Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 20 No. 10 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v20i10.49509

PAPER

A Systematic Investigation on Botnet Intrusion Detection Using Various Machine Learning Techniques

Archana Kalidindi(✉), Mahesh Babu Arrama

Department of CSE, Koneru Lakshmaiah Education Foundation, Hyderabad, Telangana, India

archana.buddaraju@klh.edu.in

ABSTRACT

The Internet of Things (IoT) is growing rapidly in an exponential manner due to its versatility in technology. This has led to many challenges in securing the IoT environment. Devices in IoT environments are vulnerable to various cyberattacks. Botnet-based attacks are predominant and widespread in nature. Due to insufficient memory and computational power, the IoT environment cannot handle the botnet attack that affects security. Identifying intrusions in IoT environments is another challenge for researchers. Finding unknown patterns in the data generated through IoT networks helps improve security in the IoT environment. Machine learning (ML) is a platform that helps identify patterns in the provided data. In this study, we present our research on classifying incoming data from the IoT as malicious or benign using machine learning techniques. We propose an ML-based botnet attack detection framework for nine commercial IoT devices that primarily target BASHLITE and Mirai botnet attacks. Rigorous pragmatic research was conducted on the N-BaIoT dataset, which was extracted from real-time IoT devices connected to a network. Using this framework, the results have been depicted, which can efficiently detect botnet attacks and can also be applied to any other types of attacks.

KEYWORDS

Internet of Things (IoT), botnet detection, machine learning (ML), N-BaIoT

1 INTRODUCTION

The basis for the evolution of Industry 4.0 was the remarkable achievements of Industry 3.0. Industry 4.0 mechanisms and methods have revolutionized the processes used in all product-based companies. This trend has driven industries to integrate various new technologies, such as the Internet of Things (IoT), artificial intelligence (AI), machine learning (ML), cloud-based computing, and more, into their operational processes. IoT is the prominent key to enabling the technology revolution [1]. The IoT is a vast universal information system comprising numerous diverse and distributed devices interconnected through the Internet. These devices can be identified, and the data from them can be sensed, processed, and shared with each other through a smart

Kalidindi, A., Arrama, M.B. (2024). A Systematic Investigation on Botnet Intrusion Detection Using Various Machine Learning Techniques. *International Journal of Online and Biomedical Engineering (iJOE)*, 20(10), pp. 18–32. https://doi.org/10.3991/ijoe.v20i10.49509

Article submitted 2024-04-04. Revision uploaded 2024-05-18. Final acceptance 2024-05-20.

© 2024 by the authors of this article. Published under CC-BY.

environment [2]. IoT is widely used in sectors such as healthcare, logistics, smart homes, smart cities, and supply chain management. According to an article in IoT Business News, the number of smart devices used in the IoT environment is projected to increase by 20 times in the near future, reaching an estimated 24.1 billion by 2030.

The coexistence and usage of these smart devices in the IoT environment have led to the generation of a huge volume of data that is shared on the network connecting all smart devices. The rapid growth and continued deployment of IoT devices have also led to a substantial increase in cybersecurity issues. Every two minutes, IoT network attacks are recorded, according to a Symantec report. According to a Forbes report, cyberattacks exceeded 2.9 billion in 2019, which is three times higher than in 2018. In a similar vein, the Kaspersky report shows a fourfold increase in malware samples from 2017 to 2018. The above exemplifies the increased vulnerability of IoT devices and their environments to botnet attacks. Despite their latent potential, IoT devices face numerous security threats, particularly botnet attacks caused by malware. There are several types of malware that exist to facilitate botnet attacks on IoT devices, such as Mirai [3], which occurred in 2016, and BASHLITE, which preceded Mirai and was reported in 2015 [4]. BASHLITE and Mirai are two popular malware programs that target authentication credentials by infecting numerous IoT devices. Both attacks turned the devices networked through Linux operating systems into remotely controlled 'bots' that are part of a botnet attack in larger networks. The attacks caused by Mirai [3] and BASHLITE are orchestrated by the domain name system provider Dyn. These attacks involve multiple distributed denial of service (DDoS) operations that attempt to deplete system resources such as CPU time, memory, and internet bandwidth [5]. Due to this, servers such as GitHub, Twitter, and many others were inaccessible. These attacks used a botnet consisting of many IoT devices, including IP security cameras, baby monitors, gateways, doorbells, etc.

Botnet attacks are controlled and commissioned by a command-and-control (C&C) mechanism. Another major attack was called Stuxnet [6], which was a worm created to damage Iran's nuclear program. The rise in IoT applications has resulted in new advancements in botnet attacks, which present security and privacy threats. Botnet attacks are very serious and severe in nature, spreading rapidly among IoT devices connected to the internet. Consequently, there is a need for a decisive methodology that helps protect IoT devices from botnet attacks. Motivated by the increasing number of vulnerabilities and information leaks in IoT devices due to botnet attacks, researchers have conducted numerous investigations leading to the identification of several potential mechanisms [7] [8]. The intrusion detection system (IDS) is a solution that helps detect botnet attacks. IDS is an application that monitors and analyzes large volumes of data to identify any malicious activity or botnet attacks on a network. Traditional IDS consist of anomaly detection methods and misuse or signature detection methods. Signature-based detection methods learn about attack signatures and are commonly found in all public IDS. IDS such as Snort and Suricata [9] fall under signature-based IDS. These IDS cannot work efficiently with unidentified attacks and modifications made to known attacks. Anomaly-based systems work efficiently with both unknown and known attacks.

The IoT environment presents challenges such as limited storage, diverse IoT networks, and varying computing capabilities. For this object, the simple IDS cannot be integrated with the IoT environment. Another issue with IoT environments is the challenge of collecting standard data because of the presence of diverse devices in the environment. To meet the requirements of efficient IDS in an IoT environment, numerous recent studies have investigated the potential of ML and DL methodologies. ML and DL are powerful tools that fall under the umbrella of AI. Flexibly dynamic systems that can learn from new inputs can be created using ML and DL [10]. Various techniques in ML and DL are used to build applications that are highly effective and efficient [11] [12]. ML and DL techniques help in modeling the system by extracting knowledge from features present in large volumes of data. They predict normal and abnormal activities based on learned patterns as part of knowledge discovery.

