Received: April 20, 2024. Revised: May 15, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Novel Framework based on Extra Tree Regression Classifier and Grid Search LSTM for Intrusion Detection in IoT and Cloud Environment

H. Kanakadurga Bella[1]* **S. Vasundra[2]**

[1]*Jawaharlal Nehru Technological University Anantapur, Ananthapuramu, India*
[2]*JNTUA College of Engineering, Ananthapuramu,*
Constituent College of Jawaharlal Nehru Technological University Anantapur, Ananthapuramu, India
* Corresponding author's Email: reddykanakadurga3@gmail.com

Abstract: Currently, the Cloud Computing (CC) and Internet of Things (IoT) have emerged as advanced technologies that enable new levels of connection and data processing. As the IoT ecosystem grows, it becomes more important to ensure the security and integrity of IoT devices and the data they create in cloud. The identification and prevention of intrusions in both cloud and IoT cloud systems has become a major challenge. In this research work, a new intrusion detection framework based on Extra Tree Regression Classifier and Grid Search Optimized Long ShortTerm Memory (ETR-GSO-LSTM) is used to identify and classify intrusions in IoT and Cloud environments. The input data is first gathered via the CIC-IDS-2018 and KDD-Dataset, which include a lot of information about network traffic and possible security issues. The data preprocessing tasks such as label encoding and data augmentation was performed to perform more amount of labelled data for analysis. Another crucial phase in the intrusion detection process is feature selection, and the ETR Classifier has shown to be a significant tool in determining the most relevant characteristics from the dataset. These chosen features assist in reducing dimensionality and improving the accuracy of the intrusion detection model. Finally, GSO-LSTM appears as a potential strategy for classification, which employs the capacity of LSTM networks to examine sequential data and find abnormalities in real time. The proposed ETR-GSO-LSTM achieves detection or classification accuracy of 99.95% and 99.9% on the CIC-IDS 2018 and NSL-KDD datasets, respectively. From the result analysis, it clearly shows that the proposed ETR-GSO-LSTM obtains better performance in all the metrices when compared to Enhanced Long-Short Term Memory with Recurrent Neural Network (ELSTM-RNN) and Unsupervised Technique Ensemble based IDS (UTENIDS).

Keywords: Cloud computing, Extra tree regression classifier, Internet of things, CIC-IDS-2018, KDD-Dataset.

1. Introduction

Internet of Things (IoT) and Cloud Computing have emerged as key technologies in today's modern world [1]. IoT is a wide network of interconnected devices, sensors, and everyday objects, all equipped with embedded technology that allows them to communicate and share data over the internet. These devices can range from smart wearable fitness trackers [2] to industrial sensors and autonomous vehicles. Concurrently, Cloud computing provides a scalable and flexible platform for storing and processing the immense amount of data generated by IoT devices. Together, they enable a wide array of applications across industries, such as smart healthcare, smart agriculture, manufacturing, transportation, etc. Moreover, these networks have the potential to drive efficiency, improve decision-making [3], and enhance the overall quality of life. However, this technological evolution also brings forth critical concerns such as security and privacy [4]. Nowadays, a vast amount of confidential information is being transmitted and stored in the cloud. Security issues and cyber threats such as data breaches, virus attacks, and unauthorized access represent a major hazard to both individuals and companies. Common attack kinds include

Distributed Denial of Service (DDoS) attacks [5], [6], where a network is overwhelmed with traffic to disrupt services, and data breaches through unauthorized access. Malware and ransomware attacks can compromise the integrity of IoT devices or Cloud servers.

Moreover, the sheer scale and diversity of IoT devices make them susceptible to botnet formation, where compromised devices are controlled by malicious actors for coordinated attacks. Intrusion Detection Systems (IDS) perform significantly in safeguarding IoT and Cloud networks [7]. These systems monitor network traffic, identify suspicious activities, and trigger alerts or countermeasures when potential threats are detected. Traditional IDS rely heavily on rule-based systems and signature-based detection, which may not be effective against novel or sophisticated attacks. While Machine Learning (ML) algorithms have been employed in IDS [8], they often struggle to keep pace with the evolving tactics of cyberattacks. On the other hand, Deep Learning (DL) excels in identifying intricate patterns within vast datasets, making it better suited for IDS. However, the effective implementation of DL, particularly Long Short-Term Memory (LSTM) networks [9], comes with its challenges. LSTM networks are sensitive to their hyperparameters, includes units per layer, and learning rates. Without optimization, LSTM-based IDS may produce high false-positive rates or miss critical threats. Without proper hyperparameter optimization, LSTM-based IDS struggled to achieve optimal detection accuracy. So, in this paper, a Grid Search Optimization algorithm is used for LSTM to fine tune its hyperparameters to obtain improved intrusion detection accuracy. The main contributions are as follows:

- In this research, two open-source standard datasets such as CIC-IDS 2018 and NSL-KDD datasets are employed, and the features from those datasets are used as input for intrusion detection.
- Pre-processing processes such as label encoding and data augmentation are performed to organize the input features for the selection of optimal features.
- Moreover, an Extra Tree regression classifier is proposed in this research for selecting the optimal features for effective intrusion detection or classification.
- Finally, using Accuracy, Precision, Recall and F1 measure, the developed model performance is validated.

The remaining of the paper is structured as follows: Section 2 presents the previous research done based on the development of IDS. The proposed methodology is briefly explained in section 3 whereas the experimental results are detailed in section 4. At last, the research conclusion is presented in section 5.

2. Related works

To improve the accuracy of a multiclass classification model, Lin [10] developed an IDS that integrated a Random Forest (RF) with an SMOTE resampling approach. The developed model was validated using UNSW-NB15 and CSE-CIC-IDS 2018 datasets. SMOTE was used to rebalance the original data by increasing the minority class samples, which resulted in fewer classification mistakes for minority classes during training. This method not only balanced the data, but also decreased the feature set, which improved intrusion detection performance. While SMOTE with RF has been proposed for unbalanced data in intrusion detection, practical issues such as possible decreases in classification performance owing to underfitting on imbalanced data due to the lowering of the majority class samples.

Kanna and Santhi [11] proposed an effective hybrid IDS called ABC-BWO-CONV-LSTM. The two-stage strategy was used, initially, feature selection was done using the Artificial Bee Colony (ABC) method. Then, for analyzing system traffic data, a hybrid DL classifier named BWO-CONV-LSTM, was constructed inside a MapReduce model. BWO modified the hyperparameters of this Convolutional and LSTM network to obtain an optimum design. Various datasets such as NSLKDD, ISCX-IDS, UNSW-NB15, and CSE-CIC-IDS2018 were used for performance analysis of the suggested model. The findings showed that this model outperformed others, with much shorter detection and training time, due to the faster training process in CNN.

