IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 September 2023, accepted 15 October 2023, date of publication 19 October 2023, date of current version 1 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3325929

RESEARCH ARTICLE

IoT Network Anomaly Detection in Smart Homes Using Machine Learning

NADEEM SARWAR[1], IMRAN SARWAR BAJWA[1], MUHAMMAD ZUNNURAIN HUSSAIN[2], MUHAMMAD IBRAHIM[1], AND KHIZRA SALEEM[1]

[1]Department of Computer Science, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[2]Department of Computer Science, Bahria University Lahore Campus, Lahore 54600, Pakistan

Corresponding author: Nadeem Sarwar (Nadeem_srwr@yahoo.com)

ABSTRACT In this modern age of technology, the Internet of Things has covered all aspects of life including smart situations, smart homes, and smart spaces. Smart homes have a large number of IoT objects that are working continuously without any interruption. Better security and authentication of these smart devices can provide peaceful environments to live in such spaces. It is important to monitor the activities of smart IoT devices to make them work fault-free. Such devices are small, consume relatively less power and resources, and are easily attackable by attackers. It is crucial to protect the integrity and characteristics of the smart home environment from external attacks. Machine Learning played a vital role in recognizing such malicious activities and attempts. Several Machine Learning approaches are available to detect the normal and abnormal behavior of IoT device traffic. This study proposed a machine learning-based anomaly detection approach for smart homes using different classifiers. Testing and evaluation are performed using the University of New South Wales (UNSW) BoT IoT dataset. Machine learning models based on four classifiers are built using an IoT devices dataset. For the Test dataset, the Weighted Precision, Recall, and F1 score of Random forest, decision tree, and AdaBoost is 1 as compared to ANN which has 0.98, 0.96, and 0.96 respectively Results show that high performance, precision, and robustness can be achieved using the proposed methodology. In this way, smart homes' security and identity of devices can be monitored and anomalies can be detected with high accuracy. Attack categories include binary class, multiclass class, and subclasses. Results show Random Forest algorithm outperforms enough to use this methodology in smart environments.

INDEX TERMS Smart homes, IoT environment, cyber security, network anomaly detection, smart environments, machine learning.

I. INTRODUCTION

The Internet has revolutionized the modern age of technology by providing everyday life facilities at the fingertip. The Internet of Things (IoT) is one of the technologies, which has changed the ideology of modern developments. IoT has been used in hospitals, agriculture, restaurants, roads, and even in our houses. IoT-based applications are known as smart applications [1]. Smart homes are based on IoT devices that are capable of capturing and facilitating every area of the house using smart sensors and controllers. These sensors are communicating through an internet connection. These devices share data for particular tasks and purposes [2]. Fig 1 shows the basic elements of an IoT network, which includes identification, sensing, communication, services, and semantics. IoT applications are increasing day by day, which enables these devices to become low-cost, energy-efficient, and compact. However, increased usage of IoT devices also increases the risk factors and threats for such networks [3]. It is important to make these devices secure and threat-free so that people can securely use such networks in smart homes. Researchers are focusing on the security of IoT applications but it is also the demand of the current age to work on the protection of IoT networks from unauthorized users (intruders) [4].

Smart Homes' technology usage is rapidly growing. This aids people and individuals to control house appliances and

The associate editor coordinating the review of this manuscript and approving it for publication was Hang Shen.

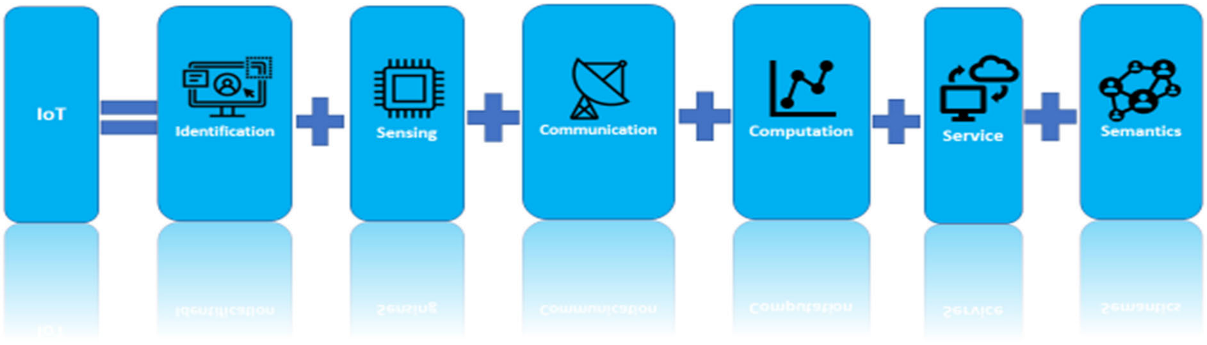


FIGURE 1. The IoT components.

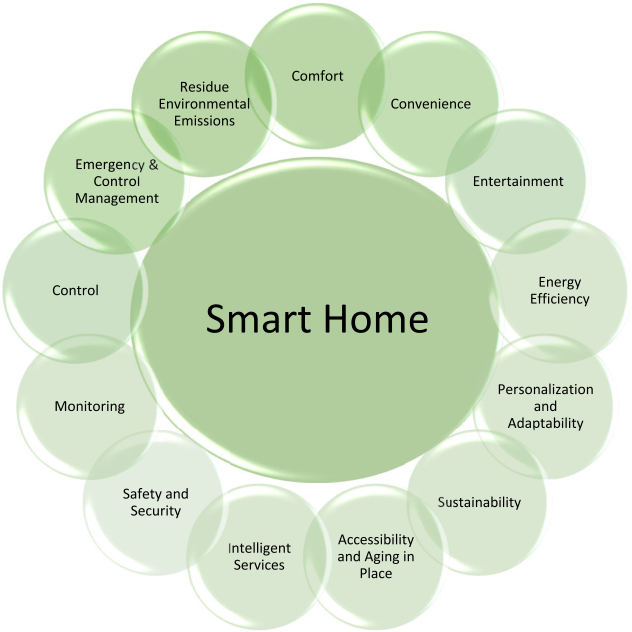


FIGURE 2. Smart homes objectives.

devices through a single platform. In this way, people will be able to easily control and monitor house devices. The basic objectives of a smart home are illustrated in Fig 2. These objectives show that comfort, security, and reliability are the key points. However, this comfort also increases the risk of external threats that should be addressed properly. One of the main concerns in smart home environment technology is to permit users to monitor, and control device security and take precautions accordingly [6] Smart home devices consist of sensors and actuators. These devices are of different sizes. Voice controllers may enable intruders to risk the security of the network. A household person will not be able to detect and protect against such intrusions and threats. In recent years, researchers have been focusing on application intrusion detection to address such issues [7].

Generally, there are two types of home automation systems available. These include a locally controlled system and a remotely controlled system. First, a local controller is used to control in-house devices. While in second, devices can be controlled from distant areas. Remotely controlled systems are operated using an internet connection. A locally controlled system can use an Ethernet, wireless connection, or Bluetooth for such purposes [9].

The rapid increase in IoT technology for smart homes also increases the potential risks of network attacks and security challenges. It is necessary to bring high-level security to the smart home environment to enable privacy and data protection. Such advancements increase the benefits and challenges of smart environments. If these smart environments are attacked by intruders of attackers, serious privacy issues to users' data can be damaged by stealing information or monitoring the activities [10].

Thereafter, a secured IoT framework for anomaly detection is needed for the protected communication of sensing devices. Smart devices make people more vulnerable to safety precautions. For some investors and business owners, data is everything to the organization [11]. Certain information must be kept secret by the government and certain private organizations. Internet of Things (IoT) nodes allow hackers to collect vital information from any significant corporation [12]. The abovementioned issues are frequently resolved by employing very simple strategies. The signature-based [13] technique is used to capture anomalies in advance. At times, the system is crosschecked with the database. This strategy, however, creates processing costs and leaves the system vulnerable to unidentified assaults. Feature selection is critical for identifying the most important traits. The da-ta-analysis-based strategy can overcome the problem of unknown threats and can be adopted faster than other alternatives. As a result, this study adopts data-driven methodologies.

Several Machine Learning (ML) approaches are discussed to address such issues. Machine Learning models analyze, recognize, and monitor traffic patterns to learn the traffic flow in a normal routine. For this determination, supervised, and unsupervised machine learning models are available [14]. To overcome the security problem in smart homes, a supervised learning-based anomaly detection model is proposed which is efficient enough to accurately predict the anomalies on time generate alerts, and, halt the working of the devices to enable persons to take precautionary measures. The main goal of this research is to develop a smart, secure, and reliable methodology for smart home environments to detect anomalies and vulnerabilities. The primary focus of our study is on identifying network anomalies, specifically in the context of IoT (Internet of Things) environments. These anomalies encompass a range of malicious activities that may include but are not limited to:

1. Data Exfiltration: Anomalies related to unauthorized data transfers or leaks from the IoT devices, indicate potential security breaches.
2. Keylogging: Detection of anomalous keystroke logging activities, which could indicate attempts to capture sensitive information.
3. OS Fingerprinting: Anomalies related to attempts to identify the operating system of devices in the network, often a precursor to targeted attacks.
4. Service Scanning: Unusual activities associated with probing or scanning services and ports on IoT devices, a common behavior of potential malicious entities seeking vulnerabilities.
5. UDP Anomalies: Abnormalities in UDP (User Datagram Protocol) communication, can be indicative of network attacks or unusual device behavior.

Now, we suggested a Machine Learning solution for smart home environments' abnormal activity monitoring and detecting malicious activities. The proposed framework is tested with six different ML algorithms. An additional crucial characteristic of this anomaly detection study is to compare simple machine learning classifiers like Decision Tree (DT), AdaBoost (ADA), and Random Forest (RF) with complex classifiers like Artificial Neural Network (ANN). The following are the primary contributions of the suggested study:

1. The dataset for the proposed methodology is evaluated for six different categories of anomalies.
2. An anomaly detection framework is implemented for analyzing anomalous activity by machine learning algorithms.
3. The proposed study has high performance and reliable predictions for anomaly detection and alert generation.
4. This study's key contribution is the development of a high-performing anomaly detection model utilizing a machine-learning classification approach.

Section II of the paper describes the literature work. The proposed approach is highlighted in section III along with dataset description, and anomaly details. Section IV discusses the implementation specifics and outcomes of the proposed methodology. Section V lastly provides the conclusion and future work.

II. BACKGROUND & LITERATURE REVIEW

in the Internet of Things (IoT) is becoming popular in every field of life including industry and research. IoT is simply defined as the objects or things that share recorded information with other objects or platforms through a communication channel like the Internet. IoT devices consist of sensors, actuators, and communication links. These sensors and devices are used nowadays to connect homes, offices, schools, universities, hospitals, and even human beings [15]. IoT refers to a smart environment. Such smart systems are becoming helpful in industry, banks, education, healthcare, and home environments Several biomedical IoT devices are available for patients to assist them in monitoring and diagnosing diseases [16]. In this way, patients will be able to understand their health parameters. Such devices are also connected to a central platform where recorded information is stored and processed further.

IoT is also assisting in managing and operating powerhouses that transform these houses into smart powerhouses. The basic objective of offering smart environments is to increase the availability, usability, and reliability of the services and applications. The Internet of Things (IoT) is designed to facilitate productive communication between the real world and its digital equivalent (also called the digital transformation or cyber-physical systems) (CPS)) [17].

