An Efficient DDoS Attack Detecting System using Levenberg-Marquardt Based Deep Artificial Neural Network Approach for IOT

Ahmed Saeed Alzahrani

Abstract: ***The Internet of Things model envisions the widespread interconnection and collaboration of smart devices over the present and future Internet environment. Threats and attacks against IoT devices and services are on the rise due to their rapid development. Distributed-Denial-of-Service (DDoS) attacks are one of the main dangerous malwares that attack targeted organizations through infected devices. Many mechanisms are developed for IoT devices in order to detect DDoS attacks. Nonetheless, the prevailing DDoS Attack Detection (DAD) methods involve time-delay and a lower detection rate. This paper proposed an efficient approach using the Levenberg-Marquardt Neural Network (LMDANN) algorithm for detecting the DDoS attacks in order to enhance prediction accuracy. In the proposed system, a MapReduce technique is used to eliminate the redundant copies. In addition, the Entropy-based Fisher's Discriminate Function (ENTFDF) method was developed to reduce the features from the extracted features, and the system suggests an LMDANN algorithm to classify DDoS attack data separately from the normal data. In this, 80% of the data is used for training, and 20% of the data is used for testing. The performance of the proposed LMDANN method was evaluated in contrast to other art of state algorithms (ANN, SVM, KNN, and ANFIS) in terms of some specific qualitative performance metrics (recall, sensitivity, f-measure, specificity, precision, accuracy, and training time). The results show that the proposed detection approach can efficiently detect the DDoS attack in the IoT environment, achieving 96.35% accuracy.***

Keywords: ***Distributed Denial of Service (DDoS), Data Deduplication, Hadoop Distributed File System (HDFS), Feature-based MinMax (F-MinMax), Entropy-based Fisher's Discriminant Function (ENTFDF), and Levenberg-Marquardt based Deep Artificial Neural Network (LMDANN).***

I. INTRODUCTION

The Internet of things is one of the most promising recent technology developments, which allowed for the grouping, processing, and exchanging of data among smart applications [1, 2]. Along with the rapid development of IoT networks, attacks on these networks have increased remarkably, particularly DDoS attacks [3, 4], which have harmed numerous IoT networks and caused significant damage. In DDoS attacks, the attacker chooses a master (called a Bot) for his attack, and an IoT device, for instance a computer, smartphone, etc., and then hacks them. By hacking those devices, the hacker will take total control of that IoT device [5]. Then, the attacker utilizes that DDoS master in order to hack several systems on the network [6, 7]. Hackers prefer such attacks because they are easily used to target large-scale networks and widespread websites in order to deactivate them [8]. As a result, a successful attack causes enormous harm to servers, as well as any devices on the network, and thus generates situations in which the authorized users of a network have no ability to access its resources or services [9, 10]. Therefore, the Intrusion Detection System (IDS) is used in order to mitigate security threats against the DDoS attacks.

The IDS is a very popular technology for detecting DDoS. However, with the enormous amounts of data in the real world, the IDS might be incapable of performing well. This indicates the necessity of Feature Extraction (FE), together with Feature Selections (FS), to minimize the data's dimensionality and to improve the IDS system's performance [11]. A Multi-variate Correlative Analysis (MCA) was employed in order to extract the features centered upon statistical analysis. MCA employs the Triangle-Area-Map (TAM) representation technique to illustrate the relationship between every traffic feature [12]. Shannon entropy [13], Kernel-based Online Anomaly Detection (KOAD) [14], and the Mahalanobis distance [15], along with the chi-square test [16], were each employed as FS techniques in order to differentiate between DDoS attacks and the normal traffic of the network. Moreover, machine learning (ML) techniques are used in information security, in which an appropriate decision is suggested based on the analysis and the proper action is automatically taken. Examples of ML techniques are artificial neural networks (ANN) [17], the Bayesian network [18], Decision Trees (DT) [19], Support Vectors Machine (SVM) [20], clustering [21], Forward Additive Neural Networks (FANN) [22], Deep Belief Networks (DBNs) [23], ensemble learning [24], etc. Many artificial neural networks (ANN) approaches have been used in the IDS area, and these approaches have many pros in terms of detecting the DDoS attack, comprising self-organizing, self-learning, robustness, and fault tolerance, along with parallelism. Research conducted in a variety of different studies offers strong evidence that DDoS is a type of attack whose volume,intensity, and mitigation expenses increase with the expanding scale of the organization. In the proposed work, a new approach has been developed in order to detect DDoS attacks based on the behaviors of network activity, applying the LMDANN for IoT. The proposed work's objectives are:

- To present an efficient F-MinMax normalization in order to improve classification accuracy.

Revised Manuscript Received on January 05, 2020.

***** Correspondence Author
Ahmed Saeed Alzahrani*, department of Computer Science, FCIT, King Abdulaziz University, Jeddah, Saudi Arabia. Email: asalzahrani@kau.edu.sa

- To present HDFS-based elimination of data redundancy.
- To introduce Entropy-based Fisher's Discriminant Function (ENTFDF) in order to perform a Feature Reduction (FR).
- To enhance the classification module's accuracy applying Levenberg-Marquardt based Deep Artificial Neural Network LMDANN technique.

The rest of the paper is organized as follows: Section 2 discusses the related works; Section 3 explains the proposed study briefly; Section 4 describes the analysis of the experiments of the proposed work; and finally, Section 5 concludes the paper with recommendations for future work.

