
# A Multiclass Attack Classification Framework for IoT Using Hybrid Deep Learning Model

**Saraladeve .L[1,*], Chandrasekar .A[2], Nithya .T[3], Mohamed Imtiaz .N[4], Kalaiarasi .S[5], Balaji Sampathkumar[6], Rajendran Thanikachalam[7], Maria Arockia Dass .J[3]**

[1]Department of Computer Science, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India

[2]Department of Computer Science and Engineering, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India

[3]Department of Computer Science and Business Systems, Rajalakshmi Institute of Technology (Autonomous), Chennai, Tamil Nadu, India

[4]Department of Information Science and Engineering, Nagarjuna College of Engineering and Technology, Bangalore, Karnataka, India

[5]Department of Computer Science & Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India

[6]Akshaya College of Engineering and Technology, Coimbatore, Tamil Nadu, India

[7]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India

Emails: saraladeve@gmail.com; dean@stjosephs.ac.in; nithya.t@ritchennai.edu.in; imtiaz4687@gmail.com; kalaiarasis.sse@saveetha.com; bbaallaajjii@gmail.com; tlrajen@gmail.com; dasscse2008@gmail.com

**Abstract**

In recent years, the Internet of Things (IoT) has emerged as one of the most significant concepts in numerous facets of our contemporary way of life. Nonetheless, addressing the concerns over the IoT's security presents the most significant obstacle to the widespread adoption of this technology. Using an Intrusion Detection System (IDS) to detect malicious activity in the networks is one of the most essential things that can be done to solve the security concerns posed by the IoT. Hence, a Deep Learning-based IDS (DL-IDS) model is designed for the multi-class classification of attacks in the IoT networks. This DL-IDS model includes data preprocessing, feature extraction, feature selection, and classification processes. The Bot-IoT and IoT-23 datasets are used as input for the research model. In preprocessing, the datasets are normalized, and the missing data are replaced. After preprocessing, the features are extracted using the Convolutional Neural Network (CNN) architecture. The features selection process is performed from the extracted features by implementing the Quantum-based Chameleon Swarm Optimization (QCSO) algorithm, which selects features from the datasets. Based on these features selected, the multi-class classification is carried out using the Deep Belief Network (DBN) for each attack presented in the datasets. The classification performance is performed individually for both datasets and evaluated using accuracy, detection rate, precision, and f1-scores. The performances of the proposed DL-IDS model are compared with the other models analyzed from the literature survey discussed in this work. The average scores obtained using the IoT-23 data set include 99.45% accuracy, 99.47% detection rate, 99.66% f1-scores, and 99.85% precision. For the Bot-IoT data, the average scores are 99.49% accuracy, 99.52% detection rate, 99.70% f1-score, and 99.88% precision.

**Keywords:** IoT; IDS; Features Selection; Multi-class Classification; CNN; QCSO; DBN



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

## 1. Introduction

IoTs are devices that are connected to one another to promote the smooth exchange of information between physical devices. It is anticipated that IoT devices will be in use more than smartphones. These devices include the most sensitive information to access, including personal data. Both the likelihood of attacks and the attack surface area will rise because of this. IoT-IDSs will need to be developed to secure communications made possible by technologies related to the IoT [1]. This is because security will be an essential supporting factor for many applications related to the IoT. Monitoring and, in some cases, thwarting the criminal actions of unauthorized users is what's meant by the term "intrusion detection." It is possible to utilize it to keep track of all actions taking place within the network. If there is any kind of attack or undesirable activity going on in the network, the IDS will identify the intrusions, notify the user, log the attacks for later use in investigations, isolate the intruder, and disconnect the intruder's connection path. Figure 1 presents an illustration of the functionalities that IDS provides [2].


**Figure 1.** Functions of IDS for IoT Networks

An IoT network's network IDS keeps monitoring all the devices' internet activity. It serves as a front line of protection, identifying potential threats while defending the network from unwanted visitors and harmful attacks. IoT Intrusion refers to any action or activity that is not authorized and which causes damage to the IoT ecosystems. An intrusion is defined as the attacks that compromises the availability, integrity, and confidentiality of information. The aim of exchanging information and linking to various networks was accomplished due to the IoT, which includes a several devices like processors, sensors, and other types of technology. As IoT incorporates many linked devices, there will be a possibility that the exchange of data is not safe, which increases a concern regarding security [3].

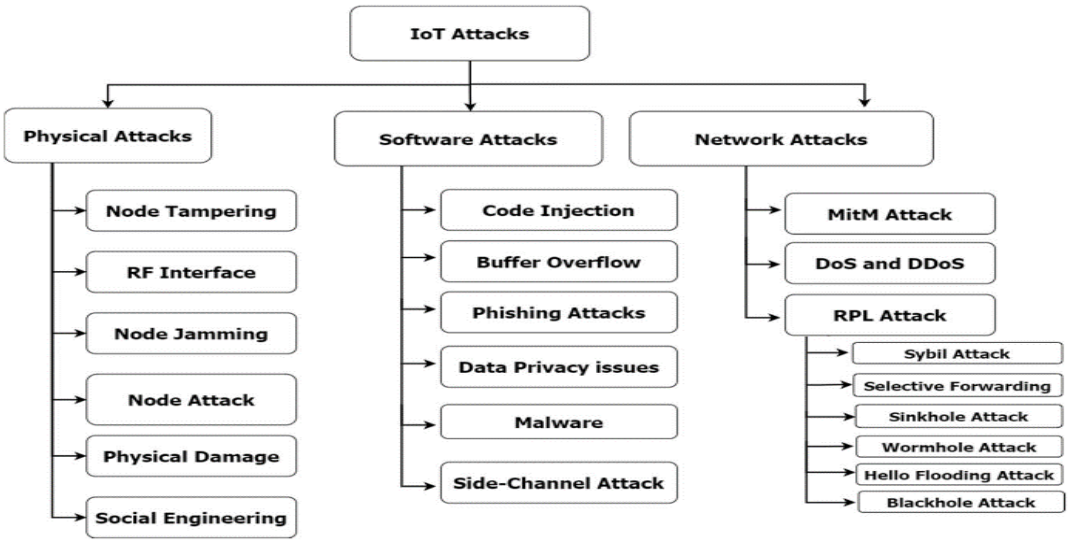


**Figure 2.** Attacks in IoT



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

The term Security of IoT refers to the safeguarding process of the data that was transferred between various networks and stored on IoT devices utilizing IoT technologies. Because these devices were linked to others with the internet, they are susceptible to vulnerabilities, which enable hackers to steal information from them. Data that is not properly secured can give rise to lot of issues and result in enormous loss for several businesses and people, mostly leading to the data loss from their respective devices [1]. The most common kinds of cyberattacks on IoT applications are depicted in Figure 2. As the IoT linked devices keeps on growing, also the potential security breaches and vulnerabilities increases. Since the nodes that make up the IoTs build dynamic topologies and carry out their activities without the assistance of humans, managing security concerns in the IoT becomes more difficult. In recent years, breakthroughs in Artificial Intelligence (AI) including machine learning (ML) and DL methodologies, were applied to improve IoT intrusion detection and prevention systems [4].