In smart homes, such devices play a significant role in day-to-day activities. One must consider the security and privacy of data shared by IoT objects. Gartner predicts that by 2020, more than 25 percent of assaults on businesses will include Internet of Things (IoT) devices [18].

Using these obstacles and numbers as a jumping-off point, we discovered that almost all studies with the intent of analyzing the difficulties encountered in the IoT area include a portion dedicated to security and privacy concerns [19]. The previous few years have seen several cyber-security problems involving networked, pervasive, and Internet of Things technology. Numerous attacks have been made against the Internet of Things (IoT), including notable Distributed Denial of Service (DDoS) attacks on Dyn's DNS (domain name system; see reference) [20]Attacks/vulnerabilities on/off self-driving cars [21], ransomware attacks [21], attacks on smart home health equipment [22], and more.

The IoT and critical infrastructures pose new security threats. The limited resources aboard the devices used to construct IoT and CPS apps make it difficult to mitigate these hazards. In addition, the gadgets are inherently insecure since they need a constant Internet connection. New instances and studies show that greater investigation is required [23]. They also reveal, sadly, that present solutions are far from being acceptable to halt the exponential development in the number and complexity of assaults [24]. Thus, the necessity for further layers of defense to make smart home IoT devices more durable and proof of such assaults is vital.

Reference [25] introduced a host-based anomaly discovery method in which device features and activities are considered using a mobile phone application. Their system uses a supervised machine learning approach to evaluate the gathered data and determine whether an app is benign or dangerous. Device metrics including CPU load, active processes, and application programming interface (API) invocations are all included in their feature vector. Based on their claimed results, their method successfully distinguishes between apps that are games and those that are tools 99.7 percent of the time. However, they fail to reveal or comment on whether the shown or mentioned programs are dangerous. Relatedly, [22] devised a method for detecting malware on smart devices at the device level. This study introduces a system called Crowdroid, which records and analyses system calls as they occur in real-time. A mobile application receives this data. The collected data, represented by Linux system calls, is sent to a central server (off-device research) where it is analyzed for harmful software. Towards a Machine Learning-based Framework for DDoS Attack Detection in Software-Defined IoT (SD-IoT) Networks [42]. This study discusses machine learning frameworks for DDoS attack detection in SD-IoT networks.

Reference [43] toward Software-Defined Networking-Based IoT Frameworks: A Systematic Literature Review, Taxonomy, Open Challenges, and Prospects [44], discusses machine learning frameworks for DDoS attack detection in SD-IoT networks.

For IoT-based smart home environments, Alippi et al. [26] presented a framework for detecting changes at the level of sensors. The approach constructs a dynamical model of the anticipated sensor-generated signal over time [22]. The system then continuously compares the predicted signals from the model with the actual signals (data streams) arriving from various sensors. Specifically, it uses a method for detecting transitions known as ''Intersection-of-Confidence'' to look for differences [27]. The authors investigated the deep learning-based solutions for IDS and achieved 95% precision and 97% Recall after various attack solutions [38]. Also was discussed about Imbalanced data issues during intrusion detection and applied SMOTE technique to the balanced dataset [39]. The author used IG and GR filter-based approaches for feature ranking and applied five ML algorithms, including kNN, ANN, bagging, J48, and Ensemble methods, to the classification of traffic features. Both classification approaches are implemented in this study, Binary and multi [40]. The NSL-KDD dataset was used for the IoT-based cyber-attack classification (Binary and Multi) [41].

The available studies for these smart homes' security and threat protection are yet not mature enough to fulfill the domain requirements. Such environments need security mechanisms, privacy mechanisms, and data transmission mechanisms with machine intelligence for a secure smart home. The proposed study consists of Machine Learning based layered architecture as shown in Fig 3, which can provide improved and reliable detection of anomalies in smart homes.

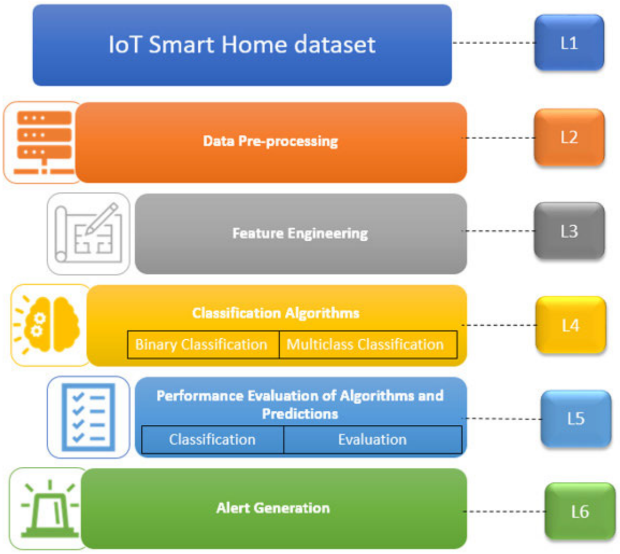


FIGURE 3. Proposed systematic architecture for Smart home IoT network anomaly detection.

At level 1, which is L1, the dataset generation and collection are performed. In the proposed study, the UNSW−2018−IoT−Botnet dataset is used. This dataset contains the data of IoT devices of a smart home. This dataset contains 0 to 668521 records that are sufficient to be used for the study. This information is forwarded to L2, which is the data pre-processing level. At this level, data is preprocessed for missing values, feature engineering, and data balancing. After preprocessing, another vital level is feature selection. The most relevant features are selected using correlation among several features. After selecting relevant and ideal features for classification, training data is used for classification. This classification is performed using different machine learning algorithms. These algorithms include AdaBoost, Decision Trees, Random Forest, LSTM, Auto Encoder, and ANN.

III. PROPOSED METHODOLOGY

Smart homes are a new technology for people to live comfortably. However, such networks and devices are prone to intrusion attacks due to a lack of security mechanisms for implementation. The speedy evolution of the Internet of Things opens new dimensions for IoT devices but poses new threats as well. The tiny sensors-based intelligent environments open a new challenge for the security and privacy of home devices. A subfield of Computational systems in which human intelligence-based algorithms are used to solve real-life problems is called Machine Learning (ML). it is reflected as a functional mare in the Big Data field. These approaches can be applied in health, business, commerce, aerospace, biomedicine, music, media, and many more real-world applications. However such advancements also increase security issues and threats to smart

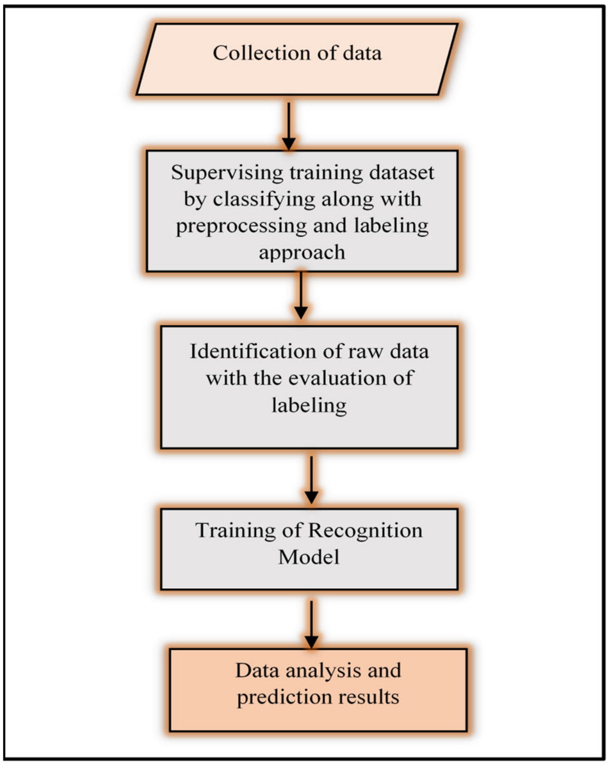

Collection of data
Supervising training dataset by classifying along with preprocessing and labeling approach
Identification of raw data with the evaluation of labeling
Training of Recognition Model
Data analysis and prediction results

FIGURE 4. **Supervised learning approach.**

home environments. Machine Learning has revolutionized the field of smart home environments by providing IoT devices in hand. In the suggested anomaly detection system, data received from the sensing devices is processed using a supervised machine-learning algorithm.

In a supervised learning approach, the algorithm is previously trained on test data and then performs prediction using this knowledge. In the unsupervised approach, previous knowledge is not known [28]. Pattern recognition and feature extraction are performed on real-world data. These two approaches can be applied according to the requirement of the problem. The following Fig 4 shows the working of the layered super-vised learning approach. In this layered process, training data with labels are provided to the classifier for learning and classification purposes. After this, testing is accomplished to assess the accuracy of the model with unlabeled data.

A subfield of Computational systems in which human intelligence-based algorithms are used to solve real-life problems is called machine learning. Machine learning is reflected as a functional mare in the Big Data field. Machine Learning approaches can be applied in health, business, commerce, aerospace, biomedicine, music, media, and many more real-world applications. However such advancements also increase security issues and threats to smart home environments.

Machine Learning has revolutionized the field of smart home environments by providing IoT devices in hand. In the proposed anomaly detection system, data received from the sensing devices is processed using a supervised machine-learning algorithm. In a supervised learning approach, the algorithm is previously trained on test data and then performs prediction using this knowledge. In the unsupervised approach, previous knowledge is not known [30]. Pattern recognition and feature extraction are performed on real-world data. These two approaches can be applied according to the requirement of the problem. Figure 4 shows the working of the layered supervised learning approach. In this layered process, training data with labels are provided to the classifier for learning and classification purposes. After this, testing is performed to evaluate the performance of the model with unlabeled data.

Fig 5 explains the system structure and flow of different parts of the suggested anomaly detection model for IoT-based smart environments. The primary stages of the suggested approach are Data Sources, Data Gathering, Processing, and Anomaly detection. The proposed framework examines the traffic and arranges this activity by the typical motions recorded for various IoT devices to unusual movement. The gathering of data is the first stage in Fig 6. The proposed study is based on one of the benchmark datasets from UNSW BoT IoT that is openly available. There are 46 attributes in this data set. Eight different types make up this dataset. Section III-A of the dataset contains a detailed description. Due to its extraction from several Internet of Things devices, the data used in this study is regarded as big data [29], [31]. Data Pre-processing is done using data cleaning, data visualization, data balancing, and feature selection.

A. DATASET DESCRIPTION

Several datasets are available for this work such as the Bot-IoT and the UNSW BoT IoT datasets. It is found that the UNSW BoT IoT dataset is more refined and has been practiced by researchers. The entire dataset is 16 GB in size, it has some similar attributes to the UNSW NB 15 dataset but contains supplementary diversity in the type of mischievous measures as shown in Fig 7. The designed methodology consists of the UNSW BoT IoT dataset because of the diverseness and authenticity of training models [30], [37]. This contains 46 different features.

In this work, 668522 records were arbitrarily nominated for selection. These selected records are used for training and testing which is 5% of the entire dataset. Data cleaning and preparation are performed for training and testing models [29]. A dataset contains nominal, numerical, and categorical features. It is important to encode the dataset for numerical values. One hot encoding and label encoding technique are used for transformation into numerical attributes. In label encoding, unique numbers are assigned to each class in the categorical column. 80% of training and 20% of test data is used for training and testing.