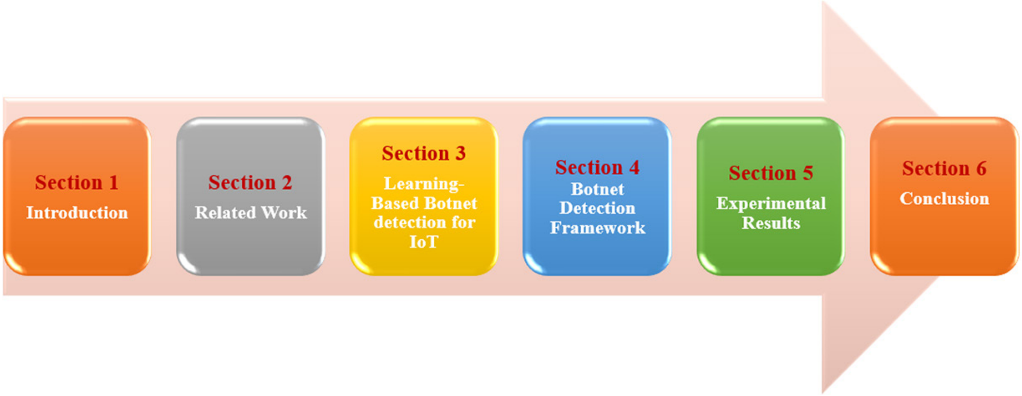


Fig. 1. Organization of study

The focus of this study is to apply ML techniques for the operational detection of botnet attacks in IoT network environments. The key contribution of this work is as follows:

- To predict attacks on nine IoT devices infected by ten different types of attacks using machine learning algorithms.
- Feature selection methods are advocated for working with relevant features. Another advantage of using a feature selection method is that it makes the entire process or system lightweight by ignoring irrelevant features.
- The proposed methodology has achieved higher accuracy with lower throughput and lightweight performance. It can analyze and work with large volumes of data.

Figure 1 illustrates the organization of the entire study of current research work to maintain a logical flow of content. Acronyms with their meanings used in the work are presented in Table 1. After discussing the introduction to the entire research work in Section 1, the remaining content of the paper is structured as follows: In Section 2, we summarize the context and motivation of this research using existing methodologies in the same domain of study. Section 3 discusses botnet detection for the IoT. The details of the methodology and framework proposed for detecting botnets in IoT environments are explained in Section 4. Section 5 validates the investigational findings, and Section 6 concludes the study by providing directions for future research.

Table 1. Acronyms with their meanings

Acronym	Meaning
IoT	Internet of Things
ML	Machine Learning
IDS	Intrusion Detection System
DL	Deep Learning
ANN	Artificial Neural Network
MLP	Multilayer Perceptron

2 RELATED WORK

In the realm of botnets, the primary concern is to enhance security measures to safeguard and protect numerous IoT devices that are being compromised. The author in [13] sheds light on the Mirai botnet as a prominent example that is vulnerable to DDoS attacks. It is high time to establish efficient and updated mechanisms for IoT devices, and IoT users indeed need to periodically update their devices. Glitches can be identified, and potential botnet activity can be mitigated by analyzing network traffic patterns. The interaction among various community members is considered a crucial issue for addressing the problems associated with such types of attacks. In line with this [14], a framework was developed, discussing three major modules: the lifecycle of a botnet and the infrastructure of a botnet, where the bot masters communicate with bots in the botnet through channels, including C&C servers used to coordinate botnet activities among various bots on different computers. The framework emphasizes the need for effective counteractions to target and disrupt the components of botnets. Lastly, the focus is on the various techniques used by botnets to compromise and control botnet components. It has been stated in [15] that deep autoencoders have the capability to extract essential data representations for identifying patterns and anomalies in botnets. Experimental results demonstrate the effectiveness of using deep autoencoders to detect IoT botnet attacks. ML holds a prominent position by providing numerous solutions to address complex problems. Among the various solutions to botnet attacks, decision trees stand out as a popular approach. They learn from data within a specific zone and build models to address issues through systematic analysis. It could adapt to a distributed environment when dealing with massive datasets. Studies by Patil present an elaborated form of a distributed decision tree algorithm introduced to outperform existing models in terms of results [16]. By utilizing various metrics, one can demonstrate superiority in the performance achieved by SAPSO-SVM. The proposed methodology in [17] involves the top 20 features of benchmark Android botnets, including two major features: one that is not corrupted by botnets and the other selected based on the best outcomes. The distance between the respective particles is calculated within the space during the execution steps of the SAPSO algorithm. This work fruitfully identified the specific and major 20 features of Android botnets by leveraging the best results obtained from analyzing the 28th Android botnet. The exploitation of network environments is increasing rapidly due to the rapid advancements in networking technologies.