### 1.1. Problem Statement

DL methods perform significantly better on huge datasets as compared to ML-based methods. DL approaches have emerged as the most suitable and extensively utilized kind of network IDS. Because it can identify patterns in raw data that had not been seen before, it finds widespread application in the field of cybersecurity. Over numerous tiers and iterations of adjustment, it discovers higher-level traits. In recent years, a great number of IDS models have been developed as a response to the widespread prevalence of security and privacy risks affecting computer networks. If IDS prevention is unsuccessful, there will be damage to the data's availability, integrity, and confidentiality. Traditional methods are incapable of providing adequate defense against more sophisticated attacks. The automatic detection of intrusions and the identification of anomalous behaviour in networks have both been suggested as possible applications for DL approaches [5]. Because of this, the research presented here presents a hybrid IDS paradigm for the IoT that is developed using the DL techniques for effective attack detection and classification.

### 1.2. Research Contributions

The contributions of the research work includes:

- A multi-class attack classification model based on DL models is proposed in this work.
- After preprocessing, the CNN model is implemented for extracting features from the Bot-IoT and IoT-23 datasets.
- A Quantum-based CSA algorithm is used to select the optimal features.
- The DBN model is implemented to perform a multi-class classification using the selected features.
- The research model performed well utilizing the features selection with DBN-based classification.
- The research model is compared with current methodologies for validation regarding detection rates, accuracy, precision, and f-measure.

The paper is framed into the following sections. Section 2 briefly analyzes the existing models related to the research work. Section 3 includes the implementation details of the proposed research models. Section 4 provides the experimentation outcomes of the research model and a comparison with existing IDS models. Section 5 presents the work conclusions and suggestions for future directions.

## 2. Related Works

This section presents an analysis of current models based on the application of the proposed research. While ML and DL-based solutions have seen widespread application in recent years for the aim of network intrusion detection, there is still a significant amount of space for improvement in both the accuracy and classifier's performances. As a result, a training algorithm was built for tuning the parameters of the DCNN so that it could more correctly detect intrusions in IoT networks. An improvement to the exploration and exploitation capabilities of the PSO approach was developed in [6] in the form of a neighbourhood search-based particles swarm optimization approach. The benefit of the model was utilized to train the DL model optimally in its role as the network IDS model to achieve higher levels of precision and performance. The provision of secure computing for the IoT absolutely requires the development of cyber security. Based on the benefits of evolutionary approaches, features extraction and selection models were presented in [7] for the IDS system. It was decided to implement the technique for the extraction of features using CNN. For choosing the useful characteristics and achieve higher levels of accuracy in the classification process, a binary implementation of the Aquila optimization was utilized as the feature selection strategy. The results have demonstrated that this model is superior to others; yet, it does have some shortcomings that may have been addressed and overcome to increase its overall performance. Because CNNs have the potential to automatically execute faster computations and extract features from data, this approach of feature extraction is regarded as one of the most effective and trustworthy available today. Using CNN for IoT features extraction with hybrid layer was built and deployed as part of the work that was reported in [8].



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

This work was done to improve anomaly identification in the IoT. In addition, an upgraded binary multi-objective Capuchin Search Algorithm was designed for the aim of performing features selection in an effective manner. This model has demonstrated superior performances in identifying intrusions in the IoT across all evaluation metrics because of improved feature extraction, more accurate feature selection, and more robust classification. The CNN was used in yet another anomaly-based IDS model for IoT network that was proposed in [9]. A multiclass classification model was developed with the assistance of the CNN architecture. The implementation of this approach included the use of 1D, 2D, and 3D CNNs. Transfer learning was employed for implementing multiclass and binary classifications using the CNN model that had been pre-trained for multiclass classification. The findings of this model could be useful for the process of designing the anomaly-based IDS for IoT that would have the higher detection rates while maintaining a lower number of false alarms.

Identification of malicious traffic by use of DL approaches has recently emerged as an essential component of network IDS. Using the recurrent neural network, a DL approach to detect anomalies in the IoT network was developed in [10]. Because the CNN model processed the input features without discarding any essential information in the process, these characteristics are exceptionally well-suited for feature learning. A DL model that combines CNN and RNN was suggested. In conclusion, a lightweight DL model for binary classification that uses Long Short-Terms Memory, Bidirectional-LSTM, and Gated Recurrent Units-based approaches was proposed. It would have been possible to improve these models' capacity for detection by employing an optimization method. The effectiveness of three DL approaches (CNN, Deep Feed Forward, and Recurrent Neural Networks) in identifying attack vector was assessed and correlated with the effectiveness of three Shallow Learning approaches (Decision Trees, Logistic Regression, and Naive Bayes) in [11]. An investigation and evaluation have been conducted into the effects that Linear Discriminant Analysis, Principal Component Analysis, and Auto-encoder models have on classification performance. Finding a combination of algorithms and classifiers that performs well, and therefore in practical applications, was far from easy and requires additional investigation. In the area of intrusion detection, DL has demonstrated some potential. It was proposed in [12] that generative DL approaches, such as bidirectional generative adversarial network and adversarial autoencoder were used for detecting intrusions according to the examination of the information from the network. This study investigated the application of generative DL approaches that could identify and classify IoT attacks automatically using the IoT-23 dataset. The accuracy was very close to zero in both models, lending credence to the hypothesis that Bidirectional Generative Adversarial Network and Adversarial Autoencoder could represent the information, but neither could produce information that originated from the same distribution. In [13], there is a description of the development of an IDS model that makes use of DL and an optimization algorithm to construct effective features extraction and selection approaches. To extract the important features, a CNN approach with one dimension was implemented. The reptile search method chose the appropriate feature subsets to lessen the data dimensionality and improved the accuracy of classification. The CNN application with RSA has a substantial impact on the process of IDS classification. To boost the overall performances, the convergence speed of the RSA may have been boosted even further.

An anomaly-based IDS for IoT network was utilized in conjunction with a CNN and gated recurrent units (GRU) for detecting and classifying multiclass and binary IoT data in [14]. The performances of the multiclass model was proved by the BoT-IoT data set, the IoT network intrusions dataset, the MQTT-IoT-IDS2020 dataset, the IoT-23 dataset, and the IoT-DS-1 and 2 data sets. The performance of the model regarding the IoT dataset was average for the various attack categories. It was possible to achieve a higher false negative value for both the scan and Mirai classes. This model was successful in achieving a higher detection score while also achieving lower false negative and positive rates across the board, including normal and attack classes. In [15], the authors developed a universal feature set with the goal of enhancing the performances of ML approaches for identifying botnet attacks, regardless of the dataset. To detect botnet attacks across three separate datasets, the features set was utilized to train ML approaches like K-Nearest Neighbor, Naive Bayesian, Logistic Regressions and Random Forests. When trained and tested using the universal feature set across a variety of datasets, these algorithms detected botnet attacks very effectively. Because there are so many kinds of botnet attacks, the ML models spanning all the datasets must be trained.

### 2.1. Research Gap

Traditional ML approaches have been found to be successful in detecting attacks, although they generally need human feature engineering. This technique is limited in its capacity to acquire all relevant information and may have challenges in being applicable to different network settings. On the other hand, DL approaches have emerged as a viable alternative for improving detection accuracy and decreasing false positives. Nevertheless, these techniques need significant computer resources for training and require clarity in their interpretation, which presents difficulties for immediate implementation in IoT systems. In addition, a thorough assessment is needed to determine their effectiveness against more complex and sophisticated attack techniques. There is an urgent requirement for the assessment of datasets that are more standardized and representative of real-world scenarios. This would enable the precise evaluation and comparison of various detection strategies. It is crucial to address



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

these areas of research that are currently lacking to make progress in improving the current state of IoT-based attack detection and protection mechanisms.