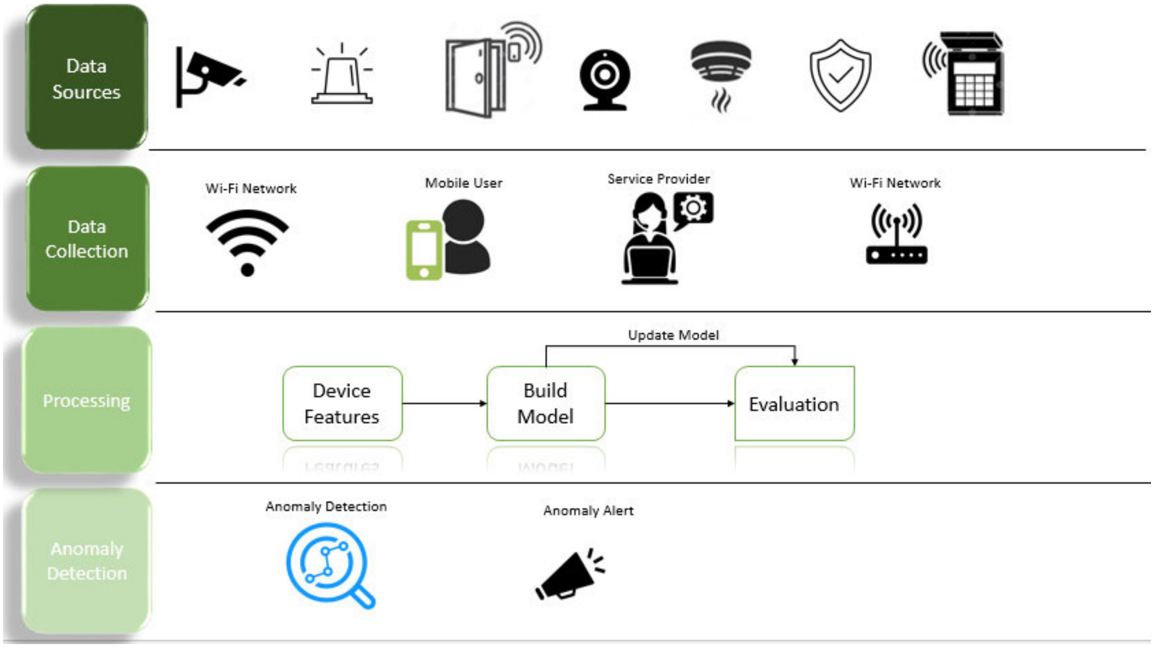


FIGURE 5. System communication channel.

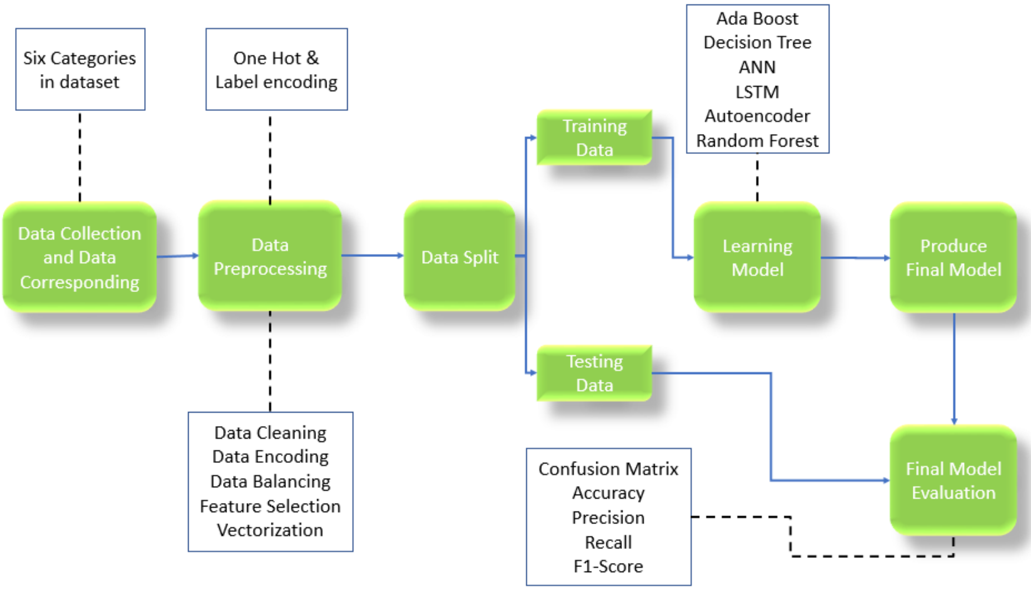


FIGURE 6. Detail description of anomaly detection framework.

1) DETAILS OF DATASET ATTACK CATEGORIES

UNSW BoT IoT dataset contains six different classes with a different number of records as shown in Table 1. There are six different types of categories considered in the proposed model. At first, binary classification is done in which there are two classes i.e. attack and normal. In multiclass classification, the attack category is subcategorized into Keylogging, Data Exfiltration, Service Scan, OS Fingerprint, and UDP. Table 2 presents the description of each type of attack, which is studied in the proposed work. A comparison of the BoT IoT dataset with other standard network datasets is given in Table 3. This shows that the BoT IoT dataset is far more realistic and reliable to test smart network environments as compared to other datasets, that were used in previous research.

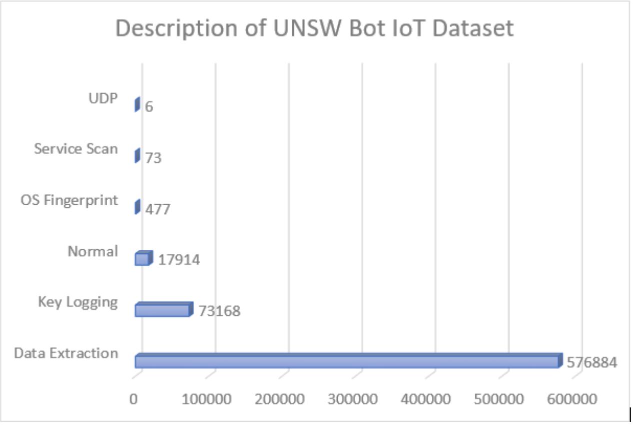


FIGURE 7. Description of UNSW BoT IoT dataset.

TABLE 1. Attack count description in the dataset.

Sr.	Class	Sub Class	No of Samples
1	Attack	Data Exfiltration	576884
2		Keylogging	73168
4		OS Fingerprint	477
5		Service Scan	73
6		UDP	6
7	Normal	….	17914

T represents true and F represents False. True shows the presence of that characteristic and False shows the absence of the characteristic.

B. FEATURE ENGINEERING

The Machine Learning models are based on exploratory data analysis and observations. A dataset must be in the appropriate format to be given to the classification model. For this purpose, categorical attributes are handled by using different transformation techniques. The dataset used in this study contains both categorical and numerical features. To convert the input into numeric characteristics, one hot encoding and label encoding technique is used. In this research, the data was transformed into a feature vector using label encoding techniques [11].

After this process, we ensure that data must be balanced so that the efficiency of the model can be achieved. 46 features selected after encoding are presented in Fig 8. In this BoT IoT dataset, there are different attack categories. Each category contains a different number of records. For efficient performance, it is important to balance the dataset. For data balancing, the oversampling technique is used. The balanced dataset after over-sampling is shown in Fig 9.

To aid in the computational efficiency and simplicity of the machine learning models, only relevant characteristics are chosen in the feature selection process using correlation as shown in Fig 10. In addition, this method can mitigate the effects of overfitting when all characteristics are used [11], below the features are extracted. Network Traffic Features:

1. These features encompass data related to network protocols, packet sizes, flow durations, and source/destination IP addresses. By analyzing patterns and variations in network traffic, the model can identify abnormal behaviors associated with potential attacks or malicious activities. For instance, sudden spikes in packet sizes or unexpected protocol usage can be indicative of an anomaly.

2.Communication Patterns: Features related to communication patterns involve attributes like the number of connections, data transfer rates, and the frequency of interactions between devices. Anomalies in these patterns, such as a device initiating an unusually high number of connections, may suggest a potential security threat.

3.Categorical Features: These features pertain to qualitative aspects of network traffic, including service types and flags. By one-hot encoding categorical features, the model can effectively handle categorical data and distinguish patterns of behavior for different services or flags. Anomalies may manifest as unexpected or malicious usage of particular services.

In feature selection, No machine learning method similar to that used by Pahl et al. has been used here for feature selection. Reference [32] for the proposed study [11]. Because of the large dataset, it is important to perform feature selection for improved performance and minimum biasedness. This process of oversampling is shown in Fig 11. IQR and SMOTE techniques are used for balancing and oversampling. SMOTE is a technique of oversampling in which artificial samples are created for the minority class. This technique helps in overcoming the overfitting problem brought on by random oversampling. IQR is used to identify outliers in the dataset. IQR (inter Quartile Range) is used to identify outliers. 1.5 IQR outlier is used to identify abnormalities.

Algorithm 1 Outlier detection and handling algorithm:

```
1. Sort the dataset D in ascending order
2. estimate the 1st and 3rd quartiles(Q1, Q3) in D features
3. calculate IQR=Q3-Q1
4. calculate lower bound = (Q1–1.5∗IQR), upper bound = (Q3+1.5∗IQR)
5. loop through the values of the dataset and check for those who fall below the lower bound and above the upper bound and mark them as outliers
6. Replace outlier values with Median value
```

Feature selection is performed using the following equation of covariance. The algorithm estimates the merit of covariance between x and y where Sd is the standard deviation.

$$P(x,y)= Cov(x,y) / (Sd (x).Sd(y)) \tag{1}$$

TABLE 2. Description of attack models.

Sr.	Type	Subcategory	Description
1	Attack	Data Exfiltration	Unauthorized removal or movement of any data from a device.
2		Keylogging	Act of tracking and recording every keystroke entry made on the network without permission.
4		OS Fingerprint	Obtain OS information of target hosts to prepare for future attacks.
5		Service Scan	The system is scanned for some information that aims to corrupt data.
6		UDP	Forward a huge amount of UDP packets to exhaust the bandwidth of target servers/ devices.

TABLE 3. Comparison of datasets models in terms of different characteristics.

Dataset	Darpa98	KDD99	ISCX	LBNL	DEFCON8	CAIDA	UNIBS	CICIDS 2017	TUIDS	UNSWNB15	BoT IoT
Realistic Traffic	No	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Realistic Testbed Configuration	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Labeled data	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	Yes
New generate features	No	Yes	Yes	No	No	No	No	Yes	Yes	Yes	Yes
Full packet capture	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes
Diverse attack scenarios	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes
IoT Traces	No	No	No	No	No	No	No	No	No	No	Yes

Algorithm 2 SMOTE algorithm:

```
1. Take the difference between a sample i and its nearest neighbor ki
2. Multiply the difference by a random number between 0 and 1
        Dif =Random$_{(0,1)}$ * (i*ki)
3. Add this difference to the sample to generate a new synthetic example the in feature space
        Add Dif to Feature space
4. Continue with next nearest neighbor up to the user-defined number.
```

Algorithm 3 Feature Selection Algorithm:

```
  An array of features _ Feature Extraction
  For i: Array of features.length
    For k : Array of data[i].length
    X=feature of i
    Y=feature of i+1
    Cov(x,y) calculates the covariance between x and y
    Calculate the standard deviation of x
    Calculate the standard deviation of y
    Correlation value calculated from the above equation
End for loop
```

C. MACHINE LEARNING MODELS

Several machine learning approaches are considered while experimenting with the proposed approach. These algorithms include AdaBoost, Decision Tree, Random Forest, Ada Boost, Auto Encoder, and Artificial Neural Network. A description of these algorithms is given below.

