Journal of Computer Science

Research Article

An Intrusion Detection System Using Optimized Deep Learning for IoT Networks

Chempavathy B. and Mouleeswaran S. K.

Department of Computer Science and Engineering, School of Engineering, Dayananda Sagar University, Bangalore, Karnataka, India

Article history
Received: 09-12-2024
Revised: 10-03-2025
Accepted: 09-05-2025

Corresponding Author:
Chempavathy B.
Department of Computer Science and Engineering, School of Engineering, Dayananda Sagar University, Bangalore, Karnataka, India
Email: chempavathyb@gmail.com

Abstract: In the rapidly evolving landscape of Internet of Things (IoT) networks, assuring security is paramount. Cyber-attacks on IoT networks are evolving in complexity and scale. Intruders employ diverse tactics, including malware, Denial of Service (DoS) attacks, along with unauthorized access. This research aims to propose an optimized Deep Learning (DL) based Intrusion Detection System (IDS) to bolster security within IoT based networks. In order to emphasize the significance of preprocessing data, a critical step is performed to ensure that data is in a suitable format and quality for effective learning and accurate intrusion detection. The preprocessing module performs data cleaning, one-hot encoding and normalization generating normalized inputs for the DL architecture. Subsequently, an efficient Transfer Learning (TL) based Deep Convolutional Neural Network (DCNN) framework is introduced. This framework, characterized by its multi-layered neural networks, autonomously learns and extracts essential features, allowing it to identify unauthorized access attempts and potential malware attacks in an automated and efficient manner. Finally, the neural network training loss is minimized using a hybrid optimization approach that combines Grey Wolf and Improved Salp Swarm algorithms (GW-ISSA). This hybrid algorithm optimizes hyperparameters, leading to faster convergence and reducing the amount of training data required. The NSLKDD dataset is used for simulation using Python, the obtained outcomes contribute to enhancing both security and resilience of IoT networks in the face of emerging threats and vulnerabilities.

Keywords: IoT, Intrusion Detection System, Cyber-Attacks, DCNN, Hybrid GW-ISSA

Introduction

Over the past decade, the IoT has gained significant prominence owing to its unique characteristics and diverse applications related to healthcare, manufacturing and smart environments (Mohy-Eddine *et al.*, 2023). Although IoT offers a wide array of applications and services, it remains vulnerable to cyber-attacks. Also, IoT is a broadened environment, and its integration method is incompatible with standard security solutions (Li *et al.*, 2021; Heidari and Jabraeil Jamali, 2023). Several elements related to IoT security, such as data authorization, privacy and accessibility oversight, have undergone improvements. These safety features are developed together with users and IoT, yet they still have security problems (Rahman *et al.*, 2020; Ngo *et al.*, 2022). As a result, it is critical to offer a unique component for assuring IoT network security. IDS is an example of a notion that is presently in usage in wireless networks. Improving network IDS characteristics supports IoT in protecting the network from attacks and other threats (Lv *et al.*, 2021).By analyzing traffic on the network and evaluating patterns of activity, IDS have been developed to recognize and respond to hacking attempts in contemporaneously (Saba *et al.*, 2022). They perform an essential part in ensuring computer system reliability and safety by identifying different types of attacks such as port scanning, DoS and malware infiltration (Chatterjee and Hanawal, 2022).

Related Works

Traditional IDS employ predetermined rules and characteristics for picking up recognized patterns of attack, yet they frequently fail to identify fresh or complex attacks. This is because they rely on predefined rules, signature-based detection and static feature sets

that are effective only against known threats (Kumar *et al.*, 2021; Alosaimi and Almutairi, 2023). To address the issues posed by cyber-attacks, researchers have deployed a variety of databases and approaches to strengthen network safety and cloud computing. The authors show training for service related IDS in incorporated industrial IoT in (Liang *et al.*, 2022; Al-Hadhrami and Hussain, 2020). When challenged with an unbalanced dataset, the proposed approach reliably identifies novel attack types. However, if the quantity of data is insufficient, categorization cannot be performed. Adversarial methods on detection of network intrusions in IoT networks have been put forward in (Qiu *et al.*, 2020; Vitorino *et al.*, 2023). This strategy makes use of the data extraction approach, allowing for an effective attack regardless of whether DL models are black boxes to adversaries. However, adversarial learning methods often rely on perturbations and approximations of the decision boundaries of deep learning models. Hence, it may retrieve traffic data for typical network flows not for malicious ones.

Protocol based depth detection of intrusions in (Zeeshan, 2022) face difficulties of public data-sets, such as mismatch characteristics and over-fitting, are handled by choosing a comparable amount of packets from every category. However, the larger forms of attack are not addressed for the overwhelming majority of risks in IoT. In (Almotiri, 2022), to develop IDS for the IoT an enhanced BP neural network design is presented. BP neural networks rely on gradient-based learning, which struggle with feature extraction for rare and low-frequency attack instances like R2L and U2R. Additionally, the backpropagation algorithm is susceptible to local minima and slow convergence, leading to suboptimal feature learning for these sophisticated intrusion types. A CNN-Based Approach for Intelligent Internet of Vehicles has been proposed in (Alladi *et al.*, 2021; Nie *et al.*, 2020). Training error and its convergence significantly enhance in this situation. Nevertheless, obtaining IDS with a lower false alarm rate and usage is still quite challenging. Additionally, DCNNs, while powerful in feature extraction, overfit to specific attack types present in the training data, reducing their ability to generalize well to unseen threats. The presence of imbalanced datasets, where normal traffic significantly outweighs intrusion instances, further complicates the model's learning process, leading to either excessive false positives or false negatives. In (El-Ghamry *et al.*, 2023; Bahaa *et al.*, 2022), the authors offer an affordable detection system for in IoT. In this case, the proposed CNN performed sufficiently with regard to precision in classification and identification time. The A-IDS, on the other hand, performs quite poorly.

