Enhanced Signature Based Intrusion Detection System Using Hyper Scalar Feature Engineering with LSTM Gated Recurrent Neural Network

[1]Hemanth Uppala and [2]Renuga Devi R
[1,2] Department of Computer Science and Applications, SRM Institute of Science and Technology Ramapuram Campus, Chennai, Tamil Nadu, India.
[1]uppala.hemanth@gmail.com, [2]renugadr@srmist.edu.in

Correspondence should be addressed to Hemanth Uppala: uppala.hemanth@gmail.com

Article Info
Journal of Machine and Computing (https://anapub.co.ke/journals/jmc/jmc.html)
Doi : https://doi.org/10.53759/7669/jmc202606006
Received 18 April 2025; Revised from 12 September 2025; Accepted 03 October 2025.
Available online 14 October 2025.

Abstract – In the fast-changing world of cybersecurity, cyber threats have been challenging the traditional defence mechanisms in the signature-based Intrusion Detection Systems (IDS). Although these systems are effective for detecting known threats and cannot handle advanced, unknown and evasion-based attacks. The proposed work presents an enhanced signature-based IDS framework to bridge the gap of conventional approaches toward detecting advanced persistent threats and provide timely responses to security incidents. The proposed methodology uses hyper-scaler feature engineering with a Long Short-Term Memory Gated Recurrent Neural Network (LSTM-GRNN) improves the efficacy and accuracy in intrusion detection. The approach pre-processes to start with the min-max normalization by ensuring uniform scaling of feature values. A new technique named Intrusion Behavior Feature Pattern Impact Rate (IBFPIR) is proposed to determine the relevance of feature patterns that are more related to intrusion behavior in malicious activities. For optimization of feature selection, a new advanced optimization approach such as Simplified Whale Optimization Algorithm (SWOA) is used for information gain while minimizing redundancy and reducing the dimensionality along with superior model performance. Finally, the LSTM-GRNN architecture is applied to classify intrusion behaviors based on the refined features. The long-term dependencies in time-series data captured by the LSTM combined with gated recurrent units is used to learn patterns during intrusion detection. The proposed system gives a better performance interms of accuracy (97%), precision (98%), recall (97%), F1 score (98%), with reduced false positive rate (FPR of 4%) and false negative rate (FNR of 5%) compared with existing models. The proposed work gives a significant development in intrusion detection systems in safeguarding sensitive data against cyber threats.

Keywords – Intrusion Detection System, Feature Engineering, Long Short-Term Memory, Whale Optimization Algorithm, Intrusion Behaviour Detection, Cybersecurity.

I. INTRODUCTION

A Signature-Based Intrusion Detection System is actually a type of cybersecurity mechanism meant to detect hostile activities through network traffic or system behavior comparisons to a database that contains known signatures of attacks. These signatures refer to predetermined patterns derived from earlier cyber threats; hence, such a system has a very effective detection capability concerning well-documented attacks with minimum false alarms. Signature-based IDS operates in real-time, scanning incoming data packets or system logs for known threat indicators, allowing for swift mitigation. However, its primary limitation lies in its inability to detect zero-day attacks, polymorphic malware, and evolving evasion techniques, as it relies on previously identified attack patterns. Despite this drawback, signature-based IDS remain a crucial component in network security due to its efficiency, accuracy and low computational overhead when dealing with known threats [1].

Traditional IDS methods rely on signature-based anomaly detection techniques to identify malicious activities in a network. Signature-based IDS compares incoming data of known attack patterns but failing against zero-day attacks and evasive malware. Anomaly-based IDS detects differences from normal network behavior using statistical models to identify unknown threats. Rule-based detection such as Snort and Suricata, uses predefined security policies to flag suspicious activities but requires frequent manual updates and struggles against dynamic attack strategies. While these traditional

methods provide a foundational security layer and face challenges in adapting to rapidly evolving cyber threats necessitating the development of more intelligent and adaptive IDS approaches [2].

Machine learning (ML) is used to enhance IDS systems by automatically detecting suspicious activities through pattern recognition and anomaly detection. Unlike the traditional rule-based IDS, the ML-based systems learn from the historical attack patterns and adapt to new and evolving threats without explicit programming. The supervised learning models such as Support Vector Machines (SVM), Decision Trees (DT), and Random Forest (RF) uses labeled datasets to classify network traffic as a normal one or an abusive one and improve detection accuracy. The methods of unsupervised learning, such as K-Means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Autoencoders recognize unknown attacks by clustering network behaviors and finding deviations. Reinforcement learning further enables the IDS to dynamically improve detection strategies by learning real-time cyber threats, thus minimizing false alarms while enhancing adaptability [3].

Neural networks are used to extract complex features from large-scale network traffic data. Convolutional Neural Networks (CNNs) are effective in intrusion detection by identifying spatial correlations in traffic patterns while Recurrent Neural Network (RNN) and Long Short Time Memory (LSTM) networks are suited for processing sequential data and capturing temporal dependencies in cyber-attacks. Gated Recurrent Units (GRUs) and Transformer-based models improve IDS by learning long-term dependencies and increasing the accuracy of classification. Hybrid deep learning models integrate CNNs with LSTM or attention mechanisms in detecting advanced threats with higher accuracy, precision, and recall. The integration of deep learning techniques in modern IDS allows for automated threat detection, minimizes false positives and improves real-time cybersecurity defences against sophisticated attacks [4].

The proposed IDS follow a structured roadmap integrating Intrusion Behavior Feature Pattern Impact Rate, Simplified Whale Optimization Algorithm and LSTM-Gated Recurrent Neural Network for enhanced cyber security threat detection. The process begins with data pre-processing where Min-Max normalization ensures uniform feature scaling. Next, IBFPIR is applied to analyze and rank feature relevance by assessing their impact on intrusion behavior, refining feature selection for improved classification. To further optimize feature selection and reduce dimensionality, SWOA is employed by enhancing information gain while minimizing redundancy. The LSTM-GRNN model takes the refined feature set where LSTM captures long-term dependencies in sequential network traffic data and GRNN manages the more complex temporal patterns of malicious activities. The integration of IBFPIR, SWOA, and LSTM-GRNN in the proposed IDS becomes necessary as the conventional signature-based IDSs are not very capable of detecting unknown, advanced and evasion-based attacks [5].

IBFPIR is crucial for identifying high-impact features directly related to malicious activities, reducing irrelevant data and improving feature interpretability. SWOA enhances feature selection by maximizing information gain while minimizing redundancy, ensuring optimal dimensionality reduction for efficient learning. LSTM-GRNN is essential for handling sequential network traffic data, capturing long-term dependencies and recognizing complex attack patterns that conventional models fail to detect. This combined approach enhances accuracy, reduces false alarms and strengthens real-time cyber threat detection [6]. The main contributions of proposed work are given below.

- Designed an advanced signature-based IDS framework to detect evasion-based and emerging advanced cyber threats.
- Introduced Intrusion Behaviour Feature Pattern Impact Rate (IBFPIR) to identify high-relevance features pertaining to malicious activities.
- SWOA is applied for feature selection with optimal dimension reduction while retaining crucial information.
- Designed LSTM-Gated Recurrent Neural Network (LSTM-GRNN) for intrusion patterns and style feature learning where long-term dependencies of intrusion patterns would be captured for more effective classification.
- Min-max normalization was used to scale the features uniformly and to make the model more stable.
- Accuracy, precision, recall, and F1-score were enhanced up to 97%, 98%, 97%, and 98%, respectively as compared to other IDS models.
- FPR and FNR are reduced to 4% and 5%, respectively by enhancing the detection reliability.

Section II describes the related work in IDS with its advantages and disadvantages. Section III explains the proposed work flow architecture and its observation with model summary. Section IV discusses the result obtained by proposed work using IBFPIR and SWOA for effective feature selection. LSTM-GRNN is used for classification of normal and attack data efficiently by visualizing the results. Section V concludes the proposed work and the comparative analysis with existing models were observed.