Wang and Ghaleb [12] proposed an attention-based CNN intrusion detection model. To boost efficiency, image creation methods were included in the model's processing loop. To improve feature usage, feature fields were structured based on significance analysis, and a more complete attention mechanism was introduced into the CNN to build the detection model. On a subset of the CSE-CIC-IDS2018 dataset, several comparison tests were carried out. The approach exhibited efficient sample data computing while retaining excellent classification accuracy in experiments. It should be noted that many susceptible target items, such as critical infrastructure, were often limited to storage

and computational power restrictions, more computational time, emphasizing the significance of their protection.

Donkol [13] proposed a novel IDS model that incorporated Likely Point Particle Swarm Optimization (LPPSO) and an improved LSTM for feature selection and classification. When the NSL-KDD dataset was reviewed during validation and testing, this technique efficiently differentiated attack data from normal data and demonstrated higher performance. Notably, the suggested approach produced a considerably lower False Alarm Rate (FAR) than LPBoost with a high detection rate. Overall, the ELSTM-RNN method improved efficiency and accuracy while efficiently resolving the gradient vanishing problem. Furthermore, thorough preparation of the NSL-KDD was methodically conducted through normalization as well as encoding.

Wang [14] introduced UTEN-IDS, a new IDS that relied on unsupervised approaches. UTEN-IDS was created to identify anomalies using an ensemble of autoencoders and an Isolation Forest approach, including preprocessing, feature grouping, and anomaly detection steps. To validate the efficacy of the suggested technique, two benchmark datasets such as CES-CIC-IDS 2018 as well as MQTT-IOT-IDS2020 datasets were utilised. The anomaly detection module utilizes a two-stage ensemble model, with autoencoders used in the first level for feature subset reconstruction and RMSE computation, and Isolation Forest used in the second level for classification based on RMSEs. However, particular attack types, such as DoS assaults (Hulk and SlowHTTPTest) and DDoS attacks (LOIC-UDP), were shown to be difficult to identify using the suggested technique.

A hybrid strategy was established by Mariama [15] to address the imbalance issue during Intrusion detecyion. This hybrid approach combines an undersampling and oversampling techniques named Tomek link with Synthetic Minority Over-Sampling (SMOTE) to minimize noise. In order to produce a more effective intrusion detection system, this research also employs two DL models, namely LSTM and CNN. Moreover, NSL-KDD, CICIDS2017, and CICIDS2018 benchmark datasets were used to verify the effectiveness of the proposed method. Though the SMOTE-SGM with CNN produced the higher attack detection rate, there is still room for improvement in terms of False Alarm Rate (FAR).

Yin [16] proposed a Temporal Convolutional Network (TCN) paired with a Transformer-based model for long-term time series prediction in network security. As input, multidimensional situational data was utilized, and TCN-Transformer units were used to process network security information fusion and prediction. To measure a risk level, the baseline datasets: UNWS-NB15 and CSE-CICIDS2018 were preprocessed, which decreases the subjective dependency of the model data processing outputs. Ablation research was carried out to test the influence of data source selection on prediction accuracy, and it was discovered that the DTW algorithm improved accuracy. However, model adjustment with the genuine curve might be improved, emphasizing the relevance of excellent input data for prediction accuracy.

The problems identified from the existing studies are listed as follows,

- It should be noted that many susceptible target items, such as critical infrastructure, were often limited to storage and computational complexities, emphasizing the significance of their protection.
- However, particular attack types, such as DoS assaults (Hulk and SlowHTTPTest) and DDoS attacks (LOIC-UDP), were shown to be difficult to identify using the suggested technique. Thus reduces the model’s detection performance
- However, the developed model adjustment with the genuine curve might be improved, emphasizing the relevance of excellent input data for prediction accuracy.
- While SMOTE with RF has been proposed for unbalanced data in intrusion detection, practical issues such as possible decreases in classification performance owing to underfitting on imbalanced data. Thus limits the capability in the model’s detection accuracy.

To address the above mentioned problems, a new ETR-GSO-LSTM is proposed to identify and classify intrusions in IoT and Cloud environments. The ETR method is used for feature selection process which selects a random threshold point instead of choosing the ideal threshold for splitting. This randomization is very useful when dealing with situations that have a wide range of numerical alternatives that might fluctuate dramatically. As a consequence, it often improves accuracy by providing a smoothing effect and lowering the computational complexity involved with determining the optimal features. After that, the LSTM hyperparameters are optimized using the GSO to enhance intrusion detection accuracy because of their crucial role in defining the network's performance. Intrusion detection often requires complicated sequential data patterns, which

LSTM networks excel at collecting. GSO algorithm iteratively explores alternative combinations of these hyperparameters, thoroughly examining their influence on model performance. This method guarantees that the LSTM network is fine-tuned to the unique properties of the intrusion detection dataset, resulting in higher detection accuracy.

3. Proposed methodology

In today's environment, the development of IDS is critical in protecting IoT and cloud systems from different kinds of threats, including grey hole attacks, wormhole attacks, etc. This IDS mainly functions at the network layer of IoT systems under difficult settings. Despite these issues, security remains a major concern in IoT contexts, prompting the development of a unique optimization-based IDS, as described in this paper. This study's framework includes four key phases: data acquisition using CIC-IDS 2018 and NSL-KDD datasets, data preprocessing using Label Encoding and Data Augmentation, feature selection using the Extra Tree Regression Classifier, and intrusion attack detection using the Grid Search Optimized LSTM model.

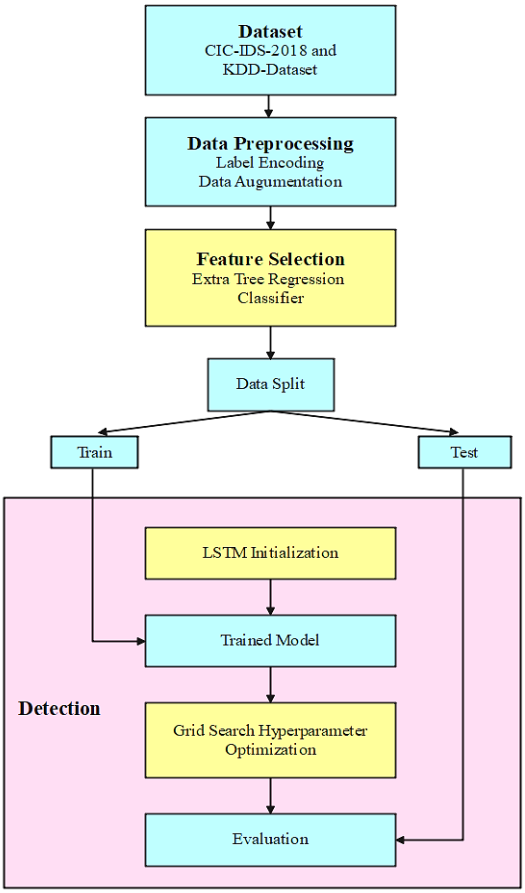

Figure. 1 Diagrammatic representation of the proposed method

The diagrammatic representation of the proposed methodology is shown in Fig. 1.