1) ADA BOOST

Ada Boost (Adaptive Boosting) is a boosting technique that is used in supervised machine learning algorithms. In this technique, instances with minimum weights are assigned higher weights. In short, instances with minimum weights are converted into strong instances by assigning more weights [33]. Equation 1 describes the AdaBoost final classifier equation.

$$\mathrm{H(x)} = Sign\left(\sum\nolimits_{t=1}^{T} \alpha_t h_t\left(x\right)\right) \tag{2}$$

In Equation 2, T represents the weak classifiers. ht (x) is the output of weak classifier "t". α_t is the weight applied to classifier 't' as obtained by the Ada Boost algorithm? The final result is thus a linear combination of all the weak classifiers, and the choice is made by inspecting the sign of this sum of values.

2) DECISION TREE

Decision Trees work by using a tree-like structure that is based on profits, outlays, and likelihoods. A schematic depicting the many consequences of a set of interconnected decisions. In most cases, a DT will have a single starting node from which several branches will emerge. The results of these actions generated new nodes, from which new cases emerged. Then it evolved into a tree, or more accurately, a flowchart.

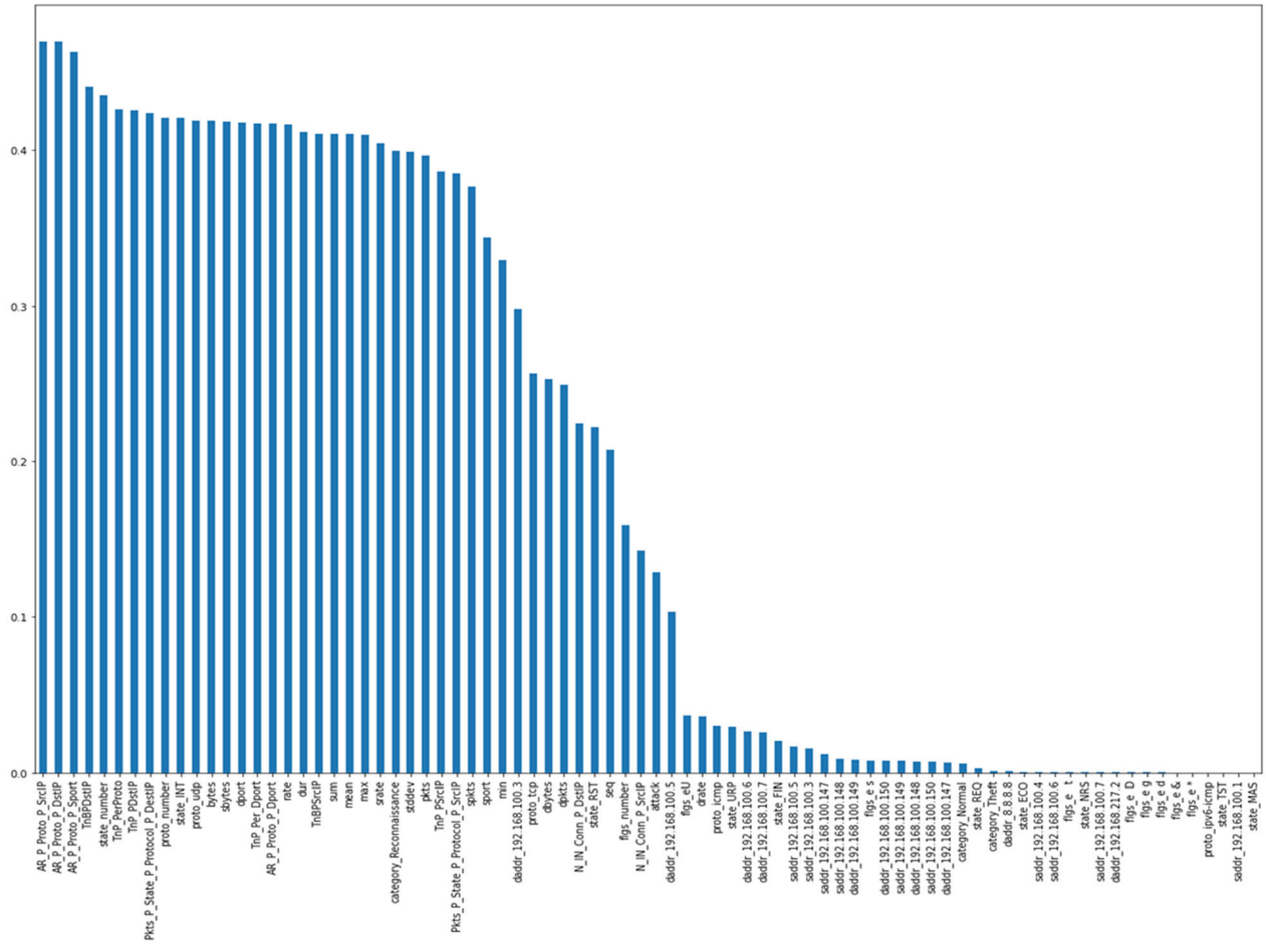


FIGURE 8. 63 Features selected after encoding.

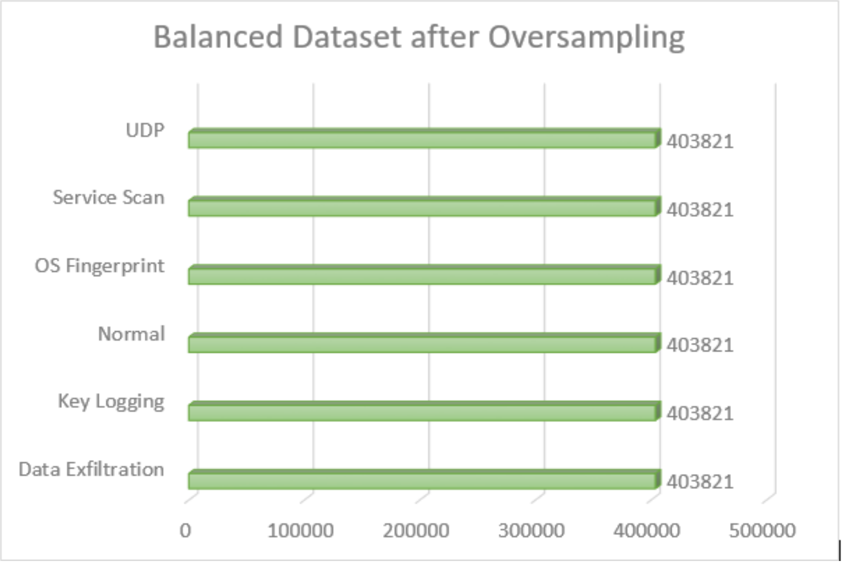


FIGURE 9. Balanced dataset after oversampling.

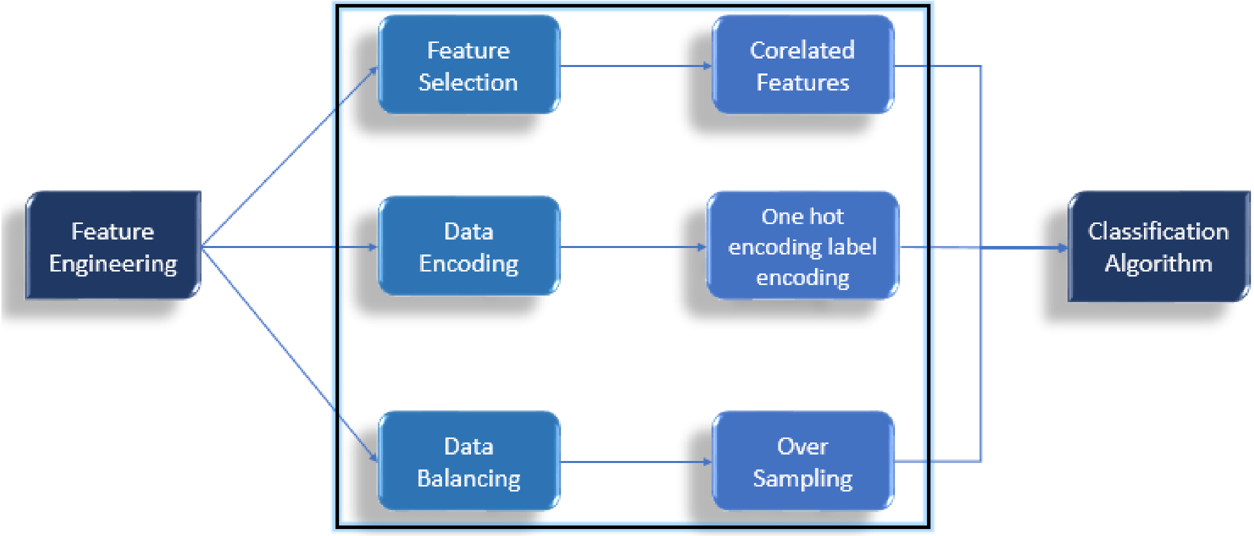


FIGURE 10. Feature engineering process in the proposed methodology.

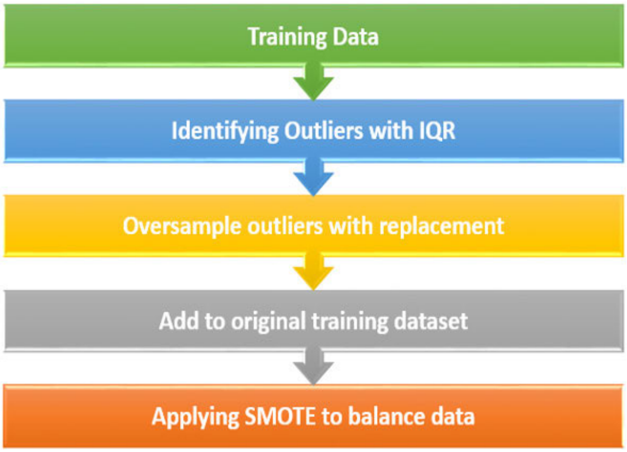

Training Data
Identifying Outliers with IQR
Oversample outliers with replacement
Add to original training dataset
Applying SMOTE to balance data

FIGURE 11. The oversampling process in the proposed methodology.

3) RANDOM FOREST

The Random Forest works using forest decision trees with a supervised learning technique. This algorithm is fast compared to other supervised learning techniques [34]. Several individual tree results are evaluated using majority voting or average calculation. This algorithm provides a good performance measure [28].

4) ARTIFICIAL NEURAL NETWORK

The ANN is a deep learning method of predicting classes or categories. We can modify many parameters in ANN as compared to other classifiers. However, error optimization time is increased in this technique [11], [35]. The amount of the chosen features of the network traffic for a host is represented by the number of input nodes of the ANN (22 or 41). The employed ANN also has two hidden layers with 25 and 30 nodes each, as well as an output layer with 29 nodes (representing 29 categories of attacks). The backpropagation (BP) computation method has been used to identify the numbers of hidden layers and nodes in them.