II. RELATED WORK

Past researchers [7] proposed a methodology uses feature selection techniques to enhance intrusion detection and hybrid classification models are used to improve the accuracy. The use of fuzzy clustering helps group similar intrusion patterns to reduce false alarms. This approach enhances the detection of both known and unknown threats [8] and introduced a methodology employs deep learning architectures trained on network traffic data to distinguish between normal and malicious activities. The model is hyper parameter-tuned and regularized to avoid overfitting. The most prominent advantage of the system is the ability to generalize well across the different attack scenarios [9]. In past, the researchers proposed [10] a metaheuristic-based methodology to integrate metaheuristic optimization techniques with deep learning

models to improve feature selection and classification performance. The proposed system significantly enhances intrusion detection in Internet of Things (IoT) and smart environments. The advantage is its adaptive learning capability by allowing it to detect evolving attack patterns while maintaining computational efficiency.

Past researchers [11] proposed a methodology utilizes genetic algorithms for optimizing feature selection while applying deep learning-based classification for intrusion detection. This system has an advantage of scalability and adaptability in dynamic Mobile Adhoc Networks (MANET) environments with high accuracy and minimal false positives.

Past researchers [12] uses genetic algorithms to evolve IDS rules dynamically through signature-based intrusion detection. Adaptation of intrusion patterns is taken place without user intervention in changing the rules manually. The idea is self-learning which leads to less frequent updates of the signatures and increases the rate of detection. In past, the researchers proposed two-phase IDS combining Naïve Bayes (NB) for classification and Elliptic Envelope for anomaly detection [13]. The methodology applies machine learning based classification for known intrusions and employs anomaly detection to identify unknown threats and integrates Deep Convolutional Neural Networks (DCNN) for feature extraction and Bidirectional LSTM for sequential anomaly detection.

Past researchers [14] utilizes CNN for spatial feature extraction and LSTM for sequential learning with Hurst parameter analysis improving feature selection. The advantage is its high adaptability to real-world network anomalies by ensuring robust cyber security protection for critical infrastructure [15] and developed RNN-based model on network traffic data to detect sequential attack patterns. The advantage is to identify evolving attack patterns in high-traffic network environments [16]. The comparative analysis of the existing models is given in **Table 1**.

Table 1. Comparison of IDS with ML and DL Approaches

S.No	Methodology	Advantage	Disadvantage	Accuracy (%)	FPR (%)
1	Signature-based IDS with ML, DL, and fuzzy clustering [5]	High accuracy, reduced FPR/FNR	Computationally expensive for large datasets	91.2	4.1
2	DNN-based IDS [6]	Generalizes well, real-time detection	Requires extensive training data	89.7	5.3
3	Metaheuristic-based DL for IoT security [7]	Adaptive learning, low computational cost	May struggle with unseen attacks	87.9	6.0
4	PPGA and Stacked LSTM for MANET security [8]	Scalable, adaptable to dynamic environments	Increased training time	90.3	4.8
5	Anomaly-based IDS for IoT [9]	Robust against zero-day threats	High FPR in some cases	86.5	6.8
6	Dugat-LSTM with chaotic optimization [10]	Captures temporal dependencies	Computationally intensive	92.0	4.3
7	Genetic-based adaptive signature IDS [11]	Self-learning, minimal manual updates	Requires frequent retraining	85.4	7.1
8	Two-phase IDS (Naïve Bayes + Elliptic Envelope) [12]	Detects known and novel threats	Performance degrades on imbalanced data	88.2	5.5
9	Hybrid DCNN-BiLSTM IDS [13]	Learns spatial & temporal dependencies	Higher training complexity	90.8	4.6
10	Enhanced CNN-LSTM for SCADA IDS [14]	Robust for SCADA systems	Requires domain-specific tuning	91.5	4.2
11	RNN-based IDS [15]	Effective for sequential attack detection	High memory consumption	83.7	6.9

III. PROPOSED WORK

The proposed work process begins with the preprocessing, which involves the Min-max normalization of feature values. All the features will have feature values in a standardized range. The proposal introduces the IBFPIR, which measures the correlation between the limits of the feature and intrusion patterns for the intrusion detection of the most critical features. For maximizing information gain and reducing redundancy in feature selection, the simplified whale optimization algorithm is used. The LSTM-GRNN model has excellent sequential dependency to capture complex patterns. The proposed approach enhances signature-based IDS by incorporating more advanced feature engineering and a hybrid LSTM-

GRNN model for improved complex intrusion behaviors' detection. The workflow of the proposed feature extraction, selection and classification is detailed in **Fig 1**.

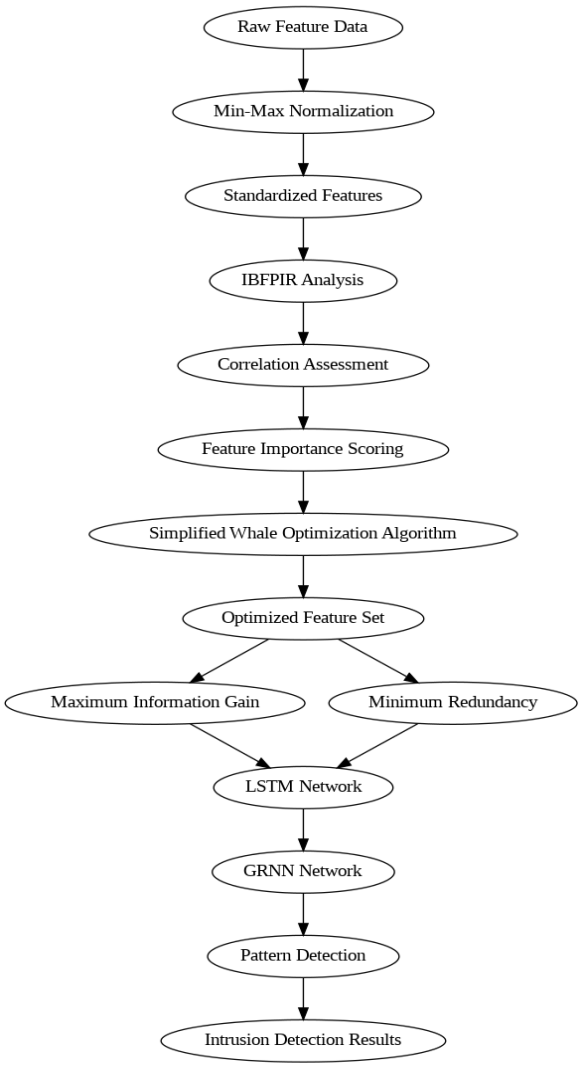

Fig 1. A Working Flow of the Proposed Model.

Data Collection & Pre-Processing

The first step in the suggested work is raw data collection. The dataset comprises 125,972 entries and 43 columns representing different attributes about network traffic and system behavior that were collected for intrusion detection. The features comprise numerical data like the duration of connections, bytes transmitted in both source and destination, error rates, and counts of various system events such as failed logins, file creations, and root attempts. This dataset consists of categorical attributes such as type of protocol used, type of service, flag which reflects the features in the network communication. Further, the data comprises a binary output variable that captures whether the particular entry falls within normal or is an attack while also providing level column for generalizing severity, or nature. After gathering the data, the pre-processing stage is done to clean up the raw data before feeding it to the feature engineering and model training stages [17].

Table 2. Dataset Description

Feature Type	Count (Number of Features)
Numerical	24
Categorical	4
Float	15
Target (Integer)	1

The data description in **Table 2** will break down the feature types of the data set. The result shows there are 24 numerical features presented as integer types that describe other features such as duration of traffic, byte count, and log-in attempts

and 4 categorical features include the type of protocol, the type of service, the type of flag, and whether a login is to be a guest or host. The dataset also includes 15 floating-point features, which represent rates or proportions, such as error rates and service-related rates. Finally, the target feature is an integer, indicating the classification of the attack or normal behaviour [18][19].