II. RELATED WORK

Doshi et al. [25] studied the feature selections approach for the DDoS Attack Detection (DAD) system. This approach offers highly accurate DAD in IoT network traffic using ML approaches and neural networks (NN) based on IoT-specific networking behaviors (in other words, time intervals betwixt packets and limited endpoints). It has been shown that home gate-way routers or any middleboxes' network might detect the local IoT device source of the attack automatically based on traffic data that was protocol-agnostic, flow-centric, and through the use of low-cost ML algorithms. Although the FS had numerous advantages, an effective algorithm for detecting attacks of varied nature using less computational time is still required. Yonghao et al. [26] developed a new detection approach called semi-supervised weighted k-means (KM). This approach identified the most effective feature sets using a Hadoop-centric hybrid FS algorithm, resolving the outliers and local optimal problems using a density-centric initial cluster center selection algorithm. This approach provided the semi-supervised K-mean algorithm using hybrid FS (SKM-HFS) for detecting attacks. It has been shown that this approach outperformed the existing approaches in terms of attack detection and the "Technique for Order Preferences-Similarity to an Ideal Solution" evaluation factor. On this account, there might be many false negatives, and therefore the victim could be blocked on account of the dangerous congestion of down-stream links.

Velliangiri et al. [27] developed an effective fuzzy and Taylor-elephant herd optimization (FT-EHO) method based on the deep belief network (DBN) classifier in order to detect DDoS attacks. The training of FT-EHO use the Taylor series and elephant herd optimization algorithm, along with a fuzzy classifier. It has been shown that the performance of FT-EHO provided better evaluation values in terms of accuracy, detection rate, precision, and recall against other approaches. However, the system was irrelevant for the targeted strikes in the application layer, which destroyed it. Wang et al. [28] uses the multilayer perceptron (MLP) combined with the sequential feature selection in order to select the optimal features for the training phase. They modeled a feedback approach for reconstructing the detector when substantial detection errors were perceived. This method indicated that it could produce comparable detection performance and correct the detector when it performed poorly. In the case of the application layer, the approach lacks the ability to distinguish attacks that had a marginal difference, such as legitimate requests, low traffic volume, etc.

Chundong et al. [29] developed an SU-Genetic approach in order to pick out the important information associated with the actual attack data. The proposed approach ranked the features using the symmetric uncertainty and picked the features by applying the genetic algorithm. The correlation evaluator with SU value was employed in genetic selection for balancing the correlation and redundancy. The experimental analysis has been conducted using the NSL-KDD dataset, and it revealed that the features decreased from 41 to 17, and the volume of the data was roughly decreased to 41%. In all three of the classification-based detections (BayesNet, J48, and RanomTree), accuracy and efficiency were enhanced by the propounded SU-Genetic feature selection method and could not differentiate the flash crowds as being DDoS attacks, thereby elevating the false alarm rate. Aamir et al. [30] rendered a clustering-centric approach in order to differentiate the data signified by the network traffic flow, which embraced both DDoS and normal traffic. The clustering approaches embraced KM and were agglomerative with FE under Principal Component Analyses. The supervised ML algorithms say that the k-Nearest Neighbor (KNN), SVM, along with Random Forest (RF), was employed in order to acquire the trained frameworks for classification. The KNN, SVM, and RF models in experiential outcomes rendered 95%, 92%, and 96.66% accurateness, respectively, under optimized parameter tuning within the provided values. It was pre-eminent in attackdetection, but the detector was computationally costly when the process-count simultaneously elevated. Thus, the above survey provided a clear illustration of detecting a DDoS. Most research studies have not fulfilled the current need of trouncing DDoS attacks, since they concentrate mainly on classifying traffic-congested attacks. It is vital to classify such attacks that developed at the phase of attack preparation, but at the same time, it should also render proactive attack detection as well. The proposed work develops a new approach for DAD centered on the characteristics of network activity utilizing the LMDANN approach for IoT.

III. DISTRIBUTED DENIAL OF SERVICE (DDOS) ATTACK DETECTION SCHEME USING LMDANN APPROACH FOR IOT

The DDoS attack is an old issue that occurred during the establishment of the network. And even though this is the 5G era, people are still struggling with this issue. Because it is still the main threat of all cyber-attacks, and the issue is becoming more and more intricate, there has been a significant amount of work done in order to mitigate DDoS attacks. The DDoS attack causes congestion inside the network; henceforth, service denial occurs. The node encompassing the resources attains unnecessary requests that block the services, and thus leads to starvation. This attack is basically a sort of dynamic attack that distorts the routing procedure. There are numerous issues in conventional DAD, such as lower accuracy, lower detection speed, etc., which is not appropriate for the instantaneous detecting, nor the processing of DDoS attacks in a big data setting. This paper proposes an efficient DDoS Detection System employing LMDANN, and the architecture of the efficient DDoS attack detection system is represented in the figure below.

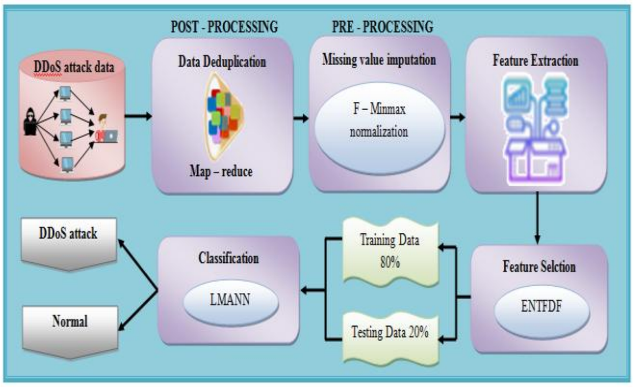


Fig 1.**Proposed DDoS attack detection model**

A. Data Collection

The initial process will be to collect the data from the CSE-CIC-IDS2018 dataset. DDOS data is extracted from several IDS datasets that were generated in various years and with various experimental DDoS traffic generation tools. The dataset is illustrated as follows:

$$D_s = \{d_1, d_2, d_3, \ldots\ldots\ldots\ldots d_k\} \qquad (1)$$

Where$D_s$signifies the IDS dataset for additional processing, and $d_k$implies the k −number of DDoS data on datasets.

B. Post-processing

After extracting the data from the dataset, post-processing is carried out in this subsequent phase. Post-processing is a crucial task through which to provide an efficient output. The dataset may have duplicate data, and in order to eliminate redundant data by performing a data deduplication operation, this phase avoids training the same data again and again.

