From the modeling modules, the boosted models are formed as shown in Fig. 3.

Algorithm 1 Spamicity score computation

```
Input:
Output: Computed spamicity score

 1: procedure FUNCTION(PageRank)
 2:     for $i = 1\ to\ n$ do
 3:         for $j = 1\ to\ 15$ do
 4:             Matrix representation $z_i$                ▷ Formulation of matrix: n*15
 5:             Set $j \leftarrow j + 1$
 6:             Set $i \leftarrow i + 1$
 7:         end for
 8:     end for
 9:     for $i = 1\ to\ 15$ do
10:         Set $V_i = \leftarrow x$      ▷ Where x is the feature importance score according to
    Table III
11:     end for                                      ▷ Machine Learning model building
12:     $p[i] \leftarrow Y$                          ▷ Where Y is the predicted constraint
13:     for $i = 1\ to\ 15$ do
14:         Compute RMSE[i]= $\sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}}$      ▷ $p_i$ is the predicted array and
    $a_i$ is the actual array
15:     end for
16:     for $i = 1\ to\ 15$ do S $\leftarrow RMSE[i] * V_i$
17:     end for
18: end procedure
```

3) eXtreme Gradient Boosting (xgboost): It is a gradient boosting system which is efficient and scalable. The package includes an effective 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 is ten times quicker than existing gradient boosting algorithms. The method of gradient boosting uses more accurate approximations to find the best tree model. It uses a number of clever tricks that make it particularly competitive with structured data in general.

The poor learner is built up in each training round and its predictions is matched with the right 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 is simply the loss function's partial derivative-so it defines the steepness of the error function. The gradient can be used to find the way to adjust the parameters of the system so that the error in the next round of learning can be minimized (maximum) by ”downgradient.”
The formula used for building this model is as follows.
xgb $\leftarrow$ xgboost(data , label, eta, max_depth,
nround, subsample, colsample_bytree, seed,
eval_metric, objective, num_class, nthread)
In this method, there are basically three types of parameters being used, i.e., general parameters (booster, num_class..), booster parameters (max_depth, gamma..), learning task parameters (base_score, objective..).

4) Generalized Linear Model with Stepwise Feature Selection: Generalized linear models (GLMs) provide a dynamic framework to explain that how a dependent variable can be interpreted through a number of explanatory (predictor) variables. The parameter dependent may be continuous or discrete, and the explanatory variables may be either empirical (co-variate) or categorical (factors). We have fitted the model by using the stepwise feature selection. This method has to be repeated until there are significant found for all effects in the equation. The equation is specified with the support glmulti function in R.

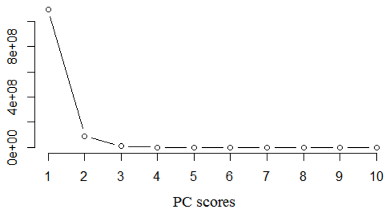

Fig. 4: Standard Deviations of Principal Components

D. Spamicity score

After the evaluation of machine learning models, we computed the spamicity score of each appliance. This score indicates the trust worthiness and reliability of the device. It is defined using Eq. 2 as follows.

$$e[i] = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}} \quad (2)$$

$$S \leftarrow RMSE[i] * V_i \quad (3)$$

In the above equations, *e[i]* is the error rate computed with the predicted and actual arrays. *S* is the spamicity score, which is computed with the support of attribute importance score and error rate. The complete procedure of spamicity score computation is described in the algorithm 1. This algorithm is implemented in R, and the computed score is presented in Table V.

E. Complexity analysis

Complexity of the algorithm is evaluated by considering all the steps with their respective iterations.

Time Complexity: Steps 2 to 8 in this algorithm are the linear matrix formulation which takes O(n) time. In the worst case, the loop in steps 2-8, 9-11, and steps 13-15 take O(n) time. In steps 10, 12, and 14, the calculation takes O(1) time. Complexity of time (TC) is calculated as below:
=> TC= O(n)+O(n)+O(n)+O(1)+O(1)+O(1)
=> TC= O(n)
Space Complexity: In this algorithm, an input that does not exceed n is fixed and thus takes O(n) space. The loops take O(n) space as well. O(1) space is taken bythe arithmetic operations. Space complexity(SC)is measured as follows:
=> SC= O(n)+O(n)+O(1)
=> SC= O(n)

