Intrusion Detection and Prevention Using Machine Learning for IoT-based WSN Network

Rajesh * and Mridul Chawla

Department of Electronics and Communication Engineering, Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonipat, Haryana 1310039, India
Email: 19001903010rajesh@dcrustm.org (R.); mridulchawla.ece@dcrustm.org (M.C.)
*Corresponding author

Abstract—**Intrusion Detection Systems (IDS) are essential for securing enterprise and IoT networks against evolving cyber threats. This study proposes a Machine Learning (ML)-based IDS framework that integrates multiple algorithms to improve detection accuracy and resilience. Using the UNSW-NB15 dataset, models including Decision Tree (DT), Random Forest (RF), CatBoost, and hybrid approaches were trained and evaluated for binary classification of network activities. To mitigate performance degradation caused by high-dimensional feature vectors, a Gini Impurity-Based Weighted Random Forest (GIWRF) was employed for feature selection, while Genetic Algorithm (GA)-based feature extraction further enhanced the model's understanding of class distributions. A total of twenty-seven features were selected based on their relevance, optimizing the learning process. Experimental results demonstrate that the hybrid model outperforms individual algorithms, achieving high accuracy in detecting various attacks, including DoS, Probe, and other network intrusions. The proposed GIWRF-Hybrid approach showed superior performance in both accuracy and loss metrics, confirming its practical applicability for real-world IoT security scenarios. The study provides insights into the design of robust ML-based IDS frameworks and underscores the importance of customized strategies and continuous improvements to enhance system resilience against increasingly sophisticated cyber-attacks. These findings contribute to strengthening IoT network defenses by combining feature selection, extraction, and hybrid classification methods within a single integrated approach.**

Keywords—**Intrusion Detection System (IDS), Machine Learning (ML), IoT, Wireless Sensor Network (WSN), Genetic Algorithm (GA), Gini Impurity-based Weighted Random Forest (GIWRF)**

I. Introduction

Driven by the increasing pace of advancement of communication technologies, Internet services, and the increasing range of network applications, network Security has emerged as a critical issue requiring effective solutions. To safeguard networks, various defense mechanisms are employed; including Intrusion Detection Systems (IDSs), firewalls, authentication techniques, and cryptographic methods [1]. IDSs examine network traffic to detect abnormal behavior and malicious digital attack [2]. An IDS alerts network administrators through notifications when suspicious activity is detected within the network. Suitable countermeasures are then employed to halt ongoing attacks and prevent future cyber-attacks [3]. Recently, machine learning approaches have demonstrated strong effectiveness in the development of Intrusion Detection Systems (IDSs). Machine Learning (ML) is a collection of scientific techniques that supports numerical pattern identification and independent analysis to gain important insights from data records [4]. The prediction accuracy of ML increases significantly as more relevant data are acquired. ML algorithms are generally divided into two main groups: supervised and unsupervised learning algorithms [5]. Supervised machine learning techniques—including K-Nearest Neighbors (KNN) [6], Decision Tree (DT)-based models [7], deep learning approaches [8], and several other algorithms—rely on labeled datasets to establish mappings between input features and corresponding output classes [7]. Unsupervised algorithms, including centroid-based clustering (e.g., k-means), probabilistic models (e.g., Gaussian mixtures), and anomaly detection approaches (e.g., isolation-based methods), are employed to uncover hidden structures in unlabeled data [9−11]. Signature-based IDSs are most commonly developed using Supervised Learning (SL) algorithms. These algorithms require labeled datasets to carry out their training processes. Anomaly-based IDSs are typically developed using Unsupervised Learning (UL) techniques. These IDSs are able to distinguish unusual data from normal data samples.

An IDS operates as a monitoring system by repeatedly scanning the network to detect and prevent intrusion attempts. It conducts an in-depth analysis of network requests—using signature-based inspection, protocol analysis, and statistical packet analysis—before determining whether they are malicious or benign. The proposed framework safeguards systems from various types of attacks, such as Distributed Denial of Service (DDoS), by permitting genuine requests while promptly flagging and alerting on potentially malicious traffic.

A network Intrusion Detection System (IDS) typically relies on two fundamental approaches: Known attack

Manuscript received July 18, 2025; revised August 20, 2025; accepted October 9, 2025; published January 29, 2026.

doi:10.12720/jcm.21.1.89-100

signature detection and abnormal activity detection. The previous identifies recognizing threats via correlation of network traffic data against predefined patterns of known attacks, whereas the latter recognizes potential intrusions by observing irregularities or deviations from established normal behaviour [12]. Signature-based detection systems identify threats using previously established attack signatures. These systems are effective against attacks whose characteristics are already known through such established patterns. However, they cannot stop newly evolved attacks because they are unable to learn from unknown patterns [13]. In anomaly-based systems, threats are identified through the detection of activities that diverge from expected norms or standard patterns. Such systems are capable of identifying previously unseen attacks by relying on models that characterize standard patterns of network behaviour [14].

Despite continuous advancements in Network-Based Intrusion Detection Systems (NIDSs), significant opportunities for improvement remain. Challenges arise from the high volume of network data, the dynamic nature of network environments, the extensive feature sets required for training, and the demand for real-time detection [15]. Redundant or irrelevant features, for instance, can prolong the training phase and reduce the accuracy of NIDSs in identifying malicious activities. Elevating the reliability of machine learning processes–driven detection models therefore requires careful feature selection and appropriate parameter optimization [16].

Several data-centric and algorithmic techniques have been used in previous studies to develop lightweight, fast, high-performance classifiers without compromising accuracy:

- The existing methodologies and research approaches for deploying IDSs in network traffic and related domains.
- A comparative evaluation of the XYZ dataset variations is carried out to offer a detailed understanding of attack categories, dataset scale, sample distribution, and their overall significance.
- The methods used for training, validating, and testing the algorithms include oversampling, feature extraction using a Genetic Algorithm, and feature selection using GIWRF and boosted LSTM.
- Various ML approaches such as knowledge-based models, transfer learning, behavioral pattern analysis and recorded pattern analysis are used to provide a thorough assessment of their effectiveness.
- A detailed discussion is presented on the advantages, limitations, and performance of the developed framework.