C. Data duplication

The process of data duplication removes redundant copies of data and minimizes storage overhead. Here, the data duplication operation is conducted using the HDFS MapReduce framework. MapReduce is one of the best, simplest, and most parallel computing techniques normally used to improve performance by checking redundant data. The MapReduce technique hides the way in which partitioning takes place, and thus helps to focus on the technique of data processing. The MapReduce algorithms have two functions, such as map function and reduce function. These functions are described below:

1) Mapping function

The map function is the first step in the MapReduce algorithm. In the map function, the input DDoS attack information document is converted into different set of data in which individual documents are split into tuples (key/value pairs), which means that the input DDoS data is forwarded to the mapper-function line by line. The mapper processes the data and generates a number of small chunks of data, which is represented as:

$$M_{fo} = map\ (D_s) \qquad (2)$$

Here,

M_{fo} - Output of the $map()$function,

D_s - Input dataset,

$map()$-Function which performs the mapping function

2) Reduce function

The reduce function is the essential function in the Hadoop tool. This reduces operation integrates these data tuples based on the key, and alters the key’s value accordingly. Once the processing is completed, a new set of output is produced and stored in the HDFS. The reduce function is represented below:

$$R_{fo} = reduce\ (M_{fo}) \qquad (3)$$

Here,

M_{fo} - Output of the mapped function

$reduce()$- Function that aids to reduces the components

R_{fo} - the reduced set of data

D. Pre-processing

In this phase, the preprocessing process is performed. Preprocessing is the process of cleaning your data in order to make it more meaningful before performing any tasks, and ensuring that it handles the data efficiently. Analyzing the data is an important step to ensure that no misleading results are obtained. So, the preprocessing process is an important step for replacing the missing values. Here, two processes are carried out, namely the missing values imputation and the normalization of the data. These are explained briefly below.

1) Missing value imputation

This is the first step in the preprocessing phase. The dataset often includes variables that have some data missing. Missing values occur when no data is stored for a given variable in the current observation. If any records have missing values, then these values will be filled by replacing the missing value for a particular attribute by the average value for that attribute. Here, the missing values will be represented by “?”.

2) F-MinMax Normalization

Normalization is a technique that organizes data for more efficient access. The system generates an effective result if normalization is applied. Here, the proposed system uses feature-based MinMax (F-MinMax) normalization, which means adjusting the data values into a specific range, such as between 0 to1 or -1 to 1, using the minimum and maximum of feature values. Initially, the number of features in the dataset isF_{nf} . Then, modify the data in order to have a lower bound of 0. To do this, subtract $min(F_{nf})$from each feature value. It is described as,

$$F_{nf} - min\ \ (F_{nf}) \qquad (4)$$

Then, modify the data in order to have an upper bound of 1. To do this, divide each value by the original range. It is represented as,

$$\frac{F_{nf}}{max\ \ (F_{nf})-min\ \ (F_{nf})} \qquad (5)$$

Finally, combine equation (4) and (5) and get the normalized value, which is described as,

$$MinMax = \frac{F_{nf} - min\ (F_{nf})}{max\ (F_{nf}) - min\ (F_{nf})} \qquad (6)$$

Based on the above steps, the missing values are replaced, and data integrity is effectively improved.

E. Feature Extraction

Feature extraction is an essential step that must be executed for the attack detection system. It begins from an initial set of measured data, and constructs derived features (values) intended to be informative and non-redundant. The proposed system extracts timestamp, mean, flow ID, flow duration, label, and other features from the dataset. In total, the proposed system extracts 84 features from the dataset. Some features are explained in brief below.

1) Timestamp

The timestamp is an important feature of the attack detection system, which is used to find the arrival time of the records. This is the time when the packet arrived at the receiving node, and is described as follows:

$$Timastamp = S_{time} - E_{time} \qquad (7)$$

WhereS_{time} indicates the starting time andE_{time} denotes the ending time for the packets to arrive at the receiver node.

2) Mean

The mean feature is the average of the record in the dataset. For a discrete set of numbers, the mean indicates the central value or the value acquired by dividing the total of the values with the number of values, which is evaluated as:

$$Mean = \frac{A_{rs}}{N_{rs}} \qquad (8)$$

WhereA_{rs} indicates the sum of the record in the dataset, and N_{rs} refers to the number of records in the dataset.

3) Flow ID

The flow ID is a type of identity. Every file has an individual flow ID, which is provided in the training data. The flow ID can be described as,

$$F^{ID} = \{F_1^{ID}, F_2^{ID}, F_3^{ID}, \ldots\ldots\ldots\ldots F_n^{ID}\} \qquad (9)$$

Where F^{ID} represents the flow ID, and F_n^{ID} denotes the n −number of flow IDs. Finally, the extracted total number of features is described as follows:

$$S_{ef} = \{s_1, s_2, s_3, \ldots\ldots\ldots\ldots\ldots s_k\} \qquad (10)$$

Where S_{ef} indicates the extracted feature set, and s_k indicates the k −number of features.

F. Feature Reduction

Feature reduction is also referred to as dimensionality reduction. Feature reduction has been used in this system for two purposes. First, for every DDoS attack, the features that best characterize the attack's behavior are reduced. Second, feature reduction could help in improving detection accuracy, reducing the false positive rate, minimizing the training time (TT), and using the classifier accurately. Reducing unnecessary features not only helps to efficiently detect DDoS attack traffic, but also shortens the response speed of the algorithm, as per the recognition rate. Here, the feature reduction process is conducted by using the Entropy-based Fisher's Discriminant Function (ENTFDF) algorithm. Fisher's Discriminant Function (FDF) is a standard technique for feature reduction in DDoS attack detection system. It projects higher-dimensional data on a line and executes classification in the 1-dimensional space. This projection increases the distance between the means of the 2 classes while also lessening the variance within each class. However, the FDF does not provide efficient results, so the entropy technique will include the FDF algorithm in order to efficiently improve the feature reduction process. The ENTFDF algorithmic procedures are explained as follows:

Step 1: First, the derived features are taken as a matrix of$\left(S_{ef}\right)_x = \left[\left(S_{ef}\right)_{1,1}, \left(S_{ef}\right)_{1,2}, \left(S_{ef}\right)_{1,3}, \ldots\ldots\ldots \left(S_{ef}\right)_{K,Y}\right]$,

where$K$ represents the amount of features and Yrepresents the dimension of$(S_{ef})_x$. The feature matrix is partitioned into $a = r$ classes as follows:

$$\left(S_{ef}\right)_x = P_i = \{p_1, p_2, p_3, \ldots\ldots\ldots p_r\} \qquad (11)$$

Where $a$signifies the classes, and $P_i$represents the i^{th}class.

