Internet of Things 7 (2019) 100059

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches

Mahmudul Hasan*, Md. Milon Islam, Md Ishrak Islam Zarif, M.M.A. Hashem

Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

ARTICLE INFO

Article history:
Received 27 January 2019
Revised 9 May 2019
Accepted 11 May 2019
Available online 20 May 2019

Keywords:
Internet of Things (IoT)
Machine Learning
Cybersecurity
Anomaly detection

ABSTRACT

Attack and anomaly detection in the Internet of Things (IoT) infrastructure is a rising concern in the domain of IoT. With the increased use of IoT infrastructure in every domain, threats and attacks in these infrastructures are also growing commensurately. Denial of Service, Data Type Probing, Malicious Control, Malicious Operation, Scan, Spying and Wrong Setup are such attacks and anomalies which can cause an IoT system failure. In this paper, performances of several machine learning models have been compared to predict attacks and anomalies on the IoT systems accurately. The machine learning (ML) algorithms that have been used here are Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Artificial Neural Network (ANN). The evaluation metrics used in the comparison of performance are accuracy, precision, recall, f1 score, and area under the Receiver Operating Characteristic Curve. The system obtained 99.4% test accuracy for Decision Tree, Random Forest, and ANN. Though these techniques have the same accuracy, other metrics prove that Random Forest performs comparatively better.

1. Introduction

With the increasing demand and growth in the Internet of Things (IoT) automated network system, the IoT models are getting complicated day by day [1,2]. People are being accustomed to data-driven infrastructure, and this is leading the research more on to Machine Learning based applications alongside IoT. IoT and Machine Learning based techniques are used in every domain of human life at present. In medicine, interpretation of ECG, disease detection using X-Ray, pattern finding in genomic data, an automated pathological system for cancer detection, brain signal modeling all these complex tasks requires the introduction of machine learning approaches [3]. The application of machine learning approaches can also cover the aerospace domain. D'Angelo et al. [4] applied content-based image retrieval technique and machine learning techniques on electrical impedance plane generated from eddy current testing. Eddy current testing is a complex task used in aircraft industries for finding out defects. Besides machine learning, IoT services are also applied to these domains. The growing complexity in IoT infrastructures is raising unwanted vulnerability to their systems. In IoT devices security breach and anomaly has become common phenomena nowadays.

* Corresponding author.
E-mail addresses: mahmudul@cse.kuet.ac.bd (M. Hasan), milonislam@cse.kuet.ac.bd (Md. M. Islam), ishrak.islam@gmail.com (Md. I.I. Zarif), hashem@cse.kuet.ac.bd (M.M.A. Hashem).

https://doi.org/10.1016/j.iot.2019.100059

IoT devices use a wireless medium to broadcast data which makes them an easier target for an attack [5]. Normal communication attack in the local network is limited to local nodes or small local domain, but attack in IoT system expands over a larger area and has devastating effects on IoT sites [6].

Thenceforth, a secured IoT infrastructure is necessary for the protection from cybercrimes. The security measures that have been used become vulnerable with the vulnerability of IoT devices. For some stakeholders and entrepreneurs, data is the money for their business. For the government and some private agency, some data are classified and confidential. Vulnerability in IoT nodes makes a backdoor for an attacker to gather confidential data from any important organization [7].

There are some trivial methods to solve the problems as mentioned above. In signature based [8] method, attacks and anomaly are previously stored in a database. Moreover, this system is checked at particular time intervals against the database. However, this methodology generates overhead in processing, and it is vulnerable to unknown threats. The advantage of data analysis based technique is that it works faster than other methodologies and it can overcome the problem raised from unknown threats. Hence, in this paper data analysis based techniques are used.

The primary goal of the system is to develop a smart, secured and reliable IoT based infrastructure which can detect its vulnerability, have a secure firewall against all cyber attacks and recover itself automatically. Here, Machine Learning based solution is proposed which can detect and protect the system when it is in the abnormal state. For this task, several machine learning classifiers have been exploited. Another key aspect of this paper is that making the realization of the fact that a simple model like Decision Tree or Random Forest can be compared with a complex network like ANN for anomaly detection.

Further analysis and comparison with other works will be briefly described in the following sections. Section 2 provides a description of other research works on IoT attack and anomaly detection. The description of the dataset, different kinds of attacks and anomalies, learning models and system frameworks are detailed in Section 3. Section 4 explains our experimental setup, result from analysis and comparison with other state of the art methods. Finally, limitation, conclusion and future scopes are presented in Section 5.

2. Literature review

There have been several similar works done in IoT fields. Still, researchers are working in this area. Pahl et al. [1] have mainly developed a detector and firewall for an anomaly of IoT microservices in IoT site. Clustering methods like K-Means and BIRCH have been implemented [9] for different microservices in this work. In clustering, different clusters were grouped in the same if the center is in the three times of standard deviation distance. The clustering model has been updated using an online learning technique. With the algorithms implemented, the overall accuracy obtained by the system is 96.3%.

A detailed description of a smart home system where security breaches were detected by deep learning method Dense Random Neural Network (DRNN) [10] have been introduced in [11]. They have mainly described Denial of Service attack and Denial of Sleep attack in a simple IoT site.

Liu et al. [5] proposed a detector for On and Off attack by a malicious network node in industrial IoT site. By On and Off attack they meant that IoT network could be attacked by a malicious node when it is in an active state or On state. Furthermore, the IoT network behaves normal when its malicious node is in the inactive or off state. The system was developed using a light probe routing mechanism with the calculation of trust estimation of each neighbor node for the detection of an anomaly.

Diro et al. [8] discussed the detection of attack using fog-to-things architecture. The authors of the paper gave a comparison study of a deep and a shallow neural network using open source dataset. This work's primary focus was to detect four classes of attack and anomaly. For four class the system got the accuracy of 98.27% for deep neural network model and accuracy of 96.75% for shallow neural network model.

Usmonov et al. [12] described the recent security problem when developing embedded technologies for the IoT. Preserving data transfer between the physical, logical and virtual components of the IoT system was also challenging. For these problems, the use of digital watermarks was proposed by the authors of this work.

Anthi et al. [13] represented an intrusion detection system for the IoT. Toward this purpose, several ML classifiers have been used for successfully identifying network scanning probing and simple forms of Denial of Service (DoS) attacks. To generate the data set, network traffic is taken for four subsequent days using the software Wireshark. For applying ML classifiers, the software Weka was used.

Ukil et al. [14] discussed the detection of anomalies in healthcare analytics based on IoT. A model of cardiac anomaly detection through a smartphone was also introduced in this paper. For the anomaly detection in healthcare; IoT sensors, medical image analysis, biomedical signal analysis, big data mining, and predictive analytics were used.

Pajouh et al. [6] presented a model for intrusion detection based on a two-layer dimension reduction and two-tier classification module. This model was also designed to identify malicious activities such as User to Root (U2R) and Remote to Local (R2L) attacks. For dimension reduction, component analysis and linear discriminate analysis have been used. NSL-KDD dataset was used to carry out the whole experiment. For detecting suspicious behaviors with the two-tier classification module, Naive Bayes and Certainty Factor version of K-Nearest Neighbor were applied.