By commissioning malicious attacks on the network, there is a high risk for hackers to exploit vulnerabilities, potentially compromising the network's security. The work in [18] introduced an innovative methodology to distinguish between regular network traffic and botnet attack traffic using artificial neural networks (ANN), gated recurrent units (GRU), and long- or short-term memory models. The GRU model is capable of handling huge datasets and provides 99.7% accuracy, but it is technically costly when used with Bot-IoT. On the other hand, with the same dataset, ANN is relatively cost-effective and provides 97% accuracy, which can effectively guide the detection of intrusive attacks in the network. In the field of networking, botnets have emerged as a major security issue that may cause significant concerns regarding personal data, infrastructure, financial data, etc. The CICIDS2017 dataset is a benchmark dataset derived from the Canadian Institute of Cyber Security, characterized by an imbalanced data distribution with a ratio of 9:1. This imbalance poses a challenge in terms of features and raises concerns about overfitting problems. In point [19], this problem is addressed using the J48 decision tree algorithm and the SMOTE technique, resulting in a significant value of 99.95%, which validates the result and proves the research determination. With an estimated 80 billion online devices by 2024 surpassing the worldwide human population, the rapid growth of IoT has brought us many more challenges in terms of security. Our responsibility has increased to ensure IoT security, which has become a critical concern. The purpose of the work in [20] is to develop a comprehensive ML algorithm-based model capable of detecting and mitigating botnet-based intrusions. To pursue the goal, the author utilized the BoT-IoT dataset and experimented with various ML models, including linear regression, logistic regression, K-nearest neighbor (KNN), and support vector machine (SVM) models. The results provide efficient values and distinguish between normal and malicious network activities, achieving high F-measure scores of 98.0%, 99.0%, 99.0%, and 99.0%, respectively, for the different models. In [4], they used the NF-BoT-IoT-v2 benchmark dataset to evaluate and predict DDoS attacks, which are typical issues in IoT networks. The dataset comprises 37,763,497 records and includes five different types of attacks. The classification tasks were conducted using WEKA and MATLAB tools. Decision tree (J48), Naive Bayes (NB), and random forest (RF) machine learning classifiers are utilized. For dimension reduction in network intrusion detection systems (NIDS). Different linear and non-linear activation functions were considered for the hidden and output layers. The Adam optimizer and mean squared error loss functions were used for learning optimization. Classification accuracies were assessed using the SVM-RBF classifier on the CICIDS2017 dataset, which contains contemporary attacks in cloud environments. The results showed that ELU achieved a low computational overhead and a negligible difference in accuracy (97.33%) compared to other activation functions. The research framework proposed for NIDS in the cloud environment [15] integrated constraints optimized stacked autoencoders (COSAE) and conventional classification techniques. Activation functions (such as rectified linear unit (ReLU), SeLU, Softplus, and ELU) are used for feature learning. The SAE+GSVM-RBF model achieves significant computational time gains compared to SAE+SVM-RBF, with marginal performance differences. COSAE+GSVM-RBF with ELU balances prediction time and accuracy. AUC results are high for all activation functions, with Softplus performing the best. The study concludes that the COSAE+GSVM-RBF with ELU is suitable for efficient NIDS with good accuracy. It addresses gradient issues and achieves a 100% F-measure for minor class labels.

3 LEARNING-BASED BOTNET DETECTION FOR IoT

Botnet detection is one of the most effective methods to protect IoT environments from unwanted infections. The goal of achieving a zero-attack day requires the capability to intelligently track all data flowing in and out of all IoT devices in the network. The ML and DL techniques will assist in analyzing the data collected from IoT components in the environment and aid in making early decisions regarding normal or malicious activities. With the help of insights gained from current events, ML-based botnet detection can intelligently anticipate unknown attacks [19]. With the help of intelligence derived from ML and DL, existing secure communication between devices in the IoT environment must advance towards a security-focused IoT environment. This is possible through learning-based techniques that are part of ML and DL [19]. Algorithms in ML can be trained to identify attacks in an IoT environment. ML algorithms are mainly categorized into three types. Supervised, unsupervised, and reinforcement learning [21], [22]. Our primary focus in this research is on supervised learning strategies. This section provides a brief overview of the various commonly used supervised ML algorithms for predicting botnet attacks, such as J48, ANN, and Naïve Bayes [21].

3.1 Artificial neural networks

Artificial neural network is a computational model that imitates and function analogously to the nerve cells in the human brain. ANN is created with the help of parallel functioning layers of neurons maintained by vector-valued functions. These layers are of three types. Input, hidden, and output layers. Multilayer perceptron (MLP) is a fully connected type of ANN. The output generated by a neuron using an activation function serves as input to another neuron. The most predominant transfer functions include sigmoid, rectified linear unit (ReLU), and tanh, as shown in equations 1 to 3 respectively.

$$g(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$g(x) = max(0,x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \tag{2}$$

$$g(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^{x} - e^{x-x}}{e^{x} + e^{-x}} \tag{3}$$

3.2 J48

Decision tree algorithms are classification models used for predictive analysis. J48 is a successor to C4.5 and a descendant of the ID3 algorithm. Data distribution can be easily understood with decision tree structures. J48 is no exception. J48 uses information gain, as shown in equation 4, and entropy, as shown in equation 5, for the construction of the tree.

$$IG(T,A) = Entropy(T) - \sum_{V \epsilon A} \frac{|T_V|}{T} * Entropy(T_V) \tag{4}$$

$$Entropy = -\sum_{i=1}^{c} P(X_i) log_b P(X_i) \tag{5}$$

3.3 Naïve Bayes

Naïve Bayes is a supervised ML algorithm that is based on the Bayesian theorem. It works with a probabilistic approach and assumes that all variables available in the dataset are correlated and play a role in classification. It works well in a dataset with high dimensionality. In Equation 6, $P(X)$ represents the prior probability, which denotes the probability of the hypothesis before observing the evidence. Similarly, $P(Y)$ stands for the marginal probability, indicating the probability of evidence [21].

$$P(X|Y) = \frac{P(X|Y)P(X)}{P(Y)} \tag{6}$$

4 BOTNET DETECTION FRAMEWORK AND METHODOLOGY

This section presents the framework for botnet detection in IoT environments and elucidates the dataset used in the research. Figure 2 illustrates the overall architecture of the project. This dataset, which contains attack data, was recently released. Devices in the IoT network were infiltrated and targeted by two well-known attacks, Mirai [3] and BASHLITE. Algorithm 1, as shown in Figure 3, was utilized to balance the instances of the aforementioned attacks in the dataset, given the unbalanced distribution of data.