Step 2:After that, the K dimensional mean vectors for the different classes were computed from the feature matrix. The mean of each class $\left(\mu^{(a)}\right)$ is calculated as,

$$\mu^{(a)} = \frac{1}{K^{(a)}} \sum_{(S_{ef})_x \in P^{(a)}} \left(S_{ef}\right)_x \qquad (12)$$

Step 3: Next, determine the total mean of all features (μ), which is defined as follows:

$$\mu = \sum_{n=1}^{a} \frac{B_n}{K}\ \mu^{(a)} \qquad (13)$$

Where B_n represents the number of samples in P_i.

Step 4: Then, compute the entropy value (E) for the mean vector of the two classes in order to improve the reduction accuracy, which is described as follows:

$$E_{\mu^{(a)}} = -\sum \rho\left(\mu^{(a)}\right) log\, \rho\left(\mu^{(a)}\right) \qquad (14)$$

$$E_{\mu} = -\sum \rho(\mu)\, log\, \rho\,(\mu) \qquad (15)$$

Step 5:Next, calculate the between-class scatter matrix $(C_b^{(a)})$ and within-class scatter matrix $(C_w^{(a)})$. It is represented as follows:

$$C_b^{(a)} = \sum_{i=1}^{a} \left(\left(S_{ef}\right)_{x_i} - E_{\mu^{(a)}}\right)\left(\left(S_{ef}\right)_{x_i} - E_{\mu^{(a)}}\right)^T \qquad (16)$$

$$C_w^{(a)} = \sum_{i=1}^{a} B_n\left(E_{\mu^{(a)}} - E_{\mu}\right)\ \left(E_{\mu^{(a)}} - E_{\mu}\right)^T \qquad (17)$$

Where $(S_{ef})_{x_i}$indicates the i^{th}feature in the a^{th}class.

Step 6:Next, evaluate the total-class scattering matrix by summing the inside class $C_b^{(a)}$and the between-class $C_w^{(a)}$, represented as follows:

$$C_{tm} = C_b^{(a)} + C_w^{(a)} \qquad (18)$$

Here, C_{tm} indicates the total class scattering matrix. Then, build a transformation matrix for each class (T_{matrix}) as follows,

$$T_{matrix} = \left(C_b^{(a)}\right)\ C_w^{(a)} \qquad (19)$$

Step 7: estimate the Raleigh quotient $(M_{error}(D))$ for minimizing the existent misclassification error, as described below:

$$M_{error}(D) = \frac{D^T C_b^{(a)} D}{D^T C_w^{(a)} D} \quad over\ D \qquad (20)$$

Step 8: After that, compute the eigenvectors $(u_1, u_2, u_3, \ldots., u_j)$ and corresponding eigenvalues $(\lambda_1, \lambda_2, \lambda_3, \ldots., \lambda_j)$ for the scatter matrices.

Then, the generalized eigenvalue problem is expressed as follows:

$$C_b^{(a)}\ u_j = \lambda_j\ C_w^{(a)}\ u_j \qquad (21)$$

Step 9: Finally, order the eigenvectors by decreasing the eigenvalue. The reduced feature set can be obtained by,

$$S_f = \left(S_{ef}\right)_x^{\ T}.u_{\ j} \qquad (22)$$

Where S_f indicates the reduced feature, which is created as a linear combination of all input features$(S_{ef})_x$.

G. Train System Using LMDANN Algorithm

Here in this phase, based on the reduced features, the training is performed. This step is primarily used to classify the data and for checking whether the data is DDoS and benign. In the proposed system, the attack detection was conducted through the use of the Levenberg-Marquardt based Artificial Neural Network (LMDANN) algorithm. There are some important advantages that interest a lot of people in ANN, such as self-learning traits, better fault tolerance, robustness, and self-organization. It is also important to detect intrusions. ANN can identify both existent and unknown attack patterns. The ANN gives satisfactory results, but it may provide less accuracy in terms of classifying between DDoS attacks or normal functioning. In order to improve classification accuracy, the Levenberg-Marquardt (LM) technique will be included in the Neural Network Algorithm. In ANN, there are three layers: input layer, hidden layer, and output layer. The LMDANN algorithmic procedures are explained as follows:

Step 1: Initially, assign the reduced features and their equivalent weights as described as below:

$$S_f = \{s_1, s_2, s_3, \ldots\ldots\ldots. s_n\} \qquad (23)$$

$$W_f = \{w_1, w_2, w_3, \ldots\ldots. w_n\} \qquad (24)$$

Step 2: After initialization, the input value is multiplied with the weight vector that is arbitrarily selected, and then the total is summed up. Mathematically, this is expressed as follows:

$$L_m = \sum_{f=1}^{n} S_f.\ W_f \qquad (25)$$

Where L_m indicates the assigned value, S_f defines the input reduced feature, and W_f represents the weight values.

Step 3:Next, determine the activation function, which helps the network to learn the complex patterns in the data. Mathematically, this is described as follows:

$$A_f^{'} = Z\ \left(\sum_{f=1}^{n} S_f.\ W_f.\ M_f\right) \qquad (26)$$

Where $A_f^{'}$ indicates the activation function. The category of the activation function utilized in this system is the Levenberg-Marquardt (LM) technique, which was chosen in order to enhance the prediction accuracy. Levenberg-Marquardt is a good substitute of the Gauss-Newton approach in terms of finding the least of a function, which is a summation of squares of input values.

$$Z\ \left(S_f\right) = \frac{1}{2}\sum_{f=1}^{n}\left(S_f\right)^2 = M_{\ f} \qquad (27)$$

Where M_f represents the Levenberg-Marquardt function's output.