5) AUTOENCODER

An example of a self-supervised learning model that can learn a compressed representation of incoming data is the autoencoder. In this, the reconstruction component of the model's design may be dropped after fitting and the model may be used until the bottleneck. A fixed-length vector that offers a compressed representation of the input data is the output of the model at the bottleneck. Neural networks known as autoencoders are capable of finding representations with low dimensions of highly dimensional input. It ought to be able to rebuild the input from the output based on this. The model can then receive input data from the domain, and the model's output at the bottleneck can be used as a feature vector in a supervised learning model.

6) LSTM

The use of LSTM Autoencoders on time series, audio, video, and text sequence data has shown that they are capable of learning a compressed representation of the sequence data. The input sequence is sequentially read by an encoder LSTM model. The hidden state or output of this model represents an internally learned representation of the full input sequence as a fixed-length vector after reading the entire input sequence. After that, this vector is sent as input to the decoder model, which interprets it as each step of the output sequence is produced. For a specific dataset of sequences, an encoder-decoder LSTM is set up to read the input sequence, encode it, decode it, and reproduce it. Based on the model's capacity to replicate the input sequence, the performance of the model is assessed.

D. PERFORMANCE EVALUATION METRICS

There are a variety of methods for gauging performance. Methods like statistics and mathematics fall under this category. The designed system's efficacy is measured using the following indicators. Indicators like this help determine the method that is most appropriate for the job at hand.

1) ACCURACY

Accuracy is the measure of correctly classified instances among all of the instances. The formula is given below in Equation 3.

$$P = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

2) PRECISION

Precision is a mathematical method to estimate the performance of the classifier. Precision is the percentage of true predictions of positive classes divided by the total prediction of the positive classes in the model. The formula for precision is given below.

$$P = \frac{TP}{TP + FP} \tag{4}$$

3) RECALL

Recall denotes the amount of correctly classified instances of the positive class. In Equation 5, R means Recall, TP symbolizes true positive predictions and FP symbolizes False positive predictions [11].

$$R = \frac{TP}{TP + TN} \tag{5}$$

4) F1 SCORE

F1-score (F1) represents the measure of the harmonic mean of recall (or TPR) and precision [36]. The formula for the F1 measure is given in equation 6.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6}$$

5) CONFUSION MATRIX

When evaluating machine learning classification models, the confusion matrix is a useful tool for getting a feel for how things will shake out. A comparison between the actual classes and the expected ones is shown in the table below. It facilitates a speedy diagnosis of misidentification between actual and anticipated classes. Prediction outcomes from a classification task may be summarized using a confusion matrix [37].

IV. IMPLEMENTATION AND RESULTS ANALYSIS

The architecture and full design specifications of the suggested Network anomaly detection framework are covered in the previous section. We will go into the specifics of the suggested system's implementation in this part. Figure 3 displays layered construction. Fig 6 displays the system in operation. Following that, section IV-B presents the analysis of the findings.

A. EXPERIMENT DESCRIPTION

The proposed methodology consists of the UNSW BoT IoT dataset because of the diverseness and authenticity of training models [30], [37]. Data Pre-processing is done using data cleaning, data balancing, and feature selection. The dataset is preprocessed by removing noisy data and missing information. After removing these anomalies, one hot encoding is performed for categorical features. In this way, each attack category has a unique column which increases the number of features as well. In our dataset, OS Fingerprint, Data Exfiltration, Service Scan, Keylogging, and UDP are categories of attack to which the dataset is classified. Figure 8 shows the details of features selected after one hot encoding. IQR outlier is used to identify abnormalities. 1.5 IQR technique is used to identify outliers in the proposed study. SMOTE (Synthetic Minority Oversampling Technique) Oversampling is performed to balance the dataset for each category. Figure 9 shows the details of oversampling for each attack category. After this process, we ensure that data must be balanced so that the efficiency of the model can be achieved. 46 features selected after encoding are presented in Fig 8. After selecting relevant features, machine learning models are applied to measure precision, recall, accuracy, and F1. Weighted mean, precision, and recall are also calculated for detailed analysis.

B. RESULT ANALYSIS

The data description section explains machine-learning approaches used on the dataset UNSW BoT IoT. Validation of data is performed using 5-fold cross-validation. Performance parameters used for result evaluation are confusion matrix; accuracy, precision, recall, and F1 score as discussed in section III-D. The accuracy comparison of Random Forest, Decision Tree, AdaBoost, LSTM, Artificial Neural Network (ANN), and Auto Encoder is shown in Table 4 and Fig 12.

C. COMPARISON OF ACCURACY

Accuracy is the measurement of correctly classified items among all items. This comparison of accuracy is shown in Table 6. In this table accuracy of ML, models including Random Forest, Decision Tree, ANN, and AdaBoost are compared. For training data, AdaBoost Decision Tree and Random Forest achieve 100% accuracy whereas ANN achieves 93.17% accuracy. For the test dataset, these algorithms achieve 99.9% and ANN achieves 95.74% accuracy. Fig 12 shows the graphical representation of accuracy, in which classifiers are shown on the x-axis and the accuracy percentage is shown on the y-axis. Training dataset accuracy is shown in yellow color and test dataset accuracy is shown in blue color. Accuracy shows that ANN performance is low compared to AdaBoost, Decision Tree, and RF. ANN performed low with training and testing datasets.

D. COMPARISON OF CONFUSION MATRICES

The confusion Matrix shows the actual values of true positive, false positive, true negative, and false negative

TABLE 4. Accuracy comparison of ML approaches.

Accuracy of ML Approaches						
Performance Metrics	Classifiers					
	AdaBoost	Decision Tree	ANN	Random Forest	LSTM	Auto Encoder
Training Data Accuracy	100%	100%	93.17%	100%	91.40%	88.30%
Testing Data Accuracy	99.99%	99.99%	95.74%	100%	90.20%	84.30%

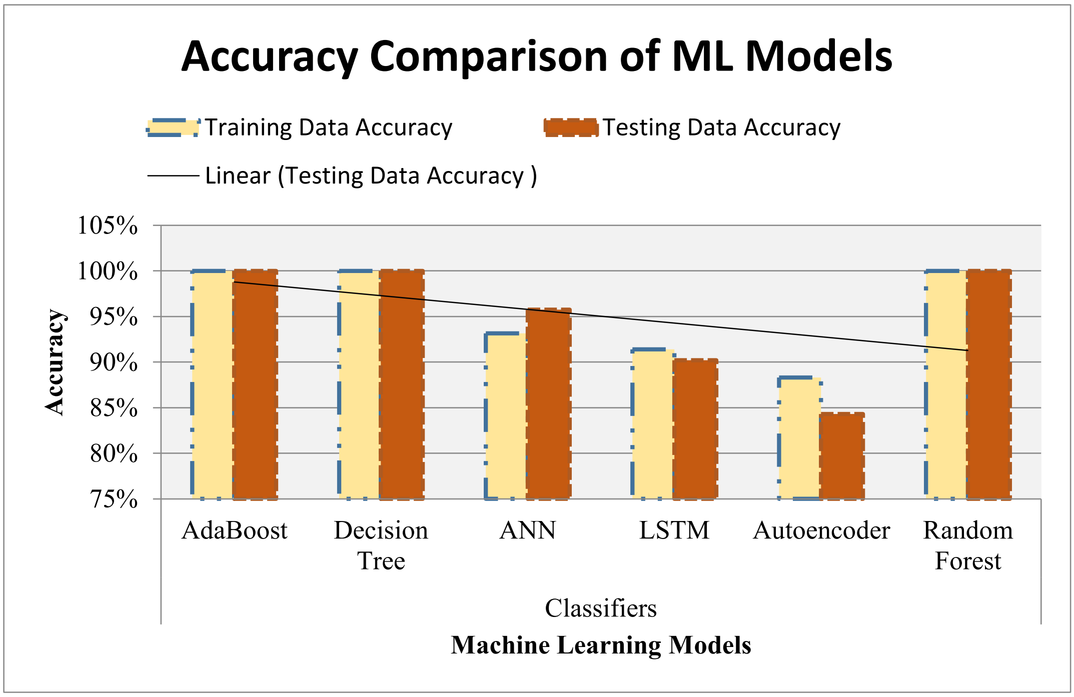


FIGURE 12. Accuracy comparison of machine learning approaches.

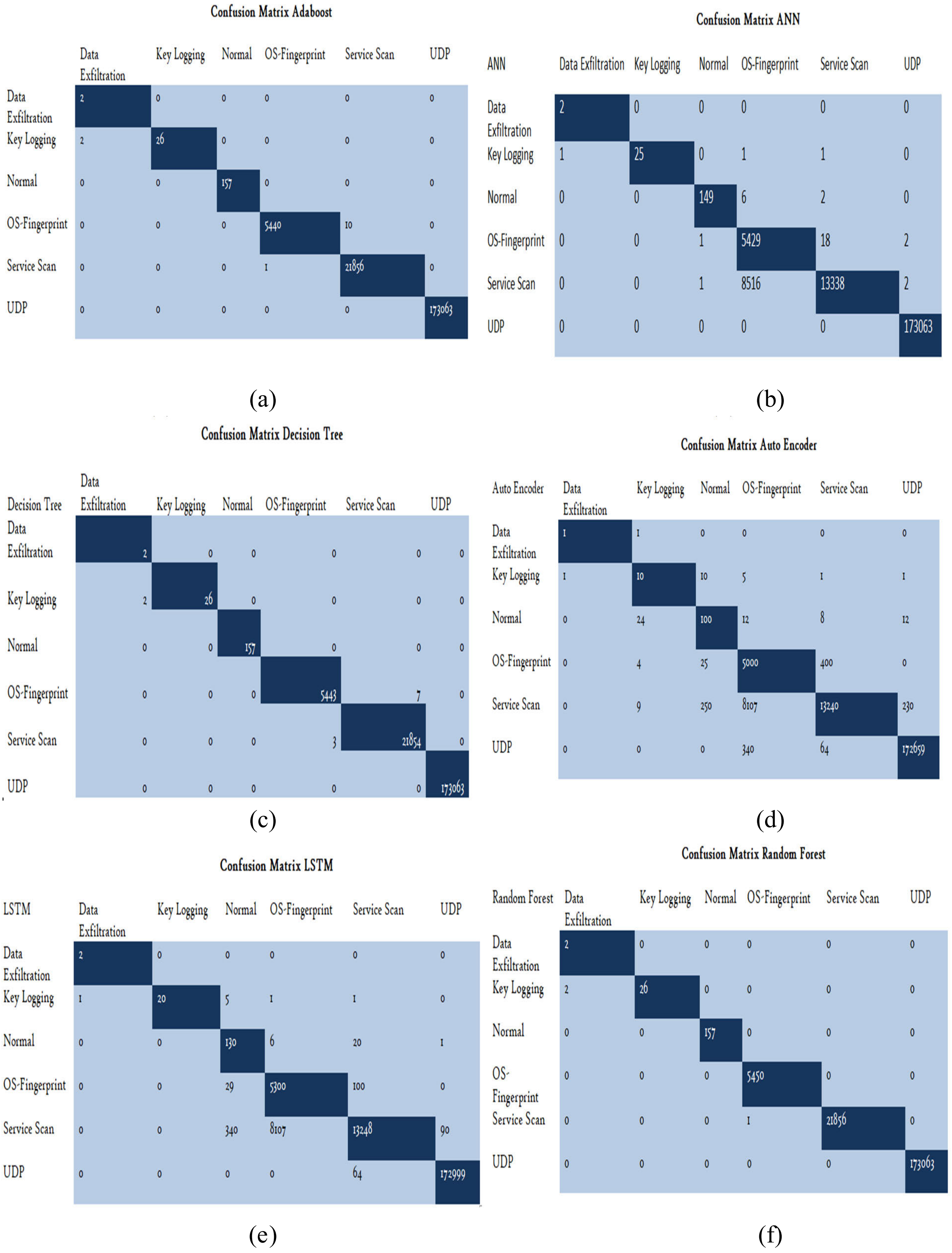
instances predicted by the specific ML model. Confusion matrices of ML techniques, which are shown in Fig 13 (a,b,c,d,e,f), RF, DT, and ADA performed well in terms of True Positive and True Negative predictions and resulted in high precision, recall, and accuracy. In ADA, 10 instances of OS_Fingerprint are misclassified out of 5450 instances, and 2 instances of keylogging are misclassified. In RF, only 2 instances of keylogging and 1 instance of Service_scan are misclassified. In DT, 2 instances of keylogging, 7 instances of OS_Fingerprint, and 3 instances of Service_scan are misclassified. Whereas in ANN, 3 instances of keylogging, 8 instances of Normal, 21 instances of OS_Fingerprint, and 8519 instances of Service_scan are misclassified.