## 3. Proposed IDS Methodology

In this research work, a DL-based hybrid intrusion detection model for identifying and categorization of attacks in IoT. This research model includes data preprocessing, features extraction, selection of features, and classification processes. The CNN model is used for extracting features from the preprocessed dataset. A Quantum-based CSO algorithm selects the best optimal subsets of features from the features extracted. Finally, the multiclass classification was done utilizing the DBN model based on the features selected. This research used Bot-IoT and IoT-23 data sets to assess the research model. Figure 3 represents the workflow of the developed IDS model. The application of DL methods can be utilized for binary classification, multi-classification, prediction, and the extraction of relatively high features to enable faster convergence times and permit data reduction. It is possible to perform a procedure known as feature engineering to acquire a low false-negative rate and enhance accuracy. During this phase of the DL model development process, this takes into consideration many approaches, for transforming non-numerical to numerical data, scaling and normalization [16].

### 3.1. Input Datasets

**Bot-IoT:** The most recent IoT network intrusion detection dataset is referred to as Bot-IoT. This dataset depicted a network environment that included both typical traffic and the botnet activity. Bot-IoT encompasses typical network traffics of IoT and four distinct forms of attack, namely denial of service, distributed denial of service, reconnaissance, and theft. Bot-testbed IoT's includes a wide variety of IoT use cases, such as the motion-activated lighting system, weather station, garage door with remote-control, smart thermostat, and smart refrigerator. Only five percent of the total Bot-IoT data set, which totals 3.6 million data, was used for the trials; the full data set, on the other hand, has over 72 million records. The five percent of the whole data set that contains the best features derived from the original data is then split into the main classes that are outlined in Table 1 [17].

**Table 1:** Bot-IoT Dataset

| Attack Classes | For Training | For Testing |
|---|---|---|
| Theft | 65 | 14 |
| Normal | 370 | 107 |
| Reconnaissance | 72919 | 18163 |
| DoS | 1320148 | 330112 |
| DDoS | 1541315 | 385309 |
| **Total** | **2934817** | **733705** |

**IoT-23:** IoT-23 is a network traffic dataset that includes 20 malicious software subsets as well as three benign software subsets. The IoT-23 data set comprises twenty distinct types of network activities to replicate a variety of applications for IoT-based systems. Collecting the network traffics from three independent IoT devices allowed for the collection of the innocuous network traffic. As shown in Table 2, the attacks that are included in the IoT-23 dataset can be broken down into nine distinct categories. The dataset includes a total of 21 feature attributes, one of which is the class label. This data set offers the community access to two separate data sets: the initial data set includes both regular and abnormal networks, whereas the other data set only comprises normal captures of IoT networks. The key advantages of the IoT-23 data set are that it accurately represents the recent trends in IoT network traffics and it is publically accessible [18].

**Table 2:** IoT-23 Dataset

| Attack Classes | For Training | For Testing |
|---|---|---|
| Normal | 3451021 | 862755 |
| Attack | 1373422 | 343356 |
| Mirai | 605 | 151 |
| File Download | 6428 | 1607 |
| HeartBeat | 10316 | 2579 |
| C&C | 19185 | 4796 |
| Torii | 27086 | 6772 |
| Port Scan | 52755890 | 13188973 |



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

| | | |
|---|---|---|
| DDoS | 16615190 | 4153798 |
| Okiru | 10974602 | 2743650 |
| **Total** | **85233745** | **21308437** |

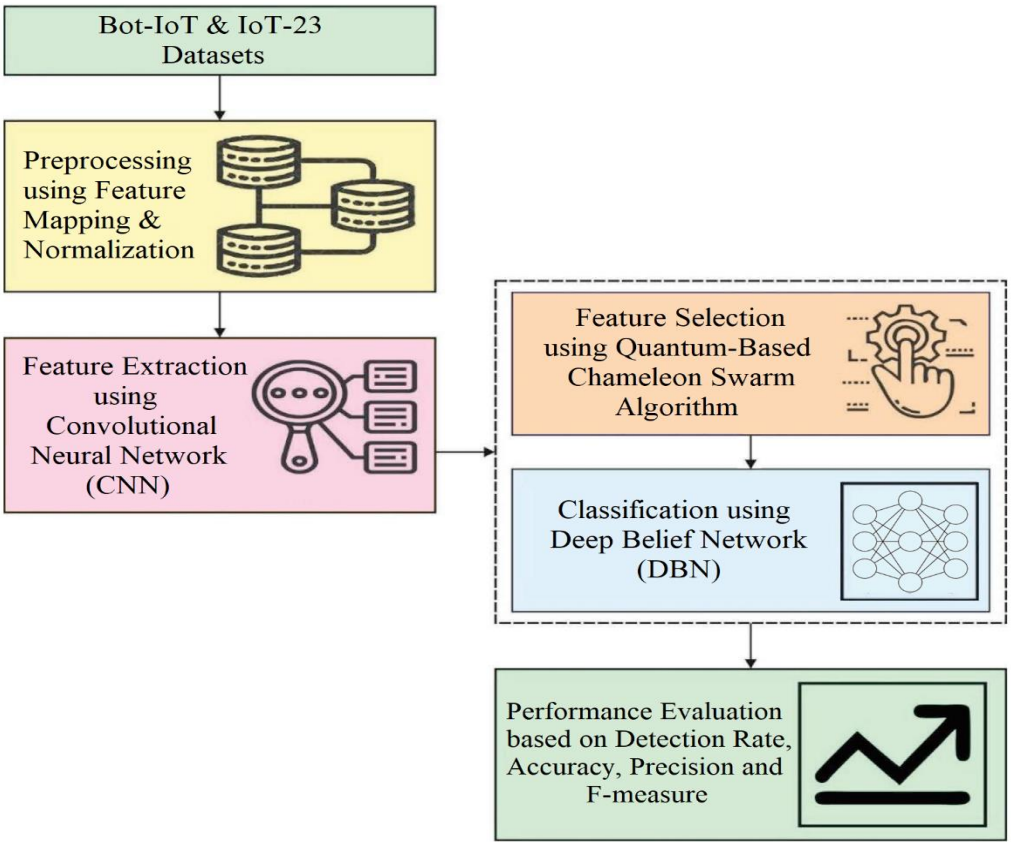


**Figure 3.** Proposed DL-based IDS Model

### 3.2. OMD-ODS-ELM for stock Trend Prediction

When it comes to the process of constructing DL models, the phase known as "data preprocessing" is an essential one. That is especially important in IoT attack data sets, which could be very extensive and difficult to understand. The characteristics of network traffic comprise both quantitative and symbolic characteristics; however, machines are only able to recognize numerical quantities. Consequently, it is necessary to first transform each symbol attribute into its corresponding numerical value. Both the Bot-IoT and the IoT-23 data sets have a total of 9 symbolic features and 36 numerical characteristics. The IoT-23 dataset also has a total of 6 symbolic features and 14 numerical features. Because the values of the network characteristics vary so widely, it is necessary to standardize the values of each feature within the tolerable ranges to decrease the impact that the different feature values have on one another. Since the DBNs need data input to lie within the scale of zero and one, all the features need to be normalized with the appropriate lowest and highest values of features and must also present among the same scale of zero and one. Also, the procedure of normalization was carried out on the test set. The minimum and maximum normalization, which is often performed between 0 and 1, is the data normalization approach that is used the most frequently. The following formula can be used to normalize each feature value:

$$x_i^* = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1)$$



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

Thus, $x_i$ denotes the initial value for the feature, and $x_{max}$ and $x_{min}$, which are derived from the input data $x$, stand for the highest and lowest possible feature values, respectively. Following the process of normalization, each data point is then scaled to fall somewhere between 0 and 1 [19]. In this instance, the datasets were segmented in such a way that 80% of them would be used for the training of the model, while the remaining 20% would be used to test the model. Using CNN as a feature extraction model, QCSO as a feature selection model, and DBN as a classifier for attack classification combines three techniques to build an effective IDS.

