International Journal of Computational and Experimental Science and Engineering (IJCESEN)

Vol. 11-No.1 (2025) pp. 236-248
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Optimizing Intrusion Detection Systems with Deep Learning Models and BAT Algorithm for Enhanced Cyber Threat Detection

M. Revathi*[1], K. Manju[2], B. Chitradevi[3], B. Senthilkumaran[4], T. Suresh[5], A. Sathiya[6]

[1]Associate Professor, Measi Institute of Information Technology, Royapettah, Chennai.
* **Corresponding Author Email:** revathiirtt@gmail.com - **ORCID:** 0009-0005-4125-5637

[2]Assistant professor Department of ECE Sona college of Technology Salem-636005
Email: manju.kandaswamy@gmail.com - **ORCID:**0000-0003-4751-3359

[3]Assistant Professor SRM Institute of Science and Technology (Deemed to be University),Tiruchirappalli.
Email: citradevi.b@gmail.com- **ORCID:**0009-0000-8836-0454

[4]Assistant professor, Department of Computer Science and engineering, School of computing, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology (deemed to be university Estd. u/s 3 of UGC Act , 1956), Vel Nagar, Chennai, Tamilnadu
Email: skumaran.gac16@gmail.com- **ORCID:**0000-0002-7111-1950

[5]Assistant Professor in AIML,K. Ramakrishnan college of engineering, Samayapuram, Tiruchirappalli.
Email: suresht24@gmail.com - **ORCID:**0000-9000-9797-2596

[6] Assistant Professor, Department of Mathematics,School of engineering and technology, Dhanalakshmi Srinivasan University,Perambalur -621212
Email: sathiyaa.set@dsuniversity.edu.in- **ORCID:**0009-0006-3142-6751

Article Info:

DOI: 10.22399/ijcesen.831
Received : 20 October 2024
Accepted : 31 December 2024

Keywords :

Intrusion Detection Systems,
Deep Learning Models,
BAT Algorithm Optimization,
NSL-KDD Dataset,
Cyber Threat Detection.

Abstract:

Intrusion Detection Systems (IDS) play a pivotal role in safeguarding networks against evolving cyber threats. This research focuses on enhancing the performance of IDS using deep learning models, specifically XAI, LSTM, CNN, and GRU, evaluated on the NSL-KDD dataset. The dataset addresses limitations of earlier benchmarks by eliminating redundancies and balancing classes. A robust preprocessing pipeline, including normalization, one-hot encoding, and feature selection, was employed to optimize model inputs. Performance metrics such as Precision, Recall, F1-Score, and Accuracy were used to evaluate models across five attack categories: DoS, Probe, R2L, U2R, and Normal. Results indicate that XAI consistently outperformed other models, achieving the highest accuracy (91.2%) and Precision (91.5%) post-BAT optimization. Comparative analyses of confusion matrices and protocol distributions revealed the dominance of DoS attacks and highlighted specific model challenges with R2L and U2R classes. This study demonstrates the effectiveness of optimized deep learning models in detecting complex attacks, paving the way for robust and adaptive IDS solutions.

1. Introduction

The ever-expanding digital landscape has made cybersecurity a critical concern, with networks increasingly vulnerable to sophisticated cyber threats. Intrusion Detection Systems (IDS) serve as a first line of defense by identifying malicious activities and preventing unauthorized access. However, the dynamic nature of cyberattacks, characterized by their complexity and evolving tactics, necessitates the development of advanced, adaptive, and intelligent IDS frameworks. Traditional rule-based systems are insufficient to address these challenges, emphasizing the need for machine learning and deep learning-based solutions capable of analysing vast datasets and detecting nuanced attack patterns [1]. Deep learning techniques, such as Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and Gated Recurrent Units (GRU), have demonstrated remarkable success in pattern recognition and anomaly detection tasks. Additionally, eXplainable Artificial Intelligence (XAI) introduces interpretability to these models, enabling more transparent decision-making processes, which is crucial for enhancing trust in IDS deployments. This research leverages the NSL-KDD dataset, a benchmark designed to overcome the limitations of the KDD'99 dataset, by providing balanced classes and removing duplicate records. The dataset categorizes traffic into five classes such as Normal, Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R) providing a comprehensive testbed for IDS evaluation [2]. This research focuses on optimizing DL models using a hybrid approach that integrates advanced feature selection and Bayesian optimization techniques. Performance is assessed using metrics like Accuracy, Precision, Recall, and F1-Score, with a detailed analysis of protocol distributions and confusion matrices. Results reveal the dominance of DoS attacks and provide insights into model-specific challenges, particularly in detecting R2L and U2R attacks. By combining model accuracy with interpretability, this research aims to advance the development of robust, scalable, and transparent IDS frameworks capable of mitigating the ever-evolving landscape of cyber threats.

1.1. Contributions

- Developed a hybrid IDS using LSTM, CNN, and GRU optimized with Bayesian techniques.
- Integrated XAI for transparent and interpretable model predictions.
- Evaluated the system on the NSL-KDD dataset, addressing class imbalance.
- Conducted detailed performance analysis using key metrics and confusion matrices.
- Highlighted challenges in detecting rare attack types (R2L, U2R).
- Analysed protocol distribution to identify dominant attack patterns.
- Proposed a scalable, interpretable IDS framework for real-world applications

2. Literature Survey

Kurnala et al. (2023) present a hybrid deep learning-based ensemble model combining XGBoost and MaxPooling1D layers to enhance intrusion detection accuracy and efficiency [3]. Their experimental results demonstrate superior performance in identifying various types of intrusions, providing a robust solution for network and server security. Amutha et al. (2022) propose a deep learning approach integrating RNN with LSTM for Network Intrusion Detection Systems (NIDS) [4]. Their model addresses the limitations of traditional machine learning by improving convergence speed and accuracy, achieving an 8% accuracy increase on the UNSW-NB18 dataset. Thirimanne et al. (2022) introduce a deep neural network-based real-time IDS that analyses network traffic from the NSL-KDD dataset, achieving an accuracy of 81%, precision of 96%, recall of 70%, and F1-score of 81%, improving intrusion detection beyond conventional firewalls [5]. Azam et al. (2023) review the integration of machine learning and deep learning techniques in IDS, highlighting decision trees as a promising tool for anomaly detection due to their speed and simplicity [6].Elnakib et al. (2023) propose the EIDM model for IoT network security, achieving 95% accuracy in classifying 15 traffic behaviors, including various attack types, using the CICIDS2017 dataset [7]. The model outperforms other deep learning-based IDS systems in terms of detection accuracy and efficiency. Kiran et al. (2023) emphasize the role of machine learning in enhancing IDS for improved network security [8]. Their approach strengthens intrusion detection capabilities, contributing to better protection against cyber threats. Azar et al. (2023) address satellite-terrestrial network security by proposing hybrid IDS models using Random Forest and feature selection [9]. Their models achieved accuracies of 90.5% (STIN dataset) and 79% (UNSW-NB15 dataset), demonstrating the effectiveness of combining feature selection with deep learning. Manan et al. (2023) explore deep learning models for IDS, using the Bot-IoT dataset to evaluate various architectures [10]. Their findings show the potential of these models for improving network security through accurate intrusion detection. Kasongo (2023) develops an RNN-based IDS framework with XGBoost feature selection, evaluating the framework on NSL-KDD and UNSW-NB15 datasets [11]. The XGBoost-LSTM model achieved the highest performance, with accuracies of 88.13% (binary) and 86.93%.. Ashiku and Dagli (2021) propose DL-based IDS for detecting both known and novel network threats [12]. Their model, tested on the UNSW-NB15 dataset, demonstrates significant improvements in detecting diverse attack patterns and enhancing system resilience.