Existing IDS for IoT networks face several challenges, including high false alarm rates, poor generalization to unseen threats, inefficiencies in handling imbalanced datasets and inability to detect sophisticated attacks. Traditional IDS methods struggle with novel and evolving cyber threats. While some advanced approaches enhance detection capabilities, they often fail to effectively extract malicious traffic patterns or suffer from overfitting, limiting their real-world applicability. Also existing networks face issues with convergence and local minima, reducing their effectiveness in detecting rare attack types. Furthermore, public datasets used for training often have mismatched characteristics, leading to overfitting and poor generalization. The proposed work addresses these limitations by introducing an optimized Deep Learning-based IDS utilizing hybrid algorithm. This hybrid optimization approach fine-tunes hyperparameters for faster convergence, reducing the need for excessive training data while improving accuracy and detection rates across multiple attack types. Additionally, robust preprocessing improves the ability of learning crucial patterns, reducing false alarms and improving generalization to emerging threats. Hence, the primary factors of the proposed work are described as below:

- To propose a unique optimized DL based IDS for the detection of attacks thereby proving enhanced security in IoT based networks.
- To propose an efficient DCNN framework which utilizes neural network capable of detecting unauthorized access attempts and identifying potential malware attacks in an automated and efficient manner.
- To minimize the training loss of neural networks by using hybrid Grey Wolf-Salp Swarm optimization which in turn selects improved hyperparameters for neural network and leads to faster convergence thereby reducing the amount of training data required. This hybrid algorithm approach monitors the progress of the optimization and automatically adjusts the learning rate or other hyperparameters to optimize convergence.

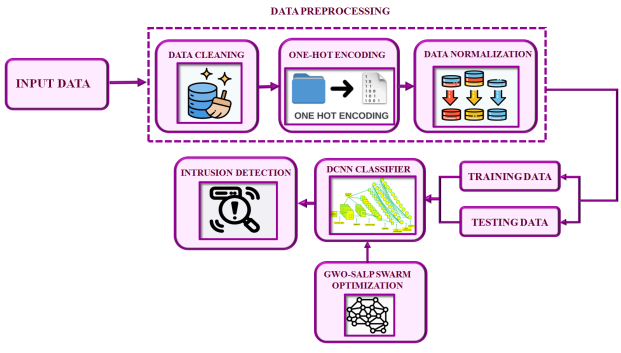

Fig. 1: IoT based IDS using GW-ISSA optimized DCNN

Proposed System Description

The explosive rise of IoT has aroused the attention of fraudsters precisely exactly as it done earlier. The increasing amount of cyber-attacks on IoT devices and

gateway media for communication backs up this assertion. Attacks against IoT, if undiscovered for a long period of time, cause severe service disruption and financial loss. It also raises the possibility of detecting the theft. Currently, detecting attacks on IoT devices is crucial for maintaining reliability, security and profitability of IoT enabled services. Figure 1 depicts a novel DL based IDS for IoT devices explained in the research.

The data preprocessing module assures that the input data is cleaned, encoded and normalized for training the DCNN classifier. The data cleaning stage is used to remove incorrect, erroneously formatted or unfinished information from dataset. Following this, the categorical information is converted into a format using a one-hot encoding phase to improve prediction accuracy.The process of using data normalization is to arrange data. It comprises generating tables and determining connections related with standards intended to safeguard data and improve database flexibility by removing inconsistent dependencies and superfluous information. The numerical values are scaled to a standardized range, improving the training efficiency of the DCNN model. Finally, the proposed DCNN classifier efficiently detects the security threats in IoT network. It automatically extracts relevant features from the data and classifies whether a given input corresponds to an intrusion or normal activity. The Hybrid GW-ISSA approach in the DCNN model is entrusted with adjusting the model parameters over the entire network. This algorithm fine-tunes the DCNN's hyperparameters, ensuring faster convergence and improved accuracy.

Data Preprocessing

The initial data collection ought to be handled prior analyzing the structure of network model. Data processing is separated into three stages: cleaning, one-hot coding, and data normalization.

Data Cleaning

It includes correcting or eradicating incorrect data from a data file, which includes the elimination of missing and invalid values in data as well as data rationality detection. It is the process of replacing, modifying, and deleting dirty data. The primary data cleaning method is depicted in Figure 2.

Step 1: Identify the data which is missing. Sometimes, faults or mistakes by humans can happen while gathering the data, which leads to the value of any of the items in the data remaining blank. In this example, the Excel tool finds the incorrect number, and location is achieved by altering a location case to a null value.

Step 2: Considering the pattern sample obtained in the present research includes a small number of missing values, the data included in the relevant section is incomplete, so row deleting is utilized for dealing with the missing values.

Step 3: After removing values which are missing, there are some visible flaws in the data. In this simulated data set, there are numerous data with contents errors, particularly in excess data with "Infinite" and "NaN". The data in these two columns, nevertheless, is supposed to speed in numerical form, which is a clear content error, so the entire row of data responsible for the error will be erased.

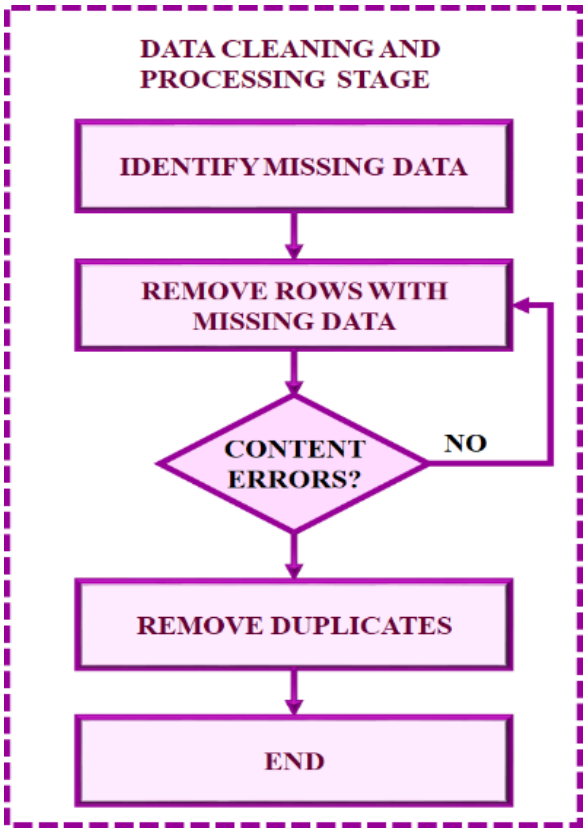

Fig. 2: Data cleaning in IoT

Data cleaning encompasses procedures that involve eliminating duplicate data, as well as detecting content errors and missing values. It has an effect on performance of model, regardless it's caused by excessive data repetition. There is no logically erroneous data detected after searching, and data duplication is inadequate, consequently no treatment measures are carried out.