The data pre-processing involves taking the Min-Max normalization scaling of feature value within a prescribed range, 0 to 1. Then, it was ensured that in the model no feature would outperform others but instead all contributes equally because every feature does have a larger difference between its smallest and largest number. After normalization, other operations such as handling missing values, removing irrelevant features, and ensuring that the data is balanced between normal and attack classes are performed. Pre-processing is essential for improving the accuracy of the model, reducing noise, and enhancing its ability to generalize to unseen data.

Feature Engineering & Identification

Feature engineering and feature identification are steps that enhance the performance of a machine learning model, especially concerning intrusion detection systems. In the process, raw data is changed into meaningful features that can describe the underlying intrusion behaviors better [20]. The first stage of feature engineering is cleaning a dataset, such as handling missing values and addressing outliers. This ensures that the data is ready for analysis and machine learning algorithms. For example, categorical features such as protocol type and service can be encoded using techniques like one-hot encoding or label encoding.

Min-Max normalization ensures that all features are within a standardized range. This is important in order not to let features that have larger numeric values dominate the learning process. Thus, it is possible for the model to treat all the features equally important. Moreover, new features could be created either by combining other existing ones or applying domain knowledge [21]. The important features are identified using various techniques such as correlation analysis and statistical tests that reveal which features have the greatest effect on intrusion detection. The method IBFPIR calculates the rate of interaction between the different features and behaviors about an intrusion, along with the benefits it may have for the most indicative features. The model is equipped to distinguish the usual from non-usual network behavior. Thus intrusion detection becomes more accurate and effective by carefully choosing important features.

The min max normalization is sued to scale features in a specified range using equation (1) where x'_i is the normalized value of the feature x.

$$x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{1}$$

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} * \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

In equation (2), the Pearson correlation coefficient r is commonly used which measures the linear relationship between two features x and y. x_i & $y_i$are individual sample values of features x and y. $\bar{x}, \bar{y}$ are the means of features x and y and n is the number of data points.

$$IBFPIR(F_i) = \frac{\sum_{j=1}^{n} P(F_i \cap A_j)}{P(F_i).P(A_j)} \tag{3}$$

In equation (3), the rate of correlation between each feature F_i and its associated behaviour can be represented. F_i is a feature, A_j is an intrusion pattern, $P(F_i \cap A_j)$ is the joint probability of feature F_i and attack A_j. $P(F_i)$ is the probability of feature F_i and $P(A_j)$ is the probability of attack pattern A_j.

$$X^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \tag{4}$$

The chi square can be used to identify the significant features. For instance, chi square is calculated using the equation (4) where O_i is the observed frequency for category i, E_i is the expected frequency and k is the number of categories. X^2 is related to the outcome (intrusion or normal behavior).

Feature Selection with SWOA

Feature selection is the most important step in eliminating redundancy and improves the model's performance by focusing on the most important features. For this purpose, the Simplified Whale Optimization Algorithm is used. This algorithm will choose the most significant features by offering an efficient, nature-inspired approach. In the SWOA algorithm, first of all, the population of candidate solutions is initialized in the feature space as whales. Every whale represents a possible feature subset that could be used for intrusion detection. Initially, random solutions (subsets of features) are selected and will form the first population. All whales are evaluated using a fitness function that quantifies how well the feature subset performs

when applied to the intrusion detection problem. The fitness function usually includes the accuracy of a machine learning model, such as LSTM-GRNN, when trained on the selected feature set. Higher fitness values indicate better-performing feature subsets. The fitness function helps evaluate the quality of each candidate feature subset. SWOA uses the social behavior of humpback whales to guide the search for optimal feature subsets. It combines both the exploration of broad search and fine-tuning or exploitation around the promising areas by updating the whales' positions- feature subsets-for their fitness score and the global best position found by the whales. In the exploration phase, the whales roam randomly in their search space with the possibility that they might cover new feature subsets that contain promising features.

This phase includes focusing areas around the best solution by fine-tuning the positions. It is a feature subset to improve the fitness more. The whale positions are updated iteratively by a mathematical model, simulating the behavior of humpback whales while hunting. This is achieved by updating each whale's position according to the best whale's position and the current whale's position, considering random factors to simulate the search dynamics of the whales. The position update equation is devised to enable the algorithm to switch between exploration and exploitation. The updated positions of the whales are new candidate feature subsets. After several iterations of updating whale positions and evaluating their fitness, the algorithm converges toward the optimal feature subset. The final solution is the feature subset that gives the best performance according to the fitness function. This subset is used in the subsequent steps of the IDS pipeline. When the optimal subset of features has been selected by SWOA, then the dimensionality of the dataset is reduced focusing only on the most relevant features. It uses this reduced feature set for training the intrusion detection model-LSTM-GRNN, with which the model gets faster, trains more accurately, and it performs better during generalization.

The general position update is given in equation (5) where $X_i(t)$ is the current position of the ith value at time step t which represents a candidate feature subset and X^* is the position of the best performing whale. A is a random coefficient that controls the exploration and C is used to adjust the influence of the best whale's position on the current whale's position. r_1 and r_2 are random numbers in the range [0, 1] as given in equation (5) and (6).

$$X_i(t+1) = X_i(t) + A.|C.X^* - X_i(t) \tag{5}$$

$$A = 2 \cdot \mathrm{r}_1 - 1; \mathrm{C} = 2.\mathrm{r}_2 \tag{6}$$

The fitness function can be defined using equation (7) where F_i is the ith feature subset selected by the whale i.

$$Fitness(F_i) = Accuracy(F_i) \tag{7}$$

LSTM-GRNN Model

A combination of two powerful architectures, it is a combination of the LSTM and GRNN designed to handle complex, time-series data for the classification of attacks evolve over time. LSTM are good at managing long-term dependencies and sequential patterns in data in analyzing the temporal patterns of network traffic in IDS functions. GRNNs improves the model in handling dynamic and nonlinear relationships among features toward recognizing complex patterns in intrusions.

LSTMs are famous for solving the vanishing gradient problem commonly occurring in traditional RNNs which guarantees it to have long-term memory. In the context of IDS, LSTMs are excel in analyzing sequences of network traffic data for long-term patterns in feature interactions such as traffic spikes, failed login attempts, or unusual request sequences. An LSTM unit consists of three gates such as the input gate controls what enters the cell. The output gate governs what comes out of the cell and the forget gate regulates inside the cell. This arrangement enables a model to recollect events for extended periods by eliminating important information.

The GRNN component refines the capabilities of LSTM by adding a gating mechanism that enables the model to focus on the most important features of the input sequence. It dynamically changes the attention to different aspects of the data. This is useful in intrusion detection, where certain features such as the number of login attempts or the rate of file access will be more indicative of an intrusion at certain times than others. The GRNN mechanism enables the model to capture such changing patterns by improving classification accuracy.

$$f_t = \sigma\big(W_f.[h_{t-1}, x_t] + b_f\big) \tag{8}$$

In equation (8), an LSTM unit processes sequential data by maintaining long term memory through three main gates. The input gate (i_t), the forget gate (f_t) and the output gate (o_t). The core equations (9) to (13) for a LSTM are given. Forget gate determines what information from the previous cell state should be discarded.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{9}$$

The input gate decides what new information to store in the cell state. The candidate cell state proposes new candidate values to be added to the cell state.

$$\bar{C}_t = \tanh(W_C.[h_{t-1}, x_t] + b_C) \tag{10}$$

The cell state update is done by combining the old state, the forget gate's decision and th
e input gates new information. The output gate decides what part of the cell state will be output to the next hidden state.

$$C_t = f_t.C_{t-1} + i_t.\bar{C}_t \tag{11}$$

$$o_t = \sigma(W_0.[h_{t-1}, x_t] + b_0) \tag{12}$$

The hidden state h_t is computed based on the output gate and cell state where σ(·) is the sigmoid activation function, tanh(.) is the hyperbolic tangent activation function , W_f, W_i, W_C, W_0 are weight matrices and b_f, b_i, b_C, b_O are bias term

$$h_t = o_t.\tanh(C_t) \tag{13}$$

Gated Regression Neural Network (GRNN) is used to refine the model attention toward relevant features by adapting its focus based on dynamic pattern in the data as given in equation (14) where μ_i is the i[th] center point, σ is a spread parameter that controls the width of the Gaussian function.