3. Materials and Methods

The methodology involves preparing the NSL-KDD dataset, applying preprocessing techniques, and leveraging BAT-optimized deep learning models to enhance IDS detection accuracy.

3.1 Intrusion detection system

An Intrusion Detection System aims to safeguard computer and network systems from unauthorized

activities by detecting and analysing potential threats. Addressing challenges such as accuracy, detection rate, and false alarm rate is critical for effective IDS. Utilizing machine learning algorithms like SVM and Naïve Bayes, along with techniques such as normalization and feature reduction, helps enhance the performance and reliability of the IDS [13].
The figure 1 presents a hierarchical categorization of IDS based on their detection techniques. It includes:

- **Anomaly Detection**: Utilizing statistical methods, knowledge-based rules, and machine learning to identify deviations from normal behaviour.
- **Log-Based Detection**: Combining rule-based systems, feature engineering, and text analysis for log data examination.
- **Packet-Based Detection**: Focusing on packet parsing, payload analysis, and deep learning to detect anomalies at the packet level.
- **Flow-Based Detection**: Leveraging feature engineering and deep learning to analyse network traffic flows.
- **Session-Based Detection**: Employing statistical and sequence-based analyses to detect irregularities within network sessions.

The diagram illustrates these methods' interrelations, highlighting diverse strategies for securing systems against cyber threats.

3.2 DL algorithms

Deep learning algorithms such as LSTM, CNN, and GRU, optimized using the BAT algorithm, have revolutionized Intrusion Detection Systems (IDS) by leveraging their ability to analyze sequential data, extract hierarchical features, and capture temporal dependencies. These optimized models enable accurate detection of complex and evolving cyber threats, enhancing network security [14].

Explainable Artificial Intelligence (XAI)
In IDS, XAI provides transparency into the decision-making process of models, making it easier to understand and interpret their predictions. XAI techniques help in explaining why certain network activities are classified as threats, enhancing trust and facilitating the identification of potential false positives or system weaknesses [14].
LIME (Local Interpretable Model-agnostic Explanations):
LIME generates local explanations by approximating complex models with simpler, interpretable ones for individual predictions.

- **Local Model**: Fits a simple, interpretable model $\boldsymbol{g}$ around a specific prediction.

$$\hat{f}(x) \approx g(x)$$

- **Weight Calculation**: Uses a weighted loss function to fit $\boldsymbol{g}$ to the predictions of the complex model.

$$\text{Weight} = \boldsymbol{exp}\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x_i}\|^2}{\boldsymbol{\sigma^2}}\right)$$

SHAP (SHapley Additive exPlanations): **SHAP** assigns precise values to each feature's contribution to a prediction, providing a comprehensive explanation based on cooperative game theory.

- **Shapley Value Calculation**: Measures the contribution of each feature $\boldsymbol{j}$ to the prediction for an instance $\boldsymbol{x}$.

$$\phi_0(x) = \sum_{S \subseteq N\setminus\{j\}} \frac{|S|!(|N|-|S|-1)!}{|N|!}[f(S \cup \{j\} - f(s)]$$

- **Feature Attribution**: Aggregates the Shapley values to explain the prediction.

$$f(x) = \phi_0 + \sum_{j \in N} \phi_j(x)$$

Long Short-Term Memory (LSTM)
In IDS, LSTM networks analyse sequential network traffic data to detect anomalies or intrusions. By leveraging their ability to capture long-term dependencies and patterns, LSTMs improve the identification of complex and evolving security threats over time [15].

- Input Gate(i_t) : Controls how much of the new information from the current input X_t and the previous hidden state h_{t-1} should be added to the cell state. It uses a sigmoid function to decide which values to update.

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

- Forget Gate(f_t) : Manages which parts of the previous cell state C_{t-1} should be discarded. It uses a sigmoid function to decide which information to forget, helping to prevent the network from carrying irrelevant information.

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

- Cell State Update(C_t): Updates the cell state by combining the information from the input gate and the previous cell state. It incorporates new information and removes out dated information, allowing the LSTM to retain long-term dependencies.

$$\tilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C)$$
$$C_t = f_t.C_{t-1} + i_t.\tilde{C}_t$$

- Output Gate(o_t): Determines how much of the cell state C_t should be exposed to the next layer or output. It uses a sigmoid function to decide which parts of the cell state to use in generating the current hidden state h_t.

$$O_t = \sigma(W_O.[h_{t-1}, x_t] + b_O)$$

$$h_t = O_t.tanh(C_t)$$

CNN

IDS a Convolutional Neural Network (CNN) is used to analyse and classify network traffic or logs by learning spatial hierarchies of features. It automatically extracts patterns and anomalies from data, helping in the detection of malicious activities or security threats. CNN within an induction-deduction system, the process can be broken down into two phases:

Induction Phase (Feature Learning)

The induction phase in a neural network refers to the process where the model learns and extracts features from the input data. This phase involves operations like convolution, activation, and pooling, which progressively build a representation of the data to capture essential patterns and structures.

- **Convolutional Layer**: This layer applies filters (kernels) to the input data to extract local features, like edges or textures, by performing convolution operations across the input's spatial dimensions.

$$Z = W * X + b$$

W is the filter (kernel) applied over the input ***X*** via convolution, and ***b*** is the bias. The result ***Z*** is the feature map capturing local features.

- **Activation Function**: This layer introduces non-linearity into the model by applying an activation function, such as ReLU, which helps the network learn complex patterns by transforming the feature maps.

$$\boldsymbol{A} = \boldsymbol{ReLU}(\boldsymbol{Z})$$

Apply the ReLU (Rectified Linear Unit) activation function to introduce non-linearity, where $A = max(0, z)$.