One-hot Coding

There are numerous discrete data among the data that the IoT has gathered. Since its data is in string design, it is regarded as digitizing each label in string format. The aforementioned information is unable to be employed by the framework without first being translated from labels to numbers. The solution to this issue is One-Hot Encoding and this process entails encoding in states. When a specific state of the result is present, a state's

corresponding number is one and the other digits are zero. It is also clear that each feature is going to turn m binary features after one-hot encoding if it possesses m potential values. Additionally, just one of these characteristics is able to be active at once, and they are all mutually exclusive. As a result, there is going to be as much data expanding the test data set's many labels into sparse label variables with appropriate dimensions.

Data Normalization

It denotes scaling data proportionately to ensure that each number falls inside the intended range. Normalizing data has several significant benefits including speed up the convergence and increasing precision. Statistics show that the experiment utilized in this paper's data has an unreliable sample size. The standard deviation isn't utilized for the purpose to more accurately imitate the data condition of actual intrusion scene. The maximum value also has to be revised as new data is added because it fluctuates. Therefore, to achieve a normalization of the data's standard deviation, this present research uses the standard deviation normalization approach.

Equation (1), where x_{norm} is the normalised data set, illustrates the way to calculate the normalisation of standard deviation. The data exhibit a Gaussian distribution once the standard deviation has been normalized, with a variance of 1 and a mean of 0.

$$x_{norm} = (x - \mu) / \sigma \tag{1}$$

Here, x is sample data set of IoT, μ is mean value, and σ is the standard deviation. When standard deviation normalization is applied, a dataset typically approximates a Gaussian distribution. Anyway, depending on the characteristics of data, the effectiveness of normalization gets varied.

On the whole, data cleaning assures that the dataset is free from missing values and inconsistencies. This facilitates the model to learn from high-quality inputs enabling improved generalization and reduced overfitting. With one-hot encoding, this model ensures the effective interpretation and utilization of discrete features. The learning process is enhanced by the transformation of categorical data into numerical data which is done by treating categorical values independently. Moreover, the normalization by standard deviation ensures that all features exhibit a variance of one and a mean of zero.

Intrusion Detection Using GW-ISSA Optimized DCNN

In this work, the DCNN framework is utilized to automatically and effectively detect unauthorized access attempts and potential malware attacks. An optimization strategy combining GW-ISSA is used to minimize the neural network training loss. This hybrid approach reduces the quantity of training data needed and intensify convergence by optimizing hyperparameters, which is explained in the detailed manner in following section.

Deep Convolutional Neural Network

A common supervised training technique for classification tasks is the DCNN classifier. An input information set is categorized using the approach based on specific requirements. Figure 3 depicts the proposed DCNN system topology, which includes an input layer with a particular feature. The algorithm begins with establishing the weight variables for every layer. The values of weights are thereafter iteratively modified across a number of epochs. Each phase begins with a forward loop over the network, where input data is processed by a series of activation, convolutional and max-pooling layers to obtain its features. After that, the compressed feature map is sent via a fully linked layer with ReLU activation.

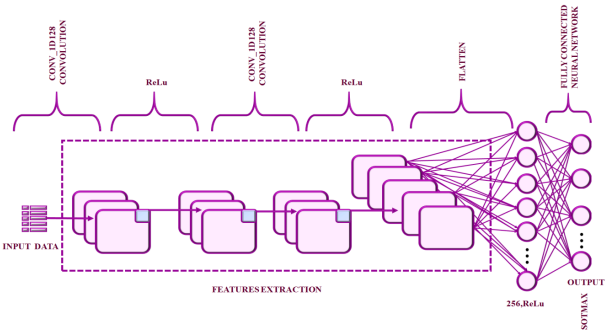

Fig. 3: DCNN structure

The network's final stage is a layer using softmax that delivers the predicted probability of each class for the source data presented. To minimize the loss, the estimated likelihoods and actual labels are used to determine the loss, and the weighting variables are revised using back propagation. This procedure is carried out for every data in the training set. The procedure evaluates an efficacy of the model on the validation data set after each epoch to track its precision and verify the model isn't overfitting. The method assesses the framework's general accuracy at the conclusion of every epoch by measuring its effectiveness on the test data set. The DCNN approach is made up of a series of totally interconnected layers, with each neuron in one layer communicating with every neuron in following one. This technique's primary purpose is to determine the optimum weight variables and biases for every single layer, enabling perfect classification of the input data.