D'Angelo et al. [15] applied Uncertainty-managing Batch Relevance-based Artificial Intelligence (U-BRAIN) on binary NSL-KDD dataset and Real Traffic Data (from Fredrico II University of Napoli). The U-Brain is a dynamic model operated on multiple machines which can handle missing data. The NSL-KDD dataset contains 41 features. From 41 features 6 features

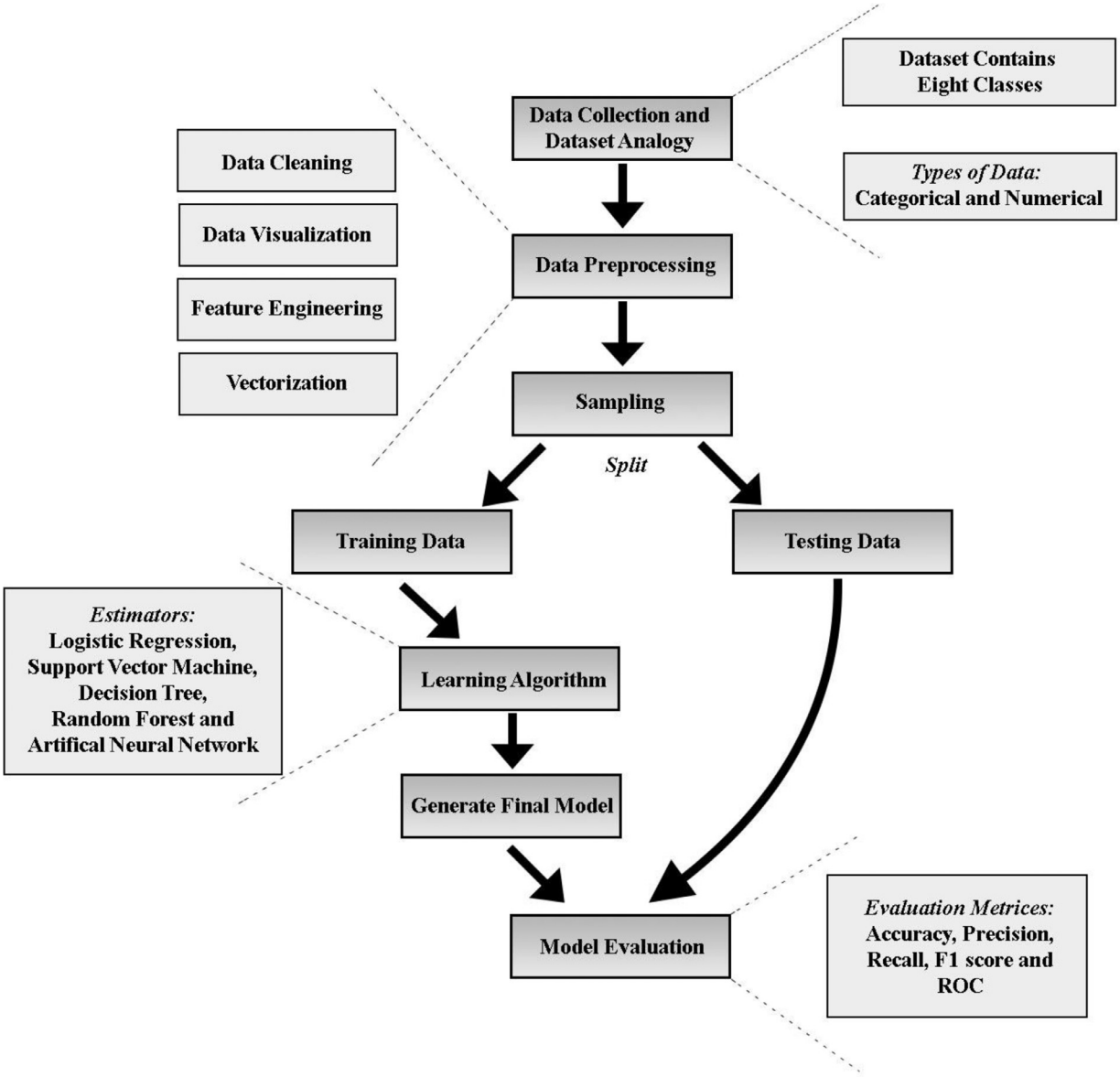

Fig. 1. Overall framework for attack and anomaly detection in IoT.

were selected using J-48 based classification algorithm. The accuracy values were 94.1% for NSL-KDD and 97.4% (10-fold training mean) for Real Traffic Data.

Kozik et al. [16] presented classification based attack detection service utilizing cloud architecture. An Extreme Learning Machines (ELM), scalable in Apache Spark cloud framework is employed on the artificial Netflow formatted data in this paper. An IoT network yielded these Netflow formatted data. The work is focused on three significant scenarios in IoT systems: scanning, command and control and infected host. Best accuracy values were found for these scenarios are 0.99, 0.76 and 0.95, respectively.

3. Materials and methods

The overall framework is a combination of several independent processes. Fig. 1 depicts the overall framework of the system. The first process of this framework is the dataset collection and dataset observation. In this process, the dataset was collected and observed meticulously to find out the types of data. Besides, data preprocessing was implemented on the dataset. Data preprocessing consists of cleaning of data, visualization of data, feature engineering and vectorization steps. These steps converted the data into feature vectors. These feature vectors were then split into 80–20 ratio into training

Table 1
Frequency distribution of considered attacks.

Attacks	Frequency Count	% of Total Data	% of Anomalous Data
Denial of Service	5780	01.61%	57.70%
Data Type Probing	342	00.09%	03.41%
Malicious Control	889	00.24%	08.87%
Malicious Operation	805	00.22%	08.03%
Scan	1547	00.43%	15.44%
Spying	532	00.14%	05.31%
Wrong Setup	122	00.03%	01.21%

Table 2
Features description.

SL	Features	Data type	Example
1	Source ID	Nominal	WashingMachine1, Battery2 etc.
2	Source Address	Nominal	\agent6\washingmachine1 etc.
3	Source Type	Nominal	Illustrated in Fig. 2(a)
4	Source Location	Nominal	Illustrated in Fig. 2(c)
5	Destination Service Address	Nominal	\agent8\lightcontroller12 etc.
6	Destination Service Type	Nominal	Illustrated in Fig. 2(b)
7	Destination Location	Nominal	Illustrated in Fig. 2(d)
8	Accessed Node Address	Nominal	\agent13\doorLock9 etc.
9	Accessed Node Type	Nominal	Illustrated in Fig. 2(e)
10	Operation	Nominal	Illustrated in Fig. 2(f)
11	Value	Continuous	0, 5.32, 19.03 etc.
12	Timestamp	Discrete	1520031600000 to 1520117999000
13	Normality	Nominal	Described in Table 1

and testing set. The training set was used in Learning Algorithm, and a final model was developed using an optimization technique. Different classifiers used in this work employed different optimization techniques. Logistic Regression used coordinate descent [17]. SVM and ANN used conventional gradient descent technique. The optimizer is not used in the case of DT and RF because these are non-parametric models. The final model was evaluated against the testing set using different evaluation metrics.

