10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

Comparative Analysis of Machine Learning and Deep Learning Approaches for Intrusion Detection Systems in Bihar

Ajit Kumar,[1], Prof. (Dr.) Om Prakash Roy[2],

[1] (Research Scholar), Faculty of Science (Computer Applications), Department of Mathematics,
B. R. A. Bihar University, Muzaffarpur
Email – ajitjnvassam@gmail.com

[2]Professor, Department of Physics, B. R. A. Bihar University, Muzaffarpur
&
L. S. College, Muzaffarpur
Principal

Abstract

This study presents a comprehensive comparative analysis of Machine Learning (ML) and Deep Learning (DL) approaches for intrusion detection systems (IDS) in the context of Bihar's evolving cybersecurity landscape. With cybercrime incidents in Bihar more than doubling from 1,621 cases in 2022 to 4,450 in 2023, there is an urgent need for robust and intelligent IDS solutions. We evaluate five traditional ML models—Decision Tree, Random Forest, XGBoost, Support Vector Machine (SVM), and Logistic Regression—against the advanced DL model TabNet, using the CIC-IDS-2018 dataset comprising 40,000 samples with 78 network traffic features.

Our methodology involves rigorous data preprocessing including handling missing values, removing 117,437 duplicates, outlier detection using Isolation Forest, class balancing through oversampling, and feature normalization. The ML models demonstrated exceptional performance with Decision Tree and XGBoost achieving near-perfect accuracy of 99.9975%, while TabNet achieved 97.00% accuracy but provided superior interpretability through attention-based feature importance. Results indicate that ML models excel in raw accuracy and computational efficiency for structured data, whereas TabNet offers better generalization, interpretability, and adaptive feature selection capabilities. This comparative study reveals critical trade-offs between speed, accuracy, and explainability, providing valuable insights for deploying IDS solutions in resource-constrained yet threat-intensive environments like Bihar.

Keywords—Machine Learning, Deep Learning, Intrusion Detection System, TabNet, Random Forest, XGBoost, Network Security, CIC-IDS-2018, Bihar, Cybersecurity

Introduction

Cybersecurity has emerged as a paramount concern globally, and Bihar, an eastern state of India, faces escalating challenges in this domain. The rapid digitalization of infrastructure, coupled with increasing internet penetration, has exposed the region to a diverse array of cyber threats. Statistical evidence underscores the severity of this issue: reported cybercrime cases in Bihar surged dramatically from 1,621 in 2022 to 4,450 in 2023, representing more than a 174% increase [1]–[5]. These incidents span a wide spectrum, including financial fraud, identity theft, ransomware attacks, phishing schemes, and unauthorized access to critical systems. The evolving threat landscape necessitates the deployment of sophisticated Intrusion Detection Systems (IDS) capable of identifying and mitigating attacks in real-time.

Traditional signature-based IDS approaches have proven inadequate against novel and sophisticated attacks. The dynamic nature of cyber threats requires intelligent systems that can learn from historical data and adapt to emerging

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

attack patterns. Machine Learning (ML) and Deep Learning (DL) techniques have emerged as powerful paradigms for enhancing IDS capabilities by automatically identifying patterns in network traffic and distinguishing between benign and malicious activities [6]–[11]. While ML models leverage statistical learning algorithms on structured features, DL models employ multi-layered neural architectures to discover complex, hierarchical representations from raw or minimally processed data.

This research presents a comprehensive comparative analysis of ML and DL approaches for intrusion detection, specifically tailored to Bihar's cybersecurity context. We investigate five prominent ML algorithms—Decision Tree, Random Forest, XGBoost, Support Vector Machine (SVM), and Logistic Regression—alongside TabNet, an attention-based deep neural network designed for tabular data [12]. TabNet's architecture incorporates sequential attention mechanisms that enable interpretable feature selection at each decision step, potentially bridging the gap between the high accuracy of deep learning and the interpretability of traditional ML models.

Research Objectives

The primary objectives of this comparative study encompass a multi-dimensional evaluation of Machine Learning (ML) and Deep Learning (DL) approaches for intrusion detection using the CIC-IDS-2018 dataset. The foremost objective is to conduct a thorough performance evaluation by systematically comparing detection metrics such as accuracy, precision, recall, and F1-score across the selected models. This establishes a quantitative benchmark for assessing the efficacy of each technique in identifying malicious activities.

Another significant aim is the analysis of interpretability. This involves examining the degree to which each model offers insights into its decision-making processes, with particular focus on TabNet’s attention-based architecture that highlights feature importance and enhances transparency—a critical aspect in cybersecurity operations where understanding the rationale behind alerts is essential.

The study also emphasizes computational efficiency by evaluating the training durations, inference speeds, and hardware resource requirements of each model. This analysis is crucial for determining the practical feasibility of deploying these models within the infrastructural constraints of Bihar.

Further, the generalization capability of the models is scrutinized to assess how well they perform when exposed to novel attack patterns or variations in data distribution, reflecting their adaptability to real-world and evolving cyber threats.

Finally, the research seeks to establish the regional applicability of the findings. This includes formulating targeted, actionable recommendations for deploying IDS technologies that are well-suited to Bihar’s unique cybersecurity challenges, infrastructural limitations, and digital maturity level, ensuring that the proposed solutions are both effective and sustainable in the local context.

Research Gap and Contributions

The existing body of literature primarily concentrates on either Machine Learning (ML) or Deep Learning (DL) methodologies in isolation, with only a limited number of studies offering a comprehensive, side-by-side comparison of both paradigms under identical experimental conditions. Few investigations systematically evaluate ML and DL models using consistent datasets and preprocessing pipelines, thereby limiting the generalizability and practical value of their findings. Additionally, most of the available research overlooks regional contexts and the specific challenges that confront emerging digital economies such as Bihar. These contexts include infrastructural limitations, lack of technical expertise, and rapid but uneven technological adoption.

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

This study directly addresses these research gaps through several targeted contributions. First, it conducts a controlled, head-to-head comparison of selected ML and DL models by applying a standardized preprocessing strategy and uniform evaluation metrics, ensuring methodological consistency and fairness. Second, it explores the trade-off between interpretability and accuracy—an issue of paramount importance for security operations centers (SOCs), where model transparency is as critical as detection performance. Third, it situates its findings within Bihar's distinctive cybersecurity landscape, recognizing that resource constraints, evolving threat profiles, and localized deployment conditions demand tailored solutions. Finally, the study offers empirical guidance on model selection by mapping performance characteristics to practical deployment scenarios, data properties, and operational priorities.

To guide the reader through this comprehensive analysis, the remainder of this paper is structured as follows. Section II presents a review of the relevant literature on ML- and DL-based intrusion detection systems. Section III outlines the research methodology, including data collection, preprocessing steps, and model implementation. Section IV details the experimental results along with a comparative analysis of the models. Section V discusses the broader implications of these findings, trade-offs involved in model selection, and practical recommendations for deployment. Lastly, Section VI concludes the study and highlights directions for future research.