Investigations into the application of ML for intrusion detection in IoT-based WSNs make a substantial contribution to improving system resilience and security. This study aims to utilize ML methods, namely anomaly detection and classification models, to identify and address different forms of intrusions in real time. The aim is to protect confidential data and maintain the uninterrupted functioning of IoT devices within the network. By using ML, it becomes possible to proactively detect suspicious actions and deviations from normal behavior patterns. This, in turn, provides strong defense mechanisms against constantly evolving cyber threats in WSNs. This contribution strengthens the protection of connected devices and facilitates the design of adaptive, intelligent intrusion detection solutions that address the specific challenges of IoT-based Wireless Sensor Network (WSN) environments.

This article is structured formulated as: the Related Works part summarizes earlier research efforts of existing scholarly works that form the foundation of this research; the Methodology discusses the design of the implemented framework; the Experimental Procedure and Findings section details the configuration used for evaluation; the Results and Discussion section highlights the outcomes and their interpretation; the Limitations and Future Work section addresses current restrictions and suggests potential directions, including multiclass IDS development, time complexity evaluation, validation on diverse IoT/WSN datasets, and the design of lightweight models for resource-constrained environments; finally, the Conclusion section outlines the main contributions of this work.

The increasing adoption of Internet of Things (IoT) applications alongside Wireless Sensor Networks (WSNs) has created significant security concerns, largely because sensor nodes operate with limited resources and are often deployed in diverse and unprotected environments. Within IoT systems, WSNs are particularly exposed to a variety of cyber-attacks, which highlights the necessity of developing strong intrusion detection and prevention strategies. Addressing these challenges, this research concentrates on IoT-driven WSNs and introduces a machine learning–based intrusion detection framework capable of identifying attacks with high accuracy while ensuring minimal computational overhead. By focusing on this setting, the proposed solution meets practical security requirements and contributes to the advancement of reliable protection mechanisms for IoT–WSN infrastructures.

II. Literature Review

Alhayali *et al.* [17] developed a more effective Intrusion Detection (ID) strategy for binary classification. Additionally, a hybrid approach combining the Rao-SVM algorithm with supervised Machine Learning (ML) techniques for Feature Subset Selection (FSS) was introduced, incorporating several optimizers, including Rao Optimization (RO), Logistic Regression (LR), Support Vector Machines (SVM), and Extreme Learning Machines (ELM). Ibraheem *et al.* [18] employed supervised ML techniques for FSS in combination with the newly developed RO method, IDS, SVM, ELM, and LR. The Rao-SVM FSS system is presented in their work along with an analysis of its parameter-free and algorithm-specific model. In Ref. [19], an intelligent IDS for Wireless Sensor Networks (WSNs) was developed, utilizing the K-Nearest Neighbors (KNN) algorithm and the Arithmetical Optimization Algorithm (AOA) from evolutionary computation. This system aimed to create an intelligent structure capable of effectively detecting and responding to Denial of Service (DoS) attacks in WSNs.

Wu *et al.* [20] proposed a feature analysis and SVM-optimized integrated web intrusion detection system, where experts analyzed common online attack characteristics. Examination of the HTTP protocol facilitated selection of relevant data attributes. Janabi and Ismail [21] developed a method integrating SVM, NTLBO, ELM, and LR algorithms using supervised ML techniques for FSS. In Ref. [22], a method was proposed to optimize the performance of Network Intrusion Detection Systems (NIDSs) using wrapper-based techniques combined with Genetic Algorithm (GA), Firefly Algorithm (FFA), Particle Swarm Optimization (PSO), and Grey Wolf Optimizer (GWO) to select features, implemented using the Anaconda Python Open Source platform. Additionally, GA, GWO, FFA, and PSO were applied to compute Mutual Information (MI) through filtering-based methods.

Bhattacharya *et al.* [23] proposed a hybrid ML approach for IDS dataset classification using Principal Component Analysis (PCA) and fireflies. IDS datasets were transformed using One-Hot encoding, and XGBoost was employed to classify the reduced data. In Ref. [24], a novel hybrid intelligent system utilizing an inverted hourglass-based encrusted network classifier was introduced for feature classification tasks. This approach was validated on three datasets to distinguish between old and new attack behaviors, employing a hybrid optimization strategy to prioritize important features. The model also utilized an up-sampled layered network architecture to improve training, enhancing its capability to detect and counter infiltration attempts. Nazir and Khan [25] proposed a new feature selection method for NIDS, termed Tabu Search Random Forest (TS-RF), employing Random Forest (RF) as the learning algorithm and Tabu Search as the search mechanism.

In 2022, a state-of-the-art IDS combining the X2 statistical model with a Bi-Directional Long Short-Term Memory (Bi-LSTM) structure was introduced and evaluated using the NSL-KDD dataset, achieving an accuracy of 95.62% [26]. Another IDS based on Deep Neural Networks (DNNs) used cross-correlation for feature extraction, demonstrating effective network attack detection [27]. A hybrid Deep Learning (DL) framework combining Convolutional Neural Networks (CNNs) for local feature extraction and Recurrent Neural Networks (RNNs) for sequential data analysis was introduced in 2021, tested on the CSE-CIC-DS2018 dataset, and achieved precision scores of up to 97.75% [28].