TABLE III: Results of Entropy-based filter

Feature	attr_importance
plugIdRef	0.76342
spaceIdRef	0.12322
manufacturer	0.23432
model	0.20345
Occupancy Type	0.10346
builtFormType	0.20998
wallAgeBand	0.43219
conditionType	0.76908
roomType	0.03076
wallType	0.38151
windowType	0.12602
fuelType	0.06642
meterType	0.47700
Heading	0.30532
Battery.Life	0.61396

TABLE IV: Summary of performance of the experimental models

Model	Precision	Recall	Accuracy	Score distribution
M1	0.650	1	79.81	Refer Fig. 5
M2	0.541	1	83.22	Refer Fig. 6
M3	0.567	1	84.35	Refer Fig. 7
M4	0.598	1	88.9	Refer Fig. 8
M5	0.513	1	91.8	Refer Fig. 9

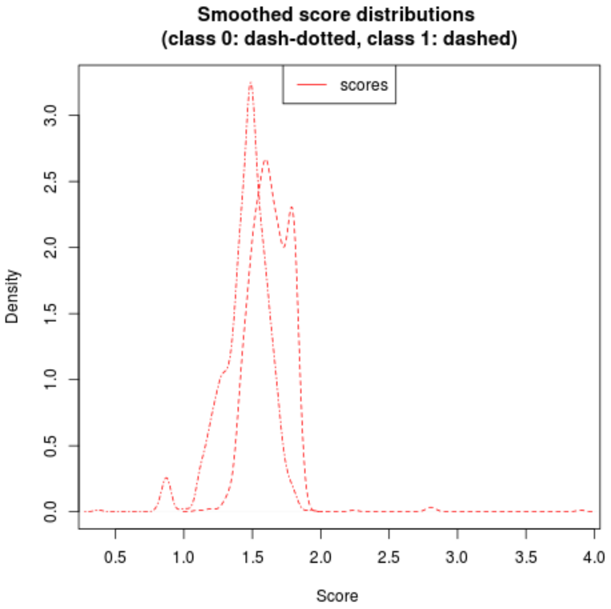

Fig. 5: Spam score distribution by Bagged Model

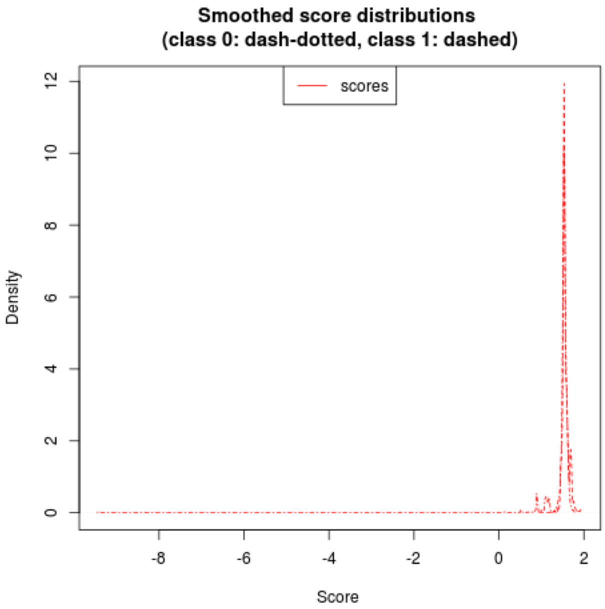

Fig. 6: Spam score distribution by Bayesian Generalized Linear Model

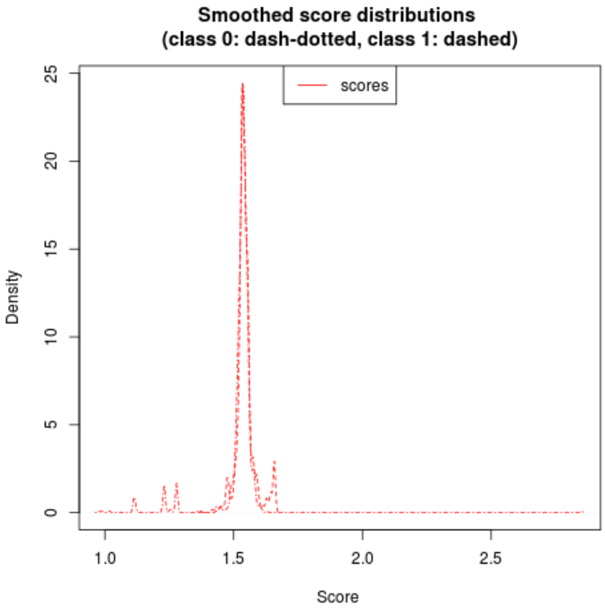

Fig. 7: Spam score distribution by Boosted Linear Model

TABLE V: Spamicity score of appliances

Appliance	Internet Connectivity	M1	M2	M3	M4	M5
Air filter	√	0.65	0.396	0.399	0.371	0.628
Alarm clock	×	0.348	0.580	0.947	0.637	0.2168
Alarm radio	×	0.246	0.607	0.686	0.633	0.175
Aquarium	×	0.671	0.709	0.143	0.878	0.489
Baby monitor	√	0.734	0.701	0.625	0.216	0.651
Bread maker	×	0.820	0.683	0.261	0.789	0.217
CD player	×	0.066	0.657	0.369	0.782	0.220
Chiller	√	0.045	0.635	0.466	0.732	0.213
Coffee grinder	×	0.081	0.283	0.046	0.074	0.020
Coffee maker	×	0.138	0.6150	0.312	0.210	0.562
DAB radio	×	0.092	0.234	0.554	0.773	0.208
Dehumidifier	×	0.160	0.106	0.608	0.761	0.223
Desktop PC	√	0.981	0.615	0.558	0.8188	0.274