3.1. Dataset collection and description

The open source dataset was collected from kaggle [18] provided by Pahl et al. [1]. They have created a virtual IoT environment using Distributed Smart Space Orchestration System (DS2OS) for producing synthetic data. Their architecture is a collection of micro-services which communicates with each other using the Message Queuing Telemetry Transport (MQTT) protocol. 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. Features “Accessed Node Type” and “Value” have 148 and 2050 missing data, respectively.

Table 1 gives a detailed picture of the distribution of different attacks and anomaly through the whole data. Descriptions of 13 features are given in Table 2. All of the features in the described dataset are object type except the timestamp which is an int64 type. Besides these the frequency distribution of several features is illustrated in Fig. 2.

1. Denial of Service (DoS): the DoS attack is caused by having too many unwanted traffic in a single source or receiver. The attacker sends too many ambiguous packets to flood out the target and make its services unavailable to other services [11]. In the dataset, 5780 samples are containing a DoS attack.
2. Data Type Probing (D.P): in this case, a malicious node writes different data type than intended data type [1]. In the dataset, there are 342 samples of Data Type Probing.
3. Malicious Control (M.C): with software vulnerabilities sometimes the attacker can gain a valid session key or somehow capture network traffic. In this way, malicious one can control the whole system [19,20]. The Dataset contains 889 samples of Malicious Control.
4. Malicious Operation (M.O): Malicious Operations are generally caused by malware. Malware means decoy activity which distracts the original operation. Device’s performances can negatively be affected by this malicious operation [21]. 805 samples of the dataset contain Malicious Operation.
5. Scan(SC): sometimes the data is acquired through hardware by scanning the system, and in this process sometimes the data can get corrupted [22]. In the dataset, 1547 samples were containing Scan.
6. Spying (SP): by Spying, the attacker exploits the vulnerabilities of the system, and they use a backdoor channel to break into the system and discovers important information [8]. In some cases, they manipulate data causing great hamper to the whole system. The dataset contains 532 samples of Spying.

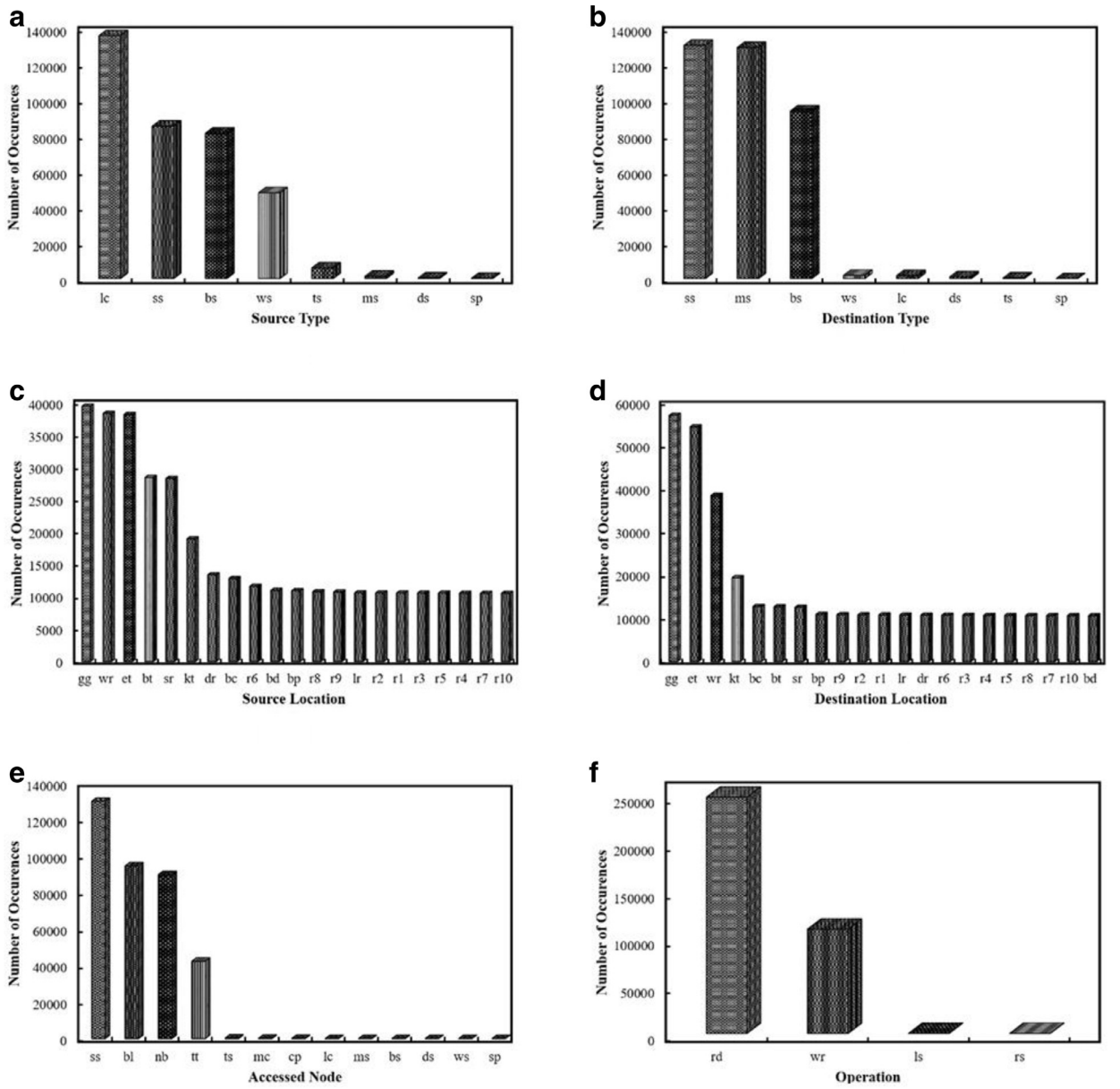

Fig. 2. Frequencies of (a) Source Type (b) Destination Type (c) Source Location (d) Destination Location (e) Accessed Node (f) Operation [lc = Light Controller, ss = Sensor Service, bs = Battery Service, ws = Washing Service, ts = Thermostat, ms = Movement Sensor, ds = Door Lock Service, sp = Smart Phone, gg = Garage, bp = Bedroom Parents, et = Entrance, lr = Living Room, sr = Showerroom, kt = Kitchen, dr = Dining Room, wr = Water Room, bd = Bedroom, bt = Bathroom, bc = Bedroom Children, r1–r10 = room1 to room10, tt = Text, nb = Number, bl = Boolean, mc = Malicious, cp = Composed, rd = Read, rs = Register Service, wr = Write, ls = Lock Sub Tree].

7. Wrong Setup (W.S): the data may also get disrupted by the wrong system setup [23]. The dataset contains 122 samples of the Wrong Setup.
8. Normal(NL): if the data is entirely correct and accurate, then the data is called normal data. Out of the 357,952 samples, 347,935 samples are of a normal class.