In 2021, an ANN-based IDS using the Flower Pollination Algorithm (FPA) on the DS2oS dataset achieved an accuracy of 99.1% [29]. Another CNN-based IDS evaluated the NSL-KDD dataset, employing Spider Monkey Optimization (SMO) and Conditional Random Field (CRF) techniques [30]. Studies from 2020 suggested ANN models for detecting normal and abnormal intrusions, utilizing Correlation-based Feature Selection (CFS) on NSL-KDD datasets [31]. In 2019, neural network-based and DNN-based IDS models were proposed, leveraging information enhancement methods to analyze NSL-KDD datasets [32, 33], while in 2018, deep learning models emerged to detect abnormal behaviors and regular cyber incidents [34].

Disha *et al.* [35] developed a feature-ranking algorithm based on Gini impurities using RF to evaluate NIDS performance on the TON-IoT dataset. While classification performance was prioritized, computational costs of feature reduction were not sufficiently addressed. Many existing datasets used for NIDS evaluation in IoT security are outdated, highlighting the need for updated benchmark datasets.

Wrapper-based Feature Selection (FS) is commonly employed to identify optimal feature subsets that improve classification performance. Shafiq *et al.* [36] proposed a wrapper-based FS algorithm and a CorrAUC approach, utilizing the Area Under the Curve (AUC) metric to select relevant features for ML algorithms. Although accuracy was lower for specific attacks, such as key-logging, the method successfully identified relevant features when evaluated on the Bot-IoT dataset [37].

Several studies focused on lightweight solutions for resource-constrained IoT networks. Liu *et al.* [38] combined one-class SVM with Particle Swarm Optimization (PSO) for attack detection, using LightGBM for model construction and PSO for feature selection. Despite efficiency gains, such FS approaches often require significant computational resources, particularly when using GA, PSO, or ML classifiers, which can be challenging for IoT systems.

Moustafa *et al.* [40] proposed an ensemble IDS using ANN, Decision Tree (DT), and Naive Bayes (NB) to extract relevant statistical flow features. Leevy *et al.* [41] employed Information Gain (IG), Chi-squared (Chi2), and Information Gain Ratio (IGR) for feature selection to improve performance metrics. However, computational cost was not a primary focus. Gavel *et al.* [42] analyzed the AWID WSN dataset using Ant Lion Optimization for feature selection, while Zhou *et al.* [43] refined FS by removing redundant features based on correlation thresholds to enhance NIDS accuracy, albeit at the cost of system complexity. Aggarwal [44] explored Random Forest classifiers with Grey-Level Co-occurrence Matrix (GLCM) feature extraction for MRI brain tumor classification, demonstrating the potential of GLCM features to improve accuracy through efficient texture analysis.

III. Research Methodology

Based on the limitations identified in the reviewed literature (Section II), the necessity for IDS capable of delivering high accuracy is apparent but also remains computationally efficient for deployment in IoT-based WSN environments. Several conventional approaches either lack effective feature selection or rely on a single technique, which often leads to overfitting and poor generalization performance. To overcome these gaps, the present work proposes a dual-strategy feature selection framework combining GIWRF and GA, followed by a lightweight LSTM-based classification model. This design leverages the strengths of state-of-the-art techniques while addressing their limitations.

Although both GIWRF and GA have been individually adopted in previous intrusion detection studies for feature selection, there is limited research that combines them in a unified framework. In this work, GIWRF is first used to obtain an initial ranking of features based on their contribution to classification performance, and GA is then applied to further refine the selected feature subset through evolutionary optimization. This two-stage selection strategy allows us to take advantage of the interpretability of GIWRF and the exploration capability of GA, which leads to a more optimal and compact feature set. The proposed combination not only improves dimensionality reduction efficiency but also enhances model performance by discarding redundant and less relevant features.

In the classification stage, a LSTM neural network is applied. The architecture includes two LSTM layers, each containing 64 hidden units, followed by a fully connected dense layer that utilizes a sigmoid activation function to perform binary classification. The sequence length for inputs is fixed at 50, while ReLU is applied as the activation function within the hidden layers. Training is carried out using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. Aimed at avoiding overfitting, a dropout layer with a rate of 0.2 is introduced between network layers. These hyperparameters were determined empirically through initial experiments and guided by insights from prior research.

During the feature ranking stage, GIWRF evaluates the relevance of features by analyzing how strongly they influence the splitting rules within decision trees. Features with higher scores are provisionally selected and passed to the GA, which evolves candidate feature subsets through crossover and mutation operators. The fitness of each candidate is evaluated using classification accuracy, and the best-performing subset is retained for training. LSTM is chosen for the classification stage because of its competence to model temporal correlation sand capture sequential relationships in network traffic data

The proposed research methodology for the “Intelligent Framework for Intrusion Detection and Prevention using Optimized Machine Learning” begins with collecting network traffic-based datasets, which serve as the foundation for subsequent analysis.

Dataset → Pre-processing → GIWRF → GA → Selected Features → LSTM Classifier → Results.

The overall architecture of the proposed framework is illustrated in Fig. 1.

A. Data Pre-processing

Within a given dataset, normalization adjusts the range of data values to enhance information processing. It is particularly useful when there is a large disparity between the maximum and minimum values, helping to alleviate algorithmic challenges. Normalization is especially effective in neural networks for classification tasks. Furthermore, when using back-propagation in neural networks, proper input normalization improves computational efficiency and accelerates training.