3.1 Description of the datasets

The CIC IDS 2018 and NSL KDD datasets comprise events of intrusion in a Cloud Internet of Things (IoT) network which are briefly discussed below. These intrusions apply to unauthorized and possibly malicious actions that represent a risk to the security and integrity of IoT devices and data in a cloud-based environment. The consequences of such intrusions may be significant, involving data breaches, service interruptions, and violations of privacy.

3.1.1. CIC-IDS 2018

The University of New Brunswick developed the CIC-IDS2018 dataset to train prediction models in the area of network intrusion detection. This dataset contains around 16,000,000 occurrences and is the most recent intrusion detection dataset available in the field of big data that is openly accessible to researchers. It provides in-depth coverage of many attack kinds and Web assaults. This multi-class dataset has a class imbalance, with attack (anomaly) traffic accounting for around 17% of the cases. The attacking infrastructure contains 50 computers, while the victim company is divided into five divisions, totaling 420 servers. The dataset contains both collected network traffic and system logs from every machine, as well as 80 features derived from the captured traffic utilising CICFlowMeter-V3 [17]. CIC-IDS2018 is available at https://registry.opendata.aws/cse-cic-ids2018.

3.1.2. NSL-KDD-dataset

The NSL-KDD was created to overcome the inherent issues with the KDD'99 dataset [18]. Notably, it provides a huge amount of records in training as well as test sets, avoiding the need to choose a small sample at random for trials. This benefit guarantees that the outcomes of multiple research initiatives are consistent and readily comparative. Intrusion assaults are classified into four types in the NSL-KDD: Probe, Remote to User (R2L) attacks, Denial of Service (DoS) attacks, and User to Root (U2R) assaults are all types of attacks. This dataset is accessible at https://www.unb.ca/cic/datasets/nsl.html. The features from the obtained datasets are given as input to the following process named data pre-processing which is briefly explained as follows.

3.2 Data preprocessing

An important stage in the data science process for dealing with difficulties such as missing or null values is data preprocessing. The dataset is now completely free of such concerns after eliminating the column with missing values. This assures that the data has been cleansed and is ready for analysis, increasing its dependability and usefulness. Missing values must be removed to avoid biases or mistakes in later analyses, allowing the data scientist to work with a more accurate dataset. There are two kinds of processes included in data pre-processing which are briefly explained as follows.

3.2.1. Label encoding

To transform the string labels into numerical representations for a neural network, do the following: Make a dictionary of label encodings that map each unique string label to a distinct integer value. Using this encoding dictionary, replace the original string labels in the original dataset with their numerical equivalents as shown in Table 1. This translation guarantees that the neural network can deal with numerical data since training and prediction need numerical input. To avoid misunderstanding, it is critical to ensure consistency in this encoding over the whole collection. This is known as label encoding, and it is commonly accomplished using Python modules such as scikit-learn's LabelEncoder. When labels are converted into numerical values, neural networks may learn and predict more effectively utilizing these encoded representations, improving their capacity to process and analyze input.

3.2.2. Data augmentation

Data augmentation is essential for reducing bias and improving fairness in machine learning datasets. It offers a more equal representation of distinct groups or categories by producing varied copies of existing data, minimizing inherent biases.

Table 1. Representation of Label Encoding

Dataset	Attack	Label
NSL-KDD	Normal	0
	Dos	1
	Probe	2
	U2R	3
	R2L	4
CIC-CID-2018	Normal	0
	Attack	1

This method increases equality and dependability in machine learning models by avoiding over-representation of certain groups and ensuring a more complete knowledge of the underlying patterns in the data. Data augmentation methods such as image rotation, translation, and text synthesis help greatly in the creation of more inclusive and representative datasets, resulting in fairer and more resilient IDS systems. After augmenting the encoded features, further processes such as feature selection were conducted to choose optimal features for precise intrusion classification.

3.3 Feature selection

The datasets that were pre-processed in the previous step are used as input for the current step, which is known as feature selection. The major goal of feature selection is discovering the optimal features for increasing classification accuracy. Here, the Wrapper Feature Selection method (WFS) is used to compute feature significance scores using the Extra-Tree Regression (ETR) approach [19]. It is a set of numerous Decision Trees (DTs), each of which is built using random samples and a portion of the dataset's characteristics. As a result, no one tree has access to the full dataset. The relevance of each feature in each decision tree is determined by its contribution to Gini impurity at each node. During the tree-building process, features are given significance scores, with higher impurity values indicating more feature relevance and lower values indicating lesser importance. Evaluating the value of each feature independently in high-dimensional datasets may be difficult and results in high computation time. The Extra-Tree technique, on the other hand, successfully addresses this difficulty by evaluating each feature's relevance by taking into account the complete dataset, class labels, and the calculated importance scores of all features in the dataset. Features with higher significance ratings are more likely to be chosen, whereas those with lower importance values are less likely to be included in the final feature subset. The authors' wrapper strategy is a robust approach that considerably improves classification accuracy. The Extra-Tree classifier is used in this approach to evaluate every feature and identify the final subset of K features. Notably, ETR provides feature significance values to each feature, often ranging from 0 to 1. The method for computing these feature significance ratings is as follows: The importance of a node in each DT is computed utilizing the Gini Importance (GI) metric, and the tree structure is binary, with

only two child nodes at each split and it is denoted in Eq. (1).

$$ni_j = w_j C_j - w_{Left(j)} C_{Left(j)} - w_{Right\ (j)} C_{Right\ (j)} \quad (1)$$

C_j - node j's polluting influence prediction;

ni_j - node j's significance;

Right (j) - child node's right division on node j.

w_j - weighted samples count reached node j;

Left (j) - child node's left division on node j;

Using Eq. (2), a DT computes the significance of each feature.

$$fi_i = \frac{\sum_{J:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ node} ni_k} \quad (2)$$

fi_i - significance of feature i;

ni_j - significance of node j.

By splitting every feature's significance value by the total collected feature significance values, standardization within [0-1] is accomplished and represented in Eq. (3).

$$norm\ fi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j} \quad (3)$$

When the total value of three trees equals one, except when all three trees are single-node structures with just one root node, the feature significance scores across all features are equally set to zero. In such cases, the mean value of the feature significance scores may be determined individually for each feature and it is denoted in Eq. (4).

$$Efi_i = \frac{\sum_{j \in all\ features} norm\ fi_j}{T} \quad (4)$$

Efi_i - mean of every feature j computed from all trees for ETR;

$norm\ fi_j$ - standardized feature importance for J feature.