The sample loss is calculated applying Eq. 2 for DCNN through the classified loss function of cross entropy.

$$Loss = -\sum_{i=1}^{output\_size} y_i * \log_i B \tag{2}$$

The relevant dimensional value in the framework's output is y_i, and the quantity of scalar values in the model output is output_size.

$$\text{Loss} = -\frac{1}{\text{output\_size}} \sum_{i=1}^{\text{output\_size}} \left(y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right) \quad (3)$$

Where y_i is the ith scalar variable output, y_i is the desired value, and $output_{size}$ is the number of scalar outputs.

$$softmax\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad (4)$$

The DCNN approach incorporates the softmax activation function, enabling the loss function to be determined using Eq. 3 in respect to weights as well as value in training set. The ReLU is deployed for remaining hidden layers, as illustrated in Eq. 5.

$$ReLu = \max\left(0, x\right) \quad (5)$$

$$z_n^m = \frac{1}{2}\left(z_n^m + \omega z_n^{m-1}\right) \quad (6)$$

Algorithm 1 DCNN Algorithm

Input: [X_1, X_2, ... , X_n]
Output: Classify Attacks W
Initialize weight parameters W for each layer l;
for each epoch e ∈ [1, E] **do**
  Shuffle the training set
  for each training (x_i, y_i) **do**
    Forward propagation
    Convolution layer: $z^{[l]} = W^{[l]} * a^{[l-1]} + b^{[l]}$
    Activation layer: $a^{[l]} = g[z^{[l]}]$ using ReLU
    Max-pooling layer: $a^{[l]} = \text{maxpool}(a^{[l-1]})$
    Flatten into a vector $a^{[l]}$
    Fully connected layer with ReLU
    $z^{[L+1]} = W^{[L+1]}a^{[L]} + b^{[L+1]}$
    $a^{[L+1]} = \text{softmax}(z^{[L+1]})$
    calculate Loss Eq. 3
    Update weight parameters
  end for
  Test model performance on validation set
end for
Test model performance on test set

The ReLU function provides a positive result if the value given is positive; else, it returns 0. By injecting nonlinearity, this improves the capacity of the model to acquire complicated relationships between input data and target variable. The research develops a novel hybrid GWO-ISSA strategy for optimizing the hyperparameters of the DCNN model for effectual classification of security attacks, which is detailed in the following subsection.

Hybrid GW-ISS Algorithm

The hybrid GW-ISS algorithm integrates GWO with ISSA for enhancing the training of deep learning models. GWO assures the hunting strategies and leadership hierarchy of grey wolves by mimicking the leadership hierarchy. Subsequently, ISSA enhances the movement and adaptive behavior of salps by refining the local search. With the integration of both these approaches, premature convergence is prevented and the optimal hyperparameter tuning is ensured. Thus convergence is accelerated and high detection accuracy is maintained.

GWO Algorithm

The GWO had been influenced by the natural behaviour and hunting tactics of grey wolves. The grey wolves maintain an established organizational hierarchy in groups. The group's leaders are known as alpha (β) wolves. The grey wolves in the following category are regarded as secondary. They assist the alphas and it are also referred to as beta (β) wolves. Additionally, delta (δ) wolves possess a lesser priority level.An omega (ω), which have to follow the leadership grey wolves, are the least important wolves. The following mathematical illustrations describe the GWO method:

Social Hierarchy Levels

The best solution in computational depiction is alpha wolf. A beta wolf is recognized as the second most acceptable response, while delta is the following finest solution. An omegas (ω) are the other populations that denote the most distant solutions. The GWO approach directs the hunting process via alpha, beta, and delta. Omegas ought to take identical steps as the wolves with higher priorities and follow these.

Surrounding the Victim

During hunting, the grey wolves surround a target. The ensuing approach is expressed analytically in (7) and (8).

$$\vec{D} = |\vec{C}.\vec{X}_p\left(t\right) - \vec{X}\left(t\right) \quad (7)$$

$$\vec{X}\left(t+1\right) = \vec{X}_p\left(t\right) - \vec{A} - \vec{D} \quad (8)$$

Where $\vec{A}$,$\vec{C}$-coefficient vectors. Victim's position is indicated by $\vec{X}_p$. The wolf's position is indicated by $\vec{X}$ and the present iteration is indicated by $'t'$. The calculation of the vectors $\vec{A}$ and $\vec{C}$ can be stated as (9) and (10) respectively.

$$\vec{A} = 2.\vec{a}.\vec{r_1}.\vec{a} \quad (9)$$

$$\vec{C} = 2.\vec{r_2} \quad (10)$$

Where $\vec{a}$ values fall gradually from 2 to 0 and $\vec{r_1}$ and $\vec{r_2}$ are randomized between [0, 1].

Hunting Process

The exploration agents have to recognize an ideal agent's location and alter their own. The modification of the agents' locations is depicted below, $\vec{D_\alpha}$, $\vec{C_1}.\vec{X_\alpha}$:

$$\left\{\begin{array}{ccc}\vec{D}_{\alpha} & & |\vec{C}_1.\vec{X}_{\alpha}-\vec{X}|\\ \vec{D}_{\beta} & = & |\vec{C}_2.\vec{X}_{\beta}-\vec{X}|\\ \vec{D}_{\delta} & & |\vec{C}_3.\vec{X}_{\delta}-\vec{X}|\end{array}\right\} \quad (11)$$

$$\left\{\begin{array}{l}\vec{X}_1=|\vec{X}_{\alpha}-\vec{A}_1.\left(\vec{D}_{\alpha}\right)|\\ \vec{X}_2=|\vec{X}_{\beta}-\vec{A}_2.\left(\vec{D}_{\beta}\right)|\\ \vec{X}_3=|\vec{X}_{\delta}-\vec{A}_3.\left(\vec{D}_{\delta}\right)|\end{array}\right\} \quad (12)$$

$$\vec{X}\left(t+1\right)=\frac{\vec{X}_1+\vec{X}_2+X_3}{3} \quad (13)$$

Exploring a Victim and Attacking

If $|A|<1$, the wolves attack their prey. Exploitation is the talent of attacking a victim, and exploration is the ability of locating a victim. If $|A|>1$, wolves depart.

Improved Salp Swarm Algorithm

Mirjalili *et al.* proposed SSA as one of several random based on population methods. SSA mimics the swarming process used by salps when hunting in the ocean. Salps form a swarm termed as a salp chain in heavy waters. The salp at the head of the chain is the leader in SSA algorithm, and the rest of the salps are termed followers. Salps, like other swarm-based algorithms, define its position in an s-dimensional search space, where s is variables in a specific problem. Hence, the positions of all salps are saved in a matrix with two dimensions termed z. It is believed that swarm's goal is a food source designated P in search space. The following is the mathematical framework for SSA: The leader salp is able to switch positions by solving the following equation:

$$z_n^1=\begin{cases}P_n+r_1\left(\left(u_n-l_n\right)r_2+l_n\right)r_3\geq 0\\ P_n-r_1\left(\left(u_n-l_n\right)r_2+l_n\right)r_3<0\end{cases} \quad (14)$$

Here, n denotes the dimension index in the search space.All symbols' definitions are provided in Table 1.