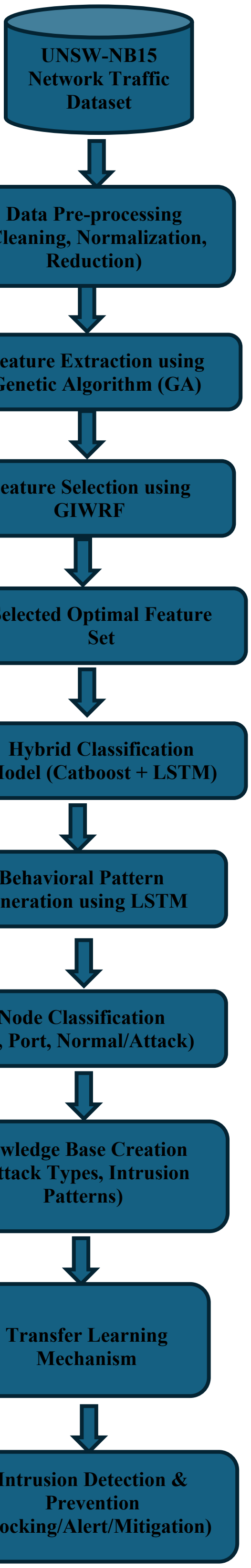
UNSW-NB15 Network Traffic Dataset
Data Pre-processing (Cleaning, Normalization, Reduction)
Feature Extraction using Genetic Algorithm (GA)
Feature Selection using GIWRF
Selected Optimal Feature Set
Hybrid Classification Model (Catboost + LSTM)
Behavioral Pattern Generation using LSTM
Node Classification (ID, Port, Normal/Attack)
Knowledge Base Creation (Attack Types, Intrusion Patterns)
Transfer Learning Mechanism
Intrusion Detection & Prevention (Blocking/Alert/Mitigation)

Fig. 1. Proposed machine learning–based framework for intrusion detection and prevention in IoT-based WSN using GA, GIWRF, and hybrid CatBoost–LSTM model.

B. Normalization

Scaling data represents a crucial step in the overall normalization procedure. This process applies the min–max technique to rescale data values within a defined range, typically [0, 1] or [−1, 1]. The following expression presents the standard normalization formula:

$$I = \frac{d - d_{MIN}}{d_{MAX} - d_{MIN}} \tag{1}$$

According to Eq. (1), the term I represents the normalized input value, indicating that it is a scaled or balanced value. Additionally, the character "d" represents the real value. "d_{MAX}" and "d_{MIN}" refer to the highest and inferior values of the input variable 'd", respectively.

C. Data Reduction

Data reduction techniques eliminate redundant information, noise, errors, and irrelevant data from a dataset. This process ensures that only pertinent and meaningful data are processed in subsequent stages, improving efficiency and reducing computational overhead.

D. Feature Extraction

The effectiveness of an Intrusion Detection System (IDS) is largely influenced by factors such as the completeness and depiction of the input dataset. Effective feature extraction is crucial for accurate detection of malicious network activity. The use of Genetic Algorithms (GA) in IDS aims to optimize feature extraction from network traffic data, enhancing both efficiency and accuracy.

At this stage, potential features available to candidate solutions are represented using binary string encoding. The GA evolutionary process includes solution evaluation, selection, crossover, and mutation operations. Every candidate solution is assessed through a fitness function that quantifies how effectively it separates legitimate network traffic from malicious behavior, and the optimization cycle proceeds until a stopping criterion is satisfied, yielding an optimal set of features that maximizes the IDS's detection and mitigation performance.

The fitness function $f(x)$ gives a measurement for assessment for potential solution x with regard to specific task achievement. In maximization problems the fitness function gains values for better solutions but it loses values for better solutions in minimization problems.

$$P(x) = \frac{f(x)}{\sum_i (x_i)} \tag{2}$$

E. Feature Selection Using GIWRF

The "Random Forest" (RF) is a classifier that combines numerous DT and offers different methods to determine the relevance of features. One method involves calculating the significance score by training the classifier. Traditional ML methods disregard possible class disparities by assuming equal significance for each category in the initial training data. With the aim of tackle this issue, RF utilizes a weight modification mechanism following the calculation of the GI, represented as $i(\tau)$, by the classifier. GI measures the degree to which a split successfully separates the entire collection of samples from both classes inside a particular node. Theoretically, it may be expressed as:

$$i(\tau) = 1 - p_p^2 - p_n^2 \tag{3}$$

where p is the percentage of favorable instances and pn is the fraction of unfavorable tests out of all samples (N) at node τ. The decrease in GI obtained from any most effective split$\Delta(\tau,M)$ is acquired collectively for all the nodes τ in the M quantity of computed within all trees of the forest for every feature independently.

F. System Training

System training employs a robust ensemble classification approach, integrating **CatBoost** and **LSTM** networks. This combination ensures the reliability, adaptability, and efficiency of the IDS.

1) Classification of nodes

The trained model is used to categorize network nodes based on their unique identifiers, such as node ID and port number. This classification enables the detection of potential threats at the node level. By analyzing node-specific information, the system can identify suspicious or abnormal activity associated with particular network entities. Proactive threat detection and mitigation strategies can then be applied, enhancing network security by addressing potential vulnerabilities and malicious behavior at an individual node level.

2) Knowledge base creation

A comprehensive knowledge base is constructed to store detailed information on various intrusion modes, attack types, and anomalous behaviors. This repository includes known threat signatures, attack vectors, and historical data on both successful and failed intrusion attempts. By continuously updating the knowledge base with new information, the system improves its ability to detect and prevent advanced attacks, offering proactive cyber security protection in dynamic network environments.

3) Prevention mechanism integration

The system incorporates preventive measures by analyzing historical intrusion data to proactively block potential threats. By evaluating past intrusion events, the system identifies recurring attack patterns, enabling it to implement preventive actions that mitigate future risks. Recognizing known attack vectors in advance allows networks to enhance security, as these vectors are automatically neutralized before causing damage.

4) Pattern-based prevention

Preventive strategies are deployed based on records of previously observed intrusion patterns. By analyzing historical intrusion data, the system identifies routine attack behaviors, which informs security actions designed to preempt predictable threats. Predictive measures strengthen network defense by preventing or neutralizing previously recorded attack methods, thereby reducing the likelihood of successful intrusions and their associated impacts.