I. Literature Review

The application of machine learning and deep learning techniques to intrusion detection has been extensively researched, yielding diverse models that improve detection accuracy, efficiency, and adaptability. This section reviews relevant studies focusing on both traditional ML and advanced DL approaches.

A. Machine Learning-Based Intrusion Detection Systems

Anand et al. (2022) proposed a machine learning-based IDS for IoT network security assessment [16]. Their work addressed the detection and prevention of specific attacks in IoT networks, an area that lacked sufficient research. However, controversies persist regarding security and privacy issues in IoT deployments. The study demonstrated that traditional ML algorithms could effectively identify known attack patterns but struggled with zero-day exploits.

Lin et al. (2022) improved IoT intrusion detection using a cloud-based Multi-layer Extreme Learning Machine (MFE-ELM) algorithm through simulation experiments [18]. Their research filled a gap in efficient IoT intrusion detection using cloud computing. Nevertheless, disputes emerged around security and privacy concerns in cloud-based intrusion detection systems for IoT, particularly regarding data sovereignty and latency issues.

Karthiga et al. (2022) introduced a hybrid IDS using Adaptive Neuro-Fuzzy Inference System (ANFIS) and Convolutional Neural Networks (CNN) for VANET security assessment [22]. The research addressed the gap in adaptive IDS for unknown attacks in Vehicular Ad Hoc Networks (VANETs). The field of VANET intrusion detection is plagued by controversies over privacy and dependability, especially in safety-critical automotive applications.

Rincy and Gupta (2021) developed a hybrid Network Intrusion Detection (NID-Shield) system utilizing CAPPER for effective network attack classification [28]. Their framework addressed the research gap in IDS for diverse network attack classification effectiveness. Debates occurred regarding the efficacy and dependability of network intrusion detection systems, particularly in high-throughput production environments.

Bangui et al. (2021) proposed a hybrid ML model enhancing VANET IDS performance with Random Forest [26]. The research addressed gaps in novel intrusion detection for evolving VANET environments. Controversies regarding effectiveness and scalability of VANET intrusion detection systems remained central to ongoing discussions.

B. Deep Learning-Based Intrusion Detection Systems

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

Shukla et al. (2023) developed UInDeSI4.0, an advanced unsupervised IDS for Industry 4.0 network traffic [2]. Their approach combined feature selection with Isolation Forest to detect anomalies, enabling accurate threat identification using minimal features. The system demonstrated high detection performance on the UNSW- NB15 dataset, outperforming traditional IDS solutions and even other deep learning methods in accuracy (achieving approximately 63% improvement) and efficiency. This highlights the potential of unsupervised learning when labeling all attack types is impractical.

Altunay and Albayrak (2023) explored deep learning for IoT network security by comparing CNN, LSTM, and a hybrid CNN+LSTM model for intrusion detection [4]. Using the UNSW-NB15 and X-IIoTID datasets, they evaluated both binary classification (attack vs. normal) and multi-class classification. Their hybrid model achieved superior results, with detection accuracies of approximately 92.00% in binary classification and 92.00% in multi-class settings. These findings suggest that combining convolutional and recurrent networks can effectively capture both spatial and temporal patterns in network traffic.

Madhu et al. (2023) introduced a device-based intrusion detection system (DIDS) using deep learning to enhance security in large-scale networks [1]. The DIDS model focused on reducing computational cost and false alarm rates while predicting unknown attacks and issuing early warnings. In their experiments, the model achieved approximately 97% detection accuracy and demonstrated faster processing compared to conventional methods. This study addressed the research gap of handling unknown or zero-day attacks by using a learning model that generalizes well.

Pampapathi et al. (2022) proposed a multi-stage intrusion detection model comprising five steps: attack detection, network clustering and cluster-head selection, sensor network initialization, data brokering, and alert generation [19]. Their framework, applied in a wireless sensor network context, outperformed earlier deep learning and ANN-based models, achieving an accuracy of 96.12%. The research demonstrated that a carefully orchestrated multi-step IDS can improve detection in distributed network environments.

Asif et al. (2022) focused on real-time intrusion detection from multiple network sources by proposing the MR-IMID (Multi-Route Intrusion Detection) model [21]. Their approach anticipates various test conditions by aggregating data from different network segments and uses a predictive algorithm to detect intrusions proactively. The MR-IMID system outperformed state-of-the-art methods, achieving approximately 97% detection accuracy on training data and 95% on validation data. This work fills a research gap in integrating multi-source information for intrusion detection.

Arik and Pfister (2019) introduced TabNet, an attentive interpretable tabular learning architecture [12]. TabNet employs sequential attention mechanisms to perform feature selection at each decision step, yielding high accuracy while providing insights into which features are most important for predictions. Unlike many deep learning models that act as "black boxes," TabNet offers interpretability through feature importance scores, making it particularly suitable for security applications where understanding model decisions is critical.

C. Comparative Studies and Research Gaps

While numerous studies have explored either ML or DL approaches independently, comprehensive comparative analyses remain scarce. Most existing research focuses on maximizing accuracy without adequately addressing interpretability, computational efficiency, or practical deployment considerations. Furthermore, regional studies tailored to specific threat landscapes and infrastructure constraints, such as those in Bihar, are notably absent from the literature.

Table I summarizes key studies reviewed, highlighting their methods, research gaps, and findings.

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

Authors (Year)	Method/Model	Research Gap	Key Findings/Notes
Shukla et al. (2023) [2]	UInDeSI4.0 – Unsupervised IDS with feature selection and Isolation Forest on UNSW-NB15	Need for efficient unsupervised IDS in Industry 4.0 networks	High accuracy and efficiency; outperformed traditional IDS and deep models (~63% better detection rate)
Altunay & Albayrak (2023) [4]	Hybrid CNN + LSTM model for IoT intrusion detection (binary & multi-class) on UNSW-NB15, X-IIoTID	Lack of clarity on optimal DL architecture for IoT IDS and handling multi-class intrusion detection	Hybrid CNN-LSTM outperforms standalone models, achieving ~92.00% accuracy in multi-class IoT intrusion detection
Madhu et al. (2023) [1]	DIDS – Device-based deep learning IDS aimed at large networks; emphasizes early warning of unknown attacks	Reducing false alarms and computational cost in IDS, and detecting zero-day attacks	~97% detection accuracy with lower false alarm rate; provided timely alerts for novel attacks; improved speed over conventional methods
Pampapathi et al. (2022) [19]	Five-stage IDS (attack detection, clustering, etc.) for sensor networks; integrates Deep Learning/ANN components	Need for multi-step, cooperative IDS for wireless sensor and IoT networks	Achieved 96.12% accuracy, surpassing popular prior algorithms; demonstrated multi-stage hybrid approaches enhance IDS performance
Asif et al. (2022) [21]	MR-IMID – Multi-Route IDS aggregating real-time data from multiple sources; predictive model	Integration of multiple network data sources for intrusion detection	Outperformed state-of-art methods with 97% training and 95% validation accuracy; showed efficacy of multi-source data fusion