Table 1: Symbols definitions

Symbol	Meaning
z_n^1	leader position in nth dimension
P_n	food source position in nth dimension
u_n	upper bound of nth dimension
l_n	lower bound of nth dimension
r_1,r_2, and r_3	random variables uniformly produced in the interval
a	current iteration
A	maximum no of iterations
znm	position of mth follower Salp in nth dimension
e	time
vo	the initial speed

$$r_1=2e^{-}\left(-\frac{4a}{A}\right)^2 \quad (15)$$

The coefficient r_1 is an important value in SSA since it balances exploration and exploitation abilities.

The following formulae are used to modify the location of the followers.

$$z_n^m=\frac{1}{2}ce^2+v_0e \quad (16)$$

Where $m\geq 2;c=v_{final}/v_o$ and $v=z-z_0/e$. Considering the t is iteration, the dispute among iterations is identical to one, and taking $v_0=0$ into account, this equation is able to be written as follows:

$$z_n^m=\frac{1}{2}\left(z_n^m-z_n^{m-1}\right) \quad (17)$$

Eqs. (13) and (16) are used to modify the position of salps. The improved solution in SSA is primarily determined by the present optimal solution. To obtain the Improved Salp Swarm Method (ISSA), an inertia weight $\omega\in[0,1]$ is inserted into SSA, similar to the PSO algorithm. The newly introduced parameter in the ISSA technique increases the rate of convergence throughout the search. It also achieves a balance among exploitation and exploration abilities regarding classification tasks, allowing it to initially avoid a vast number of local solutions and, second acquire an accurate knowledge of the best choice. The following equations describe the improved algorithm:

$$z_n^1=\begin{cases}\omega P_n+r_1\left(\left(u_n-l_n\right)r_2+l_n\right)r_3\geq 0\\ \omega P_n-r_1\left(\left(u_n-l_n\right)r_2+l_n\right)r_3<0\end{cases} \quad (18)$$

$$z_n^m=\frac{1}{2}\left(z_n^m+\omega z_n^{m-1}\right) \quad (19)$$

The flowchart of GW-ISSA approach is provided in Figure 4. The attained results of proposed system reveal that the GW-ISSA tuned DCNN has maximum accuracy for predicting the security threats in IoT networks.

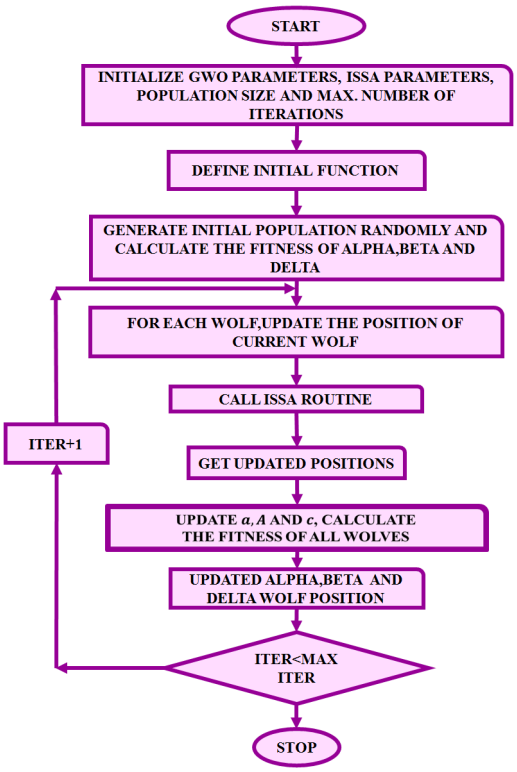

Fig. 4: Flowchart of hybrid GW-ISSA method

DOI: 10.3844/jcssp.2025.1933.1942

Materials and Methods

To develop robust and accurate IDS for IoT networks, this study utilizes a hybrid deep learning framework based on a Transfer Learning-enabled DCNN, optimized with GW-ISSA. The widely adopted NSL-KDD dataset is used as the primary benchmark. Prior to model training, a comprehensive data preprocessing pipeline is implemented, involving data cleaning**,** one-hot encoding of categorical attributes, and normalizationof numerical features to ensure consistent input scale and optimal model learning. The core of the system is the DCNN model, which integrates pre-trained deep layers via transfer learning to automatically extract meaningful, hierarchical features from network traffic data. This base network is fine-tuned for intrusion classification. To enhance detection performance, GW-ISSA is employed to optimize hyperparameters such as learning rate, batch size and filter size, ensuring rapid convergence and reduced training overhead. The entire model is implemented and evaluated using Python and multiple metrics such as accuracy, precision, recall, F1-scoreandfalse rates are computed to assess its effectiveness.

Results and Discussion

An effectual DL method based intrusion detection method is introduced in this work. The proposed hybrid GW-ISSA tuned DCNN efficaciously encounters security threats in IoT networks. The NSLKDD dataset is used in which 125973 samples with 42 features are considered. Among this 100778 samples are used for training and 25195 samples are used for testing. The proposed system is verified using Python software, the obtained results are discussed below.

A Protocol Based Intrusion Detection System (PIDS) analyses protocol in operation. In application, this system often examines the HTTPS protocol stream that connects each device to the server. In most circumstances, a PIDS is implemented at the server's front end, as seen in Figure 5.