T - total trees count. The final feature significance in ETR is normalized by splitting the significance of every feature by the sum of all feature significances in ETR and it is denoted in Eq. (5)

$$ETfi_i = \frac{Efi_i}{\sum_{k \in all\ features} Efi_k} \quad (5)$$

Then, the process begins with feature reduction applied to the input features. This reduction is guided by the feature importance scores obtained in the first part of stage 2. These feature importance scores serve as inputs to the subsequent phase. The incremental threshold values are applied systematically to select specific feature subsets. Following this feature reduction step, the reduced set of features is subjected to the GSO-LSTM to evaluate intrusion detection accuracy. Section 3.3.1 provides a detailed explanation for an in-depth understanding of the Extra-Tree algorithm.

3.3.1. Extra-tree regression classifier

ETR is an ensemble classifier builds a larger kind of binary DTs. Each tree is constructed independently of the others. ETR contains the following phases after receiving trees:

1. Selects *K* input dimensions at random. It selects an arbitrary binary splitting value *c* for each designated dimension d, denoting data points n $\in$ I with $x_{n,d}$, < c by L(left) and those with $x_{n,d}$, $\geq$ c by R(right). The primary difference between RF and this progression is that RF uses optimal splitting criterion for value *c*.
2. It computes the score for every dimension d and its s_d is denoted in Eq. (6):

$$s_d = |L| f_{score}(y_L) + |R| f_{score}(y_R) \quad (6)$$

$|L|(|R|)$ - count of data points given to the L(R) division,

$y_L(y_R)$ - y values in the L (R) division. In binary classification, using "Gini" index, F-score is computed and denoted in Eq. (7):

$$f_{score}(y) = 1 - (p_{-1}^2 + p_1^2) \quad (7)$$

For regression, f_{score} is negative of variance and is mathematically denoted in Eq. (8):

$$f_{score}(y) = -\frac{1}{n}\sum_{i=1}^{n}(y_i - mean\ (y))^2 \quad (8)$$

n - size of y and the mean (y) is mathematically denoted in (9),

$$mean\ (y) = \frac{1}{n}\sum_{i=1}^{n} \cdot (y_i) \quad (9)$$

It selects the "s_d" and saves its split value in a node. Then, it performs three steps recursively on the two resultant subtrees: Left and Right. This procedure is repeated until the minimal node size (N_{leaf}) is attained. It stores the highest standard value of classification outputs and the average value of regression tasks at a leaf node. This approach is comparable to the RF algorithm; however, it has a

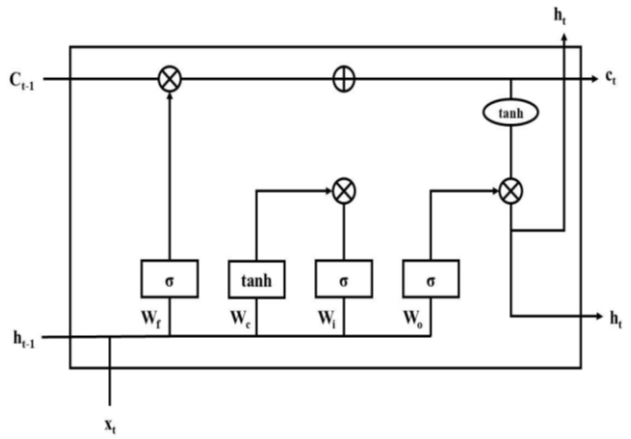

Figure. 2 Illustration of LSTM

few major differences. It selects a random threshold point instead of choosing the ideal threshold for splitting. This randomization is very useful when dealing with situations that have a wide range of numerical alternatives that might fluctuate dramatically. As a consequence, it often improves accuracy by providing a smoothing effect and lowering the computational complexity involved with determining the best cut points in existing DTs and RFs. Using the GSO-LSTM, intrusion classification is performed based on selected optimal features, which is explained briefly as follows.

3.4 Intrusion detection or classification

3.4.1. LSTM

Long Short-Term Memory (LSTM) is a kind of RNN [20] that has emerged as a powerful tool in the field of deep learning. Its unique architecture allows it to capture and remember long-range dependencies in sequential data, making it particularly well-suited for capturing intricate patterns in network traffic. In the context of intrusion detection, LSTM models can effectively analyze and classify network activities by considering the historical context of data packets and their temporal dependencies. Unlike traditional methods that rely on static rules or signatures, LSTM-based intrusion detection systems adapt dynamically to evolving attack techniques.

By learning from historical data, LSTM models can identify subtle anomalies and deviations from normal network behavior, thus enhancing the system's ability to detect both known and novel attacks. The LSTM architecture is depicted in Fig. 2.

Here,

h_t – output of LSTM,

c_t - memory cell value,

h_{t-1} - output of previous moment LSTM,

x_t – input data of LSTM at time t.

The unit computation procedure of LSTM is explained in the below points.

In Eq. (10), the patient's memory cell $\tilde{c_t}$ is calculated,

b_c - bias,

W_c - weight matrix.

$$\tilde{c_t} = \tanh\left(W_c.[h_{t-1}, x_t] + b_c\right) \quad (10)$$

In Eq. (11), the input gate i_t is computed; current input data update of memory cell state value is handled by the i_t,

b_i - bias,

W_i - weight matrix,

σ - sigmoid function

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (11)$$

In Eq. (12), the value of forget gate f_t is determined, f_t controls a memory cell state value based on prior data updates,

b_f - bias,

W_f - weight matrix

$$f_t = \sigma(W_i.[h_{t-1}, x_t] + b_f) \quad (12)$$

The current moment memory cell c_t is computed, c_{t-1} is represented as the final unit state value of LSTM, as shown in Eq. (13).

$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t} \quad (13)$$

Here, a dot product is denoted as '*'. Input as well as forget gate handles memory cell update depends on patient state value as well as final cell.

In Eq. (14), Output gate o_t value is computed, here the memory cell state output value is handled by the o_t.

b_0 - bias,

W_0- weight matrix,

$$o_t = \sigma(W_0.[h_{t-1}, x_t] + b_0) \quad (14)$$

The unit output of LSTM h_t is computed, as shown in Eq. (15).

$$h_t = o_t * \tanh(c_t) \quad (15)$$

Depending on memory cells as well as control gates, the LSTM [21] is simple to update, read, and store long-term information. The LSTM's internal parameter-sharing tool operates the output

dimensions depending on the weight matrix dimension setups.

Table 2. Hyper-parameters of LSTM

Method	Grid Search LSTM (GSO-LSTM)
Batch size	32
Optimizer	Adam
Initial Learning rate	0.01
Momentum	0.2
Decay rate	0.001
n_job	-1
CV	2

DL finds labels depend on phrases that weren't identified as well and the class probabilities are identified from the data. For two co-learning models, every data is recursively trained. In this research, the LSTM hyperparameters are optimized using the GSO to enhance intrusion detection accuracy because of their crucial role in defining the network's performance. Intrusion detection [22] often requires complicated sequential data patterns, which LSTM networks excel at collecting. The selection of hyperparameters as shown in Table 2, such as the number of LSTM batch size, learning rates, and decay rates has a substantial influence on model performance. GSO algorithm iteratively explores alternative combinations of these hyperparameters [23], thoroughly examining their influence on model performance and its functionality is explained in the following section. This method guarantees that the LSTM network is fine-tuned to the unique properties of the intrusion detection dataset, resulting in higher detection accuracy.