Step 4: After that, compute the output of the 1st hidden layer by using the following equation:

$$H_f^1 = B_f + \sum_{f=1}^{n} A_f^{'}.\ W_f \qquad (28)$$

Step 5: Then, evaluate the output unit $(O_f^{'})$ by summing up all of the weights of the input values. This is the calculation that is necessary to achieve the value of the neurons in the output layer. It is represented as follows:

$$O_f^{'} = B_f + \sum_{f=1}^{n} H_f^1.\ W_f \qquad (29)$$

Step 6: Finally, compute the loss function by using the following equation:

$$loss_f = [T_f + O_f^{'}] \qquad (30)$$

Where T_f indicates the target output of the neural network. Here, the minimum value is set as the threshold for the loss function. If the initialized threshold value meets this fitness requirement, then the output is expressed as the final output, the position of the weight value is renewed, and also the activation functions employ the same LM algorithm. Again determine the output unit based on this LMDANN algorithm, and then the output data is trained for the retrieval process. The pseudocode for the LMDANN algorithm is represented in Algorithm 1.

H. Testing Phase

After the training phase ends, the testing will be done. In this phase, 80% of the data will be given for training, and 20% of the data will be given for testing. In the testing phase, the IoT sensor values are initially taken as input. Some important features are extracted from the input values, and then the feature reduction is carried out by using the ENTFDF algorithm. Lastly, the reduced features are transmitted to the LMDANN classifier for classifying the test instance. If the classifier correctly classifies a provided class, then the process provides better results.

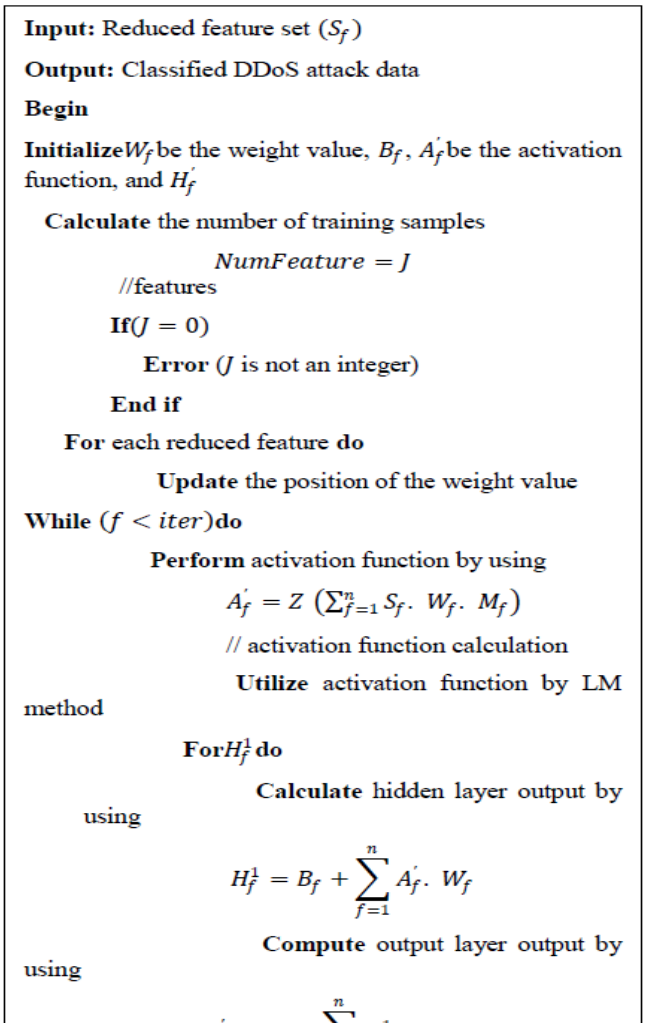
IV. RESULTS AND DISCUSSION

The performance of an efficient DDoS attack detection system utilizing the LMDANN algorithm is now analyzed, and the implementation is conducted using JAVA. Java applications are compiled into byte code, which could be run on Java virtual machines (JVM) whatever the computer architecture might be, and JAVA improves the proposed system by contrasting its results with conventional methodologies. Moreover, this section illustrates the datasets deployed, the performance matrices evaluated, the environment of the experiment, and the acquired results.

A. Dataset Description

The proposed system uses the CSE-CIC-IDS2018 dataset in the experiments. This dataset is publicly available online at the Kaggle site. In order to introduce more variance, the DDOS data is extracted from several IDS datasets that were generated in various years and with various experimental DDoS traffic generation tools. The extracted DDOS flows are integrated with "benign" flows that are extracted from the same base dataset separately and then merged into a single larger dataset.

Algorithm 1 :**Pseudocode for the LMDANN algorithm**



```
Input: Reduced feature set ($S_f$)
Output: Classified DDoS attack data
Begin
InitializeW_f be the weight value, $B_f$, $A_f'$ be the activation function, and $H_f'$
  Calculate the number of training samples
          $NumFeature = J$
      //features
      If($J = 0$)
          Error ($J$ is not an integer)
      End if
    For each reduced feature do
          Update the position of the weight value
While ($f < iter$)do
          Perform activation function by using
              $A_f' = Z\ (\sum_{f=1}^{n} S_f .\ W_f .\ M_f)$
              // activation function calculation
              Utilize activation function by LM method
          For$H_f^1$ do
              Calculate hidden layer output by using
              $H_f^1 = B_f + \sum_{f=1}^{n} A_f' .\ W_f$
              Compute output layer output by using
```

B. Performance Analysis

Here, the proposed LMDANN algorithm is contrasted with state-of-the-art Adaptive Neuro-Fuzzy Inference System (ANFIS), ANN, SVM, and KNN, based on their performances in certain metrics that are discussed below. The proposed systems' performance was measured with specificity, accuracy, sensitivity, recall, f-measure, precision, and training time (TT) metrics. The basic parameters that are evaluated are "true positive" (X_p), "true negative" (X_n), "false positive" (Y_p) and "false negative" (Y_n) values.