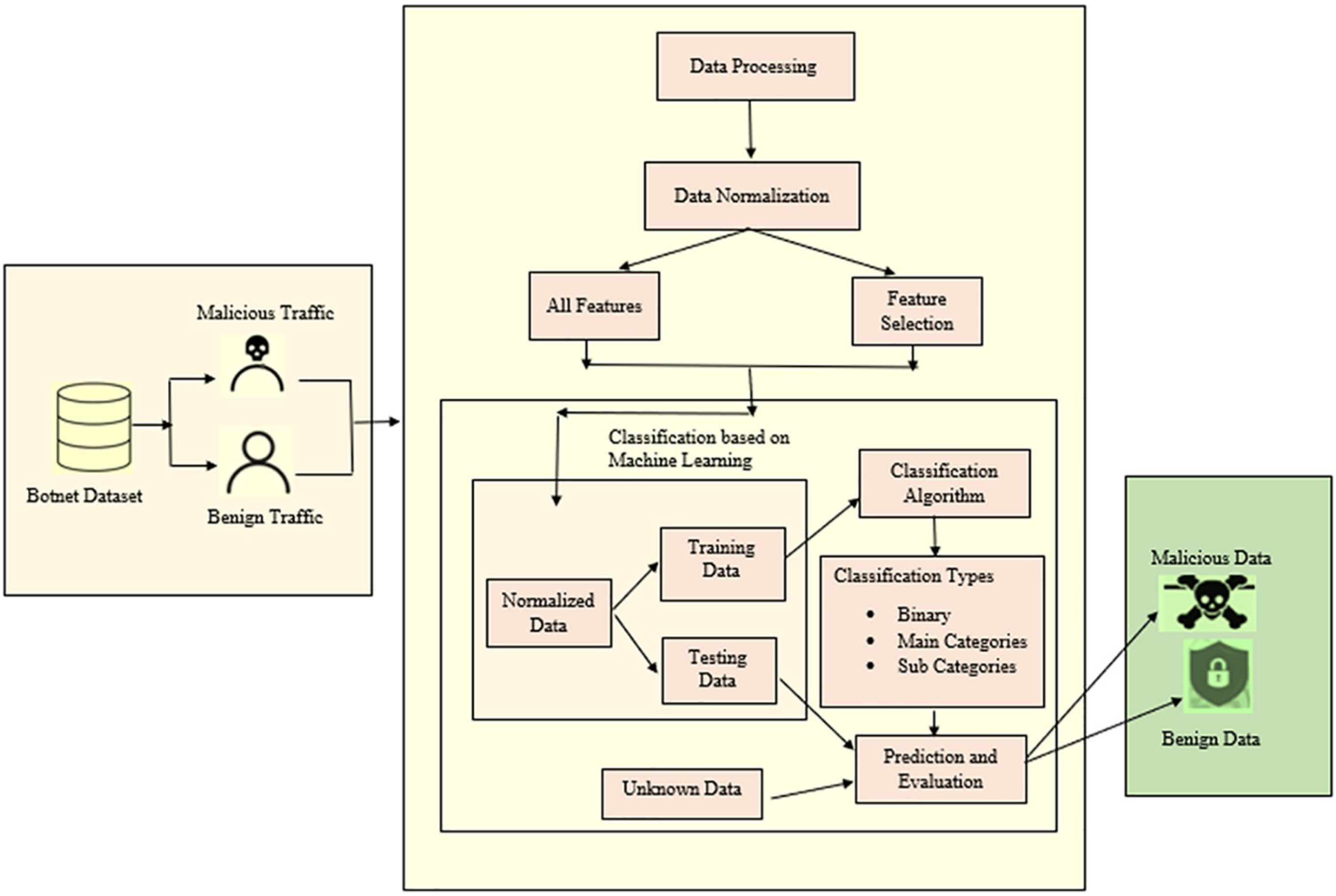
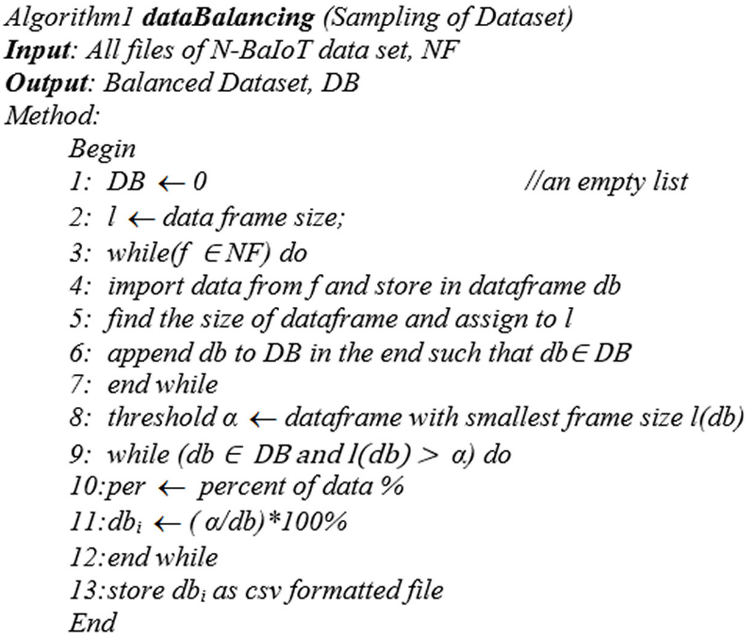


Fig. 2. Botnet detection framework



```
Algorithm1 dataBalancing (Sampling of Dataset)
Input: All files of N-BaIoT data set, NF
Output: Balanced Dataset, DB
Method:
        Begin
          DB ← 0                                   //an empty list
          l ← data frame size;
          while(f ∈NF) do
          import data from f and store in dataframe db
          find the size of dataframe and assign to l
          append db to DB in the end such that db∈ DB
          end while
          threshold α ← dataframe with smallest frame size l(db)
          while (db ∈ DB and l(db) > α) do
         per ← percent of data %
         $db_i$ ← ( α/db)*100%
         end while
         store $db_i$ as csv formatted file
        End
```

Fig. 3. Algorithm 1 sampling dataset

At first, the botnet behavior is studied manually by splitting the entire dataset into benign and malicious data. This study has helped in understanding the impact of various features in botnet attacks on IoT environments. Initially, the classifiers were trained and tested without using feature selection methods. As a result, we could measure the efficiency of using feature selection methods and explore the impact of feature selection on the performance of the entire system. As a next step, feature selection methods were also pragmatic in selecting the relevant features. Finally, as a last step, the accuracy of attack detection is reported. The model was tested for two situations: binary classification and multi-class classification. In the case of multi-class classification, the model was tested for two scenarios: one for the main attack category and the other for the subcategories of the main attack. Different stages of the model are mentioned in the following sections.

4.1 Data pre-processing

This is the initial and crucial phase of the entire process, which helps in eliminating defects in the dataset [23]. The dataset, once preprocessed, will be in the format required for further analysis. Consequently, this study proposes the following pre-processing steps: min-max normalization and label encoding.