3.4.2. Grid search hyper parameter optimization based LSTM

The selection of suitable hyperparameters [24] for an LSTM is critical in growing its performance, involving the use of efficient search algorithms. So, the GSO technique is being offered as a useful tool in this area [25].

Let x_v and θ be the validation set as well as hyperparameters of the model M_θ, and define L (M_θ , x_v) to be the loss function of M_θ. The best solution $\hat{\theta}$ of θ is calculated using Eq. (16):

$$\hat{\theta} = \arg\arg\min_{\theta} L(M_\theta, x_v) \quad (16)$$

Here L (M_θ , x_v) is the loss function.

Set $\theta = \{\theta_1, \theta_2, \ldots \theta_m\}$, $\theta_i \in S_i, i \in \{1, 2, \ldots, m\}$,

m - count of hyperparameters,

S_i - parameter space, and set $\theta_2, \theta_3 \ldots \theta_m$ the initial value $\theta'_2, \theta'_3 \ldots \theta'_m$. The GSO is detailed below:

Step 1: Input data $\{x(t)\}_{t=1}^{M}$ number. Divide the data based on 80%:10%:10% of training set, validation set and test set. Get three sets of data.

$X_{train} = \{x(t)\}_{t=1}^{0.8M}$, $X_{validation} = \{x(t)\}_{t=0.8M+1}^{0.9M}$, and $X_{test} = \{x(t)\}_{t=0.9M+1}^{M}$.

Step 2: set i = 1.

Step 3: Replace all candidate θ_i in S_i in turn with the corresponding initial values, use X_{train} to train M_θ find the best solution $\hat{\theta}$ of θ when $\hat{\theta}_i$ satisfies:

$\hat{\theta}_i = \arg\arg\min_{\theta_i} L(M_\theta, X_{validation})$

Step 4: Set $i = i + 1$, if $i < M$, go back to step 3 otherwise go to step 5.

Step 5: $\theta = \{\theta_1, \theta_2, \ldots \theta_m\}$, , use X_{test} to train M_θ. Using the loss function L (M_θ , x_v), evaluate the model M_θ.

4. Results and discussion

In this paper, the effectiveness of the ETR-GSO-LSTM model is evaluated utilizing a simulation implemented in Python 3.7 software. Furthermore, the efficacy of the created ETR-GSO-LSTM model is assessed using a set of assessment measures that include detection accuracy, F1-measure, recall, and precision. Detection accuracy serves as a straightforward assessment parameter in the context of IoT IDS, measuring the ratio of successfully anticipated observations to a total number of observations. Recall refers to the proportion of properly identified FP to the total number of TN, while precision refers to the proportion of correctly identified FP predictions to the total number of anticipated TF. In addition, the F1-measure is calculated as the mean of accuracy and recall. These assessment criteria are clearly stated in the Eqs. (17-21).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (17)$$

$$Recall = Detection\ Rate = \frac{TP}{TP+FN} \quad (18)$$

$$Precision = \frac{TP}{TP+FP} \quad (19)$$

$$F1-measure = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (20)$$

Table 3. Performance Analysis of various feature selection methods using various performances metrics

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
Actual Features	98.00	98.50	98.00	97.45
Pearson Correlation	86.00	87.00	87.00	87.57
ETR	99.95	99.95	99.95	99.95

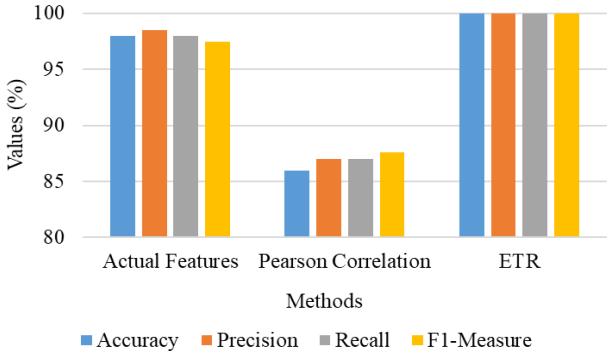

Figure. 3 Graphical depiction of results obtained from feature selection methods

$$FAR = \frac{FP}{FP + TN} \quad (21)$$

Where, $TP, TN, FP,$ and FN represent true positive values, true negative values, false positive values, and false negative values.

4.1 Performance evaluation of feature selection methods

The proposed feature selection method's (ETR) performance is evaluated using the aforementioned performance measures, and the results are shown in Table 3. This research includes evaluating performance using several feature selection strategies, such as Pearson Correlation, as well as a situation with no feature selection. Table 3 shows that the suggested ETR feature selection technique performed very well, with precision, recall, F1-measure, and classification accuracy of 99.95%. The graphical illustration of the results obtained by all the methods is depicted in Fig. 3.

4.2 Quantitative analysis of CIC-IDS 2018

The performance of the ETR-GSO-LSTM model is assessed utilizing important performance indicators such as classification accuracy, F1-measure, recall, and precision on the CIC-IDS 2018.

The model's performance is compared to that of various other classification approaches, including LSTM and GSO-LSTM. Table 4 shows that the ETR-GSO-LSTM model outperforms previous intrusion classification approaches, resulting in higher performance values. Notably, Table 5 shows that ETR-GSO-LSTM outperforms in the CIC-IDS 2018 database, with precision of 99.95%, recall of 99.95%, F1-measure of 99.95%, and classification accuracy of 99.95%. The binary classification findings of the ETR-GSO-LSTM on NSL-KDD are shown in Fig. 4.

4.3 Quantitative evaluation of the NSL-KDD database

The research assesses the efficiency of the ETR-GSO-LSTM model on the NSL-KDD utilizing important performance metrics. Table 4 compares the model's performance to that of many other classification algorithms, including SVM, Decision Tree, and RF.

When analysing Table 5, it is clear that the ETR-GSO-LSTM model excels in intrusion categorization. It delivers outstanding performance values. On the NSL-KDD, the ETR-GSO-LSTM model obtains a remarkable precision rate of 99.6%, recall rate of 99.6%, F1-measure of 99.6%, and classification accuracy of 99.6%, accurately discriminating between normal and attack classes. The binary classification findings of the ETR-GSO-LSTM on NSL-KDD are shown in Fig. 5.