1) Accuracy

Accuracy is the probability that a record, either an attack or normal, is accurately identified. The formula is described as follows:

$$Accuracy = \frac{X_p + X_n}{X_p + X_n + Y_p + Y_n} \quad (31)$$

2) Sensitivity

Sensitivity measures the rate of correct differentiation between the normal data and the attack data. Mathematically, this is represented as follows:

$$Sensitivity = \frac{X_p}{X_p + Y_n} \quad (32)$$

3) Specificity

Specificity is the rate of the accurate classification of an abnormal attack to the total classified results. The formula for calculating the specificity is described below:

$$Specificity = \frac{X_n}{X_n + Y_p} \quad (33)$$

4) Precision

Precision is the number of accurately predicted records over all predicted records for a particular class, defined as follows:

$$Pr\ e\ cision = \frac{X_p}{X_p + Y_p} \quad (34)$$

5) Recall

A recall is the number of accurately predicted attacks over all of the r3ecords available for a particular class in the dataset, represented as follows:

$$Re\ c\ all = \frac{X_p}{X_p + Y_n} \quad (35)$$

6) F-measure

The f-measure uses precision and recall for the holistic evaluation of a model, and is represented as the harmonic mean of all of them, described as follows:

$$F - measure = \frac{2\ .\ Pr\ ecision\ \ .\ Re\ call}{Pr\ ecision + Re\ call} \quad (36)$$

7) Training time

The training time is measured by evaluating the difference between the training starting time and training ending time. Overall, this is defined as the time necessary to train the dataset. Mathematically, training time is described as follows:

$$T_{time}\ (t) = T_{end}\ (t) - T_{start}\ (t) \quad (37)$$

Where $T_{end}\ (t)$ indicates the training ending time, and $T_{start}\ (t)$ denotes the training staring time.

Table 1: Performance evaluation of the proposed LMDANN algorithm with existing algorithms

Metrics	Proposed LMDANN	ANN	SVM	KNN	ANFIS
Accuracy	96.35	94.33	92.35	91.33	90.23
Sensitivity	95.33	93.56	92.32	91.45	90.33
Specificity	94.36	91.57	90.99	90.13	89.33
Precision	94.32	92.46	91.37	89.23	88.69
Recall	95.69	93.59	92.75	91.35	90.87
F-measure	95.37	93.57	92.35	91.97	91.12

The above table shows the performance evaluation of the proposed LMDANN algorithm with the existent ANN, SVM, KNN, and ANFIS algorithms. Several performance evaluation metrics, such as accuracy, sensitivity, specificity, precision, recall, and f-measure, have each been used to evaluate the performance of proposed algorithm. From the table, the existing ANFIS classifier proffers lower-level performance than the proposed LMDANN classifier. Also, the existent ANN, SVM, and KNN algorithms offer lower performance than the proposed classifiers.

However, the proposed LMDANN algorithm offers 96.35% accuracy, 95.33% sensitivity, 94.36% specificity, 94.32% precision, 95.69% recall, and 95.37% f-measure, which is greater than all of the existent classifiers. According to the obtained results, it can be observed that the proposed LMDANN approach can detect the DDoS attack quickly and with a high degree of accuracy.

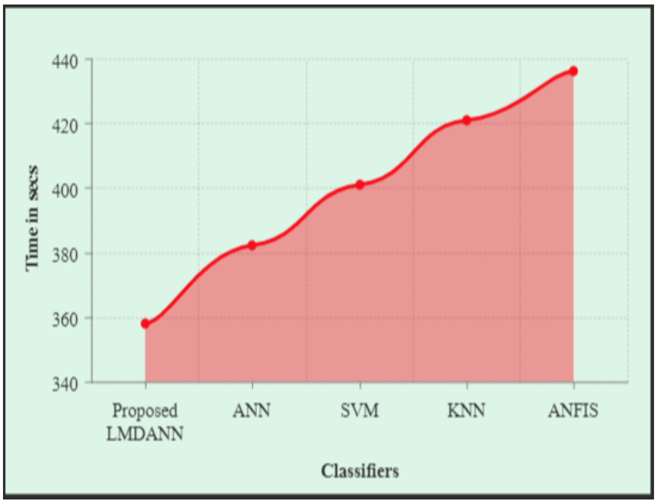


Fig2.Training time comparison for the proposed LMDANN algorithm

The figure above shows that the proposed system takes less time to train the data than the prevailing algorithms. For example, the existing ANFIS algorithm takes 436s to train the data. Likewise, the existent ANN, SVM, and KNN take 382s, 401s, and 421s, respectively, to train the data, which is even lower when contrasted with the proposed classifier. The proposed system takes only 358s to train the data. Training time is an important indicator in the proposed system for dealing with DDoS attacks. Altogether, the training time for the proposed LMDANN algorithm is much lower in comparison to the current ANN, SVM, KNN, and ANFIS algorithms.

C. Comparative Analysis

The proposed LMDANN approach can be compared to traditional algorithms using various performance metrics, such as accuracy, sensitivity, specificity, precision, recall, f-measure, and training time. These analyses are graphically represented below in Figures 3 and 4.