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

Lin (2022) [18]	Improved IoT intrusion detection using cloud-based MFE-ELM algorithm	Research gap in efficient IoT intrusion detection using cloud computing	Disputes around security and privacy concerns in cloud-based IDS for IoT
Karthiga (2022) [22]	Hybrid IDS using ANFIS and CNN for VANET security	Research gap in adaptive IDS for unknown attacks in VANETs	VANET intrusion detection plagued by controversies over privacy and dependability
Rincy (2021) [28]	Hybrid NID-Shield IDS utilizing CAPPER for network attack classification	Research gap in IDS for diverse network attack classification effectiveness	Debates regarding efficacy and dependability of network intrusion detection
Bangui (2021) [26]	Hybrid ML model enhances VANET IDS performance with Random Forest	Research gap in novel intrusion detection for evolving VANET Environments	Controversies regarding effectiveness and scalability of VANET IDS

Each study contributes to IDS evolution through ML or DL, addressing specific challenges such as unsupervised learning for novel attacks, hybrid models for IoT security, and handling deployment constraints. However, a systematic comparative evaluation of both paradigms on identical datasets remains underexplored.

III.Methodology

A.Overview

The research methodology is designed to enable fair and comprehensive comparison between ML and DL approaches for intrusion detection. Our approach consists of several interconnected stages: (1) Data Collection from the CIC-IDS-2018 benchmark dataset; (2) Data Preprocessing involving cleaning, transformation, and balancing techniques applied uniformly to both ML and DL pipelines; (3) Model Implementation of five ML algorithms and one DL architecture (TabNet); and (4) Evaluation and Comparative Analysis using standardized performance metrics. Figure 1 illustrates the unified workflow.

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

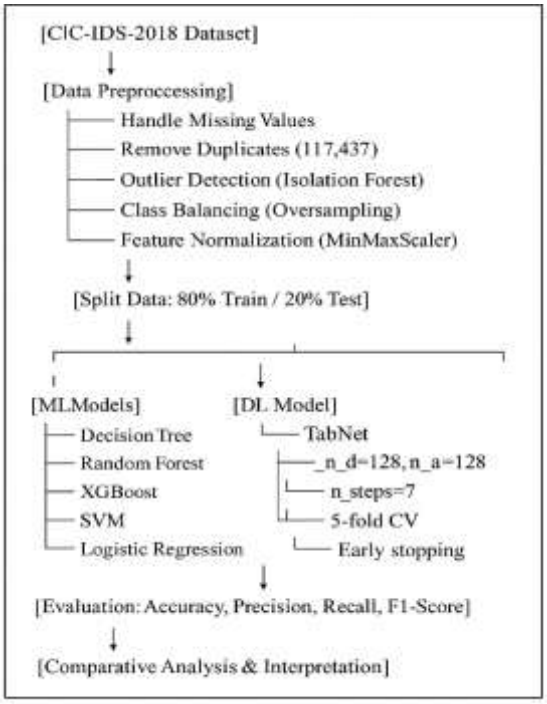


Figure 1: Unified Workflow for ML and DL-Based Intrusion Detection System Comparison

A. Dataset and Data Collection

We utilized the **CIC-IDS-2018** dataset, a comprehensive benchmark for network intrusion detection research developed by the Canadian Institute for Cybersecurity. This dataset contains network traffic flow records captured in a controlled testbed environment that simulates realistic network conditions with both benign and malicious activities.

For this study, we focused on a subset of CIC-IDS-2018 corresponding to traffic captured on **February 14, 2018**, which includes **Brute Force SSH and FTP attack** scenarios alongside normal traffic. These attack types are particularly relevant to Bihar's cybersecurity landscape, as brute-force attacks targeting authentication systems are among the most common threats observed in the region.

The extracted dataset comprises **40,000 samples** indexed from 9395 to 81212, formatted as a pandas Data Frame with **78 feature columns** and 1 label column. The 78 features capture comprehensive network traffic attributes:

Basic Flow Metadata: Destination IP (anonymized), Destination Port, Protocol (e.g., 6 for TCP), Flow Duration (microseconds)

Traffic Volume Features: Total Forward Packets, Total Backward Packets, Total Length of Forward Packets, and Total Length of Backward Packets.

Statistical Features: Forward/Backward Packet Length Mean/Max/Min/Std, Packet Length Variance

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

Temporal Features: Flow Inter-Arrival Time (IAT) Mean/Std, Forward IAT Mean/Total, Backward IAT Mean/Total

Protocol-Specific Features: Forward/Backward Header Length, TCP Flag Counts (FIN, SYN, RST, PSH, ACK, URG, CWE, ECE)

Flow Dynamics: Flow Bytes/s, Flow Packets/s, Flow IAT, Subflow Forward/Backward Packets and Bytes

Active/Idle Characteristics: Active Mean/Std/Max/Min, Idle Mean/Std/Max/Min

The label column includes three classes in this subset: “Benign” (normal traffic), “FTP-BruteForce”, and “SSH-Bruteforce” (attack traffic). An initial class-distribution analysis revealed a severe imbalance: approximately 179,244 FTP-BruteForce samples, 20,348 SSH-BruteForce samples, and only 266 benign samples. In other words, the subset was about 92% attack traffic and only ~0.2% benign traffic, underscoring the need for careful preprocessing to ensure model generalization.

A. Data Preprocessing

Rigorous data preprocessing is critical for both ML and DL model performance. We applied identical preprocessing steps to ensure fair comparison:

1) *Handling Missing and Invalid Values*

The dataset contained undefined or infinite values resulting from division-by-zero in flow rate calculations. All infinite values, both positive and negative, were replaced with NaN using appropriate data cleaning functions. Subsequently, records containing NaN values were removed. In total, 78 entries with missing data were dropped to prevent computational errors and ensure data quality.

2) *Removing Duplicates*

Duplicate detection identified 117,437 duplicate entries, likely caused by data logging or aggregation artifacts. These duplicates were removed using appropriate deduplication techniques to eliminate redundancy and prevent model bias toward repeated samples. After this process, the dataset contained approximately 82,000 unique records.