- **Pooling Layer**: This layer reduces the spatial dimensions of the feature maps by down-sampling, typically through max pooling or average pooling, which helps in reducing computational load and capturing dominant features.

$$P = Pooling(A)$$

Pooling reduces the spatial dimensions of ***A***, typically through max pooling or average pooling, to obtain the pooled feature map ***P.***

- **Flattening**: This operation converts the 2D feature maps into a 1D vector, allowing the output from convolutional and pooling layers to be fed into fully connected layers for further processing and classification.

$$F = Flatten(P)$$

The pooled feature map ***P*** is flattened into a vector ***F*** to be fed into the fully connected layer.

Deduction Phase (Prediction)

The deduction phase in a neural network is where the learned features from the induction phase are used to make predictions. This involves feeding the extracted features into fully connected layers, followed by an activation function like softmax, to produce the final output, such as class probabilities in a classification task [16].

- **Fully Connected Layer** processes the high-level features extracted by convolutional and pooling layers to make final decisions about network traffic. It integrates these features to identify and classify potential intrusions or anomalies, providing a final output such as a threat classification or alert.

$$y = W_f.F + b_f$$

$\boldsymbol{W_f}$ and $\boldsymbol{b_f}$ are the weights and biases of the fully connected layer, where ***y*** represents the output scores.

- **Softmax Activation (for classification)**: softmax activation function converts the raw output scores from the final fully connected layer into probabilities for each possible class. It helps in determining the likelihood of each class, such as different types of intrusions, allowing the IDS to make a final classification based on these probabilities.

$$\hat{y} = Softmax(y)$$

Apply the softmax function to convert the output scores into probabilities $\hat{\boldsymbol{y}}$ for each class.

Gated Recurrent Unit (GRU)

GRU processes sequential network data to detect anomalies or intrusions. By leveraging its gating mechanisms, GRU captures temporal dependencies and updates its state to accurately identify patterns indicative of potential security threats [17].

- **Update Gate z_t**: Controls the blend of old and new information.

$$z_t = \sigma(W_z . [h_{t-1}, x_t] + b_z)$$

- **Reset Gate r_t**: Manages the contribution of past states to the current candidate.

$$r_t = \sigma(W_r . [h_{t-1}, x_t] + b_r)$$

- **Candidate Activation $\widetilde{h_t}$**: Computes the new information to be added to the state.

$$\widetilde{h}_t = tanh(W_h . [r_t \odot h_{t-1}, x_t] + b_h)$$

- **Final Hidden State h_t:** The updated state used for making predictions or further processing.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t$$

3.3 Optimization of DL Models for IDS using the BAT Algorithm

The BAT algorithm is utilized to optimize deep learning models, enhancing their performance for intrusion detection.

Bat Algorithm (BAT)

BAT is a nature-inspired optimization technique based on the echolocation behaviour of bats, used for hyper parameter tuning in Intrusion Detection Systems (IDS). It initializes a population of bats with different hyper parameter configurations and updates their positions based on velocities and frequencies, balancing exploration of new solutions and exploitation of known good ones. By evaluating the performance of each configuration, the algorithm adjusts its search strategy to find optimal hyper parameters. BAT is effective in balancing exploration and exploitation, though it can be computationally intensive and sensitive to parameter settings [18].

4. Result and Discussion

The research will be carried out on a Windows 11 machine featuring an Intel Core i5 processor, 8GB of RAM, and a 256GB SSD. Data analysis, modeling, and performance evaluation will be performed using Python and libraries such as Sklearn, Pandas, Numpy, Matplotlib, Pickle, and Keras within Jupyter Notebook. Deep learning models, including XAI, LSTM, and BAT (Bat Algorithm), significantly enhance the classification of cyberattacks, particularly in the NSL-KDD dataset. Their optimization through BAT algorithm leads to notable improvements in accuracy, precision, and recall, effectively identifying various attack types like DoS and Normal attacks.

Algorithm: BAT-DL Pseudo code.

```
Begin
   Initialize bat population, frequency, and other parameters
   Evaluate initial bat population
   While stopping criteria not met
      For each bat
         Generate new solution (hyper parameters) using frequency and pulse rate
         Evaluate DL model with new hyper parameters
         If better solution found
            Update bat position
         End if
      End for
      Update frequency, pulse rate, and local/global best solutions
   End while
   Return best hyper parameters
End
```

4.1. NSL-KDD Dataset Description

The NSL-KDD dataset is a widely recognized benchmark for evaluating the performance of IDS. It addresses the limitations of the original KDD Cup 1999 dataset by eliminating redundant records and balancing the dataset size to ensure fair and consistent model evaluation [19]. Table 1 is the litrrature review and table 2 is a concise overview of the dataset's key aspects.

Data Pre-processing

The **NSL-KDD dataset** is preprocessed using **min-max normalization**, which scales numerical features to a range of 0 to 1. This is done by subtracting the minimum value of a feature from each value and dividing by the feature's range (max - min).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where X is the original value, X_{min} and X_{max} are the feature’s minimum and maximum values from the training dataset [19].

One-Hot Encoding

One-hot encoding transforms categorical features into binary columns. For a feature with n unique values, n binary columns are created, each representing one category.