3.2. Data preprocessing

Any machine learning research requires exploratory data analysis and data observation. The first task in this research was to make the dataset feed-able to any classifier. So for this reason, the first step was to handle the missing data. In the dataset, “Accessed Node Type” column and “Value” column contain missing values due to anomaly raised in data transferring. From

these two features, “Accessed Node Type” feature has categorical values on the other hand “Value” feature has continuous values. “Accessed Node Type” feature has 148 rows containing ‘NaN’ value depicted as Not a Number, and the corresponding class or label of that row are found to be anomalous. As the “Accessed Node Type” feature is categorical and removing these 148 rows might cause loss of valuable data, so the ‘NaN’ value in “Accessed Node Type” is replaced with the ‘Malicious’ value. Similarly, “Value” column also contains some unexpected data which are not continuous values. These unexpected values are transformed into meaningful continuous values that assist the classifiers to have better accuracy. Unexpected values “False”, “True”, “Twenty” and “none” in the “Value” feature are replaced by meaningful values “0.0”, “1.0”, “20.0” and “0.0”, respectively.

For feature selection, no machine learning approach has been taken here like Pahl et al. [1] because this will not have any significant impact on data analysis. Besides this timestamp column from the dataset has been removed as it has a minimal correlation to the dataset’s predictor variable normality.

In feature engineering steps, it is necessary to determine the features type in the dataset at first. The dataset contains Categorical and Numerical data. Categorical Data can be further classified into Ordinal and Nominal Values respectively while Numerical dataset into Discrete and Continuous Values. Table 2 depicts the column types. So from Table 2, it can be claimed that all columns except “Value” column and “Timestamp” column are categorical nominal variable. Moreover, “Value” column and “Timestamp” column are continuous numerical variable. The “Timestamp” column is not considered here as it was removed from the dataset.

Next vital task is to converting nominal categorical data into vectors. Categorical data can be converted into vectors in many ways. Label Encoding and One Hot Encoding is prevalent among them. In this research label encoding technique have been used to convert the data into a feature vector. Most of the features in this dataset contain nominal categorical value and many unique values. If one hot encoding were applied to these features, the number of features would have increased with a significant number, and the resulting dataset would have lots of dimensions. On the other case, by label encoding, the number of features were the same. Thus the dimension of the dataset was not increased. Besides these, one hot encoded features would have sparse features which are harder to fit in machine learning algorithm and takes a lot of processing time. Hence, label encoding is applied to the dataset.

3.3. Theoretical considerations

For data analysis part, several machine learning algorithms were used. Following are the lists of algorithms with their description.

3.3.1. Logistic Regression (LR)

Logistic Regression (LR) is a discriminative model which depends on the quality of the dataset. Given the features $X = X_1, X_2, X_3, \ldots, X_n$ ($where, X_1 - X_n = Distinct features$), weights $W = W_1, W_2, W_3, \ldots, W_n$, bias $b = b_1, b_2, \ldots, b_n$ and Classes $C = c_1, c_2, \ldots, c_n$ (in our case, we have eight classes) the equation for estimation of posterior is given in following (1) [24].

$$\text{Predicted Value: } p(y = C|X;W,b) = \frac{1}{1 + exp(-W^{transpose}X - b)} \tag{1}$$

3.3.2. Support Vector Machine (SVM)

Support Vector Machine is another discriminative model like LR. It is a supervised learning model for analyzing the data used for classification, regression, and outliers detection [25,26]. SVM is most applicable in the case of Non-Linear data. Given Input x, Class or Label c and LaGrange multipliers α; weight vector Θ can be calculated by following equation:

$$\Theta = \sum_{i=1}^{m} \alpha_i c_i x_i \tag{2}$$

The target of the SVM is to optimize the following equation:

$$\text{Maximize}_{\alpha_i} \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j c_i c_j < x_i x_j > \tag{3}$$

In Eq. (3), $< x_i, x_j >$ is a vector which can be obtained by different kernels like polynomial kernel, Radial Basis Function kernel and Sigmoid Kernel [27].

3.3.3. Decision Tree (DT)

Decision Tree allows each node to weigh possible actions against one another based on their benefits, costs, and probabilities. Overall, it is a map of the possible outcomes of a series of related choices [28]. A DT generally starts with a single node and then it branches into possible outcomes. Each of these outcomes lead to additional nodes, which branch off into other instances. So from there, it became a tree-like shape; in other words, a flowchart-like structure. Considering a binary tree Fig. 3 where a parent node is split into two children node a left child and a right child. Parent node, left child and right child contains data P_d, LC_d, RC_d, respectively [29]. Given, features x, impurity measure I(data), the number of samples

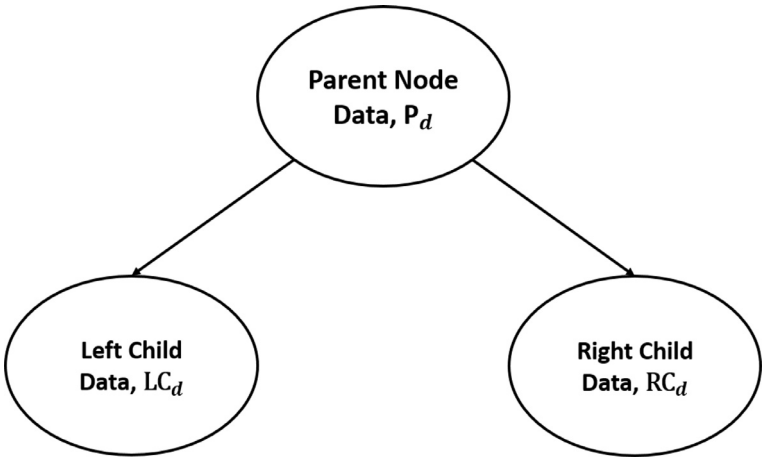

Fig. 3. Decision Tree Splitting.

in parent node P_n, the number of samples in left child LC_n and the number of samples in right child RC_n; DT's target is to maximize following Information Gain in Eq. (4).

$$\text{Information Gain}(P_d,x) = I(P_d) - \frac{LC_n}{P_n} I(LC_d) - \frac{RC_n}{P_n} I(RC_d) \tag{4}$$

Impurity Measure I(data) can be calculated in three techniques Gini Index I_G, Entropy I_H and Classification Error I_E. Following Eqs. (5)–(7) shows the calculation of different Impurity Measures.

$$I_H(n) = -\sum_{i=1}^{c} p(c|n) log_2 p(c|n) \tag{5}$$

$$I_G(n) = 1 - \sum_{i=1}^{c} p(c|n)^2 \tag{6}$$

$$I_E(n) = 1 - max\{p(c|n)\} \tag{7}$$

where, c denotes classes or labels, n denotes any node and $p(c|n)$ denotes the ratio of c with respect to n.

3.3.4. Random Forest (RF)

As the name implies, the random forest algorithm creates the forest with many decision trees. It is a supervised classification algorithm. It is an attractive classifier due to the high execution speed [30]. Many decision trees ensemble together to form a random forest, and it predicts by averaging the predictions of each component tree. It usually has much better predictive accuracy than a single decision tree. In general, the more trees in the forest the more robust the forest looks.