Dishwasher	×	0.691	0.6090	0.542	0.16	0.230
Docking station	√	0.135	0.206	0.602	0.881	0.235
Doorbuster	×	0.186	0.613	0.631	0.905	0.228
DVD player/recorder	√	0.204	0.610	0.625	0.944	0.897
Electric blanket	×	0.244	0.009	0.648	0.008	0.219
Electric heater	×	0.012	0.006	0.011	0.012	0.220
Electric toothbrush charger	×	0	0	0	0	0
Exercise machine	×	0.341	0.211	0.132	0.429	0.227
Fairy lights	×	0.402	0.578	0.062	0.921	0.230
Games console	√	0.453	0.563	0.825	0.9620	0.240
George Forman grill	√	0.486	0.558	0.840	0.985	0.235
Guitar amplifier	√	0.477	0.558	0.795	0.928	0.229
Hair tongs	×	0.507	0.5548	0.840	0.470	0.2306
Hifi	√	0.556	0.548	0.865	0.938	0.838
iPad/iPod docking station	√	0.593	0.423	0.892	0.992	0.2319
Kitchenette	√	0.621	0.535	0.917	0.987	0.230
Laptop	√	0.633	0.534	0.925	0.964	0.928
Lava Lamp	×	0.617	0.538	0.227	0.285	0.224
Microwave	√	0.637	0.531	0.938	0.933	0.225
Oven	√	0.647	0.529	0.789	0.937	0.227
PC monitor	√	0.657	0.529	0.955	0.949	0.226
Printer	√	0.667	0.528	20.798	0.946	0.227
Projector	√	0.367	0.926	0.960	0.959	0.892
Radio	×	0.686	0.525	0.344	0.610	0.229
Raspberry Pi	√	0.686	0.243	0.966	0.973	0.886
Scanner	×	0.695	0.230	0.110	0.212	0.228
Router	√	0.523	0.975	0.974	0.874	0.751
Record player	√	0.963	0.981	0.977	0.911	0.2291
Set top box	√	0.177	0.473	0.735	0.754	0.7520
Sewing machine	×	0.542	0.509	0.199	0.921	0.221
Shredder	×	0.606	0.572	0.721	0.196	0.541
Tape player	×	0.231	0.806	0.738	0.701	0.684
Telephone	×	0.770	0.739	0.751	0.707	0.005
Television	√	0.718	0.751	0.743	0.712	0.779
Toaster	×	0.105	0.211	0.657	0.123	0.231
Washing machine	√	0.729	0.725	0.809	0.778	0.992