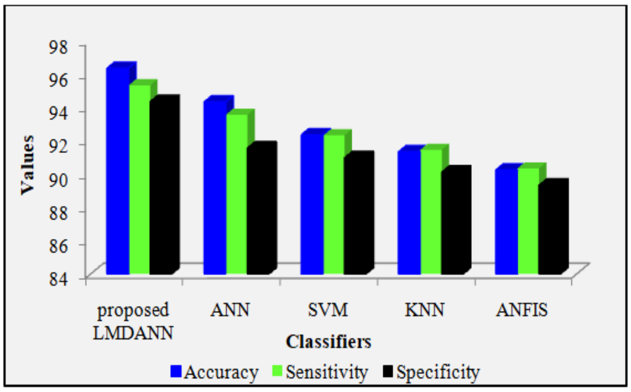


Fig3.Accuracy, sensitivity, and specificity graph for the proposed LMDANN algorithm with the conventional algorithms

In Figure 3, the obtained accuracy, sensitivity, and specificity values are displayed by comparing the proposed LMDANN technique with the existing ANN, SVM, KNN, and ANFIS techniques. Here, the proposed technique attains higher levels of performance than all of the prevailing methodologies. Based on the accuracy metric, the proposed LMDANN classifier offers 96.35% accuracy, but the prevailing ANN, SVM, KNN, and ANFIS offer accuracy levels of 94.33%, 92.35%, 91.33%, and 90.23%, respectively, which are less than the proposed classifier. Similarly, the proposed LMDANN classifier attains 95.33% sensitivity and 94.36% specificity. Therefore, it is confirmed that the LMDANN attains greater accuracy when compared to the current systems. Figure 4 demonstrates the performance of the proposed LMDANN algorithm compared to some other conventional algorithms, namely the ANN, SVM, KNN, and ANFIS algorithms. Here, the performance comparisons are done using some qualitative metrics, namely precision, f-measure, and recall. These are important performance metrics for the attack detection system. From the analysis of the DDoS attacks in this experiment, it has been found that this system has high f-measure, recall, and precision values. The existing ANFIS offers 88.69% precision, 90.87% recall, and 91.12% f-measure, which are each smaller than the proposed method offers. The proposed method attains 94.32%, 95.69%, and 95.37% of the previously mentioned values, respectively. The obtained results prove that the LMDANN classifier is the better method with which to detect DDoS attacks than the existent methodologies regarding the precision, recall, and f-measure metrics.

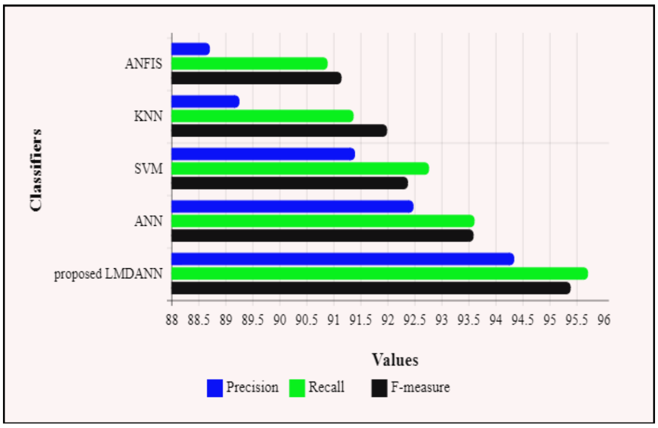


Fig 4:Comparative analysis of the proposed LMDANN algorithm with the conventional algorithms

V. CONCLUSION

The possibility of vulnerability to infamous DDoS attacks has increased with the invasion of IoT devices. DDoS attacks originating from IoT botnets represent an imminent threat for today's Internet because of attackers' ability to generate high packet volume from millions of compromised IoT devices. In this paper, an efficient DDoS attack detection system using the LMDANN approach for IoT is proposed. The proposed system has two phases, namely the training and testing phase. The training phase is comprised of five separate phases, namely post-processing, preprocessing, feature extraction, feature reduction, and the classification of the DDoS attack. Extensive experiments have been performed using the CSE-CIC-IDS2018 dataset.

The proposed LMDANN algorithm is compared to the existing ANN, SVM, KNN, and ANFIS algorithms using some specific performance evaluation metrics, namely accuracy, sensitivity, specificity, precision, recall, f-measure, and training time. From the analysis of the experimental results, the proposed LMDANN classifier attains 96.35% accuracy with less training time. The proposed system takes only 358s to train the data. Thus, the proposed system outperforms the LMDANN with the existing approaches, which proves that the proposed method outperforms other work, and demonstrates the efficiency of the proposed algorithm. Regarding future work, advanced machine algorithms can be employed for the same data sets and implemented in a real-time environment.