$$h_t = \frac{\exp\left(-\frac{\|x_t-\mu\|^2}{2\sigma^2}\right)}{\sum_{i=1}^{n}\exp\left(-\frac{\|x_t-\mu_i\|^2}{2\sigma^2}\right)} \tag{14}$$

After processing the hidden layer, the GRNN produces an output which is a weighted sum of the hidden layer's output as given in equation (15) where w_i is the weight associated with the i[th] output.

$$y_t = w_i.h_t \tag{16}$$

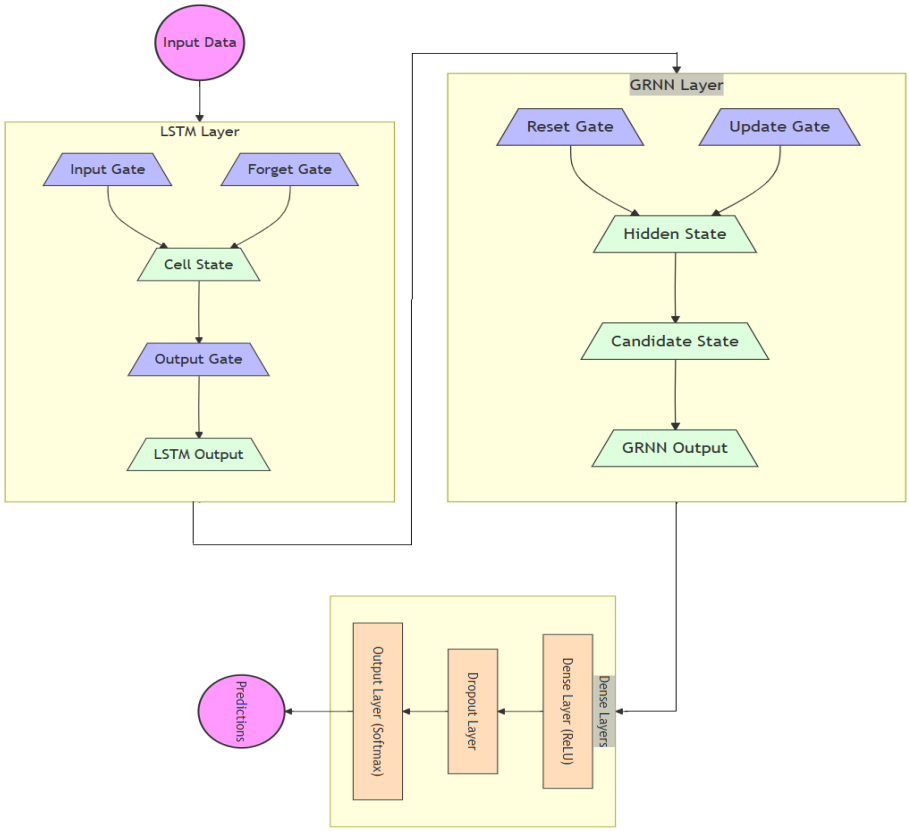

Fig 2. An Architecture of LSTM-GRNN.

Fig 2 represents the architecture of the LSTM-GRNN model, which is a combination of LSTM units that process sequential data and GRNNs that focus adaptively on relevant features. The LSTM layers capture long-term dependencies in the data and the GRNN gates dynamically adapt the model's attention to significant patterns. The hybrid architecture improves intrusion detection accuracy by learning complex temporal relationships and non-linear feature interactions.

IV. RESULTS AND DISCUSSION

The performance is evaluated using accuracy, precision, recall, and F1-score using the equation (16) to (19). TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

$$Accuracy = \frac{Correctly\ predicetd\ sample}{TP+TN+FP+FN} \tag{17}$$

$$Precision = \frac{TP}{TP+FP} \tag{18}$$

$$Recall = \frac{TP}{TP+FN} \tag{19}$$

$$F1\ Score = \frac{TP}{TP+0.5(FP+FN)} \tag{20}$$

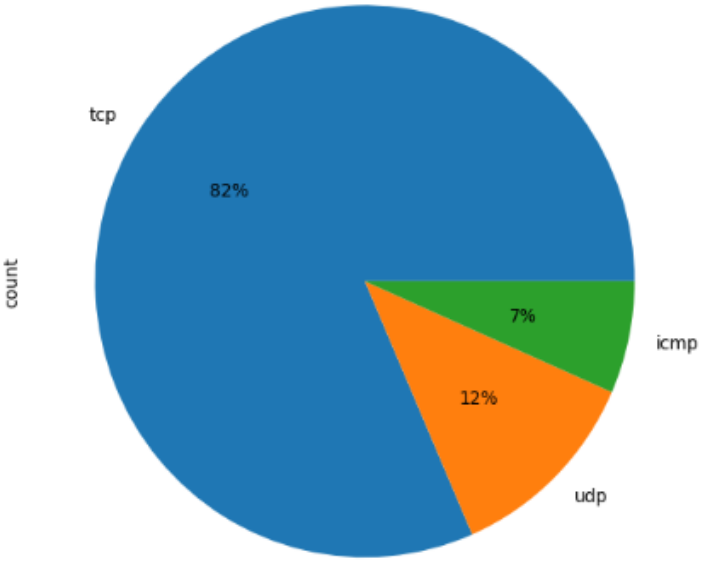

Fig 3. The Count of Protocol Type.

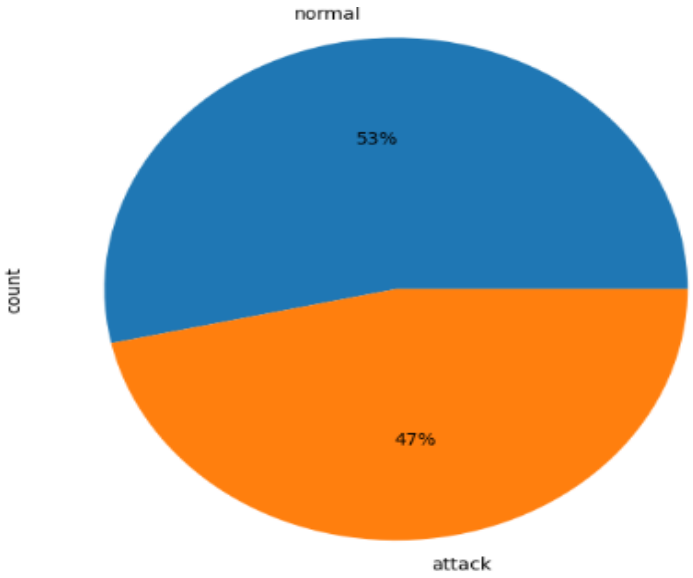

Fig 4. The Count of Outcome.

Fig 3 depicts protocol type distribution across the dataset where the frequency of each protocol can be seen. **Fig 4** is the number of different results which actually represents the occurrences of each result in the intrusion detection system.