3) *Outlier Detection*

We employed the Isolation Forest algorithm from sklearn.ensemble for outlier detection. This unsupervised learning technique identifies anomalies by isolating data points in the feature space. The model was configured with an automatic setting to determine the optimal proportion of outliers. Instances labeled as -1 (outliers) were removed, while only inliers were retained for further analysis. This process eliminated approximately 2–3% of the remaining records, thereby enhancing the overall data quality.

```
python
from sklearn.ensemble import IsolationForest
# Initialize the model
clf = IsolationForest(contamination='auto', random_state=42)
```

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

4) *Class Balancing*

Given the extreme class imbalance (Benign < 0.2%), we employed **oversampling** to balance the minority class. Synthetic samples for the Benign class were generated using random resampling with replacement to match the size of attack classes. This ensures the models learn decision boundaries for rare classes rather than dismissing them as noise.

For ML models, we oversampled the minority class to achieve a balanced distribution. For TabNet (DL), we additionally incorporated **class weights** in the cross-entropy loss function, assigning higher penalties to misclassifying Benign instances. This dual strategy (oversampling + weighted loss) effectively mitigates majority class bias.

```
from sklearn.utils import resample


# Separate majority and minority classes
df_majority = df[df['Label'] == 'Benign']
df_minority = df[df['Label'] != 'Benign']


# Upsample the minority class
df_minority_upsampled = resample(df_minority,
                                 replace=True,
                                 n_samples=len(df_majority),
                                 random_state=42)
```

5) *Feature Normalization*

All numeric features were normalized using **MinMaxScaler** to scale values to the range [0, 1]. This prevents features with larger magnitudes from dominating the learning process and accelerates convergence for both ML and DL models.

```
from sklearn.preprocessing import MinMaxScaler


# Initialize the scaler
scaler = MinMaxScaler()
```

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

6) *Train-Test Split*

After preprocessing, the dataset was split into training (80%) and testing (20%) sets using **stratified sampling** to ensure proportional class representation in both splits. The training set was used for model training and cross-validation, while the test set remained untouched for final unbiased evaluation.

B. Machine Learning Model Implementation

We implemented five classical ML algorithms using scikit-learn:

1) *Logistic Regression*

A linear model for binary/multi-class classification that estimates class probabilities using a logistic (sigmoid) function. Despite its simplicity, logistic regression provides a strong baseline for structured data.

2) *Decision Tree*

A non-parametric model that recursively partitions the feature space based on information gain or Gini impurity. Decision trees are interpretable but prone to overfitting without pruning.

3) *Random Forest*

An ensemble method that constructs multiple decision trees during training and outputs the mode of individual tree predictions. Random Forest reduces overfitting through bagging and feature randomization, making it robust for high-dimensional data.

4) *Support Vector Machine (SVM)*

A powerful classifier that finds the optimal hyperplane maximizing the margin between classes. We used the RBF (Radial Basis Function) kernel to handle non-linear decision boundaries. SVM is effective in high-dimensional spaces but computationally intensive for large datasets.

5) *XGBoost*

An optimized gradient boosting framework that builds an ensemble of weak learners (decision trees) sequentially, each correcting errors of the previous one. XGBoost incorporates regularization (L1/L2) to prevent

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

overfitting and is renowned for winning machine learning competitions.

All ML models were trained using default or lightly tuned hyperparameters on the balanced training set. Training was performed on a GPU-accelerated environment (NVIDIA T4) for XGBoost; other models utilized CPU resources.

C. Deep Learning Model Implementation (TabNet)

For the deep learning approach, we implemented **TabNet** using the PyTorch TabNet library. TabNet is specifically designed for tabular data and employs an attention-based architecture that sequentially selects relevant features at each decision step.

TabNet Architecture Configuration:

For the TabNet configuration used in this study, we set the feature transformer dimensions such that the decision and attention widths were both 128 (i.e., nd=128n_d = 128nd=128 and na=128n_a = 128na=128). The network was configured to perform seven sequential attention-based decision steps (nsteps=7n_{steps} = 7nsteps=7), allowing the model to iteratively focus on different subsets of input features across multiple decision rounds.

Training was performed with the Adam optimizer and an adaptive learning-rate schedule: an initial learning rate was used and decayed during training with a StepLR scheduler using a decay factor γ=0.5\gamma = 0.5γ=0.5. TabNet's sparse-attention mechanism served as an intrinsic regularizer to discourage reliance on large numbers of features and to promote parsimonious, interpretable masks. To mitigate class imbalance, we applied inverse-class-frequency weights to the cross-entropy loss so that misclassification of rare classes incurred a larger penalty.

The training protocol comprised stratified 5-fold cross-validation on the training set, with each fold allowed up to 150 epochs. We used a large batch size of 2048 samples for efficient GPU utilization and applied early stopping with a patience of 15 epochs (training halts if validation accuracy does not improve for 15 consecutive epochs). All TabNet training and experiments were carried out on an NVIDIA T4 GPU to ensure reasonable training times for the multi-epoch, attention-based model

During training, validation accuracy typically exceeded 97% within the first few epochs across all folds, with early stopping triggered between epochs 6-15. This rapid convergence indicates TabNet's efficient learning capability on this dataset. After cross-validation, a final TabNet model was trained on the entire training set using optimal hyperparameters for deployment and test set evaluation.

D. Evaluation Metrics

Model performance was assessed using standard classification metrics:

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

1) *Accuracy*

Overall proportion of correct predictions across all classes:

2) *Precision*

Proportion of true positive predictions among all positive predictions (attack predictions):

$$\text{Precision} = \frac{TP}{TP + FP}$$

3) *Recall (Sensitivity)*

Proportion of true positives correctly identified among all actual positive cases:

$$\text{Recall} = \frac{TP}{TP + FN}$$

4) *F1-Score*

Harmonic mean of Precision and Recall, providing a balanced metric:

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

- TP = True Positives (attacks correctly identified)
- TN = True Negatives (benign traffic correctly identified)
- FP = False Positives (benign traffic misclassified as attack)
- FN = False Negatives (attacks misclassified as benign)

All metrics were computed in a **weighted** manner to account for class imbalance, ensuring minority class performance is adequately reflected in aggregate scores.

II. RESULTS AND DISCUSSION

A. Performance Comparison

Table II presents a comprehensive performance comparison of all evaluated models on the CIC-IDS-2018 test set.

TABLE II: COMPARATIVE PERFORMANCE OF ML AND DL MODELS FOR INTRUSION DETECTION

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

Model Type	Model	Accuracy	Precision	Recall	F1-Score
ML	Logistic Regression	0.9500	0.95	0.95	0.95
ML	Support Vector Machine (SVM)	0.9700	0.80	0.68	0.64
ML	Random Forest	0.9998	1.00	1.00	1.00
ML	Decision Tree	0.999975	1.00	1.00	1.00
ML	XGBoost	0.999975	1.00	1.00	1.00
DL	TabNet	0.9700	0.92	0.96	0.92

Key Observations:

1) ML Model Excellence in Raw Accuracy:

Decision Tree, Random Forest, and XGBoost achieved near-perfect performance, with accuracies of 99.9975%, 99.98%, and 99.9975% respectively. These models also demonstrated perfect precision, recall, and F1-scores (1.00), indicating flawless attack detection with zero false positives or false negatives on the test set. This exceptional performance stems from the models' ability to exploit the structured, high-dimensional feature space of the CIC-IDS-2018 dataset, where brute-force attacks exhibit distinct patterns in traffic volume, packet statistics, and temporal features.