*3.3. Feature Extraction using CNN*

CNNs are effective DL models that can extract valuable features from raw data such as images, text, and network traffic data. These features may be extracted from any type of data. It is possible to train a CNN on the Bot-IoT and IoT-23 datasets in order to extract significant features that can then be used for the classification of attacks. CNNs can learn and represent complex features automatically, against the regular machine learning approaches that rely on the data extraction through manual. While this, the models that are based on CNN can differ related to the pooling operation, the count of fully connected layers, the size of kernel, and the initialization procedure. Moreover, the number of convolution layers can change. In this process, the key target was for learning useful representation from the original information, which provides increase in accuracy of the model. The CNN can learn more complicated representations by using the network traffic samples as training data. The CNN accomplishes pattern extraction by means of a convolution process, during which the weights are shared amongst the various layers and channels. The patterns that are extracted are local and position-invariant. After the stage of learning with the CNN architecture, the features selection approach filters the extracted features by selecting the essential attributes that maximize the classification performance. This happens after the stage of learning. The essential capability of CNNs is to share weights between different layers for minimizing the model's complexity [7].

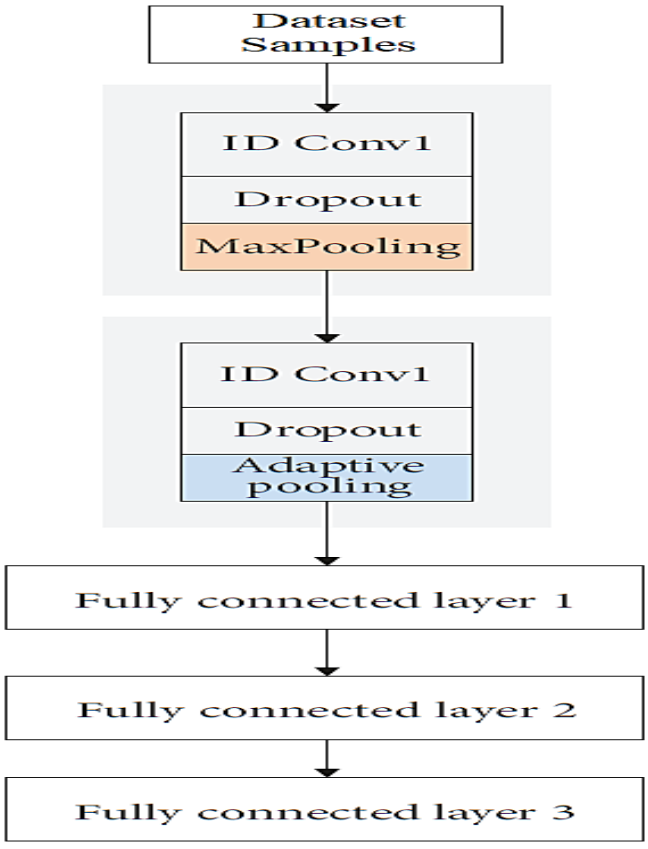


**Figure 4.** CNN Architecture

In the CNN architecture that was implemented, the Conv block was followed by the ReLU unit expressed in (2), which prevents negative or lower values from being distributed, while the pooling operators are utilized for minimizing the activation map's dimensionality $ReLU(x)$ of the $x$ that was inputted:



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

$$ReLU(x) = \max(0, x) \quad (2)$$

To simplify the model and prevent it from becoming overly specific to its data, dropout layers are applied during training with a regularization rate of 0.5. This causes a random number of neurons to be removed. In addition to that, the Conv1 layer has the kernel sizes of 1x3, 64 filters, and the stride of 1x1. The following equation provides a definition for a one-dimensional convolution that was performed on the data input $x^{l-1}$ from the layers before it:

$$x^l = W^l \cdot x^{l-1} + b^l \quad (3)$$

The output is denoted by $x^l$, while the bias and weight matrix associated with the *l*-th layers are denoted by $W^l$ and $b^l$, respectively. Thus, two distinct methods of pooling, known as adaptive average pooling and max-pool with 2x2 size, were utilized. As can be seen in Figure 4, following the completion of the final pooling process, the derived feature maps are then entered into a series of FC layers. For feature extraction, the layers FC1 to FC3 are utilized [13].

### 3.4. Feature Selection using QCSO Algorithm

The behaviour of chameleons in the natural world served as the inspiration for a form of metaheuristic optimization algorithm called the CSO. By emulating the collective intelligence of a group of chameleons, this algorithm is said to be able to figure out what the best solution is to any problem that has been presented to it. The core idea behind CSO is that the ability of chameleons to change their appearance to blend in with their environment demonstrates a behaviour that can be utilized to optimize difficult situations. They have a full stereoscopic view in all 360 degrees, which gives them a superior aptitude to track prey. The sticky tongues used for feeding by chameleons adhere to the body of their prey and perform a variety of surface actions, including entanglement and moist adhesion, all while maintaining an extremely high tongue acceleration of 2590 meters per second. The proposed algorithm is designed to imitate the natural process of hunting that chameleons engage in, which includes pursuing, capturing, and prey tracking. The algorithm depicts each possible solution as a chameleon that is part of a larger group. The chameleons will then wander around in the search space, altering their characteristics in accordance with the conditions of their immediate surroundings to locate the optimal solution. The CSO algorithm has several benefits that rival those of competing optimization methods. For instance, it can be scaled to a very large extent, it is very versatile, and it can be adapted to a wide variety of situations. It is also effective regarding the time needed for computation and can deal with issues involving a significant number of variables [20].

The CSO can be applied in the role of a features selection model for determining the aspects of a dataset are the most important. The purpose of features selection is to improve the accuracy and efficiency of a DL model by choosing a subset of the characteristics within a larger set of features that are the most relevant. In this work, the CSO algorithm is combined with the quantum-based optimization (QBO) to create the Quantum-Based CSO (QCSO) algorithm. This system can discover the most essential characteristics for classifying attacks in an effective and precise manner. The binary number was utilized for representing the characteristics in the QBO technique, and this determines either they would be chosen (1) or eliminated (0). Each property in QBO was represented by the quantum bits, also known as a Q-bit (q), here *q* stands for the binary values superposition (zero and one). Using a method that is based on quantum mechanics is the cornerstone of the proposed QCSO, which aims to improve the effectiveness of the CSO. Using QBO with the intention of improving one's capacity to strike the balances among exploitation and exploration while searching for the optimal solution was the major objective. After that, the randomly generated numeric values for each of the *N* chameleons were assigned, and the value of fitness is determined. The chameleon with the best possible value of fitness was chosen at this point. After this step is complete, the solution is improved by utilizing CSO exploitation. The process of updating will continue until the stop criteria have been satisfied, at which point it will be terminated. After that, the dimensions of the test set was reduced depending on the best solutions, and the subsequently deployed QCSO as the feature selection was evaluated using various metrics [21]. At this step, the initial chameleons, will later stand in for the population of the solutions, are created. Every solution has a total of D Q-bits. Here D was the total count of features. As a result, the solution $X_j$ could be expressed as the equation (4), which is as follows:

$$X_j = [q_{i1}|q_{i2}| \dots |q_{iD}|] = [\theta_{j1}|\theta_{j2}| \dots |\theta_{jD}|], j = 1,2 \dots m \quad (4)$$