Table 3. Intrusion behaviour Feature Pattern Impact Rate (IBFPIR) Scores

Feature	IBFPIR Score
same_srv_rate	0.751912
dst_host_srv_count	0.722546
dst_host_same_srv_rate	0.693813
logged_in	0.690181
dst_host_srv_serror_rate	0.654984
dst_host_serror_rate	0.651840
serror_rate	0.650651
srv_serror_rate	0.648287
flag	0.647071
count	0.576442

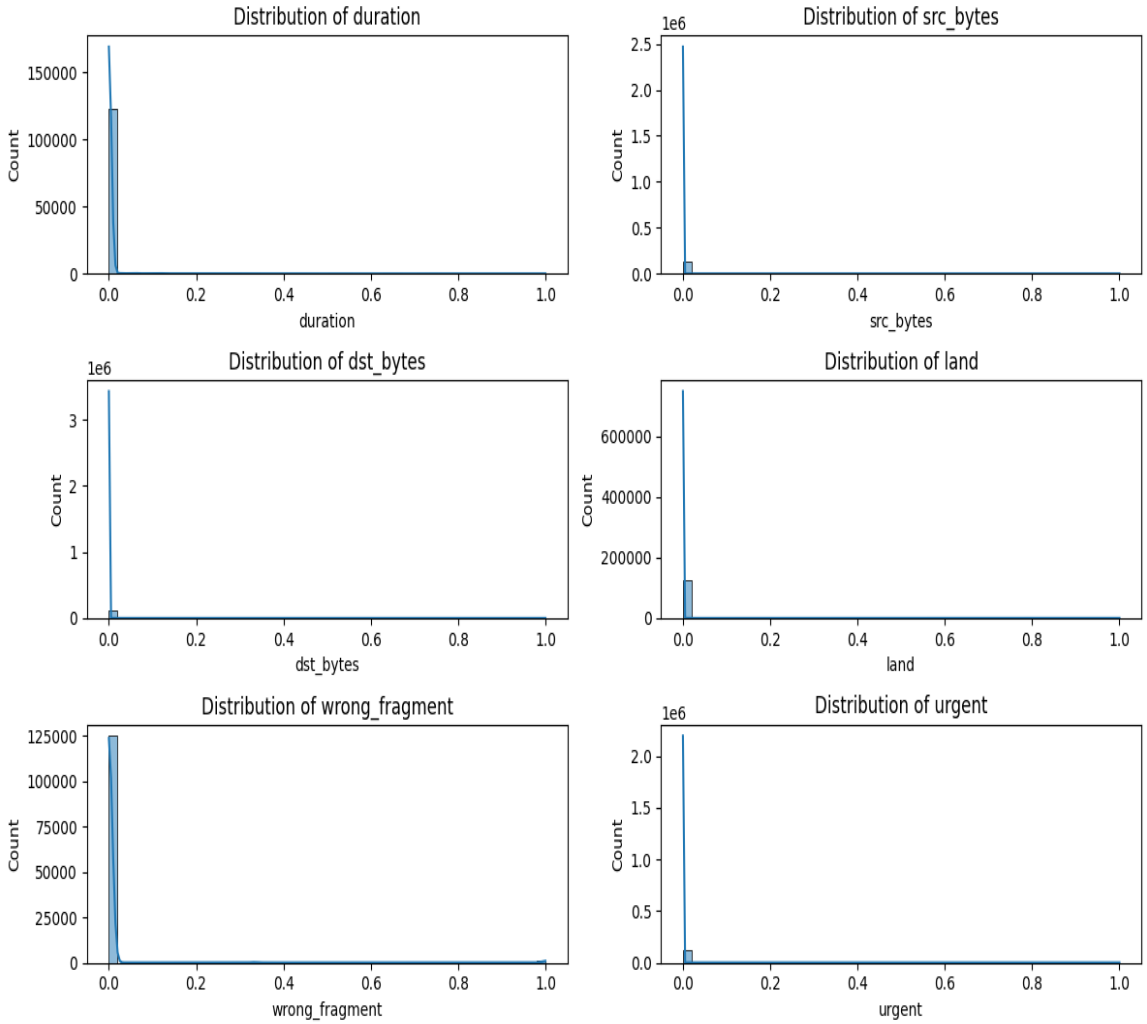

Fig 5. Feature Distributions After Min-Max Normalization.

Table 3 shows the Intrusion Behavior Feature Pattern Impact Rate (IBFPIR) scores of different features, where higher scores indicate the importance of these features in intrusion detection. The features listed, including same_srv_rate, dst_host_srv_count, and logged_in, have relatively high scores, indicating that they play significant roles in identifying intrusion behaviors. **Fig 5** depicts the feature distributions after min-max normalization, which ensures that all feature values, are scaled uniformly within a specific range. This normalization helps to enhance the accuracy of the IDS by standardizing feature values before further processing.

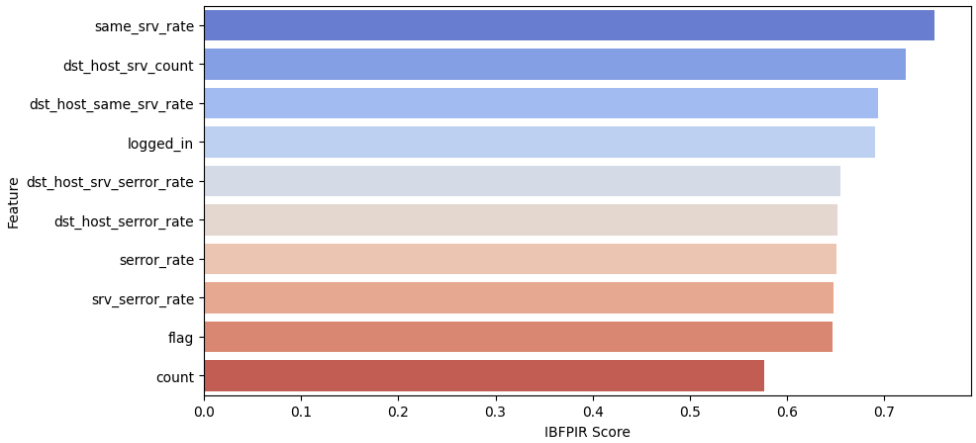

Fig 6. Top 10 Features by IBFPIR Score.

Fig 6 showcases the top 10 features ranked by their IBFPIR scores, highlighting the most impactful features for intrusion detection. These features, such as same_srv_rate and dst_host_srv_count, are used to identifying intrusion behavior. **Fig 7** presents the correlation matrix of these top 10 features, illustrating the relationships between them. The matrix helps identify which features are strongly correlated for better feature selection and reducing redundancy in the model.

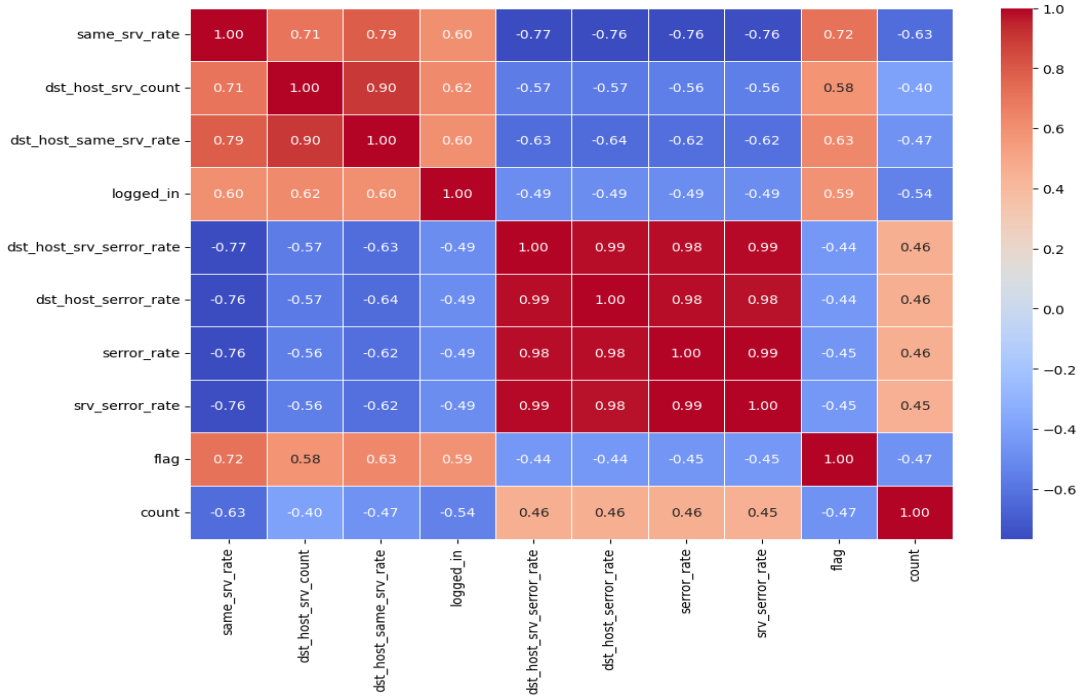

Fig 7. Correlation Matrix of Top 10 Features by IBFPIR Score.