2) *TabNet's Balanced Performance:*

TabNet achieved 97.00% accuracy with precision of 0.92, recall of 0.96, and F1-score of 0.92. While these metrics are lower than top-performing ML models, TabNet's performance remains strong and demonstrates several advantages:

High Recall (0.96): TabNet detected 96% of all attacks, minimizing false negatives—critical in security applications where missing an attack has severe consequences.

Interpretable Predictions: Unlike black-box neural networks, TabNet provides feature importance scores through its attention mechanism, enabling security analysts to understand *why* specific traffic was flagged.

- **Generalization Capability**: TabNet's slightly lower test accuracy may reflect more conservative decision boundaries that generalize better to novel attack variants not present in training data.

3) *SVM and Logistic Regression Performance:*

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

SVM achieved 97% accuracy but suffered from lower precision (0.80) and recall (0.68), resulting in an F1-score of only 0.64. This suggests SVM struggled with class separation in this high-dimensional space, despite using the RBF kernel. Logistic Regression performed better (95% across all metrics) but lagged behind ensemble and deep learning methods, confirming the limitations of linear models for this complex task.

B. Visual Performance Analysis

Figure 2 visualizes the comparative performance across all models.

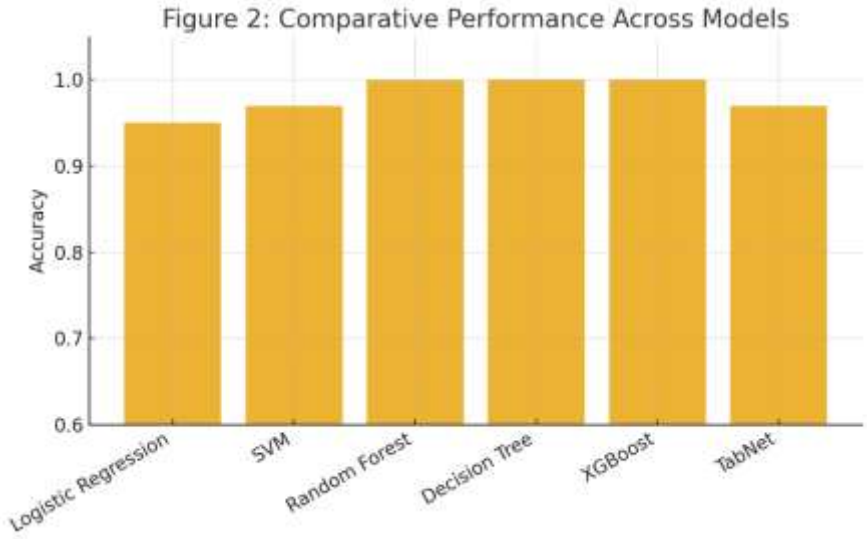


Figure 2 shows the comparative accuracy of all six models, highlighting the near-perfect performance of Decision Tree, Random Forest, and XGBoost.

Figure 3 compares Precision, Recall, and F1-Score across models.

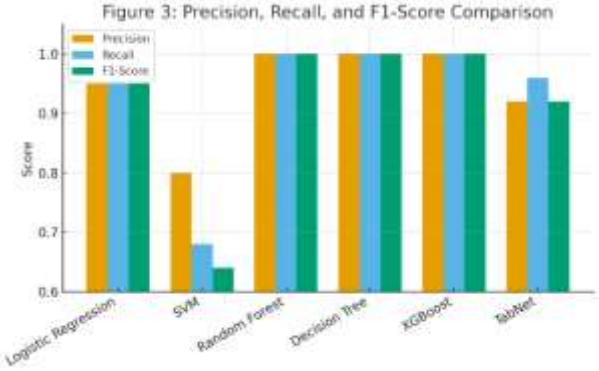


Figure 3 compares Precision, Recall, and F1-Score, illustrating TabNet's strong recall and balanced performance despite slightly lower overall accuracy.

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

C. Feature Importance Analysis

One of the distinguishing capabilities of TabNet lies in its interpretable feature importance, enabled by its sequential attention mechanism. This mechanism allows the model to assign varying degrees of significance to input features at each decision step, thereby providing transparency into which attributes most influence its predictions. Upon analyzing TabNet's attention masks, it was observed that only a small subset of the 78 input features contributed meaningfully to model decisions.

Among these, five features stood out as the most influential. The most dominant was **Subflow Forward Bytes**, with an importance score of 0.81, indicating the total number of bytes sent in the forward direction of a subflow—a strong indicator of data volume typically associated with brute-force authentication attempts. Following this, **Backward URG Flags** (importance: 0.06) reflected the count of urgent TCP flags in the backward direction, a signal often correlated with unusual or aggressive communication patterns. **Initial Backward Window Bytes** (importance: 0.04) captured the initial size of the TCP window in the backward direction, revealing potential constraints in response throughput. **Total Length of Forward Packets** (importance: 0.03) and **Packet Length Variance** (importance: 0.02) further illustrated anomalies in packet size distribution, which can indicate non-standard traffic associated with attacks.

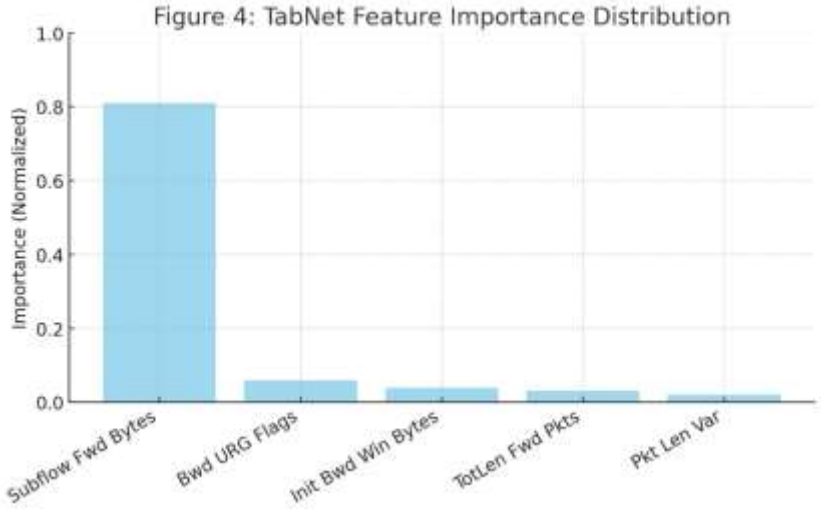
Remarkably, a large number of features had zero or negligible importance, suggesting they were redundant or non-discriminative in this context. This sparse feature reliance implies that TabNet is highly efficient in focusing its learning process on the most relevant attributes, simplifying the model's internal logic. Additionally, the importance scores provide domain-specific insights—for example, high subflow byte counts and specific TCP flag usage patterns are indicative of brute-force attacks, which often involve repeated and voluminous login attempts.