Min-max normalization. The performance of machine learning is affected when the dataset has a wide range of values. Deteriorating performance is observed when there is an imbalance in attribute values, as this leads to improper fitting of data in the model [24]. The aim of normalization is to bring all attribute values to a common scale.

Min-Max normalization is one of the most popular methods of normalization. By using this method, all attribute values will be transformed into a [0, 1] scale, where 0 represents the minimum value and 1 represents the maximum value. It is a linear transformation technique. Min-max normalization is performed using Equation 7.

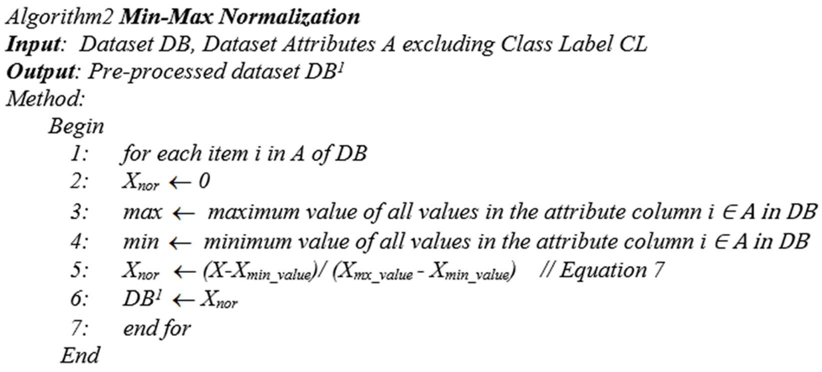
$$v' = \frac{v - min(A)}{max(A) - min(A)} \tag{7}$$

Where v' represents the normalized value, $min(A)$ and $max(A)$ are the values of the internal, which are 0 and 1, respectively. v is the original value that needs to be transformed into the specified range. Algorithm 2 presented in Figure 4 depicts the pseudocode for min-max normalization.

Label encoding. Machine learning models cannot be directly fed with categorical values. So, the categorical class label values of the dataset are managed using label encoding. Each value of the class label is assigned a unique integer number. This technique does not affect the dimensionality of the dataset. The Scikit-learn package in Python contains a label encoder that is used for label encoding [25]. This scalar uses the equation 8 for scaling.

$$y\,scaled = \frac{y - mean(CL)}{SD(CL)} \tag{8}$$

Where y is the value of the class label, and CL is the class label. Each value y of class label CL is derived by subtracting the mean of every value of class label CL and finally dividing by the standard deviation of every class label's values.

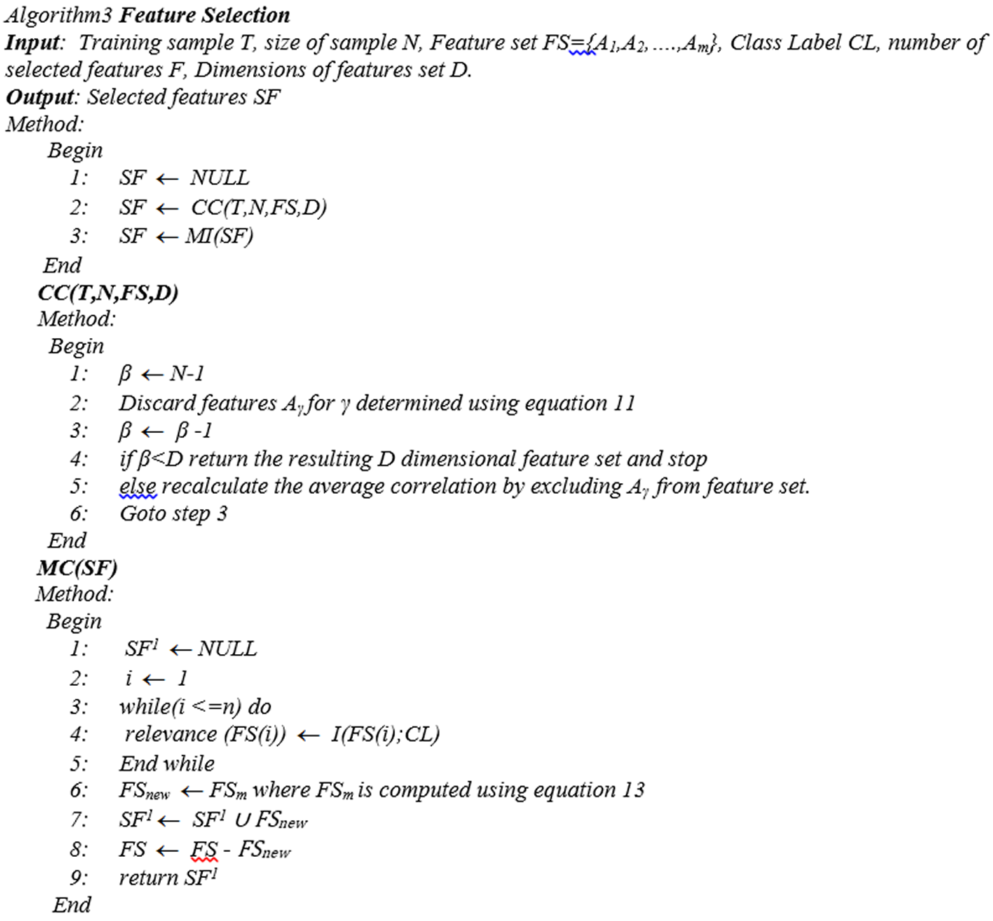


```
Algorithm2 Min-Max Normalization
Input:  Dataset DB, Dataset Attributes A excluding Class Label CL
Output: Pre-processed dataset $DB^1$
Method:
    Begin
      1:   for each item i in A of DB
      2:   $X_{nor} \leftarrow 0$
      3:   max $\leftarrow$ maximum value of all values in the attribute column i $\in$ A in DB
      4:   min $\leftarrow$ minimum value of all values in the attribute column i $\in$ A in DB
      5:   $X_{nor} \leftarrow (X-X_{min\_value})/(X_{mx\_value} - X_{min\_value})$   // Equation 7
      6:   $DB^1 \leftarrow X_{nor}$
      7:   end for
     End
```