REFERENCES

1. D. Yin, L. Zhang and K. Yang, "A DDoS Attack Detection and Mitigation With Software-Defined Internet of Things Framework," in IEEE Access, vol. 6, pp. 24694-24705, 2018, doi:10.1109/ACCESS.2018.2831284.
2. I. Ud Din, M. Guizani, S. Hassan, B.Kim, M. K. Khan, M. Atiquzzaman, and S. H. Ahmed, “The Internet of Things: A review of enabled technologies and future challenges,” IEEE Access, vol. 7, pp. 7606-7640, 2018, 10.1109/ACCESS.2018.2886601.
3. S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in IoT using SDN," 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, VIC, 2017, pp. 1-6, doi: 10.1109/ATNAC.2017.8215418.
4. X. Liu, C. Qian, W. G. Hatcher, H. Xu, W. Liao and W. Yu, "Secure Internet of Things (IoT)-Based Smart-World Critical Infrastructures: Survey, Case Study and Research Opportunities," in IEEE Access, vol. 7, pp. 79523-79544, 2019, doi: 10.1109/ACCESS.2019.2920763.
5. Z. A. Baig, S. Sanguanpong, S. N. Firdous, T. G. Nguyen, and C. So-In, “Averaged dependence estimators for DoS attack detection in IoT networks,” Future Generation Computer Systems, vol. 102, pp. 198-209, 2020, 10.1016/j.future.2019.08.007.
6. M. Zekri, S. E. Kafhali, N. Aboutabit and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), Rabat, 2017, pp. 1-7, doi: 10.1109/CloudTech.2017.8284731.
7. M. M. Salim, S. Rathore, and J. H. Park, “Distributed denial of service attacks and its defenses in IoT: a survey,” The Journal of Supercomputing, pp. 1-44, 2019, 10.1007/s11227-019-02945-z.
8. V. Adat, and B. B. Gupta, “A DDoS attack mitigation framework for internet of things,” in International conference on communication and signal processing (ICCSP), IEEE, pp. 2036-2041, 2017
9. J. Choi, C. Choi, B. Ko, and P. Kim, “A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment,” Soft Computing, vol. 18, no. 9, pp. 1697-1703, 2014.
10. A. Lohachab, B. Karambir, “Critical Analysis of DDoS—An Emerging Security Threat over IoT Networks,” Journal of Communications and Information Networks, vol. 3, no.3, 2018, 10.1007/s41650-018-0022-5.
11. K. J. Singh, K. Thongam, and T. De, “Entropy-based application layer DDoS attack detection using artificial neural networks,” Entropy, vol. 18, no. 10, pp. 350, 2016.
12. P. Redekar, and M. Chatterjee, “Hybrid technique for DDoS attack detection,” International Journal of Computer Science and Information Technologies, vol. 8, no.3, pp. 377-379, 2017.
13. Z. Liu, Y. He, W. Wang, and B. Zhang, “DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN,” China Communications, vol. 16, no. 7, pp. 144-155, 2019.
14. S. Daneshgadeh, T. Kemmerich, T. Ahmed, and N. Baykal, “A Hybrid approach to detect DDoS attacks using KOAD and the mahalanobis distance,” in IEEE 17th International Symposium on Network Computing and Applications (NCA), IEEE, pp. 1-5, 2018, 10.1109/NCA.2018.8548334.
15. S. D. Çakmakçı, T. Kemmerich, T. Ahmed, and N. Baykal, “Online DDoS attack detection using Mahalanobis distance and Kernel-based learning algorithm,” Journal of Network and Computer Applications, pp. 102756, 2020., 10.1016/j.jnca.2020.102756.
16. F. E. Ouerfelli, K. Barbaria, B. Zouari, and C. Fachkha, “Distributed detection system using wavelet decomposition and chi-square test,” in International Conference on Risks and Security of Internet and Systems, Springer, Cham, pp. 365-377, 2019.
17. G. S. Kushwah, and S. T. Ali, “Detecting DDoS attacks in cloud computing using ANN and black hole optimization,” in 2nd International Conference on Telecommunication and Networks (TEL-NET), IEEE, pp. 1-5, 2017, 10.1109/TEL-NET.2017.8343555.
18. R. F. Fouladi, C. E. Kayatas, and E. Anarim, “Frequency based DDoS attack detection approach using naive Bayes classification,” in 39th International Conference on Telecommunications and Signal Processing (TSP), IEEE, pp. 104-107, 2016, 10.1109/TSP.2016.7760838.
19. S. Lakshminarasimman, S. Ruswin and K. Sundarakantham, "Detecting DDoS attacks using decision tree algorithm," 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, 2017, pp. 1-6, doi: 10.1109/ICSCN.2017.8085703.
20. J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, “A DDoS attack detection method based on SVM in software defined network,” Security and Communication Networks, 2018, 10.1155/2018/9804061.
21. S. Bista, and R. Chitrakar, “DDoS attack detection using heuristics clustering algorithm and Naïve Bayes classification,” Journal of Information Security, vol. 9, no. 01, pp. 33, 2017.
22. J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, “Method of intrusion detection using deep neural network,” in IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, pp. 313-316, 2017, 10.1109/BIGCOMP.2017.7881684.
23. Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, “Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks,” Applied Sciences, vol. 9, no. 2, pp. 238, 2019.
24. Y. Zhong, W. Chen, Z. Wang, Y. Chen, K. Wang, Y. Li, X. Yin, X. Shi, J. Yang, and K. Li, “HELAD: A novel network anomaly detection model based on heterogeneous ensemble learning,” Computer Networks, vol. 169, pp. 107049, 2020, 10.1016/j.comnet.2019.107049.
25. R. Doshi, N. Apthorpe, and N. Feamster, “Machine learning ddos detection for consumer internet of things devices,” in IEEE Security and Privacy Workshops (SPW), IEEE, pp. 29-35, 2018, 10.1109/SPW.2018.00013.
26. Y. Gu, K. Li, Z. Guo and Y. Wang, "Semi-Supervised K-Means DDoS Detection Method Using Hybrid Feature Selection Algorithm," in IEEE Access, vol. 7, pp. 64351-64365, 2019, doi: 10.1109/ACCESS.2019.2917532.
27. S. Velliangiri, , and H. M. Pandey, “Fuzzy-Taylor-elephant herd optimization inspired Deep Belief Network for DDoS attack detection and comparison with state-of-the-arts algorithms,” Future Generation Computer Systems, 2020, 10.1016/j.future.2020.03.049.
28. M. Wang, Y. Lu, and J. Qin, “A dynamic MLP-based DDoS attack detection method using feature selection and feedback,” Computers & Security, vol. 88, pp. 101645, 2020, 10.1016/j.cose.2019.101645.
29. C. Wang, H. Yao, and Z Liu, “An efficient DDoS detection based on SU-Genetic feature selection,” Cluster Computing, vol. 22, no. 1, pp. 2505-2515, 2019.
30. M. Aamir, and S. M. Zaidi, “Clustering based semi-supervised machine learning for DDoS attack classification,” Journal of King Saud University-Computer and Information Sciences, 2019, 10.1016/j.jksuci.2019.02.003.

.AUTHORS PROFILE

Dr. Ahmed Alzahrani (Member, IEEE) received the B.S. degree in computer science from King Abdulaziz University, in 2000, the M.S. degree in information security from the University of Glamorgan, Cardiff, U.K., in 2005, and the Ph.D. degree in computer networks from the University of Bradford, U.K., in 2009. He is currently an Associate Professor with the Computer Science Department and the Vice Dean of Deanship of graduate studies for academic affairs with King Abdulaziz University. His current research interests include computer networks, networks security, quality of service routing, quantum computing, deep learning, big data, in-memory computing, and high-performance computing