5) Behavioral pattern generation with LSTM

LSTM networks are employed to model **be**havioral patterns that aid in detecting abnormal system events and potential intrusions. LSTM's capability to monitor long sequences of network activity enables it to identify unusual patterns indicative of security risks. This context-aware approach allows dynamic threat detection, as anomalies trigger immediate responses to emerging security threats.

6) Hybrid model

The proposed framework integrates **CatBoost**, a gradient boosting algorithm optimized for categorical data, with LSTM networks for sequential data management. CatBoost effectively processes structured network traffic features while reducing overfitting and supporting reliable querying, whereas LSTM captures long-term temporal dependencies in traffic flows, thereby enhancing the system's capability to identify sophisticated intrusion patterns.

The hybrid model harnesses the complementary strengths of CatBoost and LSTM. While traditional machine learning models have limited capacity to process sequential data, deep learning models require optimal feature selection for effective generalization. CatBoost processes both categorical and numerical inputs efficiently, minimizing bias-related errors, whereas LSTM captures temporal attack patterns. Combining these approaches improves overall intrusion detection accuracy, stability, and robustness compared to using either model independently.

IV. Dataset

The UNSW-NB15 dataset deployed in this research is publicly accessible benchmark dataset containing both benign and malicious traffic generated under realistic network conditions. It encompasses a broad range of contemporary attack categories commonly observed in IoT and WSN environments, making it particularly well-suited for assessing intrusion detection systems in these domains.

The dataset has been widely employed in related studies and is considered representative of real-world IoT-based WSN traffic patterns. Consequently, the findings of this study can be reasonably generalized to similar scenarios. Future work will involve testing the proposed approach on additional datasets to further validate its generality and robustness.

The experimental evaluation of the IDS utilized the UNSW-NB15 dataset for offline analysis [45]. This dataset, extensively referenced in IDS research [46], contains 27 features as summarized in Table I. Notably, UNSW-NB15 is more recent than many other benchmark datasets, making it well-suited for contemporary intrusion detection research.

TABLE I. Features of the UNSW-NB15 Dataset

Features	Value	Section feature
dbytes	int	primary
rate	int	content
sttl	int	primary
dmean	int	content
ct_state_ttl	int	general
dload	float	primary
sloss	int	primary
sinpkt	float	time
dinpkt	float	time
dur	nominal	primary
ct_dst_sport_ltm	int	connection
sbytes	int	primary
synack	float	time
dpkts	int	primary
ackdat	float	time
smean	int	connection
swin	int	content
tcprtt	float	time
ct_src_dport_ltm	int	connection
state_INT	nominal	primary
ct_srv_dst	int	connection
proto_tcp	nominal	flow
ct_srv_src	int	connection
dttl	int	primary
ct_dst_ltm	int	connection
ct_dst_src_ltm	int	connection
sload	int	primary

The dataset was divided into 70% for training, with 15% assigned for validation while the rest 15% reserved for testing the models. A verification procedure was conducted to ensure optimal performance during the training process. The dataset contains contemporary internet traffic data, encompassing both normal and abnormal instances, including modern low-profile attacks. The data is presented in a clean and structured format without unnecessary repetition, making it highly suitable for accurate IDS evaluation.

V. Result

Experiments were conducted on an HP Notebook 14-AL143TX laptop running the latest version of the Windows operating system. The system is powered by an Intel Core™ i5-7200U processor, featuring a base clock speed of 2.8 GHz and a maximum turbo boost of 3.5 GHz. Machine learning models were developed, trained, and evaluated using Pandas**,** Scikit-Learn (sklearn)**,** and other ML libraries within the Python Jupyter Notebook environment, which is freely available.

A. Evaluation Parameters

The proficiency of the formulated approach was assessed using common evaluation metrics such as accuracy, precision, and loss. Accuracy is determined by the ratio of correctly classified instances to the total number of samples in the dataset, as expressed by the following formula:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{4}$$

$$Precision = \frac{TN}{TN + FP} \quad (5)$$

True Positive (TP) indicates the count of attacks correctly detected, while True Negative (TN) represents the number of normal traffic instances accurately classified. False Positive (FP) refers to normal traffic incorrectly identified as attacks, and False Negative (FN) corresponds to attacks that are mistakenly classified as normal network traffic [47].

B. Experimental Results

This section summarizes the findings derived from the binary classifications performed by the IDS developed using machine learning techniques. The study also evaluates precision rates achieved on the used dataset and compares them with those reported in previous research. Moreover, the analysis examines the detection accuracy for the various attack types present in the dataset.

The study was carried out in two stages to evaluate the effectiveness of four ML models: Decision Tree, Random Forest, CatBoost, and the proposed Hybrid model. In the first phase, all features of the UNSW-NB15 dataset wereused to examine the models' effectiveness in detecting binary classes. During the second stage, the study applied the formulated feature selection approach, and the four models were evaluated using accuracy and loss metrics.

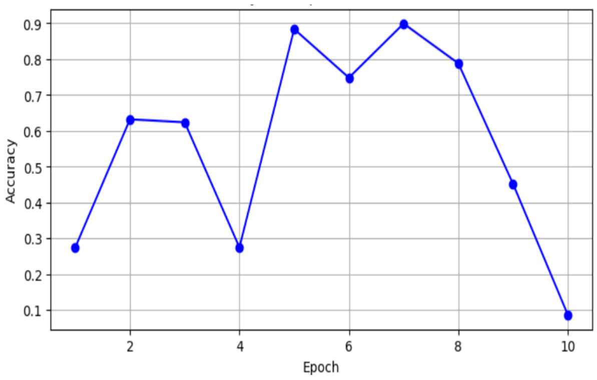

Fig. 2. Relative examination of the accuracy of RF model.