Fig. 4. Algorithm 2 min-max normalization

4.2 Feature selection

The N-BaIoT dataset is a high-dimensional vector consisting of many features and class labels, which are discussed in detail in Section 5.1. This study employs a filter feature selection technique based on correlation with mutual information. In this paper, we utilized the correlation coefficient as the initial step to measure the similarity between features of a dataset while excluding the class label. We used the absolute correlation coefficient value to identify redundant data items among different features. This coefficient will remain constant without being affected by scaling or translation.



```
Algorithm3 Feature Selection
Input:  Training sample T, size of sample N, Feature set $FS=\{A_1,A_2,\ldots,A_m\}$, Class Label CL, number of
selected features F, Dimensions of features set D.
Output: Selected features SF
Method:
    Begin
      1:   SF $\leftarrow$ NULL
      2:   SF $\leftarrow$ CC(T,N,FS,D)
      3:   SF $\leftarrow$ MI(SF)
    End
  CC(T,N,FS,D)
  Method:
    Begin
      1:   $\beta \leftarrow$ N-1
      2:   Discard features $A_\gamma$ for $\gamma$ determined using equation 11
      3:   $\beta \leftarrow \beta$ -1
      4:   if $\beta$<D return the resulting D dimensional feature set and stop
      5:   else recalculate the average correlation by excluding $A_\gamma$ from feature set.
      6:   Goto step 3
    End
  MC(SF)
  Method:
    Begin
      1:    $SF^1 \leftarrow$ NULL
      2:    i $\leftarrow$ 1
      3:   while(i <=n) do
      4:    relevance (FS(i)) $\leftarrow$ I(FS(i);CL)
      5:   End while
      6:   $FS_{new} \leftarrow FS_m$ where $FS_m$ is computed using equation 13
      7:   $SF^1 \leftarrow SF^1 \cup FS_{new}$
      8:   FS $\leftarrow$ FS - $FS_{new}$
      9:   return $SF^1$
    End
```

Fig. 5. Algorithm 3 feature selection

The correlation coefficient (μ_{ab}) between two features is computed using the Equation 9,

$$\mu_{ab} = \frac{E\{[A - E(A)][B - E(B)]\}}{\sqrt{V(A)} * \sqrt{V(B)}} \tag{9}$$

In Equation 9, the numerator is used to compute the covariance between features *A* and *B*. *E* represents the expected value of the feature. *V*(*A*) represents the variance of feature *A*, while *V*(*B*) represents the variance of feature *B*. When two features are not correlated, the value of μ_{ab} will be 0. Correlation coefficients for all pairs of features are computed, and the average correlation of a feature is calculated over β features using Equation 10.

$$\mu_{j,\beta} = \frac{1}{\beta} \sum_{i=1, i \neq j}^{\beta} | \mu_{a_i, a_j} | \tag{10}$$

The feature with highest average correlation will be eliminated at every step using Equation 11.

$$\gamma = arg\ max_j\, \mu_{j,\beta} \tag{11}$$

Mutual information is a concept that has evolved from information theory. It measures the amount of information obtained about one feature by observing the other feature. This research applies mutual information to measure the association between classes and features. We selected a subset of features, *SS*, from the pre-processed dataset, *DB*, with *Y* rows and *A* features in a way that ensures this subset has the highest mutual information value with the class label, *CL*. This is computed using Equation 12,

$$I(SS;CL) = \sum_{Y_1, Y_2, \ldots, Y_f, CL} P\left(Y_1, Y_2, \ldots, Y_f, CL\right) log\left(\frac{P\left(Y_1, Y_2 \ldots, Y_f, CL\right)}{P\left(Y_1, Y_2 \ldots, Y_f, CL\right) P(CL)} \right) \tag{12}$$

$$FS_m = arg\ max_{FS_m \in FS}\, relevance(FS_m) \tag{13}$$

Where *SS* is the final subset selected with the class label *CL* using mutual information. *O*(*M*) is the time complexity for computing mutual information, where *M* represents the total number of samples in the dataset. The feature selection algorithm is presented as Algorithm 3 in Figure 5.

4.3 Classification algorithms

This study mainly focuses on utilizing logistic regression, linear regression, SVM, stacking algorithms, voting algorithms, and decision trees for predicting botnet attacks. Random forest and XGBoost are ensembling classifiers, along with a decision tree classifier. For improving the performance of classifiers, hyperparameter optimization is conducted using random search and grid search.

5 EXPERIMENTAL RESULTS

5.1 Dataset

The N-BaIoT dataset is considered for the study, and the system is tested using this dataset. The data is collected from nine commercial IoT devices in real-time. This data is infected by two of the most common botnet attacks called Mirai [3] and BASHLITE. This dataset consists of 115 features. 23 features were extracted at discrete time intervals. The dataset is from [13].

Features of N-BaIoT are broadly classified as packet count (PC), time between packet arrivals (TPA), time delay in delivering packets (TDP), and size of packets (inbound and outbound) (PS). A total of 23 features are generated by applying statistical measures such as mean, variance, magnitude, integer, radius, covariance, and correlation coefficient to these four varieties of features. Considering five time windows, the first one with 100 milliseconds, the second one with 500 milliseconds, the third one with 1.5 seconds, the fourth one with 10 seconds, and finally the fifth window with one minute over each of the available 23 features, a total of 115 features are extracted.

5.2 Assessment metrics

The system is equipped with various metrics to analyze and study the performance of algorithms used for predicting attacks. Accuracy, precision, recall, and F1-score are metrics used to evaluate the predictive performance of a system. The description and equations are provided below.

Accuracy: Accuracy is a metric that measures the overall correctness of a classification model by calculating the ratio of correctly predicted instances to the total number of instances.

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN} * 100\%$$

Where *TP* represents true positives, *FP* represents false positives, *TN* represents true negatives, and *FN* represents false negatives.