3.3.5. Artificial Neural Network (ANN)

Artificial Neural Network (ANN) is a machine learning technique which is the skeleton of different deep learning algorithms. We can train the ANN model using raw data. Compared to other classifiers it has a large number of parameters for tuning which makes it a complex structure. It also takes a long time to optimize error than other techniques. For this reason, Neural Network algorithm instances are trained in Graphics Processing Unit using CUDA programming. Each single Neuron Node of ANN is trained with feature set $X = X_1, X_2, X_3, \ldots, X_n$ ($where, X_1 - X_n = Distinct\ features$). The features are multiplied by some random weights, $W = W_1, W_2, W_3, \ldots, W_n$ and added with bias values, $b = b_1, b_2, \ldots, b_n$. The values are then given as input in non-linear activation function [8]. Activation functions can be of several types. Following Eqs. (8)–(11) are some activation functions. In the equations, (i) means a single sample.

$$\text{Sigmoid Function: } \sigma(z) \text{ or } a(z) = \frac{1}{1+e^{-z}} \tag{8}$$

$$\text{Tanh Function: } a(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{9}$$

$$\text{Rectified Linear Unit (RELU): } a(z) = max(0, z) \tag{10}$$

$$\text{Leaky RELU: } a(z) = max(0.001 * z, z) \tag{11}$$

After applying Non-Linear function, a softmax function is applied to get initial predicted value which is shown in Eq. (12).

$$\text{Predicted Value: } \hat{y}^{(i)} = \sigma\left(W^{transpose} X^{(i)} + b\right) \tag{12}$$

Lastly from the true value and the predicted value, the loss function is calculated and weights of the whole neural network architecture is modified using the backpropagation technique, gradient descent and error got from the loss function. The equation of loss function is given in the following equation:

$$L\left(\hat{y}^{(i)}, y^{(i)}\right) = -\left(y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)\right) \tag{13}$$

3.4. Evaluation criteria

The following metrics were calculated for evaluating the performance of the developed system. Using these metrics, one can decide which technique is best suited for this work.

3.4.1. Confusion matrix

The confusion matrix is used to visualize the performance of a technique. It is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It allows for easy identification of confusion between classes. Most of the time, almost all performance measures are computed from it. A confusion matrix is a summary of prediction results on a classification problem [31]. A definition of True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) for multiple classes can be given from confusion matrix. Let C_i be any class out of the eight classes. Following are the definitions of TP, FP, FN, and TN for C_i:

- TP(C_i) = All the instances of C_i that are classified as C_i.
- FP(C_i) = All the non C_i instances that are classified as C_i.
- FN(C_i) = All the C_i instances that are not classified as C_i.
- TN(C_i) = All the non C_i instances that are not classified as C_i.

3.4.2. Accuracy

A model's accuracy is only a subset of the model's performance. Accuracy is one of the metrics for evaluating classification models. Eq. (14) depicts single class accuracy measurement.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \tag{14}$$

3.4.3. Precision

Precision means the positive predictive value. It is a measure of the number of true positives the model claims compared to the number of positives it claims. The precision value for a single class is given in the following equation:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{15}$$

3.4.4. Recall

The recall is known as the actual positive rate which means the number of positives in the model claims compared to the actual number of positives there are throughout the data. The recall value for a single class is given in the following equation:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{16}$$

3.4.5. F1 score

The F1 score can also measure a model's performance. It is a weighted average of the precision and recall of a model. The F1 Score value for a single class is given in Eq. (17).

$$\text{F1 Score} = \frac{2 * \text{True Positive}}{2 * \text{True Positive} + \text{False Positive} + \text{False Negative}} \tag{17}$$

3.4.6. Receiver operating characteristic curve

It is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate against the False Positive Rate as the value of the threshold is varied for assigning observations to a given class [31]. The calculation of True Positive Rate and False Positive Rate are given in the following equations:

Table 3
Evaluation metrics of our study.

Evaluation		Classifiers				
Metrics		LR	SVM	DT	RF	ANN
Training	Accuracy	0.983	0.982	0.994	0.994	0.994
	STD(+/-)	0.0012	0.0015	0.00081	0.00081	0.0013
	Precision	0.98	0.98	0.99	0.99	0.99
	Recall	0.98	0.98	0.99	0.99	0.99
	F1 Score	0.98	0.98	0.99	0.99	0.99
Testing	Accuracy	0.983	0.982	0.994	0.994	0.994
	STD(+/-)	0.0055	0.0064	0.016	0.014	0.021
	Precision	0.98	0.98	0.99	0.99	0.99
	Recall	0.98	0.98	0.99	0.99	0.99
	F1 Score	0.98	0.98	0.99	0.99	0.99

$$\text{False Positive Rate} = \frac{\text{Number of False Positive Samples}}{\text{Total Number of Samples}} \tag{18}$$

$$\text{True Positive Rate} = \text{Recall} = \frac{\text{Number of True Positive Samples}}{\text{Total Number of Samples}} \tag{19}$$

The threshold value is the probability value for each predicted class. The ROC curve can be drawn using binary classes. However, using one vs. rest method, it can be extended for multiple classes. The values of the true positive rate and false positive rate for each class ranges from 0 to 1.

4. Implementation and result analysis

4.1. Experimental setup

The experiment was done using HP (EliteDesk 800 G3 TWR) desktop where the operating system was Windows 10 Enterprise 64-bit (10.0, Build 17134) (17134.rs4_release.180410–1804), Processor was Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz (8 CPUs), 3.6GHz. The memory of the desktop was 16.384 GB RAM. NVIDIA GeForce GTX 1070 graphics card was used for running the program. For data analogy, cleaning and feature engineering, Pandas framework and Numpy framework; for data visualization, Matplotlib framework and Seaborn framework and lastly for data analysis, scikit-learn framework and Keras framework were used.

4.2. Result analysis

In the Data Analysis subsection, it has been described that several machine learning techniques were applied to the dataset. Five-fold cross-validation was performed on the dataset using each of these techniques. Fig. 4(a) and (b) shows how the accuracy results are converged after five-fold cross-validation. From the cross-validation, it can be inferred that RF and ANN have performed best both in training and testing accuracy. DT performed with approximate similarity to RF and ANN in the case of training. In the case of testing, the DT had most deviations than other techniques and performed poorly at first. However in the last three folds, it performed similarly to RF and ANN. SVM and LR performed weakly than other techniques in training. In the case of testing and in the first two fold, SVM and LR both performed better than other techniques and logistic regression was best among them, but at the last three folds, they performed worse than others. Table 3 represents different evaluation metrics for different techniques trained on the dataset. From Table 3, it can be seen that DT and RF have more accuracy, precision, recall, and F1 score values than other techniques. ANN also performed well in the case of evaluation. However, DT and RF are a little more accurate than ANN. On the other case, LR and SVM also do well on our dataset but not as good as other classifiers.