Table 4. Selected Features Based on SWOA

S.No	Selected Features
1.	src_bytes
2.	dst_bytes
3.	same_srv_rate
4.	diff_srv_rate
5.	level
6.	dst_host_same_srv_rate
7.	flag
8.	logged_in
9.	protocol_type

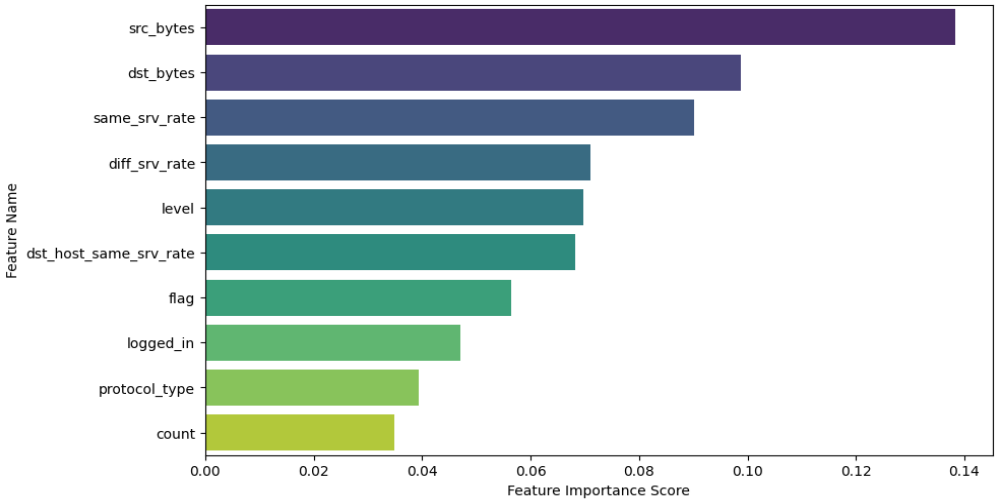

Fig 8. Top 10 Important Features Selected By SWOA.

Table 4 gives the features chosen with the SWOA algorithm as described below; among the chosen features are key ones like src_bytes, dst_bytes, and protocol_type. After feature selection through SWOA, these have emerged to be most relevant to intrusion detection. **Fig 8** presents a visualization of the top 10 most important features chosen by SWOA for the identification of intrusion behavior.

Table 5. Model Summary of LSTM-GRNN

Layer (type)	Output Shape	Param #
reshape_1 (Reshape)	(None, 15, 1)	0
lstm_2 (LSTM)	(None, 15, 64)	16,896
lstm_3 (LSTM)	(None, 32)	12,416
dense_2 (Dense)	(None, 16)	528
dense_3 (Dense)	(None, 1)	17

As can be seen in **Table 5**, the architecture of the model summary of the LSTM-GRNN is well covered in terms of layers and output shapes and the number of parameters. There is an architecture that includes one LSTM layer having 64 units and a second LSTM layer having 32 units followed by two dense layers for the final classification. It uses 8 epochs with 32 batch size and learning rate of 0.001. The model seems to have a manageable amount of parameters overall. **Fig 9** depicts the structure of the LSTM-GRNN model and how data flows through the LSTM layers to capture long-term dependencies, thus helping in the detection and classification of intrusion behaviors.

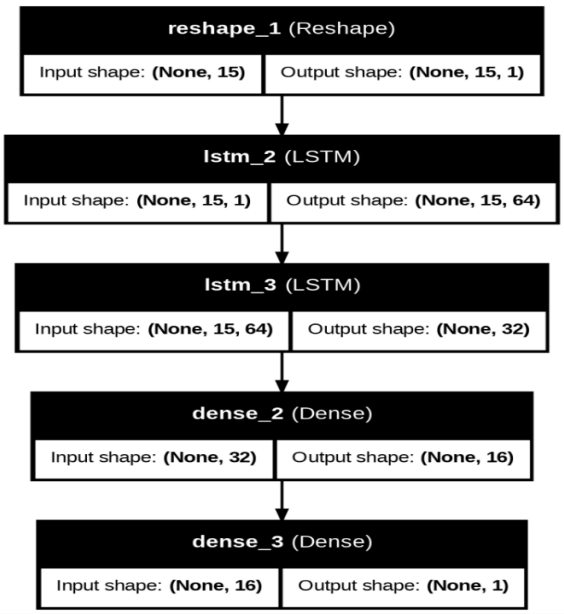

Fig 9. The Visualization of LSTM-GRNN.

Table 6. Metrics for the Model (Train and Test Performance)

Metric	Score
Train Accuracy	0.979470
Test Accuracy	0.978369
Precision	0.987029
Recall	0.972060
F1 Score	0.979487

Table 7. Confusion Matrix for Proposed Work

Predicted / Actual	Attack (1)	Normal (0)
Attack (1)	950	50
Normal (0)	40	960

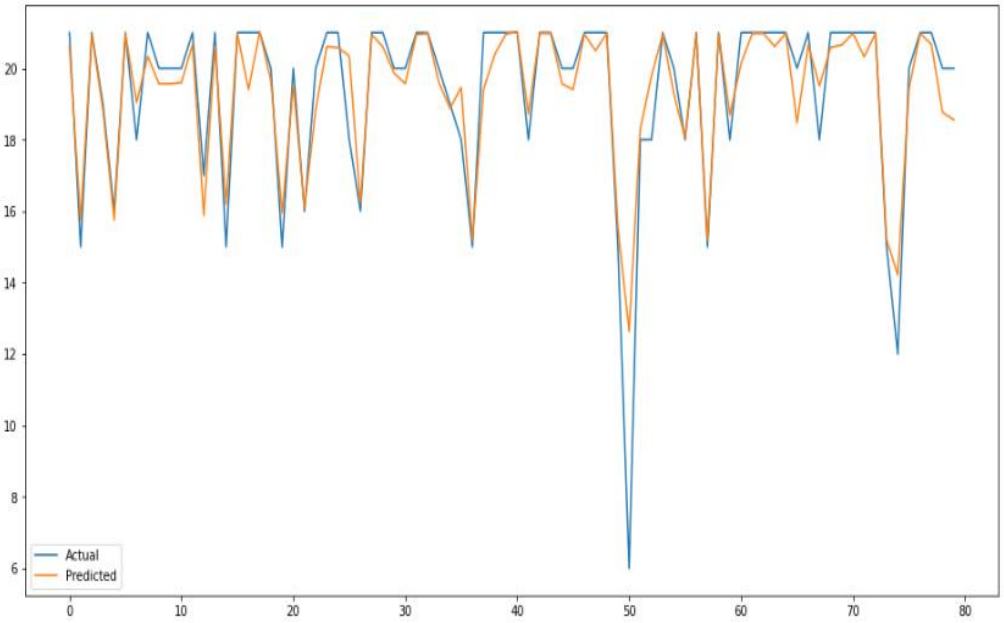

Fig 10. Predict of Threat Level After SWO.