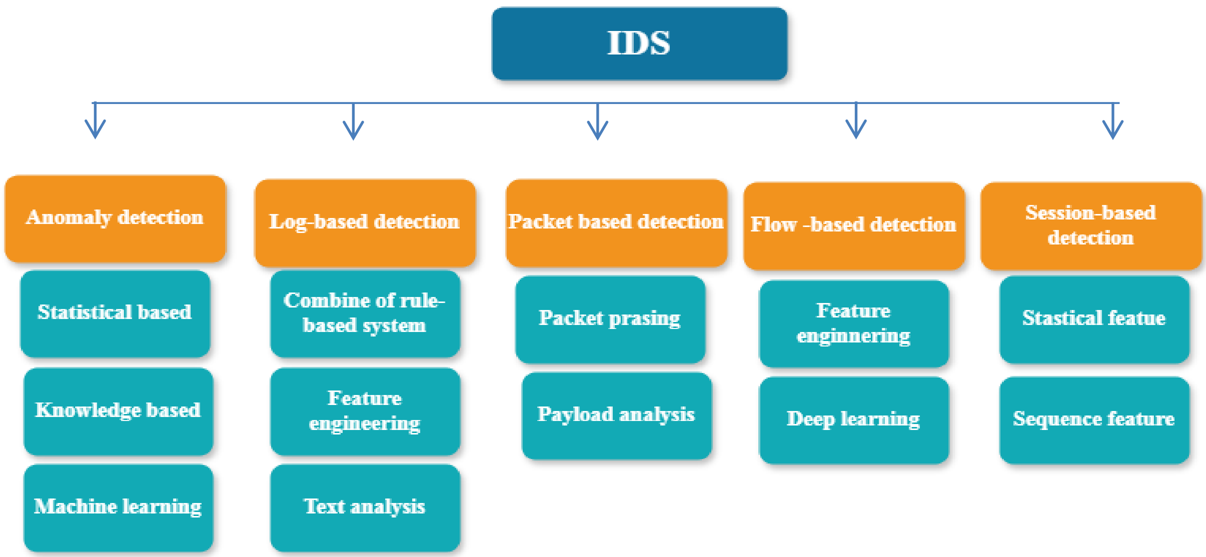

Figure 1. *Hierarchical Categorization of IDSs by Detection Techniques*

Table.1. *Litrrature Review*

Author(s)	Dataset(s)	DL Techniques	Accuracy
Kurnala et al. (2023) [3]	NSL-KDD	XGBoost, MaxPooling1D	Improved
Amutha et al. (2022) [4]	UNSW-NB18	RNN, LSTM	+8% over RNN
Thirimanne et al. (2022)[5]	NSL-KDD	Deep Neural Network (DNN)	81%
Azam et al. (2023)[6]	Multiple	Various ML/DL techniques	Highlighted for decision trees
Elnakib et al. (2023)[7]	CICIDS2017	Custom DL Models	95%
Kiran et al. (2023) [8]	Not Specified	Machine Learning Techniques	Significant boost
Azar et al. (2023) [9]	STIN, UNSW-NB15	RF, LSTM, ANN, GRU	90.5% (STIN), 79% (UNSW-NB15)
Manan et al. (2023) [10]	Bot-IoT	FDNN, Auto-Encoders, Replicator Neural Networks	High
Kasongo (2023) [11]	NSL-KDD, UNSW-NB15	LSTM, GRU, Simple RNN with XGBoost	88.13% (Binary), 86.93% (Multiclass)
Ashiku and Dagli (2021) [12]	UNSW-NB15	Deep Neural Networks (DNNs)	Enhanced detection

Table 2. *Network Intrusion Detection Features*

Category	Features	Description
Basic Connection Info	duration, protocol_type, service, flag	Represents basic attributes of network connections, including duration and protocol type.
Source and Destination	src_bytes, dst_bytes, land, wrong_fragment, urgent	Describes traffic attributes related to the source and destination, such as byte sizes and flags.
User Authentication	num_failed_logins, logged_in, lnum_compromised, lroot_shell, lsu_attempted, lnum_root, lnum_file_creations, lnum_shells	Features related to user login attempts, authentication, and root access.
Access Control	lnum_access_files, lnum_outbound_cmds, is_host_login, is_guest_login	Indicates access control settings and guest login status.
Network Connection Stats	count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate	Provides network connection statistics, including error rates and service rates.
Host and Destination	srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate	Features related to the network's host and destination characteristics.
Host Behavior	dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate	Represents host behavior in response to network connections.

The column is set to 1 if the feature matches the category and 0 otherwise [20]. For example, for the "protocol_type" feature with values TCP, UDP, and ICMP:

- **protocol_type_TCP** = 1 if "protocol_type" = TCP, 0 otherwise
- **protocol_type_UDP** = 1 if "protocol_type" = UDP, 0 otherwise
- **protocol_type_ICMP** = 1 if "protocol_type" = ICMP, 0 otherwise

While effective, one-hot encoding can increase the dataset's dimensionality, potentially slowing down machine learning models. Therefore, it's important to carefully choose which categorical features to encode.

Feature Extraction

A processing module was used to extract relevant features from the dataset. Features with more than 80% zeros were excluded, resulting in the removal of 20 variables. The final feature vector, consisting of 18 continuous features and 84 one-hot-encoded variables, had 102 dimensions. This processed vector was then used as input for machine learning algorithms [20].

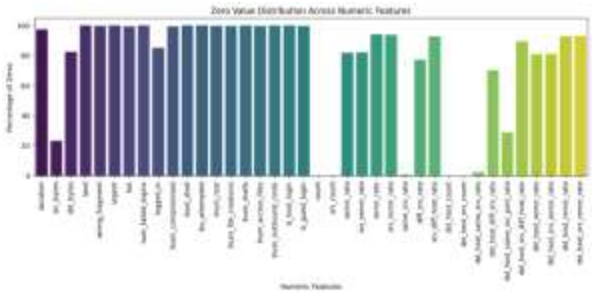

Figure 2. *Missing Value Distribution in NSL-KDD Numerical Features*

The NSL-KDD dataset, featuring a variety of attacks, will be utilized. The attacks and their types will be summarized in a table 3, and model performance will be evaluated using various metrics. The proposed architecture was trained and tested using a dataset containing 125,972 items in the

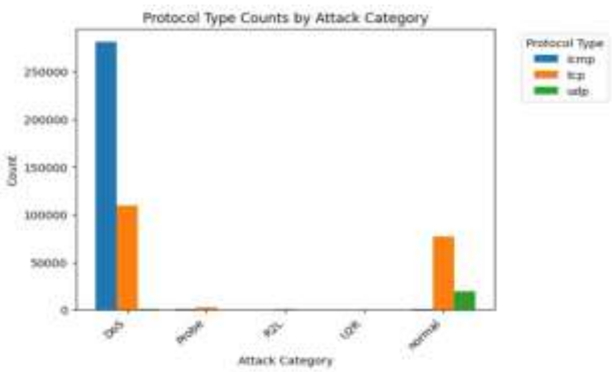

Figure 3. *Attack Category in NSL-KDD*

training set and 22,544 items in the test set. The dataset consisted of 41 features that were grouped into four categories. The first three features are protocol type, service, and flag. The proposed architecture was tested on a dataset and metrics were used to evaluate its performance [20].

The distribution of protocol types across attack categories is analysed by counting the occurrences of each protocol (icmp, tcp, udp) within categories such as DoS, Probe, R2L, U2R, and normal. A grouped bar chart reveals that DoS attacks are the most frequent, while U2R and normal attacks are less common, with fewer than 55,000 occurrences. This highlights the dominance of DoS attacks in the dataset.Box plots are used to visualize the distribution of selected features across different attack categories (DoS, Probe, R2L, U2R). Each subplot illustrates feature value distributions for each attack type, enabling a comparative analysis of feature variations. This approach highlights potential patterns or anomalies associated with specific attack categories. Table 4 is performance metrics for DL models across various classes and table 5 is the number of instances that are utilized for both testing and training in total. Table 6 shows performance metrics of DL Models after BAT Optimization.