TABLE VI: Principal components being computed by PCA method for features

Feature	PC1	PC2	PC3	PC4	PC5 ——	PC15
1	4.255091e-08	6.764816e-05	1.145414e-06	-4.126413e-07	2.332671e-04	-1.612771e-12
2	1.257375e-04	-1.348555e-04	3.608422e-12	7.430535e-12	1.237237e-12	1.480848e-04
3	4.948566e-11	4.266645e-03	-1.223795e-12	1.007857e-02	9.111890e-12	4.042344e-05
4	7.535564e-04	4.896944e-02	1.090096e-02	9.787808e-03	1.816266e-01	4.702625e-02
5	2.637138e-01	4.681924e-02	9.005530e-13	5.998283e-01	6.321595e-02	-2.265900e-14
6	1.620736e-01	8.626930e-15	2.347263e-01	6.220893e-01	-1.063215e-01	4.663576e-16
7	6.879058e-01	2.180458e-01	-2.698411e-01	3.001963e-01	-4.495283e-15	5.822666e-01
8	-9.253025e-02	-6.857088e-01	2.269870e-03	6.833382e-01	1.559366e-04	-1.408902e-01
9	6.522830e-01	-1.762764e-03	6.512795e-01	4.664394e-02	-3.218220e-01	-1.127804e-15
10	-2.196190e-02	2.877072e-05	-2.021959e-15	-1.085136e-03	-1.139319e-05	4.868426e-05
11	3.189077e-12	2.859939e-05	1.615075e-04	-1.230201e-11	-7.347401e-05	1.216977e-11
12	6.950183e-05	3.858547e-12	1.745346e-04	1.637610e-02	1.778308e-10	1.681497e-13
13	8.558204e-10	-6.920804e-07	4.439540e-14	-8.191861e-06	2.146017e-12	-5.372825e-05
14	-2.058280e-15	-3.574847e-03	1.067373e-9	5.693648e-05	-4.831610e-02	-1.984294e-09
15	6.29293e-07	3.15414e-09	5.92394e-07	-1.23342e-07	-4.15506e-07	9.95639e-07

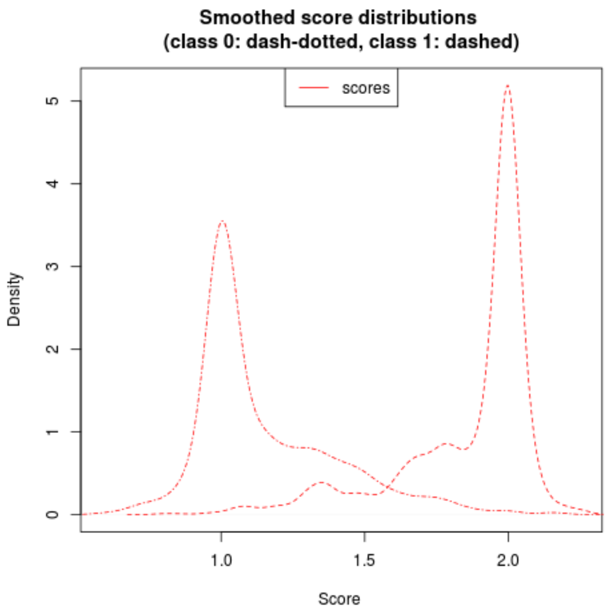

Fig. 8: Spam score distribution by eXtreme Gradient Boosting

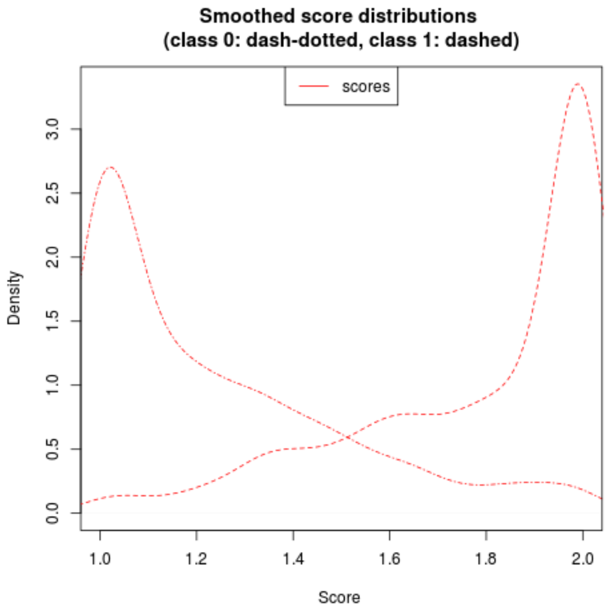

Fig. 9: Spam score distribution by Generalized Linear Model with Stepwise Feature Selection

IV. Results and discussion

The proposed approach detects the spam parameters causing the IoT devices to be effected. To get the best results, the IoT dataset is used for the validation of proposed approach as described in the next Section.