$X_j$ is the superpositions set of probability for those characteristics that were either chosen or not selected in this equation. In the subsequent stage of QCSO, the solutions were revised till they meet the standards for the final phase. Using Equation (5), the first stage that should to be made to perform this process was to transform $X_j$ into the binary format as ($BX_{j,i}$):

$$BX_{j,i} = \begin{cases} 1 & if\ rand < |\beta|^2 \\ 0 & otherwise \end{cases} \quad (5)$$



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

A value chosen at random was returned by $rand \in [0,1]$. The next step that needs to be done is to learn the classifier by making use of the training features that have been allotted at the indexes that are similar to those in the $BX_{j,i}$ array, and then to calculate the value of fitness, which was represented as follows:

$$Fit_j = \rho \times \gamma + (1 - \rho) \times \left(\frac{|BX_{j,i}|}{D}\right) \quad (6)$$

The total features selected was denoted by $|BX_{j,i}|$ in Equation 6, and the error in classification caused by the classifier is represented by $\gamma$. The value $\rho \in [0,1]$ makes the fitness score of two sections equal. The following step is to allot the optimal solution $X_b$, which should have the smallest value for the $Fit_b$ variable. The CSO operators were then utilized after that. The size of the test set is minimized by selecting just the values that correspond in the $BX_b$ matrix. After that, the classifier receives the reduced test set that it was given. The next step in the process is to determine how accurately various measurements portray the quality of the output.

*Input: The parameters include the number of solutions (N), the tested data set with features D, the count of iterations (tmax), and additional metrics.*

*Initial Phase*

*Build training and test sets, with those two sets representing, respectively, 80% and 20% of the total.*

*Apply Equation (4) to the process of building the populations, X.*

*Second Phase*

*t = 1*

*While (t < tmax)*

*The quantum representation of $X_j$ can be obtained by solving Equation (5).*

*Determine the level of $X_j$'s fitness based on the information provided by the training sample using Equation (6).*

*$X_b$ should be reserved for the optimal solution.*

*X will be kept up to date by using CSO operators.*

*t = t + 1*

*End While*

*Third Phase*

*Using $X_b$, remove any features from the testing set that are not relevant.*

*Using various metrics, QCSO's effectiveness can be evaluated.*

From the Bot-IoT dataset, pkSeqID, Seq, Mean, Stddev, Min, Max, Srate, Drate, NINConn, PSrcIP, NINConn, and PDstIP features were selected. From the IoT-23 dataset, Bwd Pkt Len Min, Pkt Len Mean, Pkt Len Min, Bwd Header Len, Pkt Size Avg, Bwd IAT Max, Flow Byts/s, Bwd Pkt Len Mean, Fwd Pkt Len Mean and Flow IAT Max features are selected.

### 3.5. Classification using DBN

The DBN is a neural network model that is frequently utilized in DL methods. This model traditionally employs the blocks of a Restricted Boltzmann machine (RBM), and it is composed of numerous RBM. When there is only one hidden layer in the RBM model, obtaining features in the information was not the appropriate method. The feature that the RBM has learned after it has been trained could be utilized as the data of input to train another RBM. As a result, the properties of a completed RBM model end up being the learnt properties of the entire training model. DBN is produced by the layer-wise learning method described here. The characteristics of the DBN input data can be derived using this method. The DBN was composed of RBM layers and organized from bottom to top with the logistic regression layers serving as the outcome of the model. Backpropagation, often known as BP, was frequently utilized in the process of training the conventional artificial neural networks (ANN). When applied to network with multiple model parameters, the BP technique may result in fewer optimizations or may cause an issue with excessive reinforcement. In DBN networks, the pre-training procedure takes the alternate sampling form and greedy wise layer. The pre-training of the RBM and all the DBNs in the greedy layers is accomplished with the help of alternative sampling [22].



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

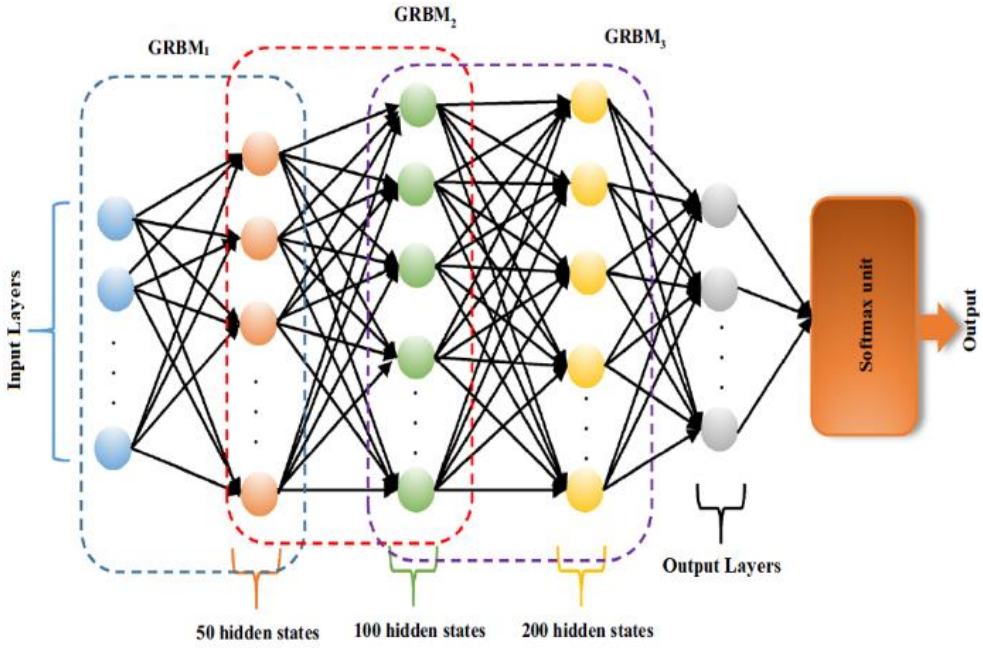


**Figure 5.** Architecture of DBN Model

The application of a classification problem to the DBN network was trailed by various learning processes like the distinctive learning and pre-training that fine-tunes the field of parameters. These procedures are carried out after the classification task has been applied to the DBN network. Following the unsupervised pre-processing in the form of greedy wise-layer, the abstract value representation of $x$ that exists in the $k$th layer is denoted by $h^K(X)$. The labelled data are utilized in the process of adjusting the $W$ parameter space. To make this change to the level of precision, the variable's final layer that prepare the appropriate label samples in the training data set are given more variables to produce higher distinguishing performance. This method of optimization is represented by the formula that is following.

$$f(h^K(X), Y) = \sum_{i=1}^{n} \sum_{j=1}^{c} T\left(h^K\left(x_i^j\right) \times y_i^j\right) \quad (7)$$

The loss function is denoted by $T$ in this context. In backpropagation, if the loss function looks like this (as shown below), the squared error function is frequently utilized.

$$r = h^K\left(x_i^j\right) \times y_i^j \quad (8)$$

Logistic regression was expanded to include multiple categories, and softmax regression is the result of such development. The DBN design features a classifier that is implemented in the Softmax layer. It is possible to utilize softmax regression with the equation $y^{(i)} = \in \{0, N\}$. In this context, "N" refers to the total number of different classes. The components that make up Softmax are the input, classifier, and output units [23].