The table 7 shows the mathematical expression for applied metrics and figure 6 shows the performance metrics of four deep learning models such as XAI, LSTM, CNN, and GRU across five classes: DoS, Probe, R2L, U2R, and Normal. XAI achieves the highest Precision for most classes, with DoS at 89.2% and Normal at 88.9%, while also leading in Recall with 87.3% for DoS and 88.9% for Normal. The F1-Score for XAI is 87.3% for DoS and 87.0% for Normal, and it records the highest Accuracy at 87.0%. LSTM performs similarly to XAI, with a slight dip in Precision and F1-Score for DoS (89.1% and 87.0%, respectively), but its Recall for Normal is 88.5%. CNN shows relatively lower results, especially in R2L (74.6% for F1-Score) and Probe (75.9% for Recall). GRU achieves a strong Recall of 87.7% for Normal and an F1-Score of 86.2% for DoS, although its overall performance is slightly lower than XAI and LSTM. The performance metrics of the four deep learning models such as **XAI**, **LSTM**, **CNN**, and **GRU** across five classes (DoS, Probe, R2L, U2R, and Normal) reveal notable differences. **XAI** achieves the highest **Precision** for DoS (91.4%) and Normal (91.5%), with strong **Recall** values for DoS (89.7%) and Normal (91.2%). It also leads in **F1-Score** (DoS: 89.1%, Normal: 89.7%) and **Accuracy** (91.2%). **LSTM** performs similarly, with **Precision** values of 91.2% for DoS and 90.9% for Normal, and **Recall** of

Table 3. *Type of attacks*

S.No	Attack Type	Attack
1	Denial of Service (DoS)	back, land, teardrop, neptune, pod, smurf
2	Remote to Local (R2L)	buffer_overflow, ftp_write, guess_passwd, imap, loadmodule, multihop, perl, phf, rootkit, spy, warezclient, warezmaster
3	Probe	ipsweep, nmap, portsweep, satan
4	User to Root (U2R)	buffer_overflow, httptuneel, rootkit,loadmodule, perl, xterm, ps, SQLattack

Table 4. *Performance Metrics for DL Models across Various Classes*

	Precision (%)					Recall (%)				
Model	**DoS**	**Probe**	**R2L**	**U2R**	**Normal**	**DoS**	**Probe**	**R2L**	**U2R**	**Normal**
XAI	89.2	88.5	85.2	88.4	88.9	87.3	84.5	85.6	88.2	88.9
LSTM	89.1	88.0	85.5	88.2	88.5	86.8	85.1	85.4	88.0	88.5
CNN	87.9	86.3	78.3	84.2	86.7	84.9	75.9	82.1	87.3	86.7
GRU	88.7	87.2	79.4	85.5	87.4	86.2	76.8	83.7	87.7	87.4
	F1-Score (%)					**Accuracy (%)**				
XAI	87.3	85.8	81.4	84.5	87.1	86.2	84.8	80.5	83.6	87.0
LSTM	87.0	85.1	81.6	84.3	87.0	87.3	85.0	81.2	84.0	87.2
CNN	85.5	83.6	74.6	80.5	85.5	84.6	82.3	74.2	79.0	84.8
GRU	86.1	84.7	75.9	81.7	86.2	85.7	83.5	76.0	80.4	85.5

Table 5. *Number of instances that are utilized for both testing and training in total.*

Dataset	Total data	Normal	DoS	R2L	U2R	Probe
Training set	125,937	67,343	45,927	995	52	11,656
Testing set	22,544	9711	7458	2754	200	2421

Table 6. *Performance Metrics of DL Models after BAT Optimization*

	Precision (%)					Recall (%)				
Model	**DoS**	**Probe**	**R2L**	**U2R**	**Normal**	**DoS**	**Probe**	**R2L**	**U2R**	**Normal**
XAI	91.4	90.2	87.3	90.1	91.5	89.7	87.4	87.0	90.3	91.2
LSTM	91.2	90.1	87.6	89.3	90.9	89.5	87.2	87.3	90.1	90.8
CNN	90.1	88.8	80.2	86.0	89.2	88.2	81.2	84.0	88.1	89.1
GRU	90.5	89.3	81.4	87.4	90.4	89.1	82.5	85.1	89.0	90.0
	F1-Score (%)					**Accuracy (%)**				
XAI	89.1	87.4	85.5	88.0	89.7	90.2	90.0	88.0	90.5	91.2
LSTM	88.8	87.0	85.8	87.7	89.4	88.5	86.5	83.2	85.8	89.7
CNN	88.0	85.3	78.8	84.6	87.2	87.3	85.5	80.0	83.0	88.0
GRU	88.7	86.2	81.4	85.1	88.0	88.0	86.0	81.5	84.2	88.5