Now considering the confusion matrices of each technique, the most optimized technique can be found. From the confusion matrices in Fig. 5 it can be concluded that RF is the best technique for this work. RF classified every class correctly except DoS and Normality classes. Out of 1178 samples of DoS, it misclassified 403 samples as Normal. Moreover, for Normal class, it misclassified 18 samples as DoS out of 69,571 samples. Confusion matrix for DT is similar to RF except for the Normal class. In the case of Normal class, it misclassified 18 samples as DoS and two samples as spying out of 69,571 samples. ANN performed similarly to DT. It only misclassified one more sample than DT. ANN correctly predicted every sample of 6 labels out of 8 labels. In the case of DoS, ANN misclassified 403 samples as Normal out of 1178 samples. While for the Normal Label, ANN misclassified 18 samples as DoS, two samples as Spying and 1 sample as Malicious Control out of 69,571 samples. LR and SVM performed very poorly in this system. Fig. 5 shows the confusion matrix of LR. LR correctly classified 775 samples of DoS class but misclassified all the remaining 403 samples as Normal class. For Data Probing, it misclassified 63 samples as Normal out of 63 samples, for Malicious Control it misclassified 159 samples as Normal out of 169 samples,

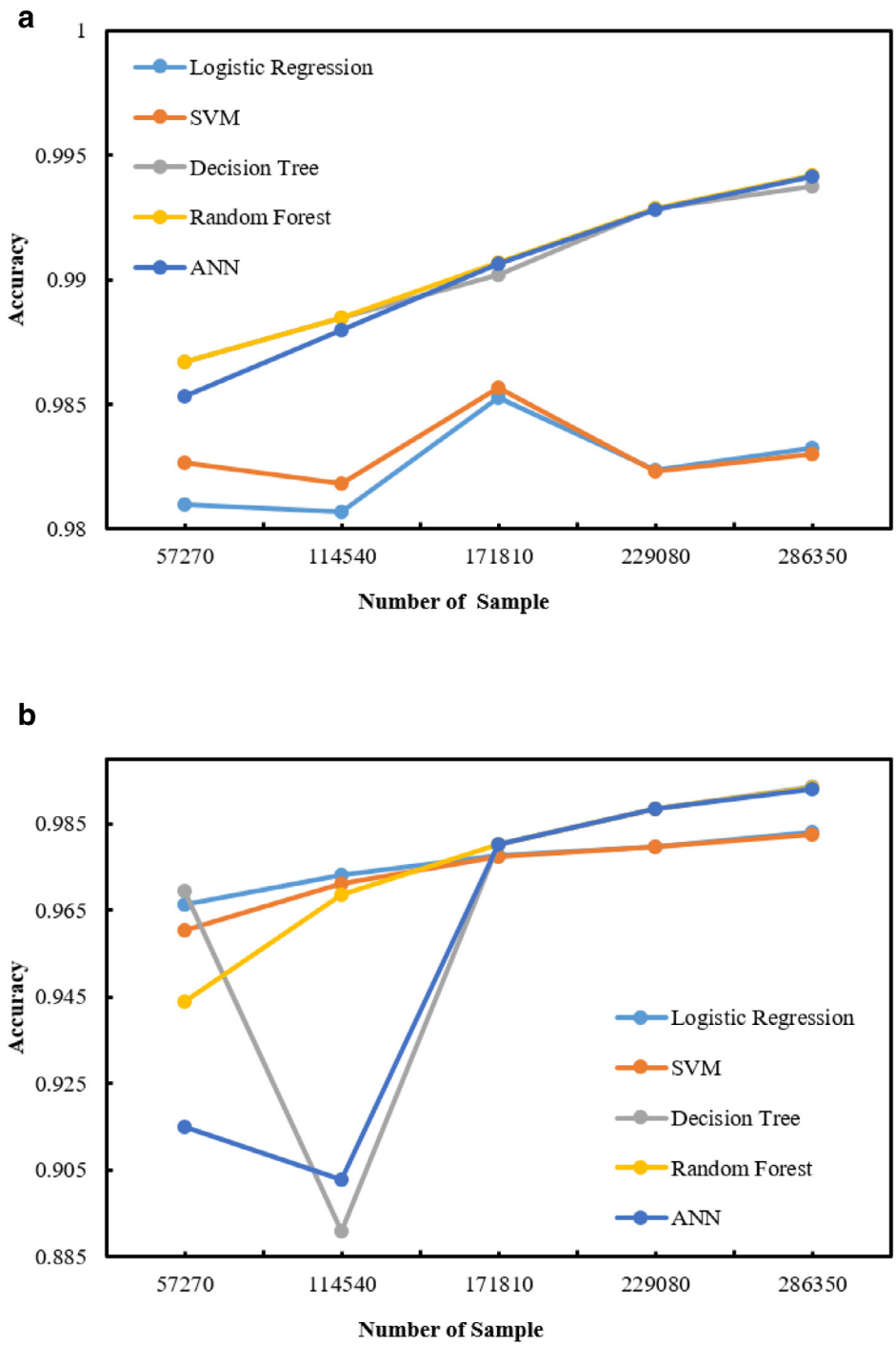

Fig. 4. (a) Training accuracy for different techniques for 5 fold cross validation (b) Testing accuracy for different techniques for 5 fold cross validation.

for Malicious operation it misclassified 77 samples as Normal out of 155 samples, for scan it misclassified 5 samples as DoS and 298 samples as Normal out of 305 samples, for spying it misclassified 120 samples as Normal out of 120 samples, for the Wrong Setup it misclassified 28 samples as Normal out of 28 samples and lastly for Normal, it misclassified 34 samples as DoS and 9 samples as Malicious Operation out of 69,571 samples. In the case of SVM, it correctly classified 775 samples of DoS class but misclassified 403 samples as Normal class. SVM misclassified all samples of data probing, spying and the wrong setup as normal, for malicious control it misclassified 159 samples as Normal out of 169 samples, for malicious

LR

	DoS	D.P	M.C	M.O	SC	SP	W.S	NL
DoS	775	0	0	0	0	0	0	403
D.P	0	0	0	0	0	0	0	63
M.C	0	0	10	0	0	0	0	159
M.O	0	0	0	78	0	0	0	77
SC	5	0	2	0	0	0	0	298
SP	0	0	0	0	0	0	0	120
W.S	0	0	0	0	0	0	0	28
NL	34	0	0	9	0	0	0	69528

SVM

	DoS	D.P	M.C	M.O	SC	SP	W.S	NL
DoS	775	0	0	0	0	0	0	403
D.P	0	0	0	0	0	0	0	63
M.C	0	0	10	0	0	0	0	159
M.O	0	0	0	33	0	0	0	122
SC	0	0	2	0	0	0	0	303
SP	0	0	0	0	0	0	0	120
W.S	0	0	0	0	0	0	0	28
NL	34	0	0	0	0	0	0	69537

DT

	DoS	D.P	M.C	M.O	SC	SP	W.S	NL
DoS	775	0	0	0	0	0	0	403
D.P	0	63	0	0	0	0	0	0
M.C	0	0	169	0	0	0	0	0
M.O	0	0	0	155	0	0	0	0
SC	0	0	2	0	305	0	0	0
SP	0	0	0	0	0	120	0	0
W.S	0	0	0	0	0	0	28	0
NL	18	0	0	0	0	2	0	69551

RF

	DoS	D.P	M.C	M.O	SC	SP	W.S	NL
DoS	775	0	0	0	0	0	0	403
D.P	0	63	0	0	0	0	0	0
M.C	0	0	169	0	0	0	0	0
M.O	0	0	0	155	0	0	0	0
SC	0	0	2	0	305	0	0	0
SP	0	0	0	0	0	120	0	0
W.S	0	0	0	0	0	0	28	0
NL	18	0	0	0	0	2	0	69553

ANN

	DoS	D.P	M.C	M.O	SC	SP	W.S	NL
DoS	775	0	0	0	0	0	0	403
D.P	0	63	0	0	0	0	0	0
M.C	0	0	169	0	0	0	0	0
M.O	0	0	0	155	0	0	0	0
SC	0	0	2	0	305	0	0	0
SP	0	0	0	0	0	120	0	0
W.S	0	0	0	0	0	0	28	0
NL	18	0	1	0	0	2	0	69550

Fig. 5. Confusion Matrix of LR, SVM, DT, RF and ANN.

operation it misclassified 122 samples as Normal out of 155 samples, for scan it misclassified 2 samples as Malicious Operation and 303 samples as Normal out of 305 samples and for Normal, it misclassified 34 samples as DoS out of 69,571 samples.