From the 5-fold cross-validation results, it can be concluded that RF performed best as shown in Fig 14. In the first fold, ANN performed with approximately 94% accuracy, and DT, RF, and ADA performed between 96% and 98%. ADA and DT are similar in the first fold whereas RF is slightly lower than these. In the second fold, ANN accuracy is below 92%, whereas ADA, DT, and RF are better than ANN. In the third fold, DT and ADA accuracy is dropped and ANN accuracy is comparatively higher than these two, Whereas RF performs highest in the third fold. In the last two folds, ADA and DT accuracy are similar and RF performed better in all of the five folds. ANN, LSTM, and Auto Encoder have less performance.

E. COMPARISON OF MACRO PRECISION, RECALL, AND F1 SCORE

Average precision, recall, and F1 score per class are calculated using Macro measures. The macro measure is used in multiclass classification in which precision is calculated for each class separately and then the average of these is calculated. Similarly, recall and F1 scores are calculated as shown in Table 5.

If there are six classes (a, b, c, d, e, f) then precision is calculated using Equation 7, and Macro precision is calculated



Confusion Matrix Adaboost

	Data Exfiltration	Key Logging	Normal	OS-Fingerprint	Service Scan	UDP
Data Exfiltration	2	0	0	0	0	0
Key Logging	2	26	0	0	0	0
Normal	0	0	157	0	0	0
OS-Fingerprint	0	0	0	5440	10	0
Service Scan	0	0	0	1	21856	0
UDP	0	0	0	0	0	173063

(a)

Confusion Matrix ANN

ANN	Data Exfiltration	Key Logging	Normal	OS-Fingerprint	Service Scan	UDP
Data Exfiltration	2	0	0	0	0	0
Key Logging	1	25	0	1	1	0
Normal	0	0	149	6	2	0
OS-Fingerprint	0	0	1	5429	18	2
Service Scan	0	0	1	8516	13338	2
UDP	0	0	0	0	0	173063

(b)

Confusion Matrix Decision Tree

Decision Tree	Data Exfiltration	Key Logging	Normal	OS-Fingerprint	Service Scan	UDP
Data Exfiltration	2	0	0	0	0	0
Key Logging	2	26	0	0	0	0
Normal	0	0	157	0	0	0
OS-Fingerprint	0	0	0	5443	7	0
Service Scan	0	0	0	3	21854	0
UDP	0	0	0	0	0	173063

(c)

Confusion Matrix Auto Encoder

Auto Encoder	Data Exfiltration	Key Logging	Normal	OS-Fingerprint	Service Scan	UDP
Data Exfiltration	1	1	0	0	0	0
Key Logging	1	10	10	5	1	1
Normal	0	24	100	12	8	12
OS-Fingerprint	0	4	25	5000	400	0
Service Scan	0	9	250	8107	13240	230
UDP	0	0	0	340	64	172659

(d)

Confusion Matrix LSTM

LSTM	Data Exfiltration	Key Logging	Normal	OS-Fingerprint	Service Scan	UDP
Data Exfiltration	2	0	0	0	0	0
Key Logging	1	20	5	1	1	0
Normal	0	0	130	6	20	1
OS-Fingerprint	0	0	29	5300	100	0
Service Scan	0	0	340	8107	13248	90
UDP	0	0	0	0	64	172999

(e)

Confusion Matrix Random Forest

Random Forest	Data Exfiltration	Key Logging	Normal	OS-Fingerprint	Service Scan	UDP
Data Exfiltration	2	0	0	0	0	0
Key Logging	2	26	0	0	0	0
Normal	0	0	157	0	0	0
OS-Fingerprint	0	0	0	5450	0	0
Service Scan	0	0	0	1	21856	0
UDP	0	0	0	0	0	173063

(f)

FIGURE 13. **Confusion Matrix comparison of different ML approaches (a,b,c,d,e,f).**

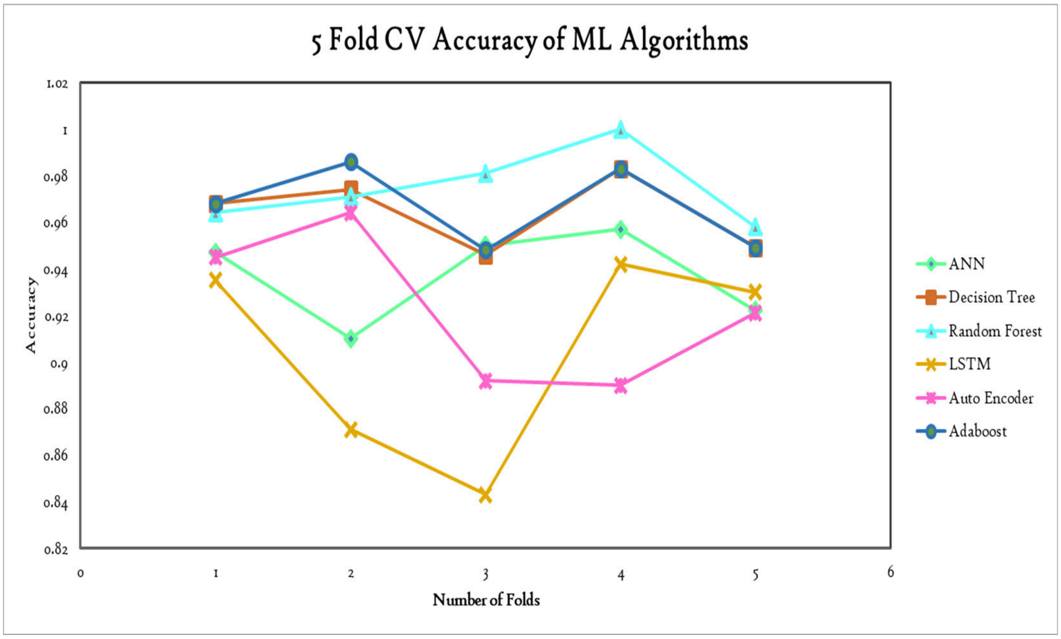


FIGURE 14. Cross-validation accuracy of decision tree, random forest, ANN, and ADA.

TABLE 5. Macro precision, recall, and F1 score of ML algorithms.

Macro precision, Recall, and F1 score of ML Algorithms								
			Classifiers					
Data	Performance Metrics		AdaBoost	Decision Tree	ANN	Random Forest	LSTM	Auto Encoder
Training Data	Macro	Precision	1.00	1.00	0.95	1.00	0.91	0.89
		Recall	1.00	1.00	0.93	1.00	0.84	0.88
		F1 Score	1.00	1.00	0.93	1.00	0.86	0.94
Test Data	Macro	Precision	0.99	0.99	0.91	0.99	0.89	0.77
		Recall	0.94	0.94	0.84	0.94	0.89	0.88
		F1 Score	1.00	1.00	0.95	1.00	0.91	0.89

using Equation 8.

$$Precision = Pa, Pb, Pc, Pd, Pe, Pf \tag{7}$$

whereas;

$$MacroPrecision = (Pa, Pb, Pc, Pd, Pe, Pf)/6 \tag{8}$$

Similarly, for recall, individual class recall is calculated and then macro recall is calculated using Equations 9 and 10 respectively.

$$Recall = Ra, Rb, Rc, Rd, Re, Rf \tag{9}$$

$$MacroRecall = (Ra, Rb, Rc, Rd, Re, Rf)/6 \tag{10}$$

A graphical comparison of Macro performance measures is presented in Fig 15. In this, it is seen that Decision Tree, Random Forest, and AdaBoost have good performance compared to ANN.

F. COMPARISON OF WEIGHTED PRECISION, RECALL, AND F1 SCORE

Weighted performance measures are presented in Table 6. In weighted measures, weights (number of instances) are attached to the measures. If the number of instances is (A, B, C, D, E, F) then weighted precision can be calculated using the following formula.

$$Precision = Pa, Pb, Pc, Pd, Pe, Pf \tag{11}$$

$$\begin{aligned} WeightedPrecision = (APa, BPb, CPc, DPd, EPe, FPf) \\ \times /(A + B + C + D + E + F) \end{aligned} \tag{12}$$

Similarly, F1 measure and Recall can be produced in this way. Fig 16 shows that weighted precision, recall, and F1 scores for RF, ADA, and DT are 1 compared to ANN which has low measures.

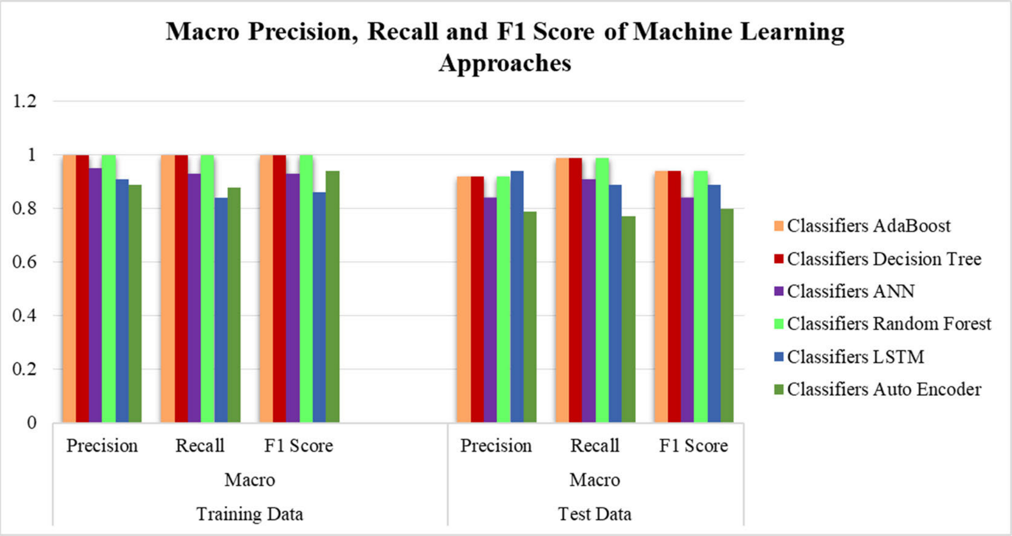


FIGURE 15. Comparison of macro precision, recall, and F1 score of ML algorithms.