Similarly, this research evaluates the efficiency of the ETR-GSO-LSTM model for multi-class classification on the NSL-KDD. Also compares the ETR-GSO-LSTM model's performance against those of several classification algorithms such as SVM, RF, and LSTM. On the NSL-KDD database, the ETR-GSO-LSTM model gets remarkable accuracy of 99.75%, 99.34%, 97.85%, and 99.76% as shown in Table 6. The model also has high precision values of 99.78%, 99.34%, 97.19%, and

Table 4. Results of Binary class classification of the proposed method on CIC-IDS 2018 database

Approaches	Performance measures (%)			
	Accuracy	Precision	Recall	F1-Measure
LSTM	97.67	96.53	97.9	96
GSO-LSTM	98.00	98.50	98.00	97.45
ETR-GSO-LSTM	99.95	99.95	99.95	99.95

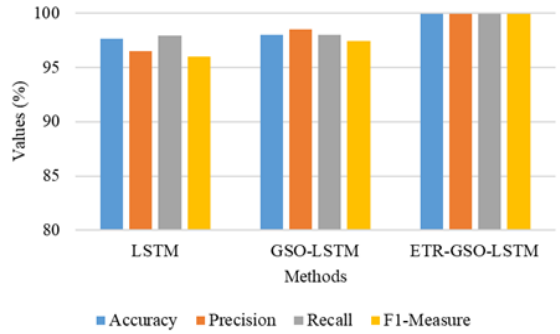

Figure. 4 Graphical illustration of classification results obtained from different classification methods

99.75%, as well as recall values of 99.66%, 99.30%, 96.64%, and 97.74% and F1-measure values of 99.72%, 99.29%, 96.69%, and 99.78% for DoS, Probe, R2L and U2R respectively. The multi-class classification findings of the ETR-GSO-LSTM model on the NSL-KDD are depicted in Fig. 6.

4.4 Comparative analysis

A comparison of the ETR-GSO-LSTM model with earlier studies is presented in Table 7 in terms of Classification Accuracy, Precision, Recall, F-measure, Detection Rate, False Alarm Rate (FAR), and Receiver Operating Characteristic (ROC). The primary parameters are batch size, initial learning rate, momentum, decay rate which determines how much the previous update influences the current update, and the how much the impact is provided at the end of each batch. Therefore, in this research parameters including Batch size (32), Learning Rate [0.01], Momentum [0.2], and Decay rate [0.001] are optimized using GSO. The best hyperparameter values are then determined by employing accuracy as a fitness function and obtained values are tabulated in Table 7.

Table 5. Results of Binary class classification of proposed method on NSL-KDD database

Model Type Class	Class	Performance measures (%)		
		Precision	Recall	F1 Measure
SVM	Normal	92.0	93.0	93.0
	Attack	91.0	91.0	91.0
Decision Tree	Normal	98.0	97.0	98.0
	Attack	95.0	96.0	95.0
RF	Normal	97.0	97.0	97.0
	Attack	99.0	98.0	99.0
ETR-GSO-LSTM	Normal	99.6	99.6	99.6
	Attack	99.6	99.6	99.6

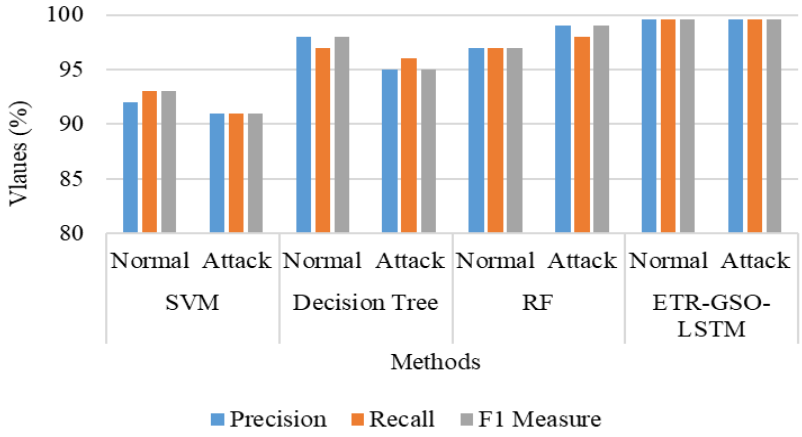

Figure. 5 Binary classification results obtained by various classification methods

In [10] researchers combined RF ensemble intrusion classification with SMOTE sampling (RF+SMOTE+C4.5) to obtain 96.53% accuracy on the CIC-IDS 2018 database. On the CIC-IDS 2018 and NSL-KDD, a combination of ABC-BWO-CONV-LSTM [11] achieved an outstanding classification accuracy of 98.25% and 98.67%, respectively. On NSL-KDD, a combination of ELSTM and RNN for IoT intrusion detection showed an astounding 99% accuracy in [13].

In addition, a UTEN-IDS [14] model for intrusion attack detection in IoT contexts was established, with an accuracy of 95.19% on the CIC-IDS 2018 database. Similarly, the SMOTETomek-

Table 6. Results of Multi-class classification of proposed method on NSL-KDD database

Model Type Class	Class	Accuracy (%)	Precision (%)	Recall (%)	F1 Measure (%)
SVM	DoS	91.61	97.44	82.88	89.56
	Probe	92.82	87.62	91.48	89.33
	R2L	83.37	89.31	64.09	67.11
	U2R	99.51	89.20	72.20	77.27
RF	DoS	99.63	99.46	99.67	99.57
	Probe	99.22	98.83	98.72	98.77
	R2L	97.55	96.79	96.24	96.51
	U2R	99.58	86.77	83.71	84.41
LSTM	DoS	99.81	99.86	99.71	99.82
	Probe	99.26	98.92	98.71	98.81
	R2L	97.77	97.26	96.55	96.79
	U2R	99.70	96.64	85.21	89.13
ETR-GSO-LSTM	DoS	99.75	99.78	99.66	99.72
	Probe	99.34	99.34	99.30	99.29
	R2L	97.85	97.19	96.64	96.69
	U2R	99.76	99.75	97.74	99.78

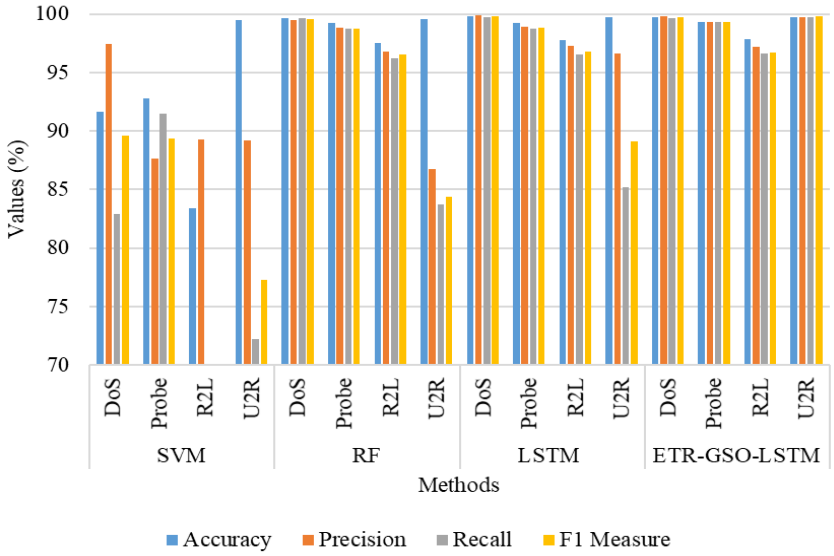

Figure. 6 Multi-class classification results obtained by various classification methods