## 4. Experimental Results and Discussion

This section provides a comprehensive examination of the experimental results obtained from the suggested DL-IDS model. The DL-IDS model is experimented and evaluated using the MATLAB 2019b simulation tool, which is installed on a laptop equipped with a Core i7 processor working at 3.20 GHz and 12GB of RAM. The laptop runs on the Windows 11 operating system. The performance measures, such as detection rate, accuracy, F1-scores, and precision are computed by considering the multi-class categorization of assaults. The DL-IDS model underwent experimental assessments and was compared with many models for validation purposes.

### 4.1. Performance Parameters

To compute the performances of the DL-IDS model in identifying intrusions in IoT based on multi-class classification, the performance evaluation was conducted using accuracy, DR, f1-scores, and precision rates. The performance indicators were assessed using the confusion matrix, which considers the values of true positives (TPs), false positives (FPs), true negatives



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

(TNs), and false negatives (FNs), as shown in the following equations. The detection rate is also referred to as recall or sensitivity.

$$Accuracy = \frac{TP+TN}{FN+TP+FP+TN} \quad (9)$$

$$F1score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (10)$$

$$DR = \frac{TP}{TP+FN} \quad (11)$$

$$Precision = \frac{TP}{TP+FP} \quad (12)$$

### 4.2. Performance Evaluation

The classification of each attack from the test set of the Bot-IoT dataset evaluated utilizing the DL-IDS model was represented in Table III. Classes in the dataset, such as Normal, Theft, Reconnaissance, DoS, and DDoS, were evaluated individually based on the result parameters. The research model obtained 99.07% accuracy in classifying normal class, 100% for Theft, 99.10% for Reconnaissance, and 99.66 for both DoS and DDoS classes. The accuracy rate was higher in classifying DoS and DDoS attacks than in other classes. The detection rate was 100% in classifying the Theft class and 99.35%, 99.67%, and 99.70% for the Reconnaissance, DoS, and DDoS classes. The DR score for the normal class was 98.92%, which is low compared to the other classes. The precision score for the Theft and Normal classes were 100% and 99.58, 99.96, and 99.86% for Reconnaissance, DoS, and DDoS classes. The F1-score obtained for the Theft class was 100%, 99.82% for DoS, 99.78% for DDoS, and 99.46% for both Normal and Reconnaissance classes. Overall, the proposed DL-IDS model obtained the best scores with all the parameters for the Theft attack class with 100% performance. Figure 6 depicts the graph of the proposed model's performance in classifying each attack in the Bot-IoT dataset.

**Table 3:** Results of the DL-IDS model on Bot-IoT Dataset

| Attack Class | Accuracy | DR | Precision | F1-score |
|---|---|---|---|---|
| Normal | 99.07 | 98.92 | 100 | 99.46 |
| Theft | 100 | 100 | 100 | 100 |
| Reconnaisance | 99.10 | 99.35 | 99.58 | 99.46 |
| DoS | 99.66 | 99.67 | 99.96 | 99.82 |
| DDoS | 99.66 | 99.70 | 99.86 | 99.78 |

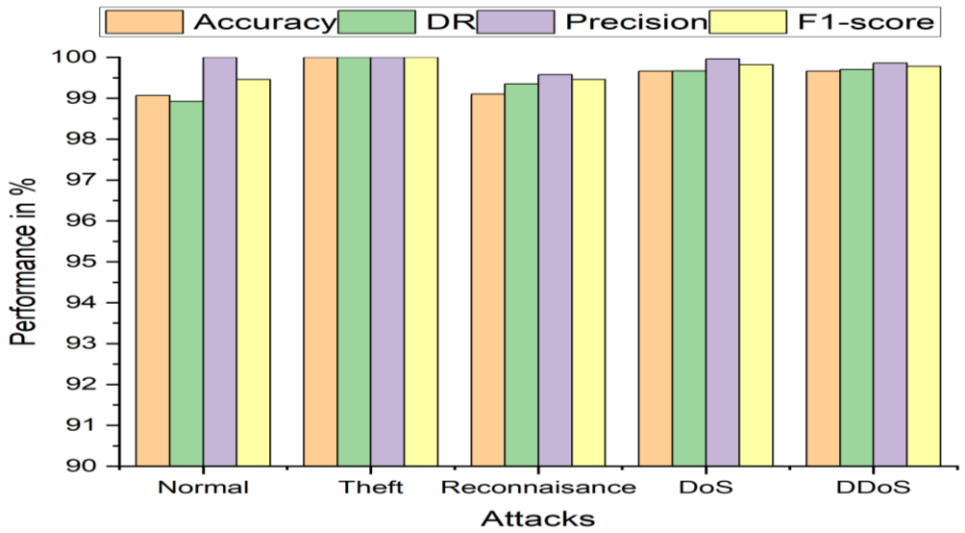


**Figure 6.** Multi-class Classification Results of DL-IDS Model using Bot-IoT Dataset

**Table 4:** Results of the DL-IDS model on IoT-23 Dataset

| Attack Class | **Accuracy** | **DR** | **Precision** | **F1-score** |
|---|---|---|---|---|



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

| | | | | |
|---|---|---|---|---|
| Attack | 99.61 | 99.67 | 99.88 | 99.77 |
| Mirai | 99.34 | 99.01 | 100 | 99.50 |
| File Download | 99.56 | 99.50 | 99.80 | 99.65 |
| HeartBeat | 99.26 | 99.51 | 99.56 | 99.54 |
| C&C | 99.04 | 99.11 | 99.75 | 99.43 |
| Torii | 99.53 | 99.67 | 99.80 | 99.73 |
| Port Scan | 99.39 | 99.39 | 99.99 | 99.69 |
| DDoS | 99.45 | 99.50 | 99.93 | 99.72 |
| Okiru | 99.58 | 99.47 | 99.95 | 99.71 |
| Normal | 99.80 | 99.88 | 99.91 | 99.89 |

Table IV represents the classification evaluation of all the attacks presented in the IoT-23 data set according to the classification parameters. The IoT-23 data set has ten attack classes ranging from Normal to Okiru, as shown in the table. The classification evaluation was performed for all the attack classes individually. Among the performances of all the classes, the highest accuracy score was obtained for the Normal class with 99.80% and 99.61% for the Attack class. The highest detection score was obtained for the Normal class with 99.88% and 99.67% for both Attack and Torii classes. The best precision score was obtained for the Mirai class with 100% and 99.99% for the Port Scan class. The best f1-score was obtained for the Normal class with 99.89% and 99.77% for the Attack class. Overall, the average scores obtained for all the attack classes were 99.45% accuracy, 99.47% DR, 99.85% precision, and 99.66% f1-scores. Figure 7 represents the graphical plot of the DL-IDS performance in classifying each attack in the IoT-23 data set. For the Bot-IoT data set, the average scores were 99.49% accuracy, 99.52% DR, 99.88% precision, and 99.70% f1-scores. Comparing the average scores of both datasets, the performances obtained for the Bot-IoT are higher than those of the IoT-23 dataset.