TABLE 6. Weighted precision, recall, and F1 score of ML algorithms.

Weighted precision, Recall, and F1 score of ML Algorithms								
Data	Performance Metrics		Classifiers					
			AdaBoost	Decision Tree	ANN	Random Forest	LSTM	Auto Encoder
Training Data	Weighted	Precision	1.00	1.00	0.95	1.00	0.90	0.83
		Recall	1.00	1.00	0.93	1.00	0.87	0.87
		F1 Score	1.00	1.00	0.93	1.00	0.76	0.82
Test Data	Weighted	Precision	1.00	1.00	0.98	1.00	0.93	0.89
		Recall	1.00	1.00	0.96	1.00	0.86	0.87
		F1 Score	1.00	1.00	0.96	1.00	0.88	0.80

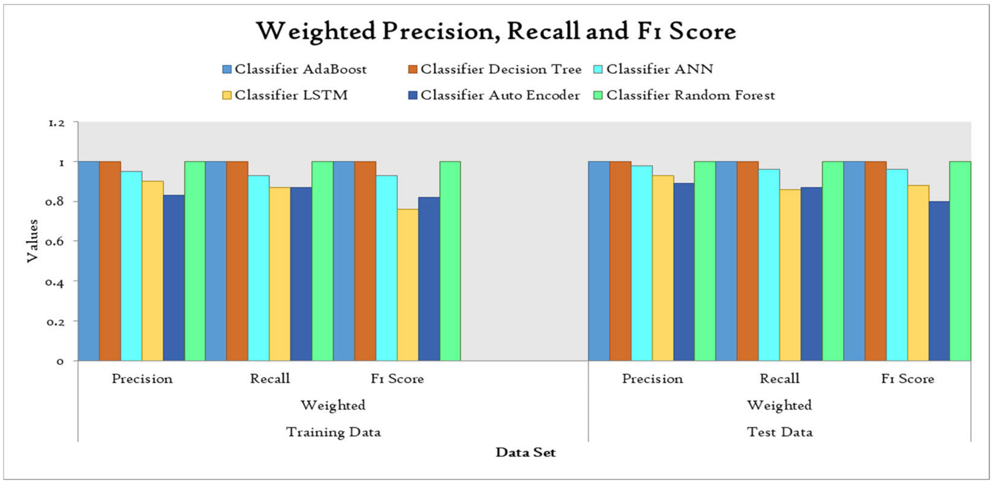


FIGURE 16. Comparison of weighted precision, recall, and F1 score of ML algorithms.

G. SUMMARY OF RIGHT CLASSIFIED AND MISCLASSIFIED INSTANCES

Table 7 and Fig 17 presents the right classified (RC) and misclassified (MC) instances with different algorithms. The total count shows the total number of instances in that particular class. Data Exfiltration has 2 instances with each classifier and the right predictions are made by each algorithm. Keylogging has 28 counts of which 2 are misclassified by

TABLE 7. Proposed methodology comparison using different algorithms.

Attacks	Total Counts	AdaBoost		Decision Tree		ANN		Random Forest		Auto Encoder		LSTM	
		RC	MC	RC	MC	RC	MC	RC	MC	RC	MC	RC	MC
Data Exfiltration	2	2	0	2	0	2	0	2	0	1	1	2	0
Keylogging	28	26	2	26	2	25	3	26	2	10	18	20	08
Normal	157	157	0	157	0	149	8	157	0	100	56	130	27
OS Fingerprint	5450	5440	10	5453	7	5429	21	5450	0	5000	429	5300	129
Service Scan	21857	21856	1	21854	3	13338	8519	21856	1	13240	8596	13248	8537
UDP	173063	173063	0	173063	0	173063	0	173063	0	172659	404	172999	64
Total MC		13		12		8551		3		9504		8765	

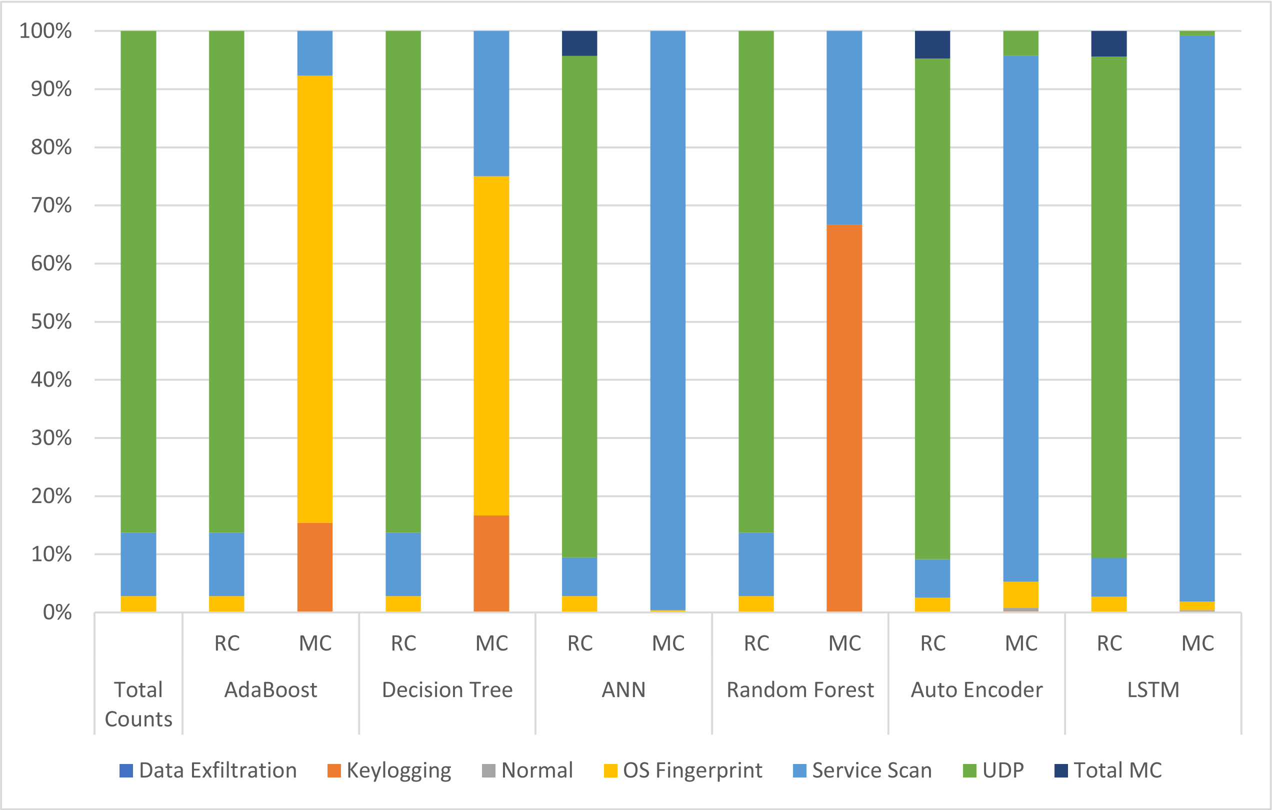


FIGURE 17. Comparison of the proposed approach with the proposed machine learning algorithms.

RF, DT, AND ADA whereas 3 are misclassified by ANN. For the normal category, ANN correctly classifies 149, and 8 instances are misclassified.

For OS Fingerprint, 5450 is the total count out of which accurately classified are 5440, and 10 instances are misclassified by ADA. For the Decision Tree, 5453 is RC and 7 is MC. For ANN, 5429 and 21 are RC and MC respectively. With RF, 5450 and 0 are RC and MC respectively. LSTM has an 8765 MC value and Autoencoder has 9504 misclassified instances. Their identification rate of the malicious node is above 80%. Results show that the proposed technique provides a detailed description of anomaly detection with large datasets. Feature engineering is performed using one-hot encoding, label encoding, oversampling, and correlation-based feature selection.

Multiclass classification is performed accurately, which is more difficult to predict than binary classification. The proposed method outperforms Random forest and is best with DT, and ADA.

The suggested method outperforms the technique in which an IoT sensors data set is utilized for experimentation and assessment, as shown by a comparison with the dataset supplied by [11], [32]. There are a total of 357,952 samples and 13 characteristics available in the aforementioned dataset from Hasan et al. There are a total of eight classes in the dataset, which consists of 347,935 Normal data and 10,017 aberrant data. There are gaps in reporting for 148 nodes in the ''Accessed Node Type'' feature and 2050 in the ''Value'' feature. In terms of precision, recall, and accuracy, our suggested methodology outperforms the status quo method. For the sake of safety and privacy, the suggested technique can be implemented in the context of smart homes.

V. CONCLUSION AND FUTURE WORK

Smart home environments are becoming popular to provide ease and comfort to users. However, increased IoT devices are more prone to attack due to a lack of security measures. It is important to monitor such devices to prevent and identify malicious activities in the network. In the proposed study, the UNSW BoT IoT dataset is used for the evaluation of the proposed methodology using a feature selection approach. This dataset is based on real-time IoT botnet traffic of smart devices. Machine Learning-based anomaly detection methodology is evaluated using six algorithms i.e. Random forest, decision trees, AdaBoost, LSTM Autoencoder, and ANN. The proposed study provides better results compared to previous approaches because of the feature selection approach which has improved the results shown in fig 13. Evaluation and validation results proved that the proposed methodology gives 100% Training dataset accuracy with Decision Tree, Random Forest, and AdaBoost. Whereas 99.9% Testing Dataset accuracy with a decision tree, and Ada-Boost. And 100% test dataset accuracy with Random forest. Weighted Precision, Recall, and F1 score of Random forest, decision tree, and AdaBoost is 1 as compared to ANN which has 0.95, 0.93, and 0.93 respectively for the Training dataset. For the Test dataset, the Weighted Precision, Recall, and F1 score of Random forest, decision tree, and Ada-Boost is 1 as compared to ANN which has 0.98, 0.96, and 0.96 respectively. It is observed that the Random forest, decision tree, and ADA work best for larger datasets and can easily identify malicious activities in the network with high accuracy. The proposed framework is best for the detection of Botnet attacks and malicious activities using RF.

The proposed work is a significant contribution to securing IoT devices from external attacks by using a Feature selection-based machine learning approach to identify malicious patterns in traffic. In this way, user privacy, security, and safety can be obtained. Our solution shows a path towards making security by design retrofittable into existing IoT installations.

In the future, the proposed methodology of anomaly detection can be tested and validated with new ML models and other datasets like CIDIDS, KDD, Bot−IoT, and CTU-13 and results can be compared. More Machine Learning classifiers can be tested. Unsupervised techniques can also be tested and compared with the algorithms used in this study. Other feature selection techniques can be tested with this methodology.

The limitation of the proposed study is that it needs more in-depth focus to interpret the behavior of the device in different environments.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

The dataset supporting the conclusions of this article is available at the repository: https://research.unsw.edu.au/projects/bot-iot-dataset.