Lastly, Fig. 6 depicts the Receiver Operating Characteristic (ROC) Curves of LR, SVM, DT, RF, and ANN respectively. From area under the curves, it can be described that DT, RF, and ANN have higher accuracy because all of the area under the curves for every class is approximately equivalent to value one. While in case of LR and SVM, only for DoS and Wrong Setup the area under the curve is equivalent to one.

4.3. Comparative study

Pahl et al. [1] acquired 96.3% accuracy value for multiclass classification using K-Means and BIRCH Clustering. Liu et al. [5] developed data packet based anomaly detector for a binary classification problem and focused on the energy consumption of each node. Their identification rate of the malicious node is above 80%. Like the data source used in this research, Diro et al. [8], Pajouh et al. [6] and D'Angelo et al. [15] have used a dataset from another popular open source. The link of the dataset is given in Table 4. Diro et al. acquired 98.27% accuracy on multiclass classification problem using Neural Network. Pajouh et al. obtained 84.82% identification rate and D'Angelo et al. obtained 94.1% testing accuracy for Binary class classification. Brun et al. [11] have not described much about the dataset and their analysis results. They have detected when the attack is occurring using time series data.

Compared to other papers, our paper provides much detailed description of the dataset. It also provides a clear explanation of the dataset preprocessing steps. The paper is focused on classifying multiple classes which is harder than binary classification. Lastly, a clear description of the evaluation metric values for each classifier is given in this paper.

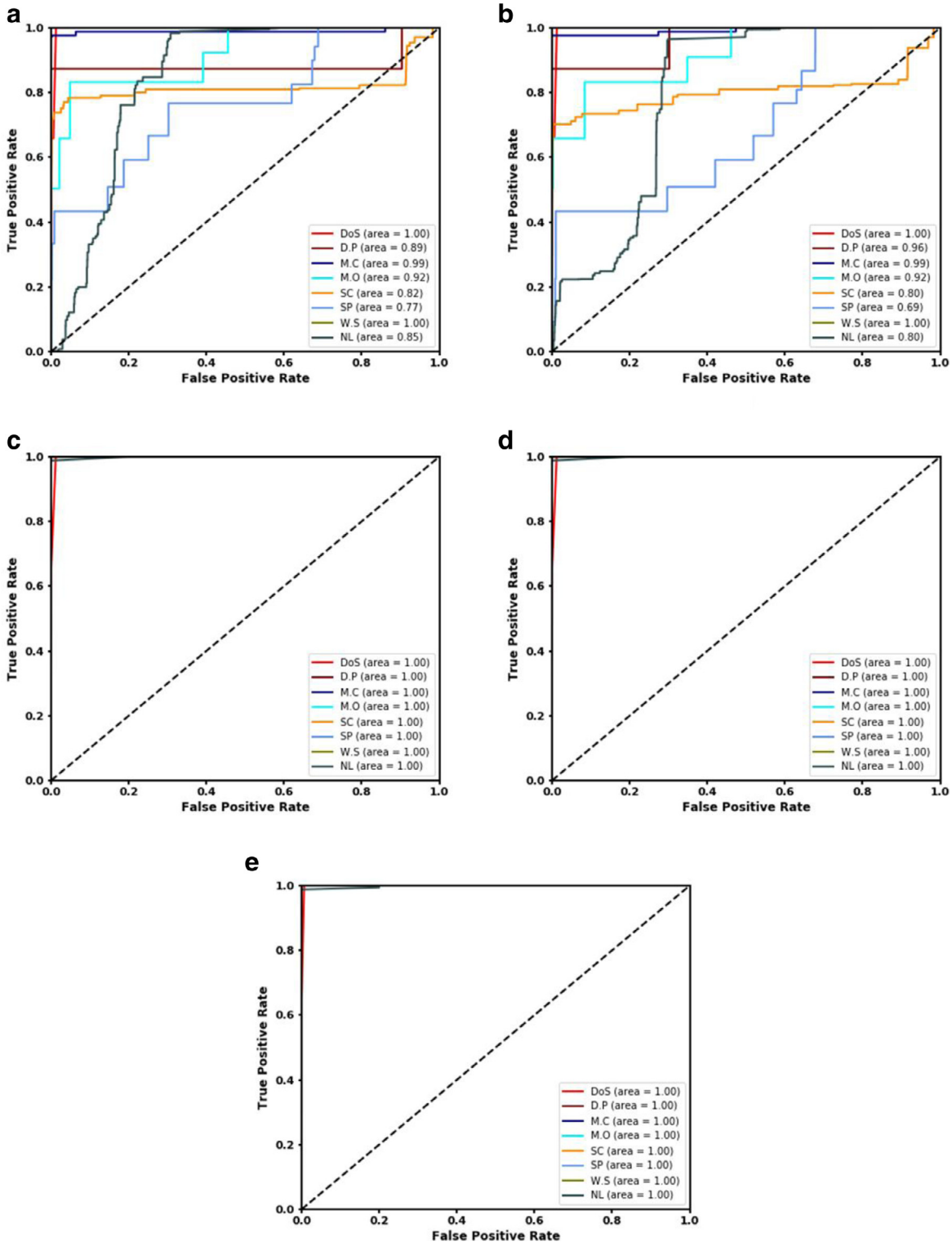

Fig. 6. ROC Curve of (a) Logistic Regression (b) Support Vector Machine (c) Decision Tree (d) Random Forest (e) Artificial Neural Network.

Table 4
Comparative description of proposed IoT attack detection with state of art.

Author and Year	Dataset	Classification Type	Mechanism	Evaluation Metric
Pahl et al. 2018 [1]	Own Synthetic	Multiclass	K-Means BIRCH Clustering	ACC = 96.3
Liu et al. 2018 [5]	Own Synthetic	Binary	Light Probe Routing	I.R. = 0.80(>)
Diro et al. 2018 [8]	NSL-KDD [32]	Multiclass	Neural Network	ACC = 98.27
Brun et al. 2018 [11]	Own	Multiclass	Random Dense Neural Network	N/A
Anthi et al. 2018 [13]	Own Synthetic	Binary	Naive Bayes	N/A
Pajouh et al. 2018 [6]	NSL-KDD [32]	Two-tier	Naive Bayes K-Nearest Neighbor	I.R. = 84.82
D'angelo et al. 2015 [15]	NSL-KDD [32] Real Traffic Data	Binary	U-BRAIN	ACC = 94.1 ACC = 97.4 (Real Data)
Our study	DS2OS traffic traces [18]	Multiclass	Logistic Regression Support Vector Machine Decision Tree Random Forest Artificial Neural Network	ACC = 98.3 ACC = 98.2 ACC = 99.4 ACC = 99.4 ACC = 99.4

I.R. = Identification Rate, ACC = Accuracy, N/A = Not Appropriately Defined

5. Conclusion

Based on the full study it was found that one should use RF technique on these kinds of dataset for solving cyberattacks on IoT network because RF predicted D.P, M.C, M.O, SC, SP, W.S attacks accurately compared to other approaches. In the case of DoS and Normal, it also predicted more samples accurately than other techniques. Hence, relying on these estimations, it can be concluded that RF is the best technique for this particular study. However, here only the classical machine learning approaches are employed over the dataset, and comparative study is given. No new algorithm is devised on this dataset. Hence, further study is needed for developing a robust detection algorithm. More analysis should be given on whole framework designing. 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, microservices 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. In this study, RF performs comparatively better with the accuracy of 99.4%. However, it does not assure that in the case of the big data and other unknown problems RF will perform this way. Hence, more study will be needed.