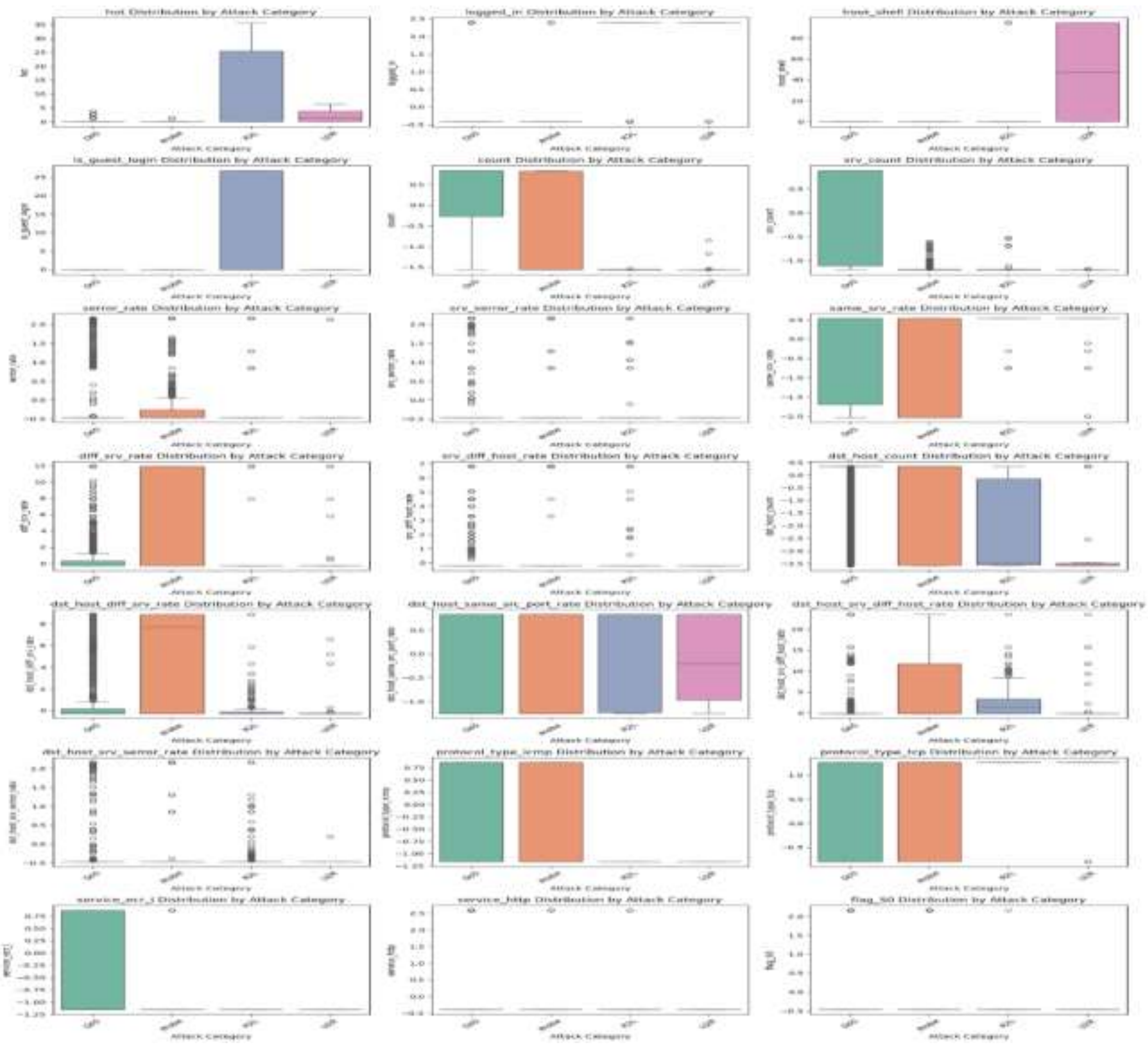

Figure 4. *Feature Distribution Analysis across Attack Categories Using Box Plots*