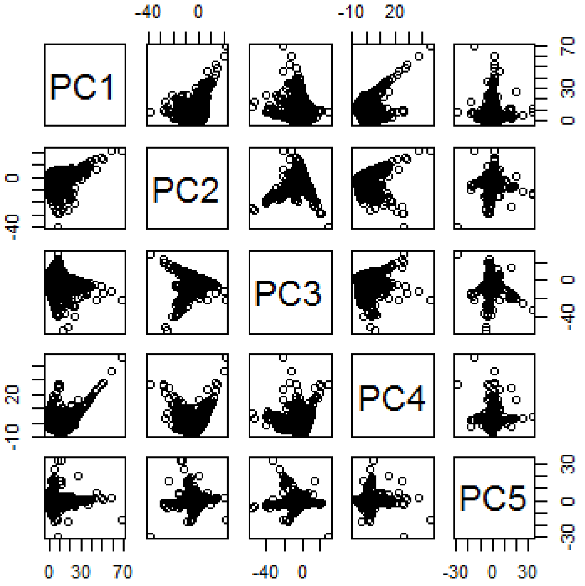

Fig. 10: Transformations of Principal Components

A. Data Collection

We have collected the smart home dataset by REFIT project [20] which is sponsored by Loughborough University. A total of twenty homes were used and advised to deploy the smart home technologies. The complete survey was conducted by the team of researchers. The experiments are varied from room to room, depending upon climate changes, floor plans, Internet supply and other attributes as shown in Fig. 11. The internal environmental conditions were captured using different sensors. There were more than 100,000 data points in each home for sensor monitoring. The survey was continued for almost 18 months. This dataset is openly available at [20].

B. Experimental setup

To perform the experiments, we use the data set traces from the source as mentioned [20]. Then, we performed the experiments on RStudio (openly free software available at [21]). The software requirements are, Operating system: Windows 7/8/10 or MacOS 10.12+ or Ubuntu 14/16/18 or Debian 8/10. Following are the results obtained.

C. Impact of data preprocessing on SDI-UML

The preprocessing involves the selection of appliances being considered for the detection of spam parameters. The main idea is to find the various spam causing factors. Firstly, the feature reduction is done. The method used for feature reduction is the Principal Component Analysis (PCA), which reduces the dimensions of data. It results in series of Principal components (PC) which corresponds to each row with each column. In the IoT dataset used in this proposal, we have 15 features, so 15 PCs are generated as shown in Table VI. The pca() works in such a way that it reduces the variance among the features. The standard deviations of PCs is presented in Fig. 4 and the transformations of PCs is presented in Fig. 10.

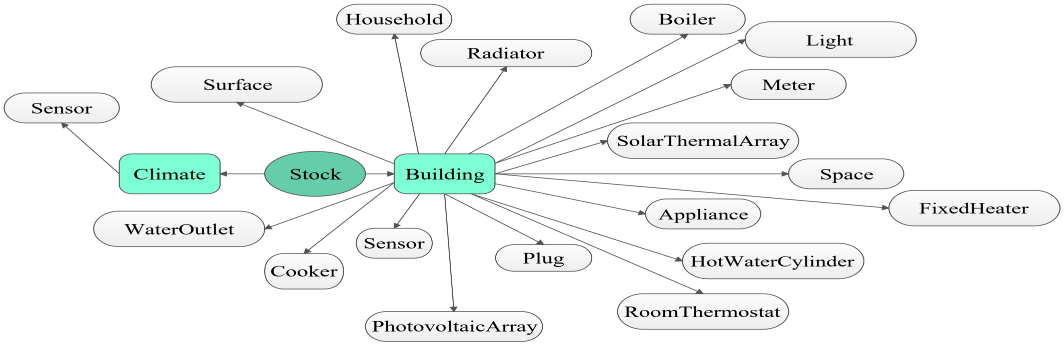

Fig. 11: Features of Smart Home dataset

After the feature extraction, the feature selection is performed. The features along with their importance score computed by entropy based filter is presented in Table III. This algorithm uses the correlation among the discrete attributes with continuous attributes to find out the weights of discrete attributes. There are three functions using this entropy based filter namely, information.gain, gain.ratio, symmetrical.uncertainty.