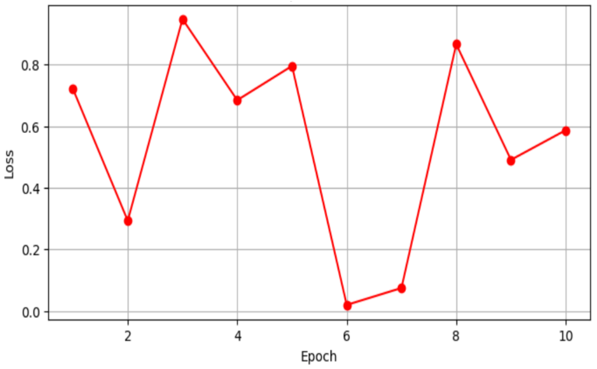

Fig. 3. Loss over the epoch of the RF model.

The accuracy of the Random Forest (RF) model in detecting WSN faults over 10 training epochs is illustrated in Fig. 2. The precision measurements exhibit varying trends throughout the learning process. The model's precision starts at 0.3 in the first epoch and peaks at 0.9 by the fifth epoch. However, precision decreases in the sixth epoch, rises again in the eighth epoch, and then sharply declines to approximately 0.1 by the tenth epoch. This instability in training accuracy suggests potential overfitting and indicates that hyperparameter tuning may be necessary to improve learning stability.

The corresponding loss values of the RF model across the same 10 epochs are presented in Fig. 3. Initially, the loss begins at approximately 0.7, decreases to 0.1 by the second epoch, and then rises sharply to 0.9 in the third epoch. Significant fluctuations continue until the loss reaches a minimum in the sixth epoch, after which it begins to rise again. This erratic behavior indicates instability in the learning process, likely caused by inappropriate learning rates or insufficient data preparation. Optimizing these parameters could lead to more stable training outcomes and reduced loss across epochs.

Fig. 4 illustrates the accuracy of the Decision Tree (DT) model across epochs in detecting and preventing incidents in an IoT-based WSN. The figure presents fluctuations in the model's accuracy over the training period. Typically, accuracy is expected to improve during initial epochs as the model extracts knowledge from the data, followed by periods of stabilization or minor variations as the model fine-tunes its performance. Observed fluctuations in accuracy indicate aspects of the learning process that may require improvements in training procedures or data preprocessing to enhance and stabilize model performance.

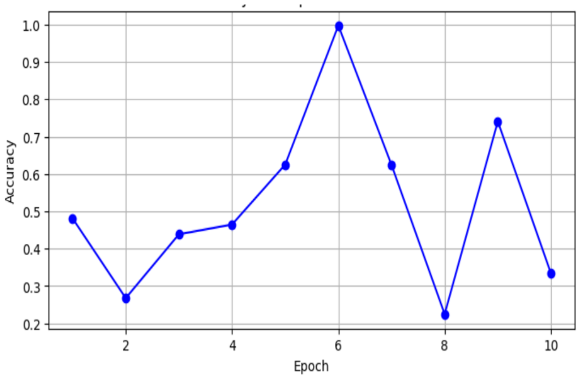

Fig. 4. Accuracy over epoch for model 2 (DT).

The loss performance of the DT model over 10 epochs is shown in Fig. 5. Initially, the loss starts at approximately 0.45 and remains relatively stable for the first three epochs. In the fourth epoch, the loss rises sharply to a peak of 0.8, before gradually decreasing. A secondary peak is observed around the ninth epoch. These fluctuations indicate that the DT model experiences learning instabilities, likely caused by overfitting and over-identification of patterns in the dataset. To achieve more stable loss values during training, the model requires optimized hyperparameter configurations and improved data preprocessing techniques.

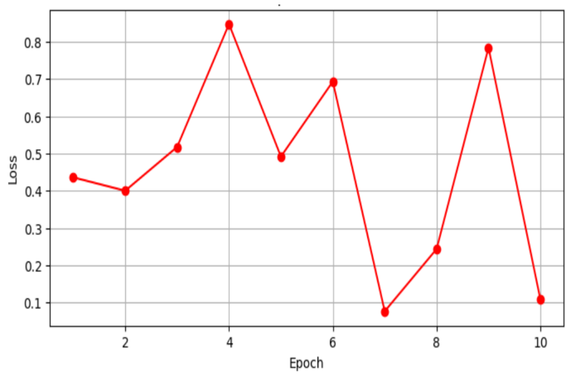

Fig. 5. Loss over Epoch of DT model.

Fig. 6 illustrates the accuracy of the CatBoost (CB) model over the training epochs. The accuracy begins at approximately 0.6 in the first epoch and reaches a peak of around 0.9 by the seventh epoch. Following this peak, accuracy declines, indicating some instability, before rising again around the eighth epoch and then sharply dropping to approximately 0.1 by the tenth epoch. This fluctuation suggests variability in the model's learning process and potential overfitting or hyperparameter-related issues.

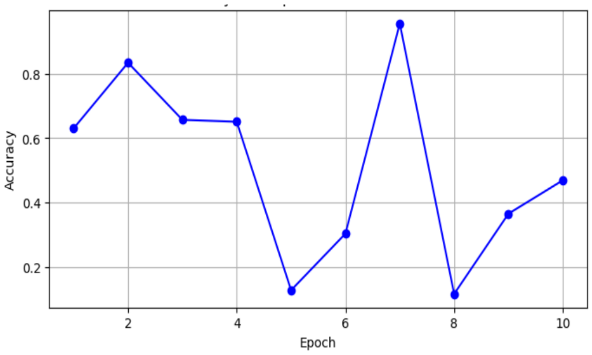

Fig. 6. Accuracy over epoch for CatBoost model.