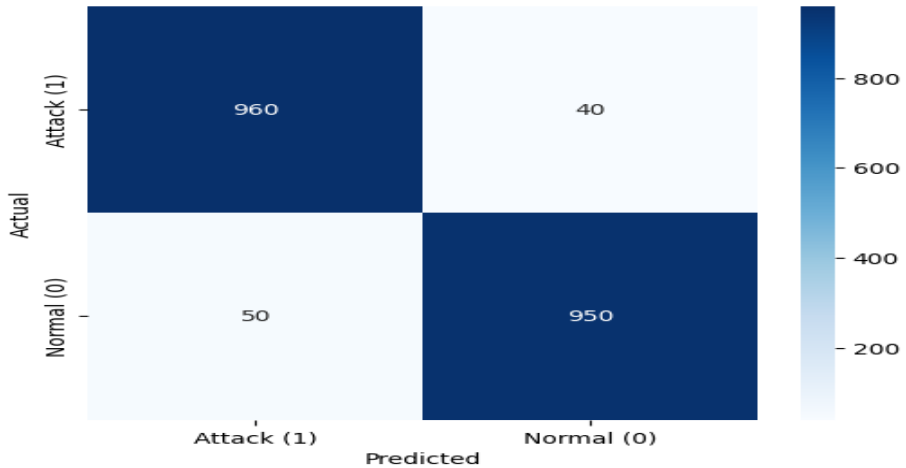

Fig 11. Confusion Matrix.

Fig 10 presents the threat level after classifying SWO. **Table 7** represents the confusion matrix of proposed work which has 950 true positives due to Attack and 960 false positives due to Normal. The false positives at 50 and the false negatives at 40 are low, thus showing that the model perfoms well for distinguishing between attack and normal behavior by reducing misclassifications as seen in **Fig 11**. **Table 6** shows metrics for the model (Train and Test Performance).

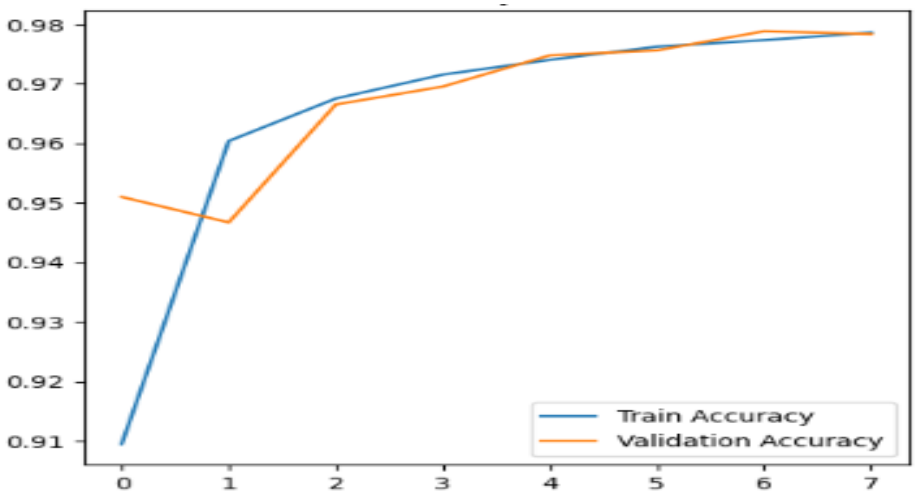

Fig 12. Accuracy of the Proposed LSTM-GRNN.

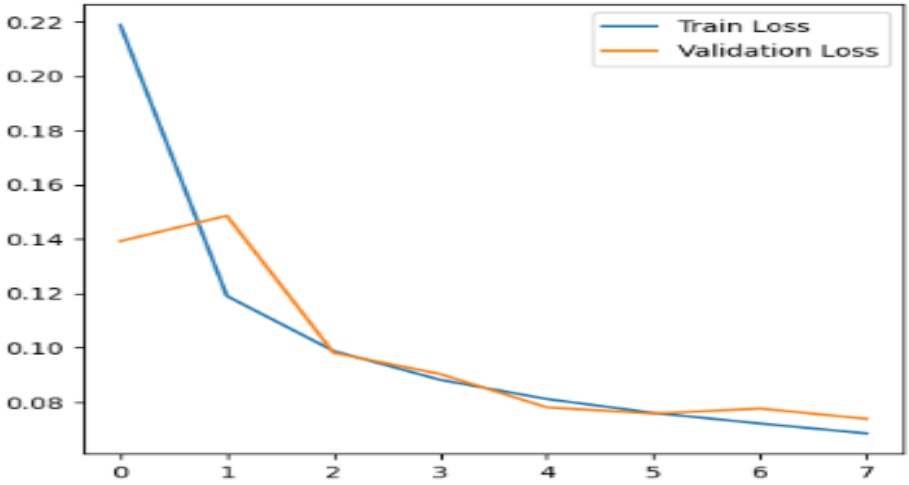

Fig 13. Loss of the Proposed LSTM-GRNN.

Fig 12 demonstrates the accuracy of the proposed LSTM-GRNN model, showing a steady increase in accuracy as the model trains. This depicts the strong learning ability of the model and its effectiveness in the correct classification of intrusion behaviors. **Fig 13** demonstrates the loss curve of the proposed LSTM-GRNN. The loss has been consistently reduced over time. This reduction in loss indicates that the model is improving in its predictions and minimizing errors.

Table 8. Comparative Analysis with Existing Models

Model	Accuracy	Precision	Recall	F1 Score	FPR	FNR
Logistic Regression	0.90	0.85	0.88	0.86	0.12	0.15
Decision Tree	0.87	0.83	0.85	0.84	0.14	0.17
Random Forest	0.92	0.89	0.91	0.90	0.09	0.11
SVM	0.89	0.84	0.87	0.85	0.13	0.16
XGBoost	0.93	0.91	0.92	0.91	0.08	0.09
CNN	0.88	0.84	0.86	0.85	0.13	0.15
LSTM	0.87	0.82	0.85	0.83	0.14	0.18
CNN-LSTM	0.90	0.86	0.88	0.87	0.11	0.14
Proposed Work(LSTM-GRNN)	**0.97**	**0.98**	**0.97**	**0.98**	**0.04**	**0.05**

Table 8 shows the comparative analysis of different models with performance metrics of accuracy, precision, recall, F1 score, False Positive Rate (FPR) and False Negative Rate (FNR). The proposed LSTM-GRNN model is found to be better than other models with maximum accuracy (97%), precision (98%), recall (97%), and F1 score (98%), and lower FPR (4%) and FNR (5%). **Fig 14** visually presents the proposed work based on the effectiveness of detection and minimizes the error rates. It performs better compared to the traditional machine learning and deep learning models like Logistic Regression, Decision Tree, SVM, XGBoost, CNN and many others.

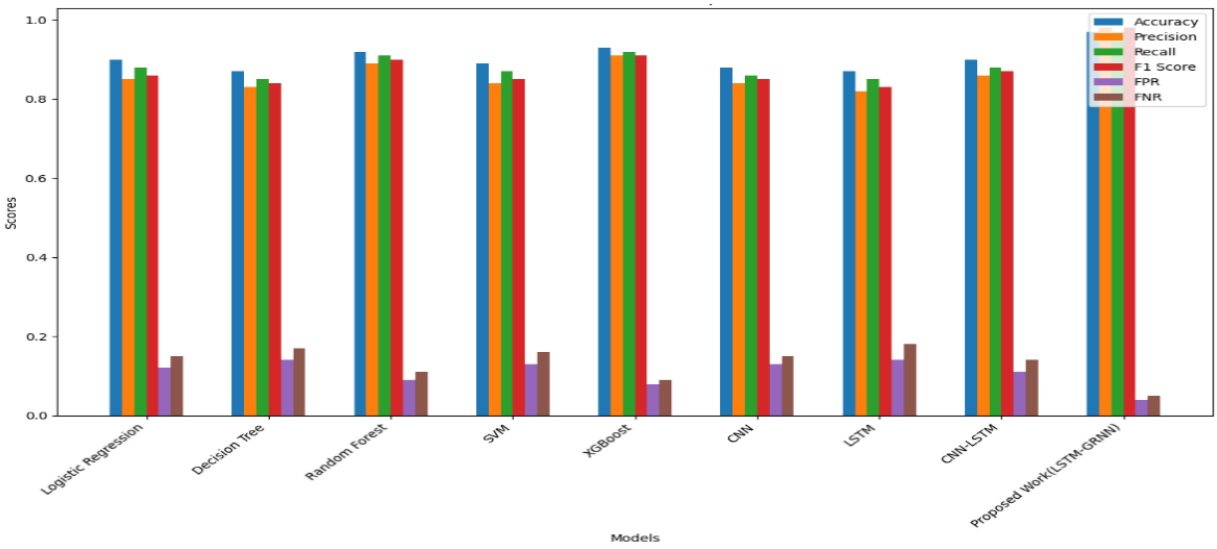

Fig 14. Comparison of Model Performance.