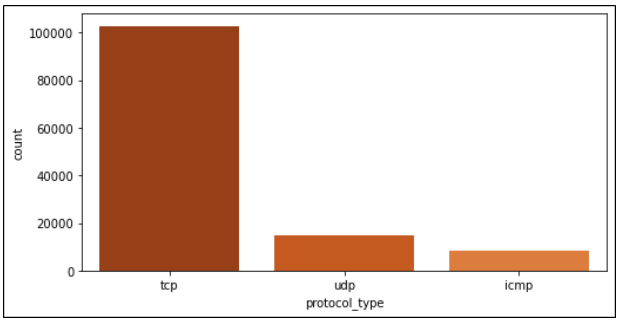

Fig. 5: IoT network Protocols

An intrusion detection system serves for monitoring network connections and devices for recognized criminal activity, unusual activity, or security policy violations. Figure 6 depicts the IoT network services representation.

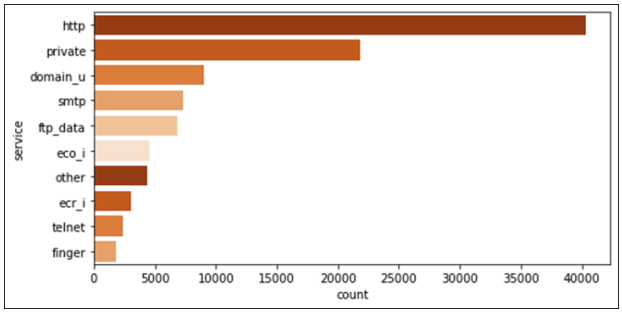

Fig. 6: IoT network services

Normal packets represent 70000 counts from the present trial as original dataset includes normal packets of data; with the DOS attack representing the largest portion with 60000 counts from the present research as actual dataset DOS packets as shown in Figure 7.

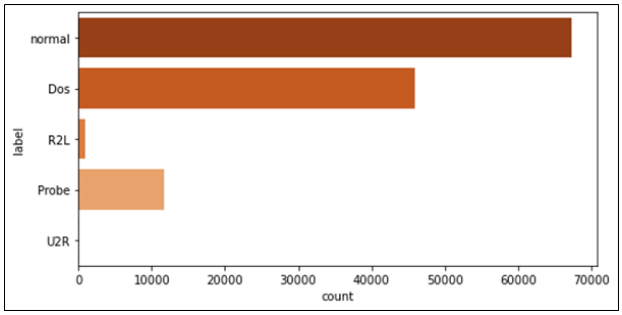

Fig. 7: IoT network intrusion attacks

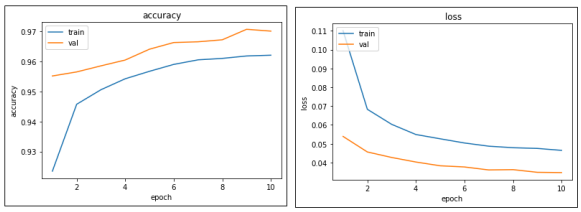

Fig. 8: Training and testing outcomes of optimized DCNN classifier

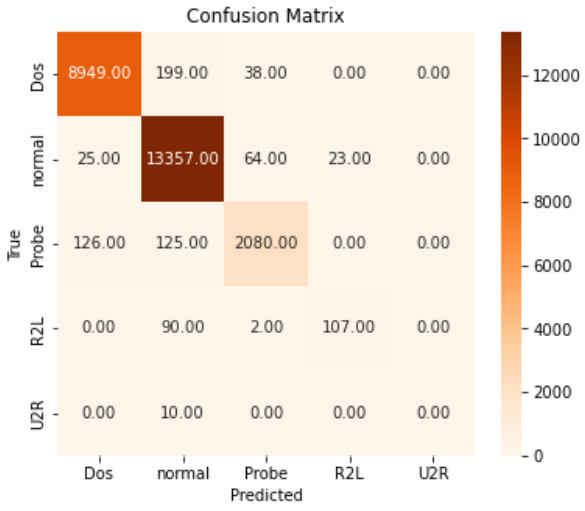

Fig. 9: Confusion Matrix

The testing & training accuracy and loss of hybrid GW-ISSA optimized DCNN is illustrated Figure 8. From the graph, the proposed classifier accomplishes high accuracy with minimized loss. Figure 9 depicts the research's findings utilizing the confusion matrix. The testing findings reveal that the majority of samples are correctly identified, especially appear on the diagonal, representing an improved categorization accuracy.

As demonstrated in Table 2, the proposed DCNNN detects security attacks such as DoS, U2R, R2L and Probe. The proposed DCNN autonomously extracts hierarchical features, making it highly effective in identifying complex intrusion patterns. Additionally, the hybrid algorithm optimizes hyperparameters dynamically resulting in faster convergence, reduced training data dependency and improved generalization. Thus Table 2 shows that in comparison to existing methods, the proposed hybrid GW-ISSA tuned DCNN-IDS strategy achieves the highest reliability for all attacks, which is illustrated in Fig. 10.

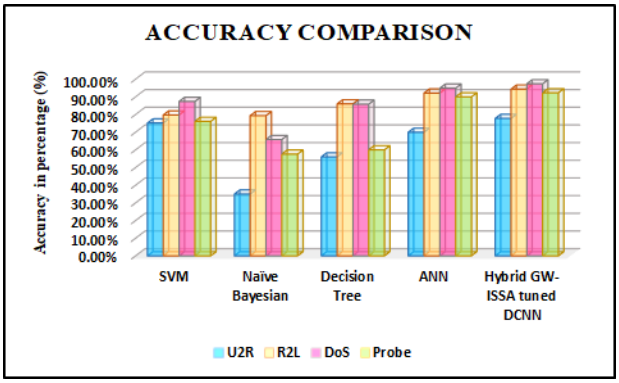

Fig. 10: Accuracy comparison

The calculated values of False Omission Rate (FOR), False Positive Rate (FPR), False Negative Rate (FNR) and False Discovery Rate (FDR) are used to prove the evaluation of the proposed model. Figure 11 displays the metrics' findings attained using GW-ISSA optimized DCNN. As illustrated in Figure 11, the proposed model outperforms the other two methods.