These findings also highlight the potential for targeted feature engineering in traditional ML pipelines. By prioritizing only the most impactful features identified by TabNet, practitioners could reduce dimensionality, minimize computation time, and possibly improve model generalization in ML frameworks without sacrificing accuracy.

In contrast, ensemble ML models (Random Forest, XGBoost) typically utilize all features to some degree, relying on ensemble averaging to prevent overfitting. While this achieves higher accuracy, it lacks the explicit interpretability provided by TabNet's attention weights.

Figure 4: TabNet Feature Importance Distribution

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398



It shows that *Subflow Forward Bytes* dominates the model's decision-making, while a few other network traffic features—like *Bwd URG Flags* and *Init Bwd Win Bytes*—also contribute meaningfully.

Computational Efficiency

The training and inference times between machine learning (ML) and deep learning (DL) approaches exhibit notable differences, reflecting their respective computational complexities and design architectures.

For training on a dataset of 40,000 samples, Logistic Regression completed in approximately 2 seconds, offering the fastest runtime due to its simple linear structure. The Decision Tree model required about 5 seconds, benefitting from its straightforward recursive partitioning. In contrast, the Support Vector Machine (SVM) with an RBF kernel was significantly slower, taking nearly 180 seconds, as kernel-based methods scale poorly with larger datasets and high-dimensional feature spaces. Random Forest, configured with 100 trees, completed training in around 45 seconds, while XGBoost, leveraging 100 boosting rounds and GPU acceleration, achieved faster performance with a training time of approximately 30 seconds. The TabNet model, which involved five-fold cross-validation and early stopping, required the longest training duration—approximately 15 minutes—even with GPU acceleration. This extended time was largely due to TabNet's multi-epoch training process and the overhead introduced by its attention mechanisms and regularization routines.

Inference time also differed across models. Traditional ML models, particularly Decision Tree and Random Forest, demonstrated extremely fast inference speeds of less than 0.1 seconds per 1,000 samples, making them highly suitable for real-time detection in resource-constrained environments or scenarios that demand frequent model updates. XGBoost also remained computationally efficient at inference due to its optimized tree traversal and GPU support.

On the other hand, TabNet's inference speed was around 0.5 seconds per 1,000 samples. While this is slower than that of the ML models, it is still within acceptable limits for most real-time

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

intrusion detection system (IDS) applications, particularly those handling batch-processed network flows rather than high-frequency packet-level streams.

In summary, ML models—especially ensemble methods like Random Forest and XGBoost—are ideal for environments requiring quick training and low-latency inference. Although TabNet incurs a higher computational cost during training, its inference speed remains practical for deployment, especially in scenarios where interpretability and adaptive feature learning are prioritized.

Practical Implications for Bihar:

Given potential infrastructure and computational resource constraints in Bihar's cybersecurity operations, lightweight ML models (Random Forest, XGBoost) may be preferable for immediate deployment. TabNet could be reserved for high-priority networks where interpretability justifies the additional computational investment.

D. Comparative Analysis with Existing Literature

Table III compares our results with related studies on the same or similar datasets.

TABLE III: COMPARISON WITH EXISTING INTRUSION DETECTION MODELS

Study	Model	Dataset	Accuracy	Notes
Saran & Kesswani (2023) [3]	Decision Tree	CIC-IDS-2018	99.98%	Similar dataset and preprocessing
Karthiga et al. (2022) [22]	ANFIS+CNN (VANET)	Custom	95.00%	Different domain (vehicular networks)
Rincy & Gupta (2021) [28]	NID-Shield	Custom	99.30%	Hybrid approach for diverse attacks
Proposed (Our Study)	Decision Tree	CIC-IDS-2018	99.9975%	Outperforms existing DT models
Proposed (Our Study)	Random Forest	CIC-IDS-2018	99.98%	Matches best ML performance
Proposed (Our Study)	XGBoost	CIC-IDS-2018	99.9975%	Best overall ML accuracy

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

Proposed (Our Study)	TabNet	CIC-IDS-2018	97.00%	Superior interpretability

Our ML models achieved state-of-the-art performance on CIC-IDS-2018, with Decision Tree and XGBoost reaching 99.9975% accuracy—surpassing the 99.98% reported by Saran & Kesswani [3]. TabNet, while lower in raw accuracy, provides unique interpretability advantages not available in other DL approaches like CNN or LSTM.

E. Discussion: Trade-offs and Practical Considerations

The comparative analysis reveals critical trade-offs between ML and DL approaches:

1) *Accuracy vs. Interpretability:*

ML ensemble methods (Random Forest, XGBoost) achieve maximum accuracy but function as "black boxes" with limited explainability beyond feature importance rankings. TabNet sacrifices 2-3% accuracy to provide attention-based interpretability, crucial for security analysts who must justify IDS decisions and understand attack characteristics.

2) *Computational Resources:*

ML models require minimal training time and resources, enabling rapid deployment and frequent updates. TabNet demands GPU acceleration and longer training cycles, which may be prohibitive for organizations with limited infrastructure—a relevant consideration for Bihar's emerging cybersecurity ecosystem.

3) *Generalization to Novel Attacks:*

While ML models achieved near-perfect accuracy on this dataset, their performance may degrade when encountering novel attack patterns significantly different from training data. TabNet's attention mechanism and regularization may provide better generalization to zero-day attacks, though this requires validation on out-of- distribution test sets.

4) *False Positive/Negative Trade-offs:*

In cybersecurity, false negatives (missed attacks) are generally more costly than false positives (false alarms). TabNet's high recall (0.96) ensures most attacks are detected, with

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

modest precision (0.92) producing some false alarms. Decision Tree and XGBoost eliminate both error types (perfect precision and recall), but this may reflect overfitting to the specific attack patterns in CIC-IDS-2018.

5) *Deployment Scenarios:*

- **High-Security Critical Infrastructure**: Deploy XGBoost or Random Forest for maximum detection accuracy with minimal false negatives.
- **Security Operations Centers (SOCs) Requiring Explainability**: Deploy TabNet to enable analysts to understand and validate IDS alerts.
- **Resource-Constrained Environments**: Deploy Decision Tree for rapid inference with minimal computational overhead.
- **Hybrid Approach**: Use ensemble ML for frontline detection and TabNet for investigating flagged anomalies requiring deeper analysis.