V. CONCLUSION

The proposed framework of LSTM-GRNN coupled with IBFPIR feature engineering and driving feature selection process by SWOA improves intrusion detection significantly by increased accuracy, precision, recall, and F1 score while also reducing false positive and false negatives. The designed model effectively discovers known and complex cyber threats within network traffic as it exploits its long-term dependence and optimizes the relevance feature. The approach ensures robustness against evolving attack patterns, making it a valuable advancement in IDS. The proposed LSTM-GRNN framework with IBFPIR-based feature engineering and SWOA-driven feature selection significantly enhances intrusion detection by improving accuracy at 97%, precision at 98%, recall at 97%, and F1 score at 98% while reducing the FPR of 4% and FNR of 5%. By exploiting long-term dependencies in network traffic and optimizing feature relevance, the model effectively identifies both known and advanced cyber threats. The approach is robust against changing attack patterns by making it a valuable advancement in IDS. Future work will focus on real-time deployment and adaptive learning mechanisms to further enhance IDS performance in dynamic cybersecurity environments.

CRediT Author Statement

The authors confirm contribution to the paper as follows:
Conceptualization: Hemanth Uppala and Renuga Devi R; **Methodology:** Hemanth Uppala; **Software:** Renuga Devi R; **Data Curation:** Hemanth Uppala; **Writing-Original Draft Preparation:** Hemanth Uppala and Renuga Devi R; **Visualization:** Hemanth Uppala; **Investigation:** Renuga Devi R; **Supervision:** Hemanth Uppala; **Validation:** Renuga Devi R; **Writing- Reviewing and Editing:** Hemanth Uppala and Renuga Devi R; All authors reviewed the results and approved the final version of the manuscript.

Data Availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Conflicts of Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing Interests

The authors declare no conflict of interest.

References

[1]. Y. Otoum and A. Nayak, “AS-IDS: Anomaly and Signature Based IDS for the Internet of Things,” Journal of Network and Systems Management, vol. 29, no. 3, Mar. 2021, doi: 10.1007/s10922-021-09589-6.
[2]. B. Nawaal, U. Haider, I. U. Khan, and M. Fayaz, “Signature-Based Intrusion Detection System for IoT,” Cyber Security for Next-Generation Computing Technologies, pp. 141–158, Nov. 2023, doi: 10.1201/9781003404361-8.
[3]. M. Sajid et al., “Enhancing intrusion detection: a hybrid machine and deep learning approach,” Journal of Cloud Computing, vol. 13, no. 1, Jul. 2024, doi: 10.1186/s13677-024-00685-x.
[4]. S. Hizal, U. Cavusoglu, and D. Akgun, “A novel deep learning-based intrusion detection system for IoT DDoS security,” Internet of Things, vol. 28, p. 101336, Dec. 2024, doi: 10.1016/j.iot.2024.101336.

[5]. O. H. Abdulganiyu, T. A. Tchakoucht, and Y. K. Saheed, "RETRACTED ARTICLE: Towards an efficient model for network intrusion detection system (IDS): systematic literature review," Wireless Networks, vol. 30, no. 1, pp. 453–482, Sep. 2023, doi: 10.1007/s11276-023-03495-2.
[6]. Andy Victor Amanoul and Adnan Mohsin Abdulazeez, "Enhanced Intrusion Detection System Using Deep Learning Algorithms : A Review," Indonesian Journal of Computer Science, vol. 13, no. 3, Jun. 2024, doi: 10.33022/ijcs.v13i3.4002.
[7]. S. M. S. Bukhari et al., "Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability," Ad Hoc Networks, vol. 155, p. 103407, Mar. 2024, doi: 10.1016/j.adhoc.2024.103407.
[8]. Z. T. Pritee, M. H. Anik, S. B. Alam, J. R. Jim, M. M. Kabir, and M. F. Mridha, "Machine learning and deep learning for user authentication and authorization in cybersecurity: A state-of-the-art review," Computers & Security, vol. 140, p. 103747, May 2024, doi: 10.1016/j.cose.2024.103747.
[9]. Yogesh and L. M. Goyal, "Retraction Note: Deep learning based network intrusion detection system: a systematic literature review and future scopes," International Journal of Information Security, vol. 24, no. 1, Nov. 2024, doi: 10.1007/s10207-024-00947-4.
[10]. U. Ahmed et al., "Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering," Scientific Reports, vol. 15, no. 1, Jan. 2025, doi: 10.1038/s41598-025-85866-7.
[11]. F. S. Alrayes, M. Zakariah, S. U. Amin, Z. I. Khan, and J. S. Alqurni, "Network Security Enhanced with Deep Neural Network-Based Intrusion Detection System," Computers, Materials & Continua, vol. 80, no. 1, pp. 1457–1490, 2024, doi: 10.32604/cmc.2024.051996.
[12]. Malibari et al., "A novel metaheuristics with deep learning enabled intrusion detection system for secured smart environment," Sustainable Energy Technologies and Assessments, vol. 52, p. 102312, Aug. 2022, doi: 10.1016/j.seta.2022.102312.
[13]. M. Deivakani, M. S. Sheela, K. Priyadarsini, and Y. Farhaoui, "An intelligent security mechanism in mobile Ad-Hoc networks using precision probability genetic algorithms (PPGA) and deep learning technique (Stacked LSTM)," Sustainable Computing: Informatics and Systems, vol. 43, p. 101021, Sep. 2024, doi: 10.1016/j.suscom.2024.101021.
[14]. B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly based network intrusion detection for IoT attacks using deep learning technique," Computers and Electrical Engineering, vol. 107, p. 108626, Apr. 2023, doi: 10.1016/j.compeleceng.2023.108626.
[15]. R. Devendiran and A. V. Turukmane, "Dugat-LSTM: Deep learning-based network intrusion detection system using chaotic optimization strategy," Expert Systems with Applications, vol. 245, p. 123027, Jul. 2024, doi: 10.1016/j.eswa.2023.123027.
[16]. K. Shafi and H. A. Abbass, "An adaptive genetic-based signature learning system for intrusion detection," Expert Systems with Applications, vol. 36, no. 10, pp. 12036–12043, Dec. 2009, doi: 10.1016/j.eswa.2009.03.036.
[17]. Dataset collection: Kaggle repository: https://www.kaggle.com/code/essammohamed4320/intrusion-detection-system-with-ml-dl/input
[18]. M. Vishwakarma and N. Kesswani, "A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection," Decision Analytics Journal, vol. 7, p. 100233, Jun. 2023, doi: 10.1016/j.dajour.2023.100233.
[19]. Hnamte and J. Hussain, "DCNNBiLSTM: An Efficient Hybrid Deep Learning-Based Intrusion Detection System," Telematics and Informatics Reports, vol. 10, p. 100053, Jun. 2023, doi: 10.1016/j.teler.2023.100053.
[20]. A.Balla, M. H. Habaebi, E. A. A. Elsheikh, Md. R. Islam, F. E. M. Suliman, and S. Mubarak, "Enhanced CNN-LSTM Deep Learning for SCADA IDS Featuring Hurst Parameter Self-Similarity," IEEE Access, vol. 12, pp. 6100–6116, 2024, doi: 10.1109/access.2024.3350978.
[21]. S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," Computer Communications, vol. 199, pp. 113–125, Feb. 2023, doi: 10.1016/j.comcom.2022.12.010.