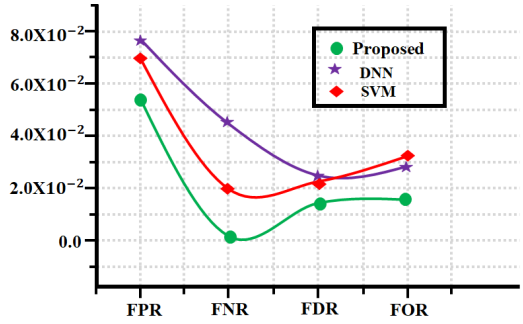

Fig. 11: Obtained values of FNR, FPR, FOR, and FDR values

The proposed GW-ISSA optimized DCNN and DLHA (Wisanwanichthan and Thammawichai, 2021), Adaptive Ensemble (Gao *et al.*, 2019), TES-IDS (Tama *et al.*, 2019) and CNN-BiLSTM (Jiang *et al.*, 2020) techniques 'performance outputs are shown in Figure 12 and Table 3 correspondingly. The proposed DCNN autonomously learns hierarchical features with greater accuracy. The hybrid algorithm optimizes hyperparameters effectively, enhancing model convergence and reducing overfitting. The combination of optimized feature learning and intelligent hyperparameter tuning makes the proposed approach highly effective in accurately detecting and classifying cyber threats in IoT networks. The comparison results demonstrate the proposed approach's superiority over competing methods in terms of all performance measures.

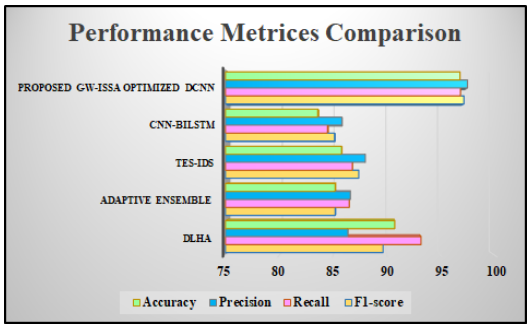

Fig. 12: Performance Metrics Comparison

Table 2: Intrusion detection comparison of accuracy

Methods	Attacks			
	U2R	R2L	DoS	Probe
SVM (Mubarakali *et al*., 2020)	75.20%	79.70%	87.40%	76.10%
Naïve Bayesian (Mehmood *et al*., 2018)	35.00%	79.35%	65.79%	57.61%
Decision Tree (Guezzaz *et al*., 2021)	56.00%	85.99%	85.84%	60.00%
ANN (Ramadevi *et al*., 2019)	69.92%	92.12%	94.92%	89.97%
Hybrid GW-ISSA tuned DCNN	77.78%	94.34%	97.34%	92.25%

Table 3: Comparison of performance metrics

	DLHA (Wisanwanichthan and Thammawichai, 2021)	Adaptive Ensemble (Gao *et al*., 2019)	TES-IDS (Tama *et al*., 2019)	CNN-BiLSTM (Jiang *et al*., 2020)	Proposed GW-ISSA Optimized DCNN
F1-score	89.65	85.20	87.39	85.14	97.16
Recall	93.17	86.50	86.80	84.49	96.84
Precision	86.38	86.60	88.00	85.82	97.48
Accuracy	90.73	85.20	85.79	83.58	96.79

Conclusion

This research seeks to present an innovative optimized DL based IDS to improve security in IoT based networks. To highlight the importance of data preparation, a vital step is conducted for ensuring that data is in a proper format and appropriate for effective learning and reliable intrusion detection. The preprocessing module cleans data, does one-hot encoding and normalizes it to generate normalized inputs for the deep learning framework. Additionally, an effective DCNN architecture, distinguished by its multi-layered neural networks, automatically trains and extracts crucial features, enabling it to detect unauthorized entry attempts and possible malware threats in a computerized reliable manner. In the end, the neural network training loss is minimized using a GW-ISSA.This approach optimizes hyperparameters, resulting in quicker convergence and less training data adopted. The achieved results demonstrate that the proposed work attains highest accuracy rate of 96.79% in IoT network intrusion detection. This level of precision ensures that malicious activity and unauthorized access within IoT environments are detected swiftly and reliably, reducing false alarms. Overall, the proposed system enhances the reliability, resilience and security of IoT networks, paving the way for safer deployment of connected devices

Author's Contributions

Chempavathy B.: Contributed to the conceptualization, data curation, methodology, and writing of the original draft of the manuscript.

Mouleeswaran S. K.: Contributed to the methodology, project administration, supervision, validation, and review and editing of the manuscript.

References

Al-Hadhrami, Y., & Hussain, F. K. (2020). Real time dataset generation framework for intrusion detection systems in IoT. *Future Generation Computer Systems*, *108*, 414–423. https://doi.org/10.1016/j.future.2020.02.051

Alladi, T., Kohli, V., Chamola, V., Yu, F. R., & Guizani, M. (2021). Artificial Intelligence (AI)-Empowered Intrusion Detection Architecture for the Internet of Vehicles. *IEEE Wireless Communications*, *28*(3), 144–149. https://doi.org/10.1109/mwc.001.2000428

Almotiri, J. (2022). DDoS Intrusion Detection Model for IoT Networks using Backpropagation Neural Network. *International Journal of Advanced Computer Science and Applications*, *13*(6). https://doi.org/10.14569/ijacsa.2022.0130682

Alosaimi, S., & Almutairi, S. M. (2023). An Intrusion Detection System Using BoT-IoT. *Applied Sciences*, *13*(9), 5427. https://doi.org/10.3390/app13095427

Bahaa, A., Sayed, A., Elfangary, L., & Fahmy, H. (2022). A novel hybrid optimization enabled robust CNN algorithm for an IoT network intrusion detection approach. *PLOS ONE*, *17*(12), e0278493. https://doi.org/10.1371/journal.pone.0278493