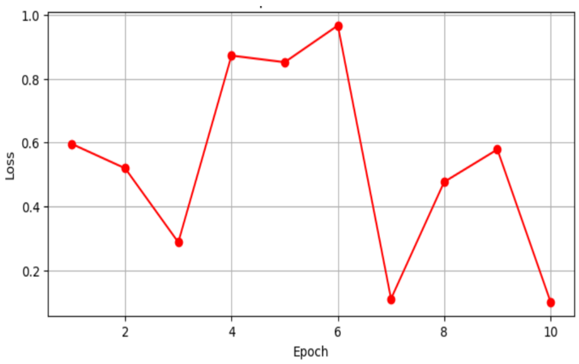

Fig. 7. Loss of Catboost model over epoch.

Fig. 7 shows the loss progression of the CatBoost model across 10 epochs. The loss starts at roughly 0.6 and reduces to nearly 0.3 by the third epoch, indicating initial learning improvements. However, the loss unexpectedly increases to nearly 1.0 by the fourth epoch, followed by alternating peaks and troughs, including a notable spike in the sixth epoch and a rapid decrease in the subsequent epoch. These fluctuations indicate substantial variability during training, reflecting instability in the model's learning process and emphasizing the need for further hyperparameter tuning and optimization to achieve stable and reliable performance.

Fig. 8 illustrates the accuracy of the hybrid model across training epochs. The model initially achieves an accuracy of approximately 0.2, peaks at around 0.85 during the third epoch, and subsequently drops to zero by the fifth epoch, indicating a significant decline in performance. Accuracy fluctuates throughout the epochs, with notable peaks observed during the third and seventh epochs. Overall, the pattern highlights instability in the model's ability to consistently detect intrusions, suggesting that further refinement of the model structure and training configuration parameters is necessary.

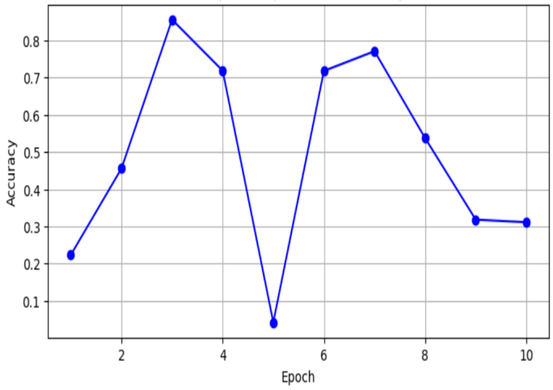

Fig. 8. Accuracy of model 4 (Hybrid model).

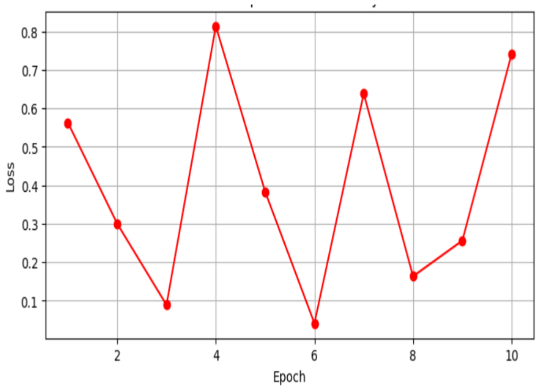

Fig. 9. Loss over epoch for Hybrid model.

Fig. 9 presents the loss progression of the hybrid model trained for intrusion detection in IoT-based WSNs. The loss exhibits substantial fluctuations, indicating instability in the learning process. Initially, the loss is around 0.5, decreasing to approximately 0.1 by the third epoch, reflecting notable early progress. However, a sharp increase occurs during the fourth epoch, peaking at roughly 0.98, followed by alternating downward and upward trends in subsequent epochs. These variations emphasize the need for improved hyperparameter tuning and training optimization to achieve more stable and reliable model performance. The comparative performance of different machine learning models in terms of accuracy and loss is presented in Table II.

TABLE II. RELATIVE EXAMINATION OF THE ACCURACY AND LOSS OF THE FOUR MODELS

Epoch	Accuracy				Loss			
	RF model	DT model	Cat Boost model	Hybrid model	RF model	DT model	Cat Boost model	Hybrid Model
1	28.6%	49%	62%	21%	75%	42%	59%	56%
2	63.6%	27%	82%	46%	28%	40%	54%	30%
3	63%	44%	65%	86%	90%	51%	29%	9%
4	28%	48%	64%	71%	70%	86%	86%	82%
5	89%	62%	%14	5%	80%	49%	82%	39%
6	75%	71%	30%	99%	2%	69%	98%	5%
7	90%	62%	98%	78%	10%	8%	9%	62%
8	80%	23%	10%	55%	85%	25%	50%	18%
9	45%	75%	38%	32%	50%	79%	58%	25%
10	10%	32%	30	31%	60%	11%	8%	75%
Average	**57.2%**	**49.3%**	**49.3%**	**55.4%**	**55%**	**46%**	**53.3%**	**40.1%**

Fig. 10 compares the accuracy of the four machine learning models across training epochs. Performance fluctuates for all models, with the hybrid model consistently achieving higher accuracy than the others in most epochs. The CatBoost model occasionally approaches similar accuracy levels but is generally outperformed by the hybrid approach. The Decision Tree and Random Forest models exhibit greater variability and often lag behind in accuracy. Overall, the hybrid model demonstrates superior and more stable performance, highlighting its effectiveness in reliably detecting intrusions in IoT-based WSNs.

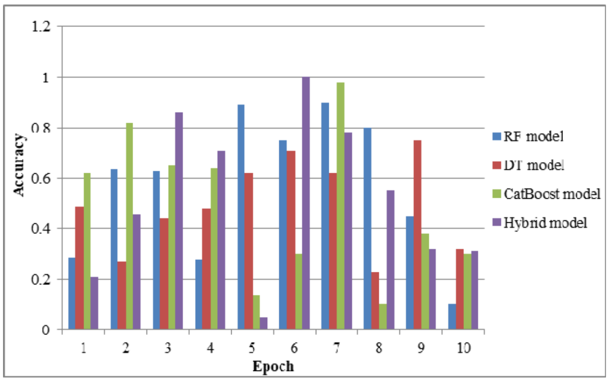

Fig. 10. Comparative research on the accuracy of several models.