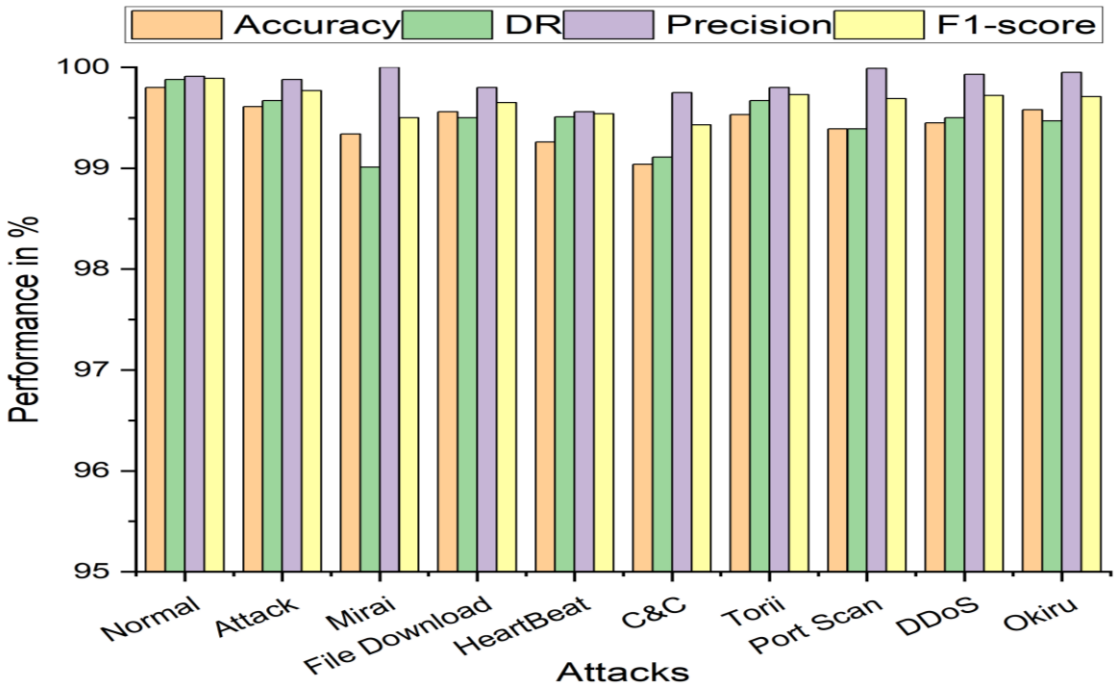


**Figure 7.** Multi-class Classification Results of DL-IDS Model using IoT-23 Dataset

**Table 5:** Comparison of the DL-IDS Model's Performance

| **Models** | **Accuracy** | **DR** | **Precision** | **F1-score** |
|---|---|---|---|---|



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

| | | | | |
|---|---|---|---|---|
| NSBPSO-DCNN [6] | 98.86 | 99.03 | NA | NA |
| AQU-CNN [7] | 98.92 | 98.90 | 98.91 | 98.90 |
| CNN-1D [9] | 99.30 | 99.20 | 99.26 | 99.23 |
| RSA-CNN [13] | 99.02 | 99.03 | 99.09 | 99.07 |
| Deep DCA [17] | 98.73 | 98.36 | 99.17 | 98.77 |
| LSTM-DSAE [18] | 99.10 | 92.00 | 100 | 95.00 |
| MVO-CNN [13] | 99.03 | 98.96 | 99.00 | 98.98 |
| DL-IDS (Bot-IoT) | 99.49 | 99.52 | 99.88 | 99.70 |
| DL-IDS (IoT-23) | 99.45 | 99.47 | 99.85 | 99.66 |

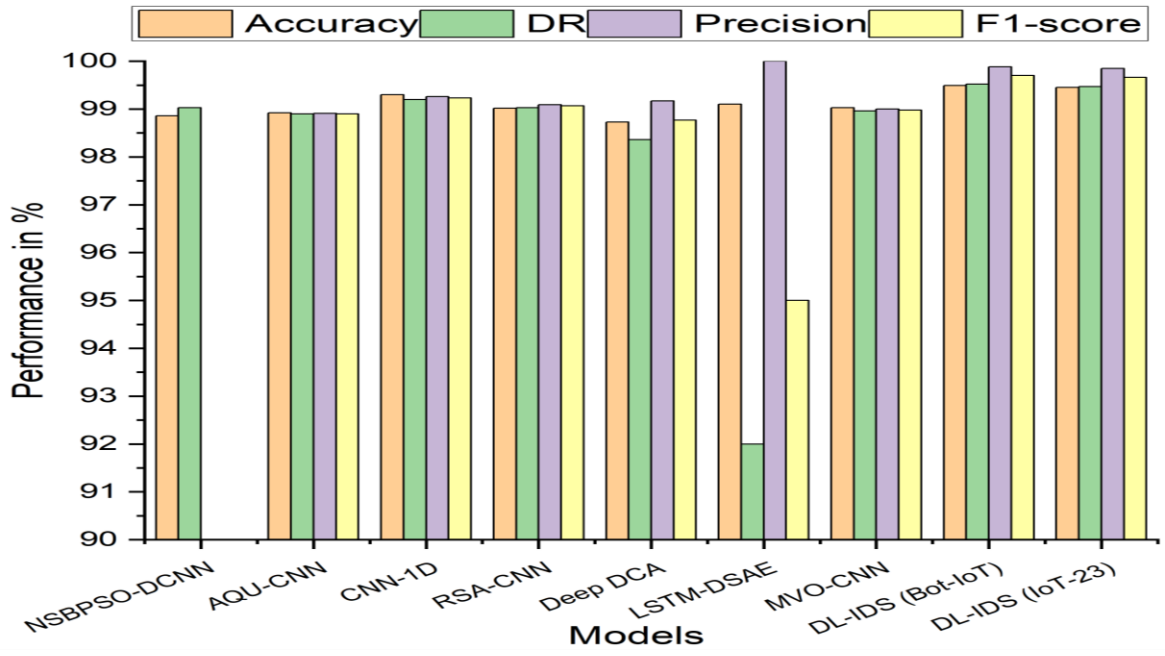


**Figure 8.** Comparison of Performance Analysis

The comparison of the DL-IDS model's performances with existing models has presented in Table V. The comparison includes the average accuracy, DR, f1-scores, and precision scores of the DL-IDS model and was compared with the models analyzed from the literature review. The DL-IDS model's best accuracy score was 99.49%, which was 0.1% to 0.7% higher than the compared models. The CNN-1D has a close accuracy score of 99.30% and the Deep DCA methodology has the least accuracy with 98.73%. The DR of the DL-IDS was 99.52%, which was 0.3% to 7.5% enhanced than the current models in this research. The CNN-1D methodology has a close DR of 99.20%, and the LSTM-DSAE methodology has the least DR of 92%. The precision value of the DL-IDS model was 99.88%, which was 0.6% to 0.9% higher than the compared methodologies except for the LSTM-DSAE. The LSTM-DSAE model has a precision score of 100%. The f1-score of the DL-IDS model was 99.70%, which was 0.4% to 4.7% higher than the current methodologies. The CNN-1D methodology has a close f1-score of 99.23% compared to the proposed model, and the LSTM-DSAE model has the least f1-score with 95%. According to this comparison, the proposed DL-IDS model has overcome the compared models in terms of multi-class classification.