D. Impact of machine learning models on SDI-UML

The dataset is trained with five different machine learning models with the features mentioned in Table III. Each model produces a spamicity score of each appliance which indicates the probability of appliance to be effected by spam. Table IV provides the summary of performance of all the five machine learning models, being used for experiments. Table V illustrates the selected appliances, each with their spamicity scores. The distribution of these spamicity score by the various models is presented in Fig 5, 6, 7, 8, 9. The evaluation is done to compute the accuracy, precision, and recall.

V. Conclusion

The proposed framework, detects the spam parameters of IoT devices using machine learning models. The IoT dataset used for experiments, is pre-processed by using feature engineering procedure. By experimenting the framework with machine learning models, each IoT appliance is awarded with a spam score. This refines the conditions to be taken for successful working of IoT devices in a smart home. In future, we are planning to consider the climatic and surrounding features of IoT device to make them more secure and trustworthy.

References

[1] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, “Iot security: ongoing challenges and research opportunities,” in *2014 IEEE 7th international conference on service-oriented computing and applications*. IEEE, 2014, pp. 230–234.

[2] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “Blockchain for iot security and privacy: The case study of a smart home,” in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*. IEEE, 2017, pp. 618–623.

[3] E. Bertino and N. Islam, “Botnets and internet of things security,” *Computer*, no. 2, pp. 76–79, 2017.

[4] C. Zhang and R. Green, “Communication security in internet of thing: preventive measure and avoid ddos attack over iot network,” in *Proceedings of the 18th Symposium on Communications & Networking*. Society for Computer Simulation International, 2015, pp. 8–15.

[5] W. Kim, O.-R. Jeong, C. Kim, and J. So, “The dark side of the internet: Attacks, costs and responses,” *Information systems*, vol. 36, no. 3, pp. 675–705, 2011.

[6] H. Eun, H. Lee, and H. Oh, “Conditional privacy preserving security protocol for nfc applications,” *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 153–160, 2013.

[7] R. V. Kulkarni and G. K. Venayagamoorthy, “Neural network based secure media access control protocol for wireless sensor networks,” in *2009 International Joint Conference on Neural Networks*. IEEE, 2009, pp. 1680–1687.

[8] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, “Machine learning in wireless sensor networks: Algorithms, strategies, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[9] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[10] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, “Evaluation of machine learning classifiers for mobile malware detection,” *Soft Computing*, vol. 20, no. 1, pp. 343–357, 2016.

[11] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, “A system for denial-of-service attack detection based on multivariate correlation analysis,” *IEEE transactions on parallel and distributed systems*, vol. 25, no. 2, pp. 447–456, 2013.

[12] Y. Li, D. E. Quevedo, S. Dey, and L. Shi, “Sinr-based dos attack on remote state estimation: A game-theoretic approach,” *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 632–642, 2016.

[13] L. Xiao, Y. Li, X. Huang, and X. Du, “Cloud-based malware detection game for mobile devices with offloading,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2742–2750, 2017.

[14] J. W. Branch, C. Giannella, B. Szymanski, R. Wolff, and H. Kargupta, “In-network outlier detection in wireless sensor networks,” *Knowledge and information systems*, vol. 34, no. 1, pp. 23–54, 2013.

[15] I. Jolliffe, *Principal component analysis*. Springer, 2011.

[16] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[17] L. Yu and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 856–863.

[18] A. H. Sodhro, S. Pirbhulal, and V. H. C. de Albuquerque, “Artificial intelligence driven mechanism for edge computing based industrial applications,” *IEEE Transactions on Industrial Informatics*, 2019.

[19] A. H. Sodhro, Z. Luo, G. H. Sodhro, M. Muzamal, J. J. Rodrigues, and V. H. C. de Albuquerque, “Artificial intelligence based qos optimization for multimedia communication in iov systems,” *Future Generation Computer Systems*, vol. 95, pp. 667–680, 2019.

[20] L. University, “Refit smart home dataset,” https://repository.lboro.ac.uk/articles/REFIT_Smart_Home_dataset/2070091, 2019 (accessed April 26, 2019).

[21] R, “Rstudio,” 2019 (accessed October 23, 2019).