Conflict of interest

None.

Acknowledgment

We would like to thank Marc-Oliver Pahl, François-Xavier Aubet and Stefan Liebald [1,2] for providing open source dataset that we have used in this study.

References

[1] M.-O. Pahl, F.-X. Aubet, All eyes on you: distributed multi-dimensional IoT microservice anomalydetection, in: Proceedings of the 2018 Fourteenth International Conference on Network and Service Management (CNSM)(CNSM 2018), 2018. Rome, Italy
[2] M.-O. Pahl, F.-X. Aubet, S. Liebald, Graph-based IoT microservice security, in: Proceedings of the NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018, pp. 1–3.
[3] R.C. Deo, Machine learning in medicine, Circulation 132 (20) (2015) 1920–1930.
[4] G. D'Angelo, M. Laracca, S. Rampone, Automated Eddy current non-destructive testing through low definition Lissajous figures, in: Proceedings of the 2016 IEEE Metrology for Aerospace (MetroAeroSpace), IEEE, 2016, pp. 280–285.
[5] X. Liu, Y. Liu, A. Liu, L.T. Yang, Defending on–off attacks using light probing messages in smart sensors for industrial communication systems, IEEE Trans. Ind. Inf. 14 (9) (2018) 3801–3811.
[6] H.H. Pajouh, R. Javidan, R. Khayami, D. Ali, K.-K.R. Choo, A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks, IEEE Trans. Emerg. Top. Comput. (2016).
[7] I. Poyner, R. Sherratt, Privacy and security of consumer IoT devices for the pervasive monitoring of vulnerable people. (2018).
[8] A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for internet of things, Future Gen. Comput. Syst. 82 (2018) 761–768.

[9] C.C. Aggarwal, J. Han, J. Wang, P.S. Yu, A framework for clustering evolving data streams, in: Proceedings of the Twenty-ninth International Conference on Very Large Data Bases-Volume 29, VLDB Endowment, 2003, pp. 81–92.
[10] E. Gelenbe, Y. Yin, Deep learning with dense random neural networks, in: Proceedings of the International Conference on Man–Machine Interactions, Springer, 2017, pp. 3–18.
[11] O. Brun, Y. Yin, E. Gelenbe, Y.M. Kadioglu, J. Augusto-Gonzalez, M. Ramos, Deep learning with dense random neural networks for detecting attacks against IoT-connected home environments, in: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London. Recent Cybersecurity Research in Europe. Lecture Notes CCIS, in: 821, 2018.
[12] B. Usmonov, O. Evsutin, A. Iskhakov, A. Shelupanov, A. Iskhakova, R. Meshcheryakov, The cybersecurity in development of IoT embedded technologies, in: Proceedings of the 2017 International Conference on Information Science and Communications Technologies (ICISCT), IEEE, 2017, pp. 1–4.
[13] E. Anthi, L. Williams, P. Burnap, Pulse: an adaptive intrusion detection for the internet of things (2018).
[14] A. Ukil, S. Bandyoapdhyay, C. Puri, A. Pal, Iot healthcare analytics: The importance of anomaly detection, in: Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), IEEE, 2016, pp. 994–997.
[15] G. D'Angelo, F. Palmieri, M. Ficco, S. Rampone, An uncertainty-managing batch relevance-based approach to network anomaly detection, Appl. Soft Comput. 36 (2015) 408–418.
[16] R. Kozik, M. Choraś, M. Ficco, F. Palmieri, A scalable distributed machine learning approach for attack detection in edge computing environments, J. Parallel Distrib. Comput. 119 (2018) 18–26.
[17] Z. Allen-Zhu, Z. Qu, P. Richtárik, Y. Yuan, Even faster accelerated coordinate descent using non-uniform sampling, in: Proceedings of the International Conference on Machine Learning, 2016, pp. 1110–1119.
[18] M.-O. Pahl, F.-X. Aubet, DS2OS traffic traces, 2018, (https://www.kaggle.com/francoisxa/ds2ostraffictraces). [Online; accessed 29-December-2018].
[19] J. Liu, Y. Xiao, C.P. Chen, Authentication and access control in the internet of things, in: Proceedings of the 2012 Thirty-second International Conference on Distributed Computing Systems Workshops (ICDCSW), IEEE, 2012, pp. 588–592.
[20] Z.-K. Zhang, M.C.Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, S. Shieh, Iot security: ongoing challenges and research opportunities, in: Proceedings of the 2014 IEEE Seventh International Conference on Service-Oriented Computing and Applications (SOCA), IEEE, 2014, pp. 230–234.
[21] J. Milosevic, N. Sklavos, K. Koutsikou, in: Malware in IoT software and hardware, 2016.
[22] W. Ding, Study of smart warehouse management system based on the IoT, in: Proceedings of the Intelligence Computation and Evolutionary Computation, Springer, 2013, pp. 203–207.
[23] W. Leister, T. Schulz, Ideas for a trust indicator in the internet of things, in: Proceedings of the First International Conference on Smart Systems, Devices and Technologies, SMART, 2012, pp. 31–34.
[24] U.S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, M. Stanley, A brief survey of machine learning methods and their sensor and IoT applications, in: Proceedings of the 2017 Eighth International Conference on Information, Intelligence, Systems and Applications (IISA), IEEE, 2017, pp. 1–8.
[25] L. Wang, Support Vector Machines: Theory and Applications, 177, Springer Science & Business Media, 2005.
[26] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[27] C.A. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowl. Discov. 2 (1998) 121–167.
[28] S.R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE Trans. Syst. Man. Cybern. 21 (3) (1991) 660–674.
[29] S. Raschka, Python Machine Learning, Packt Publishing Ltd, 2015.
[30] T.K. Ho, Random decision forests, in: Proceedings of the third international conference on Document analysis and recognition, 1995, 1, IEEE, 1995, pp. 278–282.
[31] M. Hossin, M. Sulaiman, A review on evaluation metrics for data classification evaluations, Int. J. Data Mining Knowl. Manag. Process 5 (2) (2015).
[32] R.P. Lippmann, D.J. Fried, I. Graf, J.W. Haines, K.R. Kendall, D. McClung, D. Weber, S.E. Webster, D. Wyschogrod, R.K. Cunningham, et al., Evaluating intrusion detection systems: the 1998 Darpa off-line intrusion detection evaluation, in: Proceedings of the DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00, 2, IEEE, 2000, pp. 12–26.