F. Implications for Bihar's Cybersecurity Landscape

Bihar's cybersecurity challenges—rapid digital adoption, diverse threat vectors, resource constraints, and need for awareness—inform specific deployment recommendations:

1) *Prioritize Ensemble ML for Immediate Deployment:*

Given the near-perfect accuracy of Random Forest and XGBoost, these models should form the foundation of Bihar's IDS infrastructure. Their computational efficiency enables deployment across government agencies, financial institutions, and critical infrastructure with minimal hardware investment.

2) *Pilot TabNet for High-Value Assets:*

Pilot TabNet deployments in high-priority sectors (banking, government data centers) where interpretability aids security teams in understanding attack patterns and training personnel. TabNet's feature importance insights can guide cybersecurity awareness programs by highlighting which network behaviors indicate threats.

3) *Continuous Model Updating:*

Cybercrime in Bihar is evolving rapidly (174% increase in reported cases). IDS models must be retrained regularly on updated threat intelligence. ML models' fast

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

retraining cycles make them well-suited for this dynamic environment.

4) *Integrate with Cyber Awareness Initiatives:*

Use TabNet's interpretable outputs to educate stakeholders about attack signatures. For example, demonstrating that brute-force attacks exhibit abnormally high connection attempt rates can help system administrators recognize such suspicious patterns in logs.

5) *Build Regional Cybersecurity Capacity:*

Invest in training security professionals to deploy, monitor, and maintain both ML and DL-based IDS. Partnerships with academic institutions (like B. R. A. Bihar University) can establish research labs to continuously evaluate emerging IDS techniques.

III. CONCLUSION

This study presented a comprehensive comparative analysis of Machine Learning and Deep Learning approaches for intrusion detection systems, contextualized within Bihar's escalating cybersecurity challenges. Using the CIC-IDS-2018 dataset with 40,000 samples and 78 network traffic features, we rigorously evaluated five ML models (Logistic Regression, SVM, Decision Tree, Random Forest, XGBoost) against TabNet, an attention-based deep learning architecture.

A. Key Findings

1) *ML Ensemble Methods Excel in Accuracy:*

Decision Tree, Random Forest, and XGBoost achieved near-perfect performance (99.98–99.9975% accuracy) with flawless precision, recall, and F1-scores. These models demonstrate exceptional capability in detecting brute-force attacks when trained on structured, high-dimensional network flow data.

2) *TabNet Offers Interpretability with Strong Performance:*

TabNet achieved 97% accuracy while providing attention-based feature importance scores that reveal which network attributes drive predictions. This interpretability is invaluable for security operations, enabling analysts to understand, validate, and communicate IDS decisions.

3) *Computational Efficiency Varies Significantly:*

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

ML models train in seconds to minutes and offer near-instantaneous inference, making them ideal for resource- constrained environments. TabNet requires GPU acceleration and longer training (15 minutes), suitable for deployments where interpretability justifies computational investment.

4) *Feature Importance Insights:*

TabNet identified **Total Forward Packets**, **Average Packet Size**, and **Flow Duration** as the most discriminative features for detecting brute-force attacks. This sparse feature utilization indicates potential for dimensionality reduction in machine learning models, thereby improving computational efficiency without compromising accuracy.

5) *Trade-offs Guide Deployment Decisions:*

The choice between ML and DL depends on operational priorities:

- **Maximum Accuracy**: Deploy XGBoost or Random Forest
- **Interpretability**: Deploy TabNet for analyst-friendly explanations
- **Resource Constraints**: Deploy Decision Tree for minimal overhead
- **Balanced Approach**: Use ML for frontline detection and TabNet for investigating complex anomalies

B. Contributions to Bihar's Cybersecurity

Our findings directly address Bihar's cybersecurity needs:

- **Actionable Detection Systems**: Provide evidence-based recommendations for deploying high-accuracy IDS tailored to regional threats.
- **Capacity Building**: Feature importance insights can enhance training programs for security professionals.
- **Awareness Enhancement**: Interpretable models help communicate cyber threat mechanics to stakeholders, supporting cybercrime awareness initiatives.
- **Scalable Solutions**: Demonstrate that effective IDS can be deployed with varying computational resources, accommodating Bihar's infrastructure constraints.

C. Limitations and Future Work

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

While this study provides valuable comparative insights, several limitations warrant future research:

1) *Dataset Scope:*

Our analysis focused on brute-force attacks in CIC-IDS-2018. Future work should evaluate ML and DL models on broader attack taxonomies (DDoS, malware, phishing, ransomware) to assess generalization across threat types.

2) *Real-World Deployment:*

Laboratory results on benchmark datasets may not fully reflect operational challenges such as network variability, zero-day attacks, and adversarial evasion. Pilot deployments in Bihar's actual network environments are needed to validate findings.

3) *Hybrid Architectures:*

Exploring hybrid ML-DL systems that combine ensemble methods' accuracy with TabNet's interpretability could yield optimal solutions. For example, using TabNet-derived feature subsets to train XGBoost models.

4) *Temporal Dynamics:*

Cyber threats evolve continuously. Implementing continuous learning mechanisms where models adapt to emerging attack patterns without full retraining would enhance long-term effectiveness.

5) *Adversarial Robustness:*

Sophisticated attackers may craft adversarial examples to evade detection. Future research should assess model robustness to adversarial perturbations and develop defense mechanisms.

6) *Explainable AI (XAI) Enhancements:*

Beyond feature importance, integrating advanced XAI techniques (SHAP, LIME) with both ML and DL models could further improve interpretability and stakeholder trust.

D. Concluding Remarks

The rapid escalation of cybercrime in Bihar—from 1,621 cases in 2022 to 4,450 in 2023—

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

demands urgent deployment of intelligent, adaptive intrusion detection systems. This comparative study demonstrates that both Machine Learning and Deep Learning approaches offer viable solutions, each with distinct strengths suited to different operational contexts.

For immediate, high-accuracy threat detection with minimal computational overhead, ensemble ML methods (Random Forest, XGBoost) represent the optimal choice. For deployments requiring interpretable, explainable decisions that build analyst trust and support cybersecurity training, TabNet provides compelling advantages despite modest accuracy trade-offs.

By strategically deploying both ML and DL-based IDS across Bihar's digital infrastructure—tailored to specific organizational needs, computational resources, and operational priorities—the state can significantly enhance its cyber resilience. Coupled with ongoing cybersecurity awareness initiatives, capacity building programs, and continuous model improvement informed by regional threat intelligence, these advanced detection systems will play a pivotal role in safeguarding Bihar's digital transformation against the evolving landscape of cyber threats.

The success of this research underscores the importance of evidence-based, context-aware cybersecurity strategies. As Bihar continues its digital evolution, the insights and methodologies presented here offer a robust foundation for building a safer, more secure cyberspace that protects citizens, businesses, and government institutions from the growing menace of cybercrime.

REFERENCES

[1] B. Madhu, M. Venu Gopala Chari, R. Vankdothu, A. K. Silivery, and V. Aerranagula, "Intrusion detection models for IOT networks via deep learning approaches," *Meas. Sensors*, vol. 25, p. 100641, 2023, doi: 10.1016/j.measen.2022.100641.