Chatterjee, S., & Hanawal, M. K. (2022). Federated learning for intrusion detection in IoT security: a hybrid ensemble approach. *International Journal of Internet of Things and Cyber-Assurance*, *2*(1), 62. https://doi.org/10.1504/ijitca.2022.124372

El-Ghamry, A., Darwish, A., & Hassanien, A. E. (2023). An optimized CNN-based intrusion detection system for reducing risks in smart farming. *Internet of Things*, *22*, 100709. https://doi.org/10.1016/j.iot.2023.100709

Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access*, *7*, 82512–82521. https://doi.org/10.1109/access.2019.2923640

Guezzaz, A., Benkirane, S., Azrour, M., & Khurram, S. (2021). A Reliable Network Intrusion Detection Approach Using Decision Tree with Enhanced Data Quality. *Security and Communication Networks*, *2021*, 1–8. https://doi.org/10.1155/2021/1230593

Heidari, A., & Jabraeil Jamali, M. A. (2023). Internet of Things intrusion detection systems: a comprehensive review and future directions. *Cluster Computing*, *26*(6), 3753–3780. https://doi.org/10.1007/s10586-022-03776-z

Jiang, K., Wang, W., Wang, A., & Wu, H. (2020). Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network. *IEEE Access*, *8*, 32464–32476. https://doi.org/10.1109/access.2020.2973730

Kumar, P., Gupta, G. P., & Tripathi, R. (2021). A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks. *Journal of Ambient Intelligence and Humanized Computing*, *12*(10), 9555–9572. https://doi.org/10.1007/s12652-020-02696-3

Li, B., Wu, Y., Song, J., Lu, R., Li, T., & Zhao, L. (2021). DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber–Physical Systems. *IEEE Transactions on Industrial Informatics*, *17*(8), 5615–5624. https://doi.org/10.1109/tii.2020.3023430

Liang, W., Hu, Y., Zhou, X., Pan, Y., & Wang, K. I.-K. (2022). Variational Few-Shot Learning for Microservice-Oriented Intrusion Detection in Distributed Industrial IoT. *IEEE Transactions on Industrial Informatics*, *18*(8), 5087–5095. https://doi.org/10.1109/tii.2021.3116085

Lv, Z., Qiao, L., Li, J., & Song, H. (2021). Deep-Learning-Enabled Security Issues in the Internet of Things. *IEEE Internet of Things Journal*, *8*(12), 9531–9538. https://doi.org/10.1109/jiot.2020.3007130

Mehmood, A., Mukherjee, M., Ahmed, S. H., Song, H., & Malik, K. M. (2018). NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks. *The Journal of Supercomputing*, *74*(10), 5156–5170. https://doi.org/10.1007/s11227-018-2413-7

Mohy-eddine, M., Guezzaz, A., Benkirane, S., & Azrour, M. (2023). An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. *Multimedia Tools and Applications*, *82*(15), 23615–23633. https://doi.org/10.1007/s11042-023-14795-2

Mubarakali, A., Srinivasan, K., Mukhalid, R., Jaganathan, S. C. B., & Marina, N. (2020). Security challenges in internet of things: Distributed denial of service attack detection using support vector machine-based expert systems. *Computational Intelligence*, *36*(4), 1580–1592. https://doi.org/10.1111/coin.12293

Ngo, D.-M., Lightbody, D., Temko, A., Pham-Quoc, C., Tran, N.-T., Murphy, C. C., & Popovici, E. (2022). HH-NIDS: Heterogeneous Hardware-Based Network Intrusion Detection Framework for IoT Security. *Future Internet*, *15*(1), 9. https://doi.org/10.3390/fi15010009

Nie, L., Ning, Z., Wang, X., Hu, X., Cheng, J., & Li, Y. (2020). Data-Driven Intrusion Detection for Intelligent Internet of Vehicles: A Deep Convolutional Neural Network-Based Method. *IEEE Transactions on Network Science and Engineering*, *7*(4), 2219–2230. https://doi.org/10.1109/tnse.2020.2990984

Qiu, H., Dong, T., Zhang, T., Lu, J., Memmi, G., & Qiu, M. (2020). Adversarial Attacks against Network Intrusion Detection in IoT System. *IEEE Internet of Things Journal*, *8*(13), 10327–10335. https://doi.org/10.1109/JIOT.2020.3048038

Rahman, M. A., Asyhari, A. T., Leong, L. S., Satrya, G. B., Hai Tao, M., & Zolkipli, M. F. (2020). Scalable machine learning-based intrusion detection system for IoT-enabled smart cities. *Sustainable Cities and Society*, *61*, 102324. https://doi.org/10.1016/j.scs.2020.102324

Ramadevi, R., Krishnamoorthy, N. R., Marshiana, D., Kumaran, S., & Aarthi, N. (2019). Development of Intrusion Detection System for Security Threats in Internet of Things Using Artificial Neural Network. *Journal of Computational and Theoretical Nanoscience*, *16*(8), 3242–3245. https://doi.org/10.1166/jctn.2019.8170

Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers and Electrical Engineering*, *99*, 107810. https://doi.org/10.1016/j.compeleceng.2022.107810

Tama, B. A., Comuzzi, M., & Rhee, K.-H. (2019). TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System. *IEEE Access*, *7*, 94497–94507. https://doi.org/10.1109/access.2019.2928048

Vitorino, J., Praça, I., & Maia, E. (2023). Towards adversarial realism and robust learning for IoT intrusion detection and classification. *Annals of Telecommunications*, *78*(7–8), 401–412. https://doi.org/10.1007/s12243-023-00953-y

Wisanwanichthan, T., & Thammawichai, M. (2021). A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM. *IEEE Access*, *9*, 138432–138450. https://doi.org/10.1109/access.2021.3118573

Zeeshan, M., Riaz, Q., Bilal, M. A., Shahzad, M. K., Jabeen, H., Haider, S. A., & Rahim, A. (2022). Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets. *IEEE Access*, *10*, 2269–2283. https://doi.org/10.1109/access.2021.3137201