Precision: Precision is the measure of the proportion that predicts positive instances that are true positives, that signifies the facility of the model to avoid false positives.

$$Precision = \frac{TP}{TP + FP} * 100\%$$

Recall: Recall, which is also termed sensitivity or true positive rate, measures the proportion of actual positive instances that are correctly identified by the model.

$$Recall = \frac{TP}{TP + FN} * 100\%$$

F-score: The F-score, also termed the F1-score, is a metric that pools precision and recall into a single value to weigh the model's performance. It stabilizes the trade-off between precision and recall.

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} *100\%$$

Kappa score: The Kappa score, or Cohen's Kappa, is a metric that assesses the contract between the predicted and actual labels, taking into account the contract that could occur by chance alone. It is useful to balance the datasets.

5.3 Analysis and evaluation

The entire dataset is divided into two groups. Group 1 consists of 80% of the data for training, while Group 2 consists of 20% of the data for testing. Knowledge is acquired through the use of various algorithms with the training dataset. The efficiency of the system is evaluated using a testing group. Figure 6 depicts the correlation matrix, which is a square matrix displaying the pairwise correlations between the different classes. Each component illustrates the strength and direction of the linear relationship between two variables.

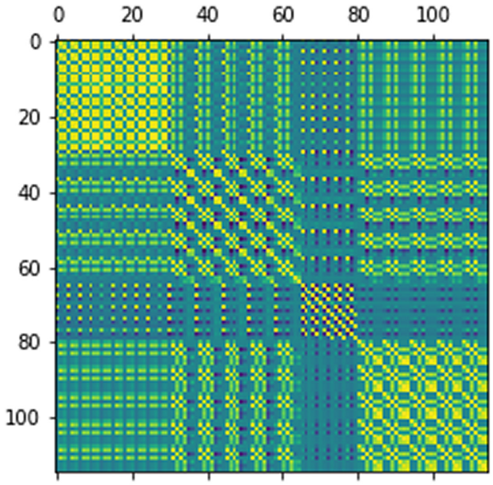


Fig. 6. Correlation matrix

Figure 6 presents the confusion matrix for the linear regression algorithm, decision tree algorithm, stacking algorithm, voting classifier, and SVM. The confusion matrix is a table used for multi-class classification problems. It summarizes the performance of a classification model by displaying the counts of true positives, true negatives, false positives, and false negatives for each class.

Table 2. Comparison of algorithms

Algorithm	Accuracy	Precision	Recall	F1 Score	Kappa Score
Decision Tree	90.8	95.3	90.8	87.7	89.8
Logistic Regression	64.3	60.8	64.3	60.4	60.7
Linear Regression	48.5	14.0	18.1	13.6	10.0
SVM	81.8	91.0	81.8	79.4	79.9
Voting Classifier	90.8	95.3	90.8	87.7	89.8
Stacking Classifier	90.8	89.8	90.8	87.7	89.8

Table 2 shows the investigative results comparing the algorithms using various measures such as kappa score, F1 score, recall, precision, and accuracy. The decision tree yields better results, as depicted in Figure 7, enhancing transparency and clarity.

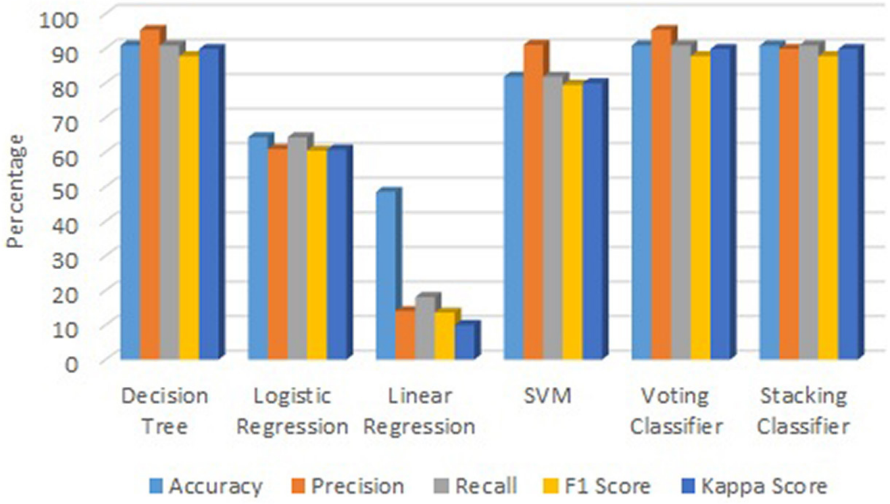


Fig. 7. Comparison of all algorithms

6 CONCLUSION

The widespread use of IoT devices in networks has led to an increase in botnet attacks. There is a need for an effective attack detection system to protect data transmitted by IoT devices within a network. We proposed an efficient system utilizing machine learning algorithms to decrease the time and cost of identifying vulnerable attacks caused by botnets. Experimental results from Section 5 demonstrate that the voting classifier exhibits good accuracy, precision, recall, F-score, and Kappa score. The achieved values for these metrics are 90.8, 95.3, 90.8, 87.7, and 89.8, respectively. This work can be further extended to develop new techniques that can identify time-related features and their temporal behavior in environments where botnet attacks are possible.

7 REFERENCES

[1] C. Zhang and Y. Chen, “A review of research relevant to the emerging industry trends: Industry 4.0, IoT, blockchain, and business analytics,” *Journal of Industrial Integration and Management*, vol. 5, no. 1, pp. 165–180, 2020. https://doi.org/10.1142/S2424862219500192

[2] S. Cirani, M. Picone, and L. Veltri, “mjCoAP: An open-source lightweight Java CoAP library for internet of things applications,” in *Interoperability and Open-Source Solutions for the Internet of Things*, *Lecture Notes in Computer Science*, I. Podnar Žarko, K. Pripužić, and M. Serrano Eds., Springer International Publishing, 2015, vol. 9001, pp. 118–133. https://doi.org/10.1007/978-3-319-16546-2_10

[3] A. Marzano *et al.*, “The evolution of Bashlite and Mirai IoT botnets,” in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 00813–00818. https://doi.org/10.1109/ISCC.2018.8538636

[4] Y. Meidan *et al.*, “N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders,” *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018. https://doi.org/10.1109/MPRV.2018.03367731

[5] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013. https://doi.org/10.1109/SURV.2013.031413.00127