REFERENCES

[1] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, ''A vision of IoT: Applications, challenges, and opportunities with China perspective,'' *IEEE Internet Things J.*, vol. 1, no. 4, pp. 349–359, Aug. 2014.

[2] I. Lee and K. Lee, ''The Internet of Things (IoT): Applications, investments, and challenges for enterprises,'' *Bus. Horizons*, vol. 58, no. 4, pp. 431–440, Jul. 2015.

[3] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, ''IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?'' *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 41–49, Sep. 2018.

[4] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, ''Network intrusion detection for IoT security based on learning techniques,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, 3rd Quart., 2019.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, ''Internet of Things: A survey on enabling technologies, protocols, and applications,'' *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[6] N. Balta-Ozkan, R. Davidson, M. Bicket, and L. Whitmarsh, ''Social barriers to the adoption of smart homes,'' *Energy Policy*, vol. 63, pp. 363–374, Dec. 2013.

[7] A. Sperotto and A. Pras, ''Flow-based intrusion detection,'' in *Proc. 12th IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM) Workshops*, May 2011, pp. 958–963.

[8] B. Ali, ''Internet of Things based smart homes: Security risk assessment and recommendations,'' Dept. Comput. Sci., Elect. Space Eng., Luleå Univ. Technol., Luleå, Sweden, Tech. Rep., 2016. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1032194/FULLTEXT02.pdf

[9] S. Bin, L. Yuan, and W. Xiaoyi, ''Research on data mining models for the Internet of Things,'' in *Proc. Int. Conf. Image Anal. Signal Process.*, Apr. 2010, pp. 127–132.

[10] J. Steinberg, ''These devices may be spying on you (even in your own home),'' 2014. [Online]. Available: https://www.forbes.com/sites/josephsteinberg/2014/01/27/these-devices-may-be-spying-on-youeven-in-your-own-home/#73cc4556b859

[11] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, ''Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches,'' *Internet Things*, vol. 7, Sep. 2019, Art. no. 100059.

[12] I. K. Poyner and R. S. Sherratt, ''Privacy and security of consumer IoT devices for the pervasive monitoring of vulnerable people,'' in *Proc. Living Internet Things, Cybersecurity IoT*, 2018, pp. 1–5.

[13] A. A. Diro and N. Chilamkurti, ''Distributed attack detection scheme using deep learning approach for Internet of Things,'' *Future Gener. Comput. Syst.*, vol. 82, pp. 761–768, May 2018.

[14] M. H. Alsharif, A. H. Kelechi, K. Yahya, and S. A. Chaudhry, "Machine learning algorithms for smart data analysis in Internet of Things environment: Taxonomies and research trends," *Symmetry*, vol. 12, no. 1, p. 88, Jan. 2020.

[15] R. Porkodi and V. Bhuvaneswari, "The Internet of Things (IoT) applications and communication enabling technology standards: An overview," in *Proc. Int. Conf. Intell. Comput. Appl.*, Mar. 2014, pp. 324–329.

[16] K. Saleem, I. S. Bajwa, N. Sarwar, W. Anwar, and A. Ashraf, "IoT healthcare: Design of smart and cost-effective sleep quality monitoring system," *J. Sensors*, vol. 2020, pp. 1–17, Oct. 2020.

[17] C. Klötzer, J. Weißenborn, and A. Pflaum, "The evolution of cyber-physical systems as a driving force behind digital transformation," in *Proc. IEEE 19th Conf. Bus. Informat. (CBI)*, vol. 2, Jul. 2017, pp. 5–14.

[18] S. Garg, K. Kaur, G. Kaddoum, and K. R. Choo, "Toward secure and provable authentication for Internet of Things: Realizing Industry 4.0," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4598–4606, May 2020.

[19] B. Khalfi, B. Hamdaoui, and M. Guizani, "Extracting and exploiting inherent sparsity for efficient IoT support in 5G: Challenges and potential solutions," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 68–73, Oct. 2017.

[20] M. D. Donno, N. Dragoni, A. Giaretta, and M. Mazzara, "AntibIoTic: Protecting IoT devices against DDoS attacks," in *Proc. Int. Conf. Softw. Eng. Defence Appl.*, 2016, pp. 59–72.

[21] T. Dargahi, A. Dehghantanha, P. N. Bahrami, M. Conti, G. Bianchi, and L. Benedetto, "A cyber-kill-chain based taxonomy of crypto-ransomware features," *J. Comput. Virol. Hacking Techn.*, vol. 15, no. 4, pp. 277–305, Dec. 2019.

[22] A. Albasir, "Detection of anomalous behavior of IoT/CPS devices using their power signals," Dept. Elect. Comput. Eng., University of Waterloo, Waterloo, ON, Canada, Tech. Rep., 2020. [Online]. Available: https://uwspace.uwaterloo.ca/bitstream/handle/10012/16545/Albasir_Abdurhman.pdf?sequence=3&isAllowed=y

[23] F. Ullah, Q. Javaid, A. Salam, M. Ahmad, N. Sarwar, D. Shah, and M. Abrar, "Modified decision tree technique for ransomware detection at runtime through API calls," *Sci. Program.*, vol. 2020, pp. 1–10, Aug. 2020.

[24] Q. Gou, L. Yan, Y. Liu, and Y. Li, "Construction and strategies in IoT security system," in *Proc. IEEE Int. Conf. Green Comput. Commun., IEEE Internet Things IEEE Cyber, Phys. Social Comput.*, Aug. 2013, pp. 1129–1132.

[25] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, "'Andromaly': A behavioral malware detection framework for Android devices," *J. Intell. Inf. Syst.*, vol. 38, no. 1, pp. 161–190, Feb. 2012.

[26] C. Alippi, V. D'Alto, M. Falchetto, D. Pau, and M. Roveri, "Detecting changes at the sensor level in cyber-physical systems: Methodology and technological implementation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1780–1786.

[27] C. Alippi, G. Boracchi, and M. Roveri, "A just-in-time adaptive classification system based on the intersection of confidence intervals rule," *Neural Netw.*, vol. 24, no. 8, pp. 791–800, Oct. 2011.

[28] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A brief survey of machine learning methods and their sensor and IoT applications," in *Proc. 8th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Aug. 2017, pp. 1–8.

[29] M. Hassan Aysa, A. Abdu Ibrahim, and A. Hamid Mohammed, "IoT ddos attack detection using machine learning," in *Proc. 4th Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, Oct. 2020, pp. 1–7.

[30] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.

[31] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques," in *Proc. Int. Conf. Mobile Netw. Manag.*, 2017, pp. 30–44.

[32] M.-O. Pahl and F.-X. Aubet, "All eyes on you: Distributed multi-dimensional IoT microservice anomaly detection," in *Proc. 14th Int. Conf. Netw. Service Manag. (CNSM)*, 2018, pp. 72–80.

[33] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. Berlin, Germany: Springer, 2009.

[34] A. Krogh, "What are artificial neural networks?" *Nature Biotechnol.*, vol. 26, no. 2, pp. 195–197, Feb. 2008.

[35] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, and F. Sabrina, "Improving performance of autoencoder-based network anomaly detection on NSL-KDD dataset," *IEEE Access*, vol. 9, pp. 140136–140146, 2021.

[36] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manag. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015.

[37] J. M. Peterson, J. L. Leevy, and T. M. Khoshgoftaar, "A review and analysis of the bot-IoT dataset," in *Proc. IEEE Int. Conf. Service-Oriented Syst. Eng. (SOSE)*, Aug. 2021, pp. 20–27.

[38] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the Internet of Things," *Sensors*, vol. 19, no. 9, p. 1977, Apr. 2019, doi: 10.3390/s19091977.

[39] S. Alosaimi and S. M. Almutairi, "An intrusion detection system using BoT-IoT," *Appl. Sci.*, vol. 13, no. 9, p. 5427, Apr. 2023, doi: 10.3390/app13095427.

[40] K. Albulayhi, Q. Abu Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method," *Appl. Sci.*, vol. 12, no. 10, p. 5015, May 2022, doi: 10.3390/app12105015.

[41] Q. Abu Al-Haija and S. Zein-Sabatto, "An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks," *Electronics*, vol. 9, no. 12, p. 2152, Dec. 2020, doi: 10.3390/electronics9122152.

[42] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106432.

[43] S. Siddiqui, S. Hameed, S. A. Shah, I. Ahmad, A. Aneiba, D. Draheim, and S. Dustdar, "Toward software-defined networking-based IoT frameworks: A systematic literature review, taxonomy, open challenges and prospects," *IEEE Access*, vol. 10, pp. 70850–70901, 2022.

[44] M. Khalid, S. Hameed, A. Qadir, S. A. Shah, and D. Draheim, "Towards SDN-based smart contract solution for IoT access control," *Comput. Commun.*, vol. 198, pp. 1–31, Jan. 2023.

NADEEM SARWAR is currently pursuing the Ph.D. degree in computer science with The Islamia University of Bahawalpur, Pakistan. He is an Assistant Professor with the Department of Computer Science, Bahria University Lahore Campus, Lahore, Pakistan. He has 12 years of teaching and research experience. He has published more than 60 international and national journal/conference publications in the last five years. He is an Editorial Board Member of various research journals, such as *Computers, Materials and Continua*, *Applied Computational Intelligence and Soft Computing*, *PLOS ONE*, *IET Software*, *Security and Communication Networks*, *International Journal of Telemedicine and Applications*, *Journal of Healthcare Engineering*, *Milestone Transactions on Medical Technometrics*, *Journal of Mathematics and Computer Science*, and *SCIREA Journal of Computer*. He worked for more than 25 journals as a reviewer and PC member.

IMRAN SARWAR BAJWA received the Ph.D. degree in computer science from the University of Birmingham, U.K. He has more than 18 years of teaching and research experience in various universities of Pakistan, Portugal, and U.K. He is currently an Associate Professor with the Department of Computer Science and Information Technology, The Islamia University of Bahawalpur. He is the author/editor of 11 books published by IEEE, Springer, and IGI Global. He has more than 200 articles and 2800 citations in Google Scholar. In addition, he has more than 135 articles in Scopus and 1500 citations of the work in Scopus. His Google H-index is 30 and Scopus H-index is 21. His personal impact factor is more than 140. His current research interests include intelligent systems, the IoT, and data analytics. He has been an associate editor and a guest editor of various IEEE and Elsevier journals. He has very good programming skills in Java and C#.

MUHAMMAD ZUNNURAIN HUSSAIN is currently an Assistant Professor with the Department of Computer Science, Bahria University Lahore Campus, Lahore, Pakistan. He has ten years of proven a Network and Security Specialist with comprehensive lifecycle advisory, client development, management, and technology consulting for Fortune 100 clients. His research interests include computer networks, information security, the IoT security, cloud computing, and machine learning.

MUHAMMAD IBRAHIM received the Ph.D. degree in computer science from The Islamia University of Bahawalpur, Pakistan. He has more than ten years of teaching and research experience in various institutes of Pakistan. He is currently a Lecturer with the Department of Computer Science, The Islamia University of Bahawalpur. His current research interests include artificial intelligence, the IoT, big data, and machine learning.

KHIZRA SALEEM is currently a Lecturer with the Department of Computer Science, The Islamia University of Bahawalpur. Her research interests include machine learning, the IoT, and cloud computing.