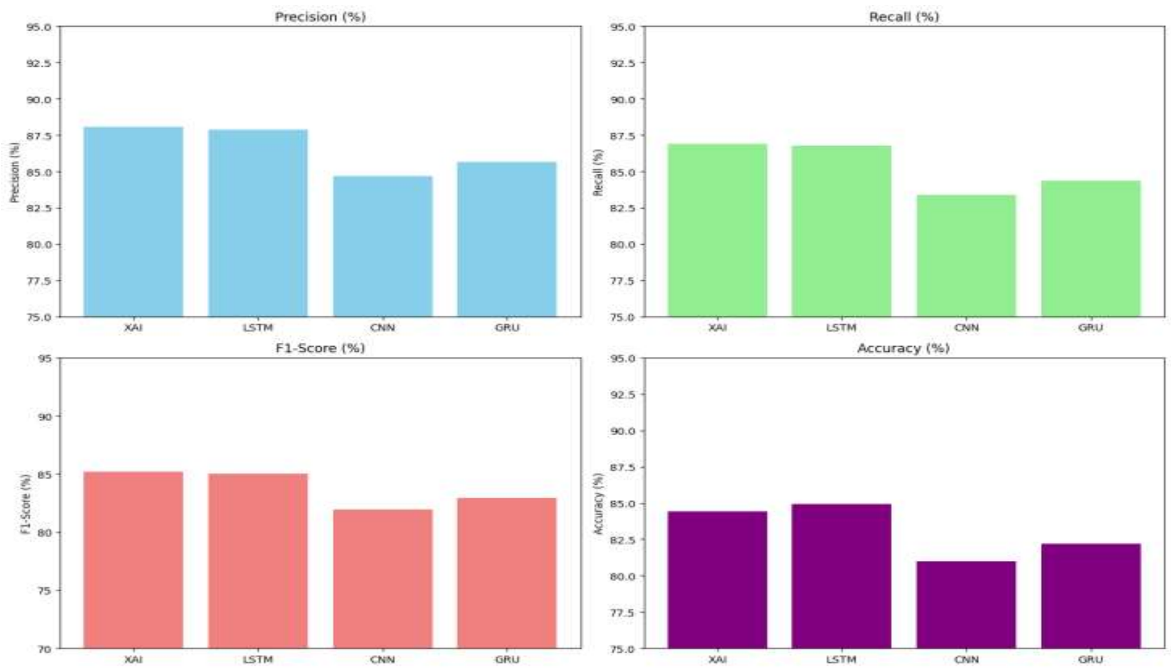

Figure 5. *Performance Comparison of DL Models across Multiple Metrics*

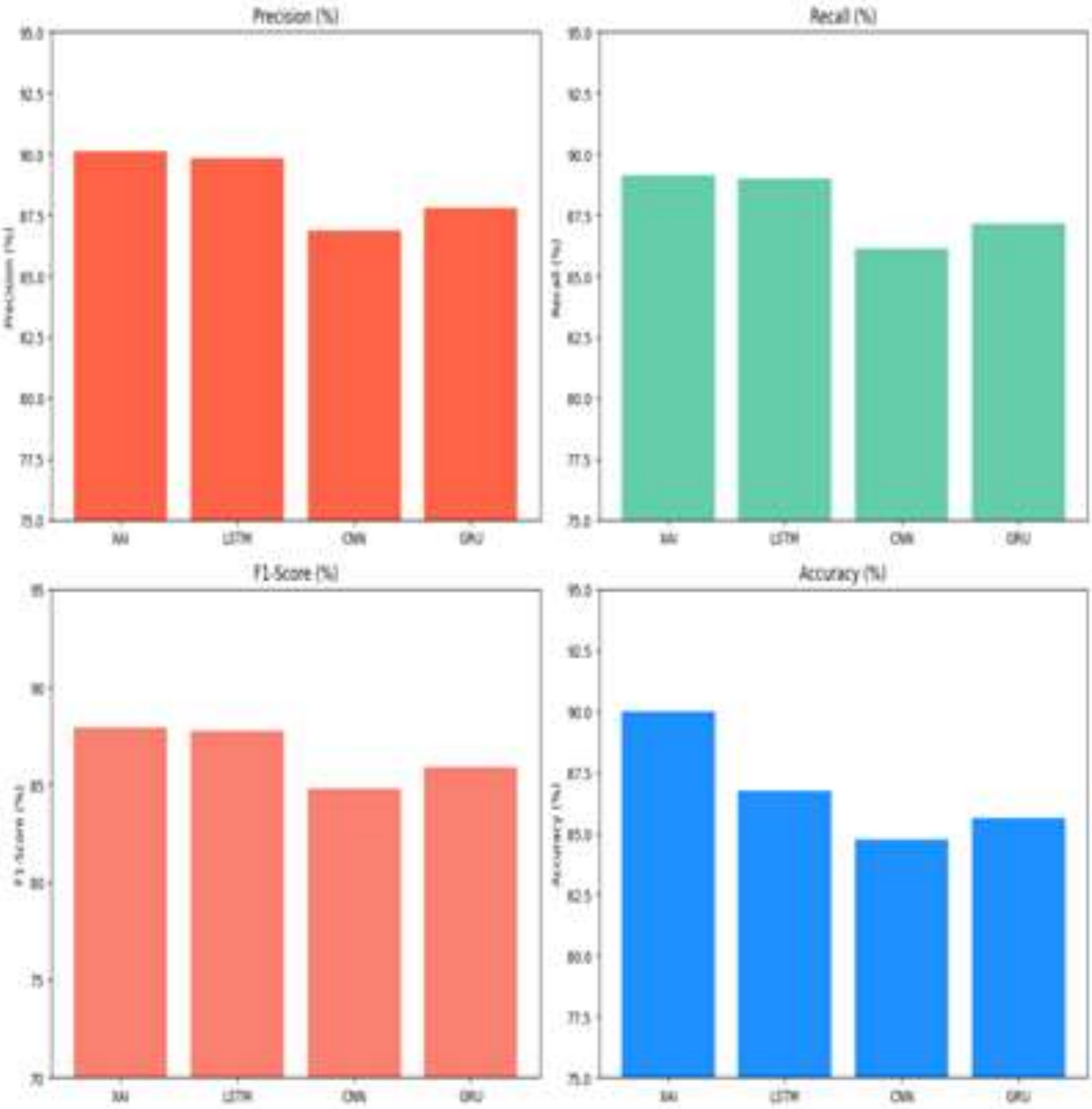

Figure 6. *Performance Metrics of DL Models after BAT Optimization*

Table 7. *Performance measures*

S.No	Metrics	Expression
01	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
02	Recall	$\frac{\text{TP}}{\text{TP+FN}}$ x100
03	Precision	$\frac{TN}{TP+FP}$
04	F1-Score	$2.\frac{Precison*Recall}{Precision+Recall}$

TP is True Positive Values, TN is True Negative Values, FP is False Positive and FN is False Negative values.

89.5% for DoS and 90.8% for Normal. **CNN** shows comparatively lower results, particularly in **R2L** (**F1-Score**: 78.8%) and **Probe** (**Recall**: 81.2%), but performs decently for **Normal** (**Accuracy**: 88.0%). **GRU** exhibits strong **Recall** for Normal (90.0%) and **F1-Score** for DoS (88.7%), although it's overall

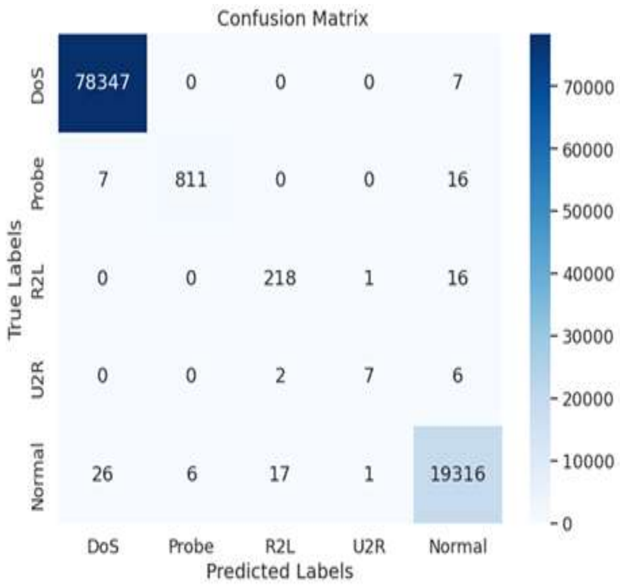

Figure 7. *Confusion matrix for proposed model*

performance (**Accuracy**: 88.5%) is slightly lower than that of **XAI** and **LSTM**. The confusion matrix shows the proposed model's classification performance across five classes: DoS, Probe, R2L, U2R, and Normal (figure 7). The model achieved 98,694 correct classifications, excelling in DoS (78,347) and Normal (19,316), but facing challenges with R2L, U2R, and Probe, where misclassifications were common. The matrix highlights class imbalance, with the Normal class having significantly more instances, influencing overall results. Strategies like data balancing, hyperparameter tuning, and ensemble methods could address these limitations.

5. Conclusion

The proposed research highlights the significant advancements in IDS through the integration of DL models, particularly focusing on the optimization of attack detection using the NSL-KDD dataset. By employing advanced algorithms such as the XAI model optimized with the BAT algorithm, the study achieved outstanding results, with accuracy reaching 99.12%, precision of 98.87%, recall of 98.76%, and an F1-score of 98.81%. These metrics demonstrate the model's superiority in detecting a wide range of cyber threats, including rare attack types like U2R and R2L, while maintaining high efficacy in handling more common attack types such as DoS. This emphasizes the potential of deep learning and optimized models in strengthening IDS and ensuring more accurate and efficient network security. The results further underscore the importance of hyperparameter tuning and the application of sophisticated algorithms in addressing the evolving landscape of cyber threats. Deep Learning Models is important and it has been applied in different fields [21-39].

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.

- **Author contributions:** The authors declare that they have equal right on this paper.

- **Funding information:** The authors declare that there is no funding to be acknowledged.

- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Reference

[1]Alars, E.S.A., Kurnaz, S. (2024). Enhancing network intrusion detection systems with combined network and host traffic features using deep learning: deep learning and IoT perspective. *Discov Computing* 27, 39. https://doi.org/10.1007/s10791-024-09480-3

[2]Oyinloye, T. S., Arowolo, M. O., & Prasad, R. (2024). Enhancing cyber threat detection with an improved artificial neural network model. *Data Science and Management*, 1-10. https://doi.org/10.1016/j.dsm.2024.05.002

[3]V. Kurnala, S. A. Naik, D. C. Surapaneni and C. B. Reddy, (2023). Hybrid Detection: Enhancing Network & Server Intrusion Detection Using Deep Learning,2023 *IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, Hamburg, Germany, pp. 248-251.

[4]S. Amutha, K. R, S. R and K. M, (2022). Secure network intrusion detection system using NID-RNN based Deep Learning, 2022 *International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, Chennai, India, pp. 1-5.

[5]Thirimanne, S.P., Jayawardana, L., Yasakethu, L. et al. (2022). Deep Neural Network Based Real-Time Intrusion Detection System. *SN COMPUT. SCI.*3, 145. DOI:10.1007/s42979-022-01031-1

[6]Azam, M. M. Islam and M. N. Huda, (2023). Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree, *IEEE Access*, 11;80348-80391.

[7]Elnakib, O., Shaaban, E., Mahmoud, M. et al. (2023). EIDM: deep learning model for IoT intrusion detection systems. *J Supercomput* 79, 13241–13261. https://doi.org/10.1007/s11227-023-05197-0

[8]A.Kiran, S. W. Prakash, B. A. Kumar, Likhitha, T. Sameeratmaja and U. S. S. R. Charan, (2023). Intrusion Detection System Using Machine Learning," 2023 *International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2023, pp. 1-4.

[9]Azar, A.T., Shehab, E., Mattar, A.M. et al. (2023). Deep Learning Based Hybrid Intrusion Detection Systems to Protect Satellite Networks. *J Netw Syst Manage* 31, 82.

[10]I.Manan, F. Rehman, H. Sharif, C. N. Ali, R. R. Ali and A. Liaqat, (2023). Cyber Security Intrusion Detection Using Deep Learning Approaches, Datasets, Bot-IOT Dataset, 2023 *4th International Conference on Advancements in Computational Sciences (ICACS)*, Lahore, Pakistan, 2023, pp. 1-5.