[6] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy Magazine*, vol. 9, no. 3, pp. 49–51, 2011. https://doi.org/10.1109/MSP.2011.67

[7] P. Kumar, A. Moubayed, A. Refaey, A. Shami, and J. Koilpillai, "Performance analysis of SDP for secure internal enterprises," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, Marrakesh, Morocco, 2019, pp. 1–6. https://doi.org/10.1109/WCNC.2019.8885784

[8] H. Hindy *et al.*, "A taxonomy of network threats and the effect of current datasets on intrusion detection systems," *IEEE Access*, vol. 8, pp. 104650–104675, 2020. https://doi.org/10.1109/ACCESS.2020.3000179

[9] S. A. R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to Snort system," *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018. https://doi.org/10.1016/j.future.2017.10.016

[10] A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya, and A. Shami, "E-learning: Challenges and research opportunities using machine learning & data analytics," *IEEE Access*, vol. 6, pp. 39117–39138, 2018. https://doi.org/10.1109/ACCESS.2018.2851790

[11] A. Moubayed, M. Injadat, A. Shami, and H. Lutfiyya, "Student engagement level in an e-learning environment: Clustering using K-means," *American Journal of Distance Education*, vol. 34, no. 2, pp. 137–156, 2020. https://doi.org/10.1080/08923647.2020.1696140

[12] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Systematic ensemble model selection approach for educational data mining," *Knowledge-Based Systems*, vol. 200, p. 105992, 2020. https://doi.org/10.1016/j.knosys.2020.105992

[13] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017. https://doi.org/10.1109/MC.2017.201

[14] J. Leonard, S. Xu, and R. Sandhu, "A framework for understanding botnets," in *International Conference on Availability, Reliability and Security,* Fukuoka, Japan, 2009, pp. 917–922. https://doi.org/10.1109/ARES.2009.65

[15] A. Mughaid *et al.*, "Utilizing machine learning algorithms for effectively detection IoT DDoS attacks," in *Lecture Notes in Networks and Systems*, Springer Nature Switzerland, 2023, pp. 617–629. https://doi.org/10.1007/978-3-031-33743-7_49

[16] S. Patil and U. Kulkarni, "Accuracy prediction for distributed decision tree using machine learning approach," in *3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2019, pp. 1365–1371. https://doi.org/10.1109/ICOEI.2019.8862580

[17] M. Moodi, M. Ghazvini, H. Moodi, and B. Ghavami, "A smart adaptive particle swarm optimization–support vector machine: Android botnet detection application," *The Journal of Supercomputing*, vol. 76, pp. 9854–9881, 2020. https://doi.org/10.1007/s11227-020-03233-x

[18] R. Biswas and S. Roy, "Botnet traffic identification using neural networks," *Multimedia Tools and Applications*, vol. 80, pp. 24147–24171, 2021. https://doi.org/10.1007/s11042-021-10765-8

[19] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020. https://doi.org/10.1109/COMST.2020.2988293

[20] A. D. Khaleefah and H. M. Al-Mashhadi, "Detection of IoT botnet cyber attacks using machine learning," *Informatica*, vol. 47, no. 6, pp. 55–64, 2023. https://doi.org/10.31449/inf.v47i6.4668

[21] S. Ray, "A quick review of machine learning algorithms," in I*nternational Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, 2019, pp. 35–39. https://doi.org/10.1109/COMITCon.2019.8862451

[22] G. Shaheamlung, H. Kaur, and M. Kaur, "A survey on machine learning techniques for the diagnosis of liver disease," in *International Conference on Intelligent Engineering and Management (ICIEM)*, London, UK, 2020, pp. 337–341. https://doi.org/10.1109/ICIEM48762.2020.9160097

[23] E. A. Felix and S. P. Lee, "Systematic literature review of preprocessing techniques for imbalanced data," *IET Software*, vol. 13, no. 6, pp. 479–496, 2019. https://doi.org/10.1049/iet-sen.2018.5193

[24] A. Mahfouz, A. Abuhussein, D. Venugopal, and S. Shiva, "Ensemble classifiers for network intrusion detection using a novel network attack dataset," *Future Internet*, vol. 12, no. 11, p. 180, 2020. https://doi.org/10.3390/fi12110180

[25] T. Al-Shehari and R. A. Alsowail, "An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques," *Entropy*, vol. 23, no. 10, p. 1258, 2021. https://doi.org/10.3390/e23101258

8 AUTHORS

Archana Kalidindi is a research scholar in the Department of Computer Science and Engineering at Koneru Lakshmaiah Deemed to be University, Hyderabad, India. She holds a Bachelor's Degree in Information Technology, which was completed in 2007, and a Master's Degree in software engineering, which is obtained in 2011. Currently, she is pursuing her Ph.D. in computer science with a specialization in the field of machine learning and deep learning. Her research interests lie in the areas of deep learning, data science, and machine learning, and software engineering. Throughout her academic journey, she has published numerous research articles in prestigious journals and conferences, contributing to the advancement of knowledge in her field. She is a highly committed and motivated teacher who possesses excellent problem-solving abilities, is focused on achieving her objectives, and has a genuine dedication to fostering the social and academic progress of each student she works with (E-mail: archana.buddaraju@klh.edu.in).

Dr. Mahesh Babu Arrama earned his Doctorate in Computer Science and Technology from Sri Krishna Deveraya University, Anatapur, Andhrapradesh. He is currently working as a Professor in the CSE Department at Koneru Lakshmaiah Deemed University, Hyderabad. He has more than 20 years of experience in teaching, and industry research. He published eight research publications in reputed journals and participated in various National and International Conferences. He served as a senior manager in HCL, and he was also involved in the Project IWTS (Infantry Weapon Training Simulator) of the Indian Army. Apart from research and training activities, he also served in many administrative roles, like Associate Dean for Skill Development of the Current University and Center Academic Head with Aptech Computer Education. His research interests include cloud computing, security, and the IoT. He published various articles in the Namaste Telangana Newspaper on career guidance. He is a dedicated, resourceful, and goal-driven professional educator with a solid commitment to the social and academic growth and development of every student (E-mail: maheshbabu.a@klh.edu.in).