Table 7. Comparative findings of the ETR-GSO-LSTM and the prior published studies

Models	Database	Performance Measures (%)					
		Classification Accuracy	Precision	Recall/Detection Rate	F-measure	FAR	ROC
RF+SMOTE+C4.5 [10]	CIC-IDS 2018	96.53	-	-	-	-	95.72
ABC-BWO-CONV-LSTM [11]	CIC-IDS 2018	98.25	97.48	98.67	98.18	2.52	-
	NSL-KDD	98.67	97.48	99.85	98.73	7.5	-
ELSTM-RNN [13]	CIC-IDS 2018	98.75	-	-	-	-	99.87
	NSL-KDD	99	-	-	-	1.02	-
UTEN-IDS [14]	CIC-IDS 2018	95.19	-	98.75	96.79	-	-
SMOTETomek-CNN-LSTM [15]	CIC-IDS 2018	98.17	95	-	94	-	-
	NSL-KDD	99.70	-	-	-	-	-
ETR-GSO-LSTM	CIC-IDS 2018	99.95	99.95	99.95	99.95	0.09	98.72
	NSL-KDD	99.92	99.75	99.98	99.88	0.05	99.98

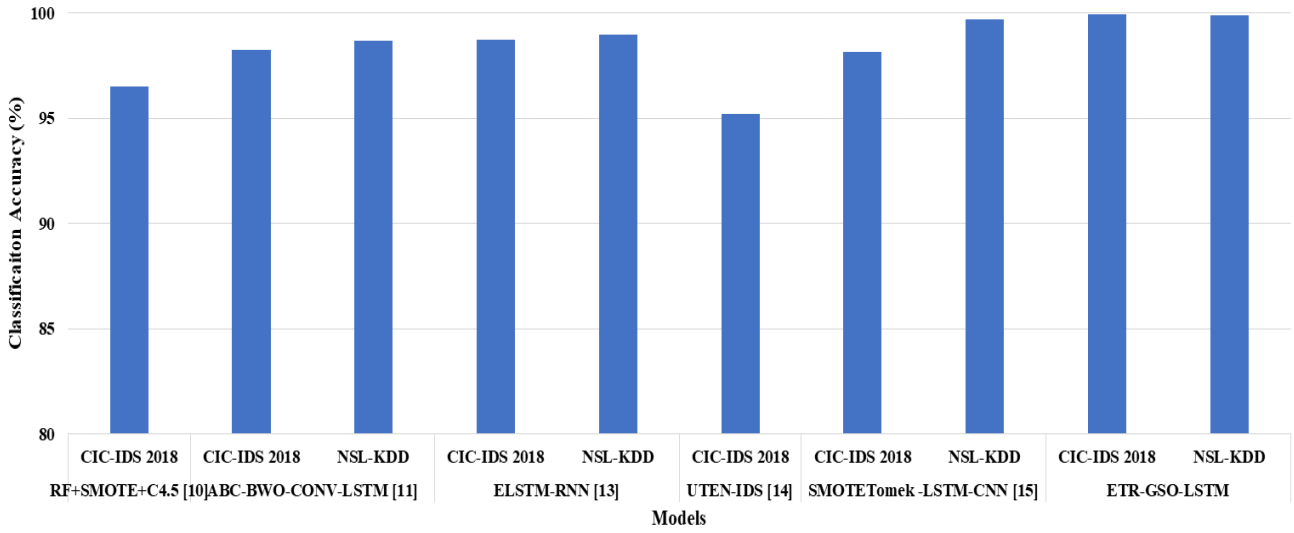

Figure. 7 Comparison analysis of ETR-GSO-LSTM with existing models in terms of Classification Accuracy

CNN-LSTM [15] model for intrusion attack detection in IoT contexts was established, with an accuracy of 98.17% and 99.70% on the CIC-IDS 2018 and NSL-KDD database respectively. On the other hand, the ETR-GSO-LSTM model stands out by obtaining a greater classification accuracy, with 99.95% and 99.9% on the CIC-IDS 2018 and NSL-KDD datasets, respectively and are graphically depicted in Fig. 7. Additionally, the ETR-GSO-LSTM outperforms the compared existing techniques [10, 11, 13, 14, 15] by obtaining higher Precision, Recall, F-measure, ROC, of 99.95%, 99.95%, 99.95%, 99.88%, 98.72%, lower FAR of 0.09% in CIC-IDS 2018 dataset and 99.92%, 99.75%, 99.98%, 98.88% and lower FAR of 0.05% in NSL-KDD. While conducting the experiment on these following settings, batch size of 32, initial learning rate of 0.01, momentum of 0.2, decay rate of 0.001. From the result analysis, it clearly shows that the proposed ETR-GSO-LSTM obtains better performance in all the metrices when compared to existing techniques.

5. Conclusion

This research has presented a novel and highly effective approach for addressing the critical challenges of intrusion detection in IoT and Cloud environments. The proposed intrusion detection framework such as ETR-GSO-LSTM employs effective techniques and datasets (CIC-IDS-2018 and KDD-Dataset) to achieve significant results. Initially, Data preprocessing, which includes label encoding and data augmentation, is crucial in preparing the input data for analysis. The Extra Tree Regression (ETR) classifier is proposed for feature selection which improves the model's performance by selecting the optimal features. Then, for effective intrusion classification, an LSTM network is proposed where the hyper parameters are optimized using the GSO algorithm. This method optimizes the hyperparameters of LSTM networks to analyze sequential data and detect abnormalities in real-time, making it particularly suitable for intrusion detection in dynamic environments. The impressive detection and classification accuracies of 99.95% and 99.9% on the CIC-IDS 2018 and NSL-KDD datasets, respectively, surpass the achievements of prior research in this field. As a direction for future work, it is essential to address practical challenges like the potential decline in classification performance.