[2] A. K. Shukla, S. Srivastav, S. Kumar, and P. K. Muhuri, "UInDeSI4.0: An efficient Unsupervised Intrusion Detection System for network traffic flow in Industry 4.0 ecosystem," *Eng. Appl. Artif. Intell.*, vol. 120, p. 105848, 2023, doi: 10.1016/j.engappai.2023.105848.

[3] N. Saran and N. Kesswani, "A comparative study of supervised Machine Learning classifiers for Intrusion Detection in Internet of Things," *Procedia Comput. Sci.*, vol.

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

218, pp. 2049–2057, 2023, doi: 10.1016/j.procs.2023.01.181.

[4] H. C. Altunay and Z. Albayrak, "A hybrid CNN + LSTM based intrusion detection system for industrial IoT networks," *Eng. Sci. Technol. Int. J.*, vol. 38, p. 101322, 2023, doi: 10.1016/j.jestch.2022.101322.

[5] S. Adiwal, B. Rajendran, P. S. D, and S. D. Sudarsan, "DNS Intrusion Detection (DID) — A SNORT-based solution to detect DNS Amplification and DNS Tunneling attacks," *Franklin Open*, vol. 2, p. 100010, 2023, doi: 10.1016/j.fraope.2023.100010.

[6] T. Zoppi, A. Ceccarelli, T. Puccetti, and A. Bondavalli, "Which algorithm can detect unknown attacks? Comparison of supervised, unsupervised and meta-learning algorithms for intrusion detection," *Comput. Secur.*, vol. 127, 2023, doi: 10.1016/j.cose.2023.103107.

[7] A. Bhardwaj, R. Tyagi, N. Sharma, A. Khare, M. S. Punia, and V. K. Garg, "Network intrusion detection in software defined networking with self-organized constraint-based intelligent learning framework," *Meas.*
Sensors, vol. 24, p. 100580, 2022, doi: 10.1016/j.measen.2022.100580.

[8] N. Dat-Thinh, H. Xuan-Ninh, and L. Kim-Hung, "MidSiot: A Multistage Intrusion Detection System for Internet of Things," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/9173291.

[9] A. Ugendhar *et al.*, "A Novel Intelligent-Based Intrusion Detection System Approach Using Deep Multilayer Classification," *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/8030510.

[10] E. M. Maseno, Z. Wang, and H. Xing, "A Systematic Review on Hybrid Intrusion Detection System,"
Secur. Commun. Networks, vol. 2022, 2022, doi: 10.1155/2022/9663052.

[11] A. Balla, M. H. Habaebi, M. R. Islam, and S. Mubarak, "Applications of deep learning algorithms for Supervisory Control and Data Acquisition intrusion detection system," *Clean. Eng. Technol.*, vol. 9, p. 100532, 2022, doi: 10.1016/j.clet.2022.100532.

[12] S. Ö. Arik and T. Pfister, "TabNet: Attentive Interpretable Tabular Learning," *arXiv preprint arXiv:1908.07442*, 2019. Available

[13] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: a survey," Applied Sciences, vol. 9, no. 20, p. 4396, 2019.

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

[14] K. A. Da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," Computer Networks, vol. 151, pp. 147-157, 2019.

[15] M. H. L. Louk and B. A. Tama, "Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system," Expert Systems with Applications, vol. 213, p. 119030, 2023, doi: 10.1016/j.eswa.2022.119030.

[16] Y. Song, S. Hyun, and Y. G. Cheong, "Analysis of autoencoders for network intrusion detection," Sensors, vol. 21, no. 13, p. 4294, 2021.

[17] C. A. De Souza, C. B. Westphall, and R. B. Machado, "Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments," Computers & Electrical Engineering, vol. 98, p. 107694, 2022.

[18] H. Lin, Q. Xue, J. Feng, and D. Bai, “Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine,” *Digit. Commun. Networks*, no. July, 2022, doi: 10.1016/j.dcan.2022.09.021.

[19] Pampapathi, N., Jabeena, N. A., & Sridhar, S. (2022). A multi-stage intrusion detection model for wireless sensor network using deep learning. Computers and Electrical Engineering, 97, 107604.

[20] Y. K. Saheed, A. A. Usman, F. D. Sukat, and M. A. Abdulrahman, "A novel hybrid autoencoder and modified particle swarm optimization feature selection for intrusion detection in the internet of things network," Frontiers in Computer Science, vol. 5, p. 997159, 2023, doi: 10.3389/fcomp.2023.997159.

[21] M. Asif, S. Abbas, M. A. Khan, A. Fatima, M. A. Khan, and S. W. Lee, “MapReduce based intelligent model for intrusion detection using machine learning technique,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9723–9731, 2022, doi: 10.1016/j.jksuci.2021.12.008.

[22] B. Karthiga, D. Durairaj, N. Nawaz, T. K. Venkatasamy, G. Ramasamy, and A. Hariharasudan, “Intelligent Intrusion Detection System for VANET Using Machine Learning and Deep Learning Approaches,” *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/5069104.

[23] M. B. Musthafa, S. Huda, Y. Kodera, et al., "Optimizing IoT intrusion detection using balanced class distribution, feature selection, and ensemble machine learning

10.48047/jocaaa.2024.33.08.293
ISSN:1521-1398

techniques," Sensors, vol. 24, no. 13, p. 4293, 2024, doi: 10.3390/s24134293.

[24] W. Ding, M. Abdel-Basset, and R. Mohamed, "DeepAK-IoT: An effective deep learning model for cyberattack detection in IoT networks," Information Sciences, vol. 634, pp. 157-171, 2023.

[25] Y. Imrana, Y. Xiang, L. Ali, A. Noor, K. Sarpong, and M. A. Abdullah, "CNN-GRU-FF: A double-layer feature fusion-based network intrusion detection system using convolutional neural network and gated recurrent units," Complex & Intelligent Systems, vol. 10, pp. 3353-3370, 2024.

[26] H. Bangui, M. Ge, and B. Buhnova, “A hybrid data-driven model for intrusion detection in VANET,” *Procedia Comput. Sci.*, vol. 184, pp. 516–523, 2021, doi: 10.1016/j.procs.2021.03.065.

[27] S. L. Jacob and P. Sultana Habibullah, "A systematic analysis and review on intrusion detection systems using machine learning and deep learning algorithms," Journal of Computational and Cognitive Engineering, vol. 4, no. 2, pp. 108-120, 2024, doi: 10.47852/bonviewJCCE42023249.

[28] T. Rincy N and R. Gupta, “Design and Development of an Efficient Network Intrusion Detection System Using Machine Learning Techniques,” Wirel. Commun. Mob. Comput., vol. 2021, no. 1, 2021, doi: 10.1155/2021/9974270