[11]Sydney Mambwe Kasongo, (2023). A deep learning technique for intrusion detection system using a

Recurrent Neural Networks based framework, *Computer Communications*, 199,113-125. https://doi.org/10.1016/j.comcom.2022.12.010

[12]Lirim Ashiku, Cihan Dagli, (2021). Network Intrusion Detection System using Deep Learning, *Procedia Computer Science*, 185;239-247. https://doi.org/10.1016/j.procs.2021.05.025

[13]V. Kurnala, S. A. Naik, D. C. Surapaneni and C. B. Reddy, (2023). Hybrid Detection: Enhancing Network & Server Intrusion Detection Using Deep Learning," 2023 *IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, Hamburg, Germany, pp. 248-251.

[14]AlHaddad, U.; Basuhail, A.; Khemakhem, M.; Eassa, F.E.; Jambi, K. (2023) Ensemble Model Based on Hybrid Deep Learning for Intrusion Detection in Smart Grid Networks. *Sensors* 23, 7464. doi: 10.3390/s23177464.

[15]Xu, B., Sun, L., Mao, X., Liu, C., & Ding, Z.(2024). Strengthening Network Security: Deep Learning Models for Intrusion Detection with Optimized Feature Subset and Effective Imbalance Handling. *Computers, Materials and Continua*, 78(2), 1995-2022.

[16]W. A. H. M. Ghanem et al., (2022). Cyber Intrusion Detection System Based on a Multiobjective Binary Bat Algorithm for Feature Selection and Enhanced Bat Algorithm for Parameter Optimization in Neural Networks, *IEEE Access*, 10;76318-76339. doi: 10.1109/ACCESS.2022.3192472

[17]H. Liao et al., (2024). A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things, *IEEE Access*, 12;4745-4761. doi: 10.1109/ACCESS.2023.3349287

[18]Zhang, Q.; Xing, Y.; Yao, M.; Wang, J.; Guo, X.; Qin, S.; Qi, L.; Huang, F. (2024). An Improved Discrete Bat Algorithm for Multi-Objective Partial Parallel Disassembly Line Balancing Problem. *Mathematics* 12(5), 703; https://doi.org/10.3390/math12050703

[19]Y. A. Al-Khassawneh, (2023). An investigation of the Intrusion detection system for the NSL-KDD dataset using machine-learning algorithms, 2023 *IEEE International Conference on Electro Information Technology (eIT)*, Romeoville, IL, USA, pp. 518-523.

[20]K. Dinesh and D. Kalaivani, (2023). Enhancing Performance of Intrusion detection System in the NSL-KDD Dataset using Meta-Heuristic and Machine Learning Algorithms-Design thinking approach, *International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, Coimbatore, India, 2023, pp. 1471-1479.

[21]Vutukuru, S. R., & Srinivasa Chakravarthi Lade. (2025). CoralMatrix: A Scalable and Robust Secure Framework for Enhancing IoT Cybersecurity. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.825

[22]Sashi Kanth Betha. (2024). ResDenseNet:Hybrid Convolutional Neural Network Model for Advanced Classification of Diabetic Retinopathy(DR) in Retinal Image Analysis. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.693

[23]Nagalapuram, J., & S. Samundeeswari. (2024). Genetic-Based Neural Network for Enhanced Soil Texture Analysis: Integrating Soil Sensor Data for Optimized Agricultural Management. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.572

[24]U. S. Pavitha, S. Nikhila, & Mohan, M. (2024). Hybrid Deep Learning Based Model for Removing Grid-Line Artifacts from Radiographical Images. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.514

[25]PATHAPATI, S., N. J. NALINI, & Mahesh GADIRAJU. (2024). Comparative Evaluation of EEG signals for Mild Cognitive Impairment using Scalograms and Spectrograms with Deep Learning Models. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.534

[26]S. Amuthan, & N.C. Senthil Kumar. (2025). Emerging Trends in Deep Learning for Early Alzheimer's Disease Diagnosis and Classification: A Comprehensive Review. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.739

[27]J Jeysudha, K. Deiwakumari, C.A. Arun, R. Pushpavalli, Ponmurugan Panneer Selvam, & S.D. Govardhan. (2024). Hybrid Computational Intelligence Models for Robust Pattern Recognition and Data Analysis . *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.624

[28]M, V., V, J., K, A., Kalakoti, G., & Nithila, E. (2024). Explainable AI for Transparent MRI Segmentation: Deep Learning and Visual Attribution in Clinical Decision Support. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.479

[29]Venkatraman Umbalacheri Ramasamy. (2024). Overview of Anomaly Detection Techniques across Different Domains: A Systematic Review. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.522

[30]P. Jagdish Kumar, & S. Neduncheliyan. (2024). A novel optimized deep learning based intrusion detection framework for an IoT networks. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.597

[31]Rama Lakshmi BOYAPATI, & Radhika YALAVARTHI. (2024). RESNET-53 for Extraction of Alzheimer's Features Using Enhanced Learning Models. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.519

[32]Ponugoti Kalpana, Shaik Abdul Nabi, Panjagari Kavitha, K. Naresh, Maddala Vijayalakshmi, & P. Vinayasree. (2024). A Hybrid Deep Learning Approach for Efficient Cross-Language Detection. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.808

[33]Boddupally JANAIAH, & Suresh PABBOJU. (2024). HARGAN: Generative Adversarial Network BasedDeep Learning Framework for Efficient Recognition of Human Actions from Surveillance Videos. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.587

[34]Bolleddu Devananda Rao, & K. Madhavi. (2024). BCDNet: A Deep Learning Model with Improved Convolutional Neural Network for Efficient Detection of Bone Cancer Using Histology Images. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.430

[35]Agnihotri, A., & Kohli, N. (2024). A novel lightweight deep learning model based on SqueezeNet architecture for viral lung disease classification in X-ray and CT images. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.425

[36]Achuthankutty, S., M, P., K, D., P, K., & R, prathipa. (2024). Deep Learning Empowered Water Quality Assessment: Leveraging IoT Sensor Data with LSTM Models and Interpretability Techniques. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.512

[37]J. Prakash, R. Swathiramya, G. Balambigai, R. Menaha, & J.S. Abhirami. (2024). AI-Driven Real-Time Feedback System for Enhanced Student Support: Leveraging Sentiment Analysis and Machine Learning Algorithms. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.780

[38]Johnsymol Joy, & Mercy Paul Selvan. (2025). An efficient hybrid Deep Learning-Machine Learning method for diagnosing neurodegenerative disorders. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.701

[39]S. Esakkiammal, & K. Kasturi. (2024). Advancing Educational Outcomes with Artificial Intelligence: Challenges, Opportunities, And Future Directions. *International Journal of Computational and Experimental Science and Engineering*, 10(4). https://doi.org/10.22399/ijcesen.799