## 5. Conclusion

In this research, a DL-based IDS model has been proposed for the multi-class attacks classification in IoT networks. The proposed DL-IDS model includes data preprocessing, feature extraction, feature selection, and classification processes. The



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

IoT-23 and Bot-IoT data sets were utilized as input for the research model. In preprocessing, the datasets were normalized, and the missing data were replaced. After preprocessing, the features were extracted using the CNN model. The features selection was carried out from the extracted features by implementing the QCSO algorithm, which selected 12 features from the Bot-IoT data set and 10 features from the IoT-23 data set. Using these features selected, the multi-class classification was made using the DBN for each attack presented in the datasets. The classification performance was performed individually for both datasets and evaluated using accuracy, detection rate, precision, and f1-scores. The performances of the DL-IDS model were compared with the models analyzed from the literature survey discussed in this work. The average scores obtained using the IoT-23 dataset include 99.45% accuracy, 99.47% DR, 99.85% precision, and 99.66% f1-scores. For the Bot-IoT data, the average scores were 99.49% accuracy, 99.52% DR, 99.88% precision, and 99.70% f1-scores. The proposed DL-IDS model has outperformed the other compared models regarding multi-class attack classification. The limitations of this work include the number of training and testing instances that could have been increased in some attack classes, which could improve the model's performance. Considering the limitations, the training and test sample can be increased in the future, and the Recurrent Neural Network or deep transfer learning models can be used for the feature extraction process. Also, the DL-IDS model can be deployed for real-time attacks detections to validate the efficiency of the model.

**References**


**[1]** A. Khraisat and A. Alazab, "A critical review of intrusions detections system in the internet of things: techniques, deployment strategy, validation strategy, attack, public dataset and challenge," *Cybersecurity*, vol. 4, no. 18, pp. 1-27, 2021.

**[2]** A. A. Anitha and L. Arockiam, "A Review on Intrusions Detections System to Secure IoT Network," *International Journal of Computer Networks and Applications*, vol. 9, no. 10, pp. 38-50, 2022.

**[3]** E. Gyamfi and A. Jurcutt, "Intrusions Detections in Internet of Things System: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Dataset," *Sensors*, vol. 22, no. 3744, pp. 1-33, 2022.

**[4]** Kumar, I., Kumar, A., Kumar, V.D.A. et al. Dense Tissue Pattern Characterization Using Deep Neural Network. Cogn Comput (2022). https://doi.org/10.1007/s12559-021-09970-2.

**[5]** A. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba, and S. A. Bahaj, "Deep Learning for Intrusions Detections and Security of Internet of Things (IoT): Current Analysis, Challenge, and Possible Solution," *Security and Communication Networks*, vol. 2022, 4016073, pp. 1-13, 2022.

**[6]** S. Baniasadi, O. Rostammi, D. Marttín, and M. Kavah, "A Novel Deep Supervised Learning-Based Approach for Intrusions Detections in IoT System," *Sensors*, vol. 22, no. 4459, pp. 1-17, 2022.

**[7]** A. Fatani, A. Dahau, M. A. A. Al-qanes, S. Lui, and M. A. Elaaziz, "Advanced Features Extractions and Selections Approach Using Deep Learning and Aquila Optimizers for IoTs Intrusions Detections Systems," *Sensors*, vol. 22, no. 140, pp. 1-20, 2022.

**[8]** S. Hemamalini ,V. D. Ambeth Kumar ,R. Venkatesan,S. Malathi. (2023). Relevance Mapping based CNN model with OSR-FCA Technique for Multi-label DR Classification. Journal of Fusion: Practice and Applications, 11 ( 2 ), 90-110.

**[9]** I. Ullah and Q. H. Mahmoud, "Design and Developments of a Deep Learning-Based Model for Anomaly Detections in IoT Network," *IEEE Access*, vol. 9, pp. 103906-103926, 2021.

**[10]** I. Ullah and Q. H. Mahmoud, "Design and Developments of RNN Anomaly Detections Models for IoT Network," *IEEE Access*, vol. 10, pp. 62722-62750, 2022.

**[11]** Malathi S, Arockia Raj Y, Abhishek Kumar, V D Ashok Kumar, Ankit Kumar, Elangovan D, V D Ambeth Kumar*, Chitra B & a Abirami (2021) Prediction of cardiovascular disease using deep learning algorithms to prevent COVID 19, Journal of Experimental & Theoretical Artificial Intelligence, DOI: 10.1080/0952813X.2021.1966842

**[12]** N. Abdalgawad, A. Sajjun, Y. Kadoura, I. A. Zualkkernan, and F. Aloal, "Generative Deep Learning to Detect Cyberattack for the IoT-23 Datasets," *IEEE Access*, vol. 10, pp. 6430-6441, 2022.

**[13]** A. Dahou et al., "Intrusions Detections Systems for IoT Based on Deep Learning and Modified Reptile Search Algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, 6473507, 2022.

**[14]** I. Ullah, A. Ullah, and M. Sajjad, "Towards a Hybrid Deep Learning Model for Anomalous Activities Detections in Internet of Things Network," *IoT*, vol. 2, pp. 428–448, 2021.

**[15]** Ruphitha, S.V., Ambeth Kumar, V.D., " Predictive analysis of postpartum haemorrhage using deep learning technique", Advances in Parallel Computing, 2021, 38, pp. 168–172.

**[16]** Sherubha, "Graph Based Event Measurement for Analyzing Distributed Anomalies in Sensor Networks", Sådhanå(Springer), 45:212, https://doi.org/10.1007/s12046-020-01451-w




DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024

**[17]** Piyush K. Pareek, Pixel Level Image Fusion in Moving objection Detection and Tracking with Machine Learning ",Fusion: Practice and Applications, Volume 2 , Issue 1 , PP: 42-60, 2020

**[18]** Shivam Grover, Kshitij Sidana, Vanita Jain, "Egocentric Performance Capture: A Review", Fusion: Practice and Applications, Volume 2, Issue 2 , PP: 64-73, 2020.

**[19]** Abdel Nasser H. Zaied, Mahmoud Ismail and Salwa El- Sayed, A Survey on Meta-heuristic Algorithms for Global Optimization Problems, Journal of Intelligent Systems and Internet of Things,Volume 1 , Issue 1 , PP: 48-60, 2020

**[20]** Mahmoud H.Alnamoly, Ahmed M. Alzohairy, Ibrahim M. El-Henawy, "A survey on gel images analysis software tools, Journal of Intelligent Systems and Internet of Things,Volume 1 , Issue 1 , PP: 40-47, 2021.

**[21]** M. A. Elaziz et al., "A Quantum-Based Chameleon Swarm for Features Selection," *Mathematics*, vol. 10, no. 3606, pp. 1-17, 2022.

**[22]** A. A. Süzen, "Developing a multi-level intrusions detections systems using hybrid-DBN," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1913–1923, 2021.

**[23]** A. Jahanjoo, M. Naderan and M. J. Rashti, "Detection and multi-class classifications of falling in elderly people by deep belief networks algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 4145–4165, 2020.



DOI: https://doi.org/10.54216/JCIM.150112
Received: February 03, 2024 Revised: April 24, 2024 Accepted: July 26, 2024