Notation

Parameter	Definition
C_j	node j's polluting influence prediction
ni_j	node j's significance
Right (j)	child node's right division on node j
w_j	weighted samples count reached node j;
Left (j)	child node's left division on node j;
fi_i	significance of feature i;
ni_j	significance of node j.
Efi_i	mean of every feature j computed from all trees for ETR
norm fi_j	standardized feature importance for J feature.
T	total trees count.
$\|L\|(\|R\|)$	count of data points given to the left (right) division
$y_L(y_R)$	y values in the left (right) division
p_{-1}	distribution of samples with y = 1
p_1	distribution of y = 1
n	size of y and the mean (y)
s_d	dimension with the greatest score
h_t	output of LSTM cell,

c_t	memory cell value,
h_{t-1}	output of previous moment LSTM cell,
x_t	input data of LSTM cell at time t
i_t	input gate
f_t	forget gate
h_t	unit output of LSTM
b_i	bias
W_i	weight matrix
σ	sigmoid function
TP	true positive
TN	true negative
FP	false positive
FN	false negative

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

References

[1] F.K. Shaikh, S. Karim, S. Zeadally, and J. Nebhen, "Recent Trends in Internet-of-Things-Enabled Sensor Technologies for Smart Agriculture", *IEEE Internet of Things Journal*, Vol. 9, No. 23, pp. 23583-23598, 2022.

[2] E.B. Asher, N. Panda, C.T. Tran, and M. Wu, "Smart wearable device accessories may interfere with implantable cardiac devices", *HeartRhythm Case Reports*, Vol.7, No. 3, 167-169, 2021.

[3] K.B. Muhammad, T.R. Soomro, J. Butt, H. Saleem, M.A. Khan, and S. Saleem, "IoT and cloud based smart agriculture framework to improve crop yield meeting world's food needs", *IJCSNS*, Vol. 22, No. 6, p. 7, 2022.

[4] N. Torres, P. Pinto, and S.I. Lopes, "Security vulnerabilities in LPWANs—an attack vector analysis for the IoT ecosystem", *Applied Science*, Vol. 11, No. 7, p. 3176, 2021.

[5] U. Islam, A. Muhammad, R. Mansoor, M.S. Hossain, I. Ahmad, E.T. Eldin, J.A. Khan, A.U. Rehman, and M. Shafiq, "Detection of distributed denial of service (DDoS) attacks in IOT based monitoring system of banking sector using machine learning models", *Sustainability*, Vol.14, No. 14, p. 8374, 2022.

[6] S. ur Rehman, M. Khaliq, S.I. Imtiaz, A. Rasool, M. Shafiq, A.R. Javed, Z. Jalil, and A.K. Bashir, "DIDDOS: An approach for detection and identification of Distributed Denial of Service (DDoS) cyberattacks using Gated Recurrent Units (GRU)", *Future Generation Computer Systems*, Vol.118, pp. 453-466, 2021.

[7] V. Hnamte, and J. Hussain, "Dependable intrusion detection system using deep convolutional neural network: A Novel framework and performance evaluation approach", *Telematics and Informatics Reports*, Vol. 11, p. 100077, 2023.

[8] R. Shrestha, A. Omidkar, S.A. Roudi, R. Abbas, and S. Kim, "Machine-learning-enabled intrusion detection system for cellular connected UAV networks", *Electronics*, Vol.10, no. 13, p. 1549, 2021.

[9] A. Meliboev, J. Alikhanov, and W. Kim, "Performance evaluation of deep learning based network intrusion detection system across multiple balanced and imbalanced datasets", *Electronics*, Vol. 11, No. 4, p. 515, 2022.

[10] H.-C. Lin, P. Wang, C. Kuo-Ming, L. Wen-Hui, and Y. Zong-Yu, "Ensemble Learning for Threat Classification in Network Intrusion Detection on a Security Monitoring System for Renewable Energy", *Applied Science*, Vol. 11, No. 23, p. 11283, 2021.

[11] P.R. Kanna, and P. Santhi, "Hybrid intrusion detection using mapreduce based black widow optimized convolutional long short-term memory neural networks", *Expert Systems with Application*, Vol. 194, p. 116545, 2022.

[12] Z. Wang, and F.A. Ghaleb, "An Attention-Based Convolutional Neural Network for Intrusion Detection Model", *IEEE Access*, Vol. 11, pp. 43116-43127, 2023.

[13] A.A.E.-B. Donkol, A.G. Hafez, A.I. Hussein, and M.M. Mabrook, "Optimization of Intrusion Detection Using Likely Point PSO and Enhanced LSTM-RNN Hybrid Technique in Communication Networks", *IEEE Access*, Vol. 11, pp. 9469-9482, 2023.

[14] Y. Wang, G. Sun, X. Cao, and J. Yang, "An intrusion detection system for the internet of things based on the ensemble of unsupervised techniques", *Wireless Communications and Mobile Computing*, Vol. 2022, p. 8614903, 2022.

[15] M. Mbow, H. Koide, and K. Sakurai, "An intrusion detection system for imbalanced dataset based on deep learning", In: *2021 Ninth*

International Symposium on Computing and Networking (CANDAR), pp. 38-47, 2021.

[16] K. Yin, Y. Yang, C. Yao, and J. Yang, "Long-Term Prediction of Network Security Situation Through the Use of the Transformer-Based Model", *IEEE Access*, Vol. 10, pp. 56145-56157, 2022.

[17] I. Sharafaldin, A.H. Lashkari, and A.A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", In: *Proc. of 4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal, pp. 108-116, 2018.

[18] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", In: *Proc. of Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, Ottawa, ON, Canada, pp. 1-6, 2009.

[19] E. B. Wegayehu, and F. B. Muluneh, "Super ensemble based streamflow simulation using multi-source remote sensing and ground gauged rainfall data fusion", *Heliyon*, Vol. 9, No. 7, p. e17982, 2023.

[20] A.A. Awad, A.F. Ali, and T. Gaber, "An improved long short term memory network for intrusion detection", *Plos one*, Vol. 18, No. 8, p. e0284795, 2023.

[21] H. Güney, "Preprocessing Impact Analysis for Machine Learning-Based Network Intrusion Detection", *Sakarya University Journal of Computer and Information Sciences*, Vol. 6, No. 1, pp. 67-79, 2023.

[22] H.K Bella, and S. Vasundra, "A study of Security Threats and Attacks in Cloud Computing," In: *Proc. of 2022 4th International Conference on Smart Systems and Inventive Technology, Tirunelveli India*, pp. 658- 666, 2022.

[23] H.K. Bella, and V. Sanjeevulu, "Intrusion Detection Using Pareto Optimality Based Grasshopper Optimization Algorithm with Stacked Autoencoder in Cloud and IoT Networks", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 269-279, 2023, doi: 10.22266/ijies2023.0630.21.

[24] H.K. Bella, and S. Vasundra, "Intrusion Detection Using Bat Optimization Algorithm and DenseNet for IoT and Cloud Based Systems", *International Journal on Artificial Intelligence Tools*, Vol. 33, No. 2, 2024.

[25] H.K. Bella, and S. Vasundra, "Healthcare Intrusion Detection using Hybrid Correlation-based Feature Selection-Bat Optimization Algorithm with Convolutional Neural Network: A Hybrid Correlation-based Feature Selection for Intrusion Detection Systems", *International Journal of Advanced Computer Science & Applications*, Vol. 15, No. 1, 2024.