Fig. 11 presents the loss values for the four machine learning models. Lower loss values indicate higher model efficacy, reflecting a smaller difference between predicted and actual results. The hybrid model consistently exhibits the lowest loss values across multiple epochs, emphasizing its robustness and effectiveness in intrusion detection. The CatBoost model occasionally shows higher loss values but generally maintains strong performance. In contrast, the Random Forest (RF) and Decision Tree (DT) models display greater volatility, with higher and more fluctuating loss values. These observations underscore the reliability of the hybrid model in minimizing prediction errors, making it a more effective tool for preventing unauthorized access in IoT-based WSNs.

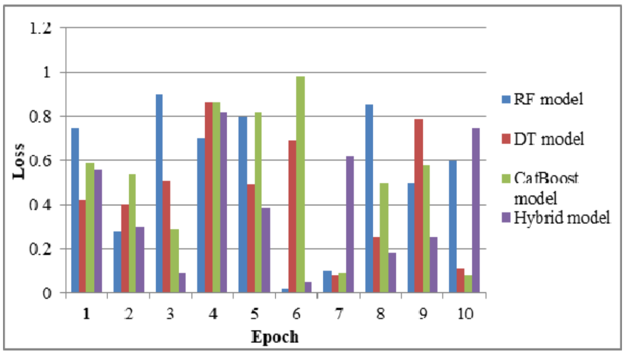

Fig. 11. Analysis of the loss of various ML models.

Table III Offers a holistic overview evaluation of the formulated method's effectiveness in detecting specific attack types, including DoS, Probe, RPL Rank Attack, Sybil Attack, and Blackhole Attack. The hybrid model achieves high performance across all evaluation metrics, maintaining accuracy, precision, recall, and F1−Score levels between 97% and 99%. Notably, detection of Sybil attacks reaches a success rate of 99.1%, comparable to the detection rates for other attack types. The system demonstrates a minimum average Intrusion Detection Rate (IDR) of 98%, reflecting its ability to identify and prevent threats with minimal false alarms. Overall, the hybrid model provides effective and stable security, ensuring robust protection for IoT-based WSNs against cyber-attacks.

TABLE III. OVERALL PERFORMANCE OF THE HYBRID MODEL FOR VARIOUS ATTACKS

Attack Type	Accuracy (%)	Precision (%)	Recall (%)	F1−Score (%)	IDR %
DoS	98.3%	98.1%	98.4%	98.25%	99.1%
Probe	97.9%	97.8%	97.9%	97.85%	98.7%
RPL Rank Attack	98.5%	98.3%	98.6%	98.45%	98.9%
Sybil Attack	99.1%	99.0%	99.2%	99.1%	99.5%
Blackhole	97.8%	97.6%	97.9%	97.75%	98.3%
Average	98.32%	98.16%	98.4%	98.28%	98.9%

Fig. 12 depicts the performance contrast between the developed hybrid model and various existing models. The hybrid system demonstrates exceptional detection metrics across multiple attack types, maintaining accuracy, precision, recall, and F1−Score levels consistently above 97–99%**.** Detection of Sybil attacks achieves optimal performance, while all other attacks also show consistent and effective identification. Both normal and malicious behaviors are efficiently classified based on the input data, demonstrating the system's strong capability to safeguard IoT-based WSN networks against cyber-attacks.

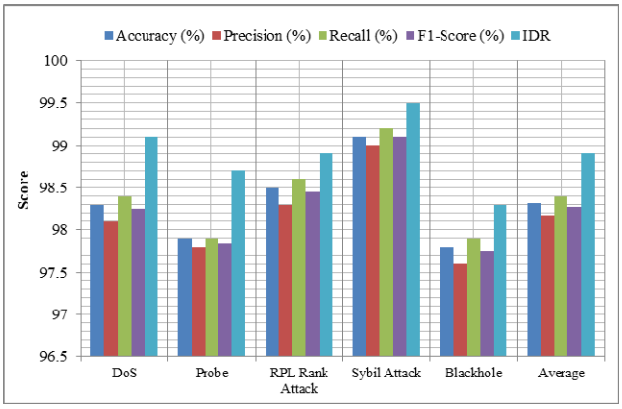

Fig. 12. Performance of Hybrid ML-based Intrusion Detection in IoT-WSN.

C. Comparative Analysis

Table IV shows a contrast of various intrusion detection systems, including DNN, Naive Bayes (NB), DRNN, DCNN, and KNN-PSO, against the proposed hybrid machine learning paradigm. The proposed model outperforms all others, achieving 98.32% accuracy, 98.16% precision, 98.4% recall, and 98.28% F1−Score. While the NB model shows competitive performance, the DCNN model exhibits the lowest accuracy at 89.1%. These findings indicate that the developed hybrid model outperforms others in effectively detecting intrusions within IoT-enabled WSN networks.

TABLE IV. Performance Comparison of Different Models

Models	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DNN [48]	93.74%	93.712 %	93.824%	93.472%
NB [49]	97.14%	96.72%	96.33%	97.94%
DRNN [50]	94.27%	92.18%	93.29%	92.29%
DCNN [51]	89.1%	89.23%	88.2%	89.1%
KNN-PSO [52]	96.42%	95.35%	98.36%	95.42%
Proposed model	98.32%	98.16%	98.4%	98.28%

Fig. 13 presents the performance metric scores of different intrusion detection models evaluated in this research. The figure highlights performance degradation in the DCNN model, whereas the proposed hybrid model consistently achieves the highest scores across all metrics. These trends validated that the formulated method considerably augments intrusion detection in IoT-based WSN networks, validating its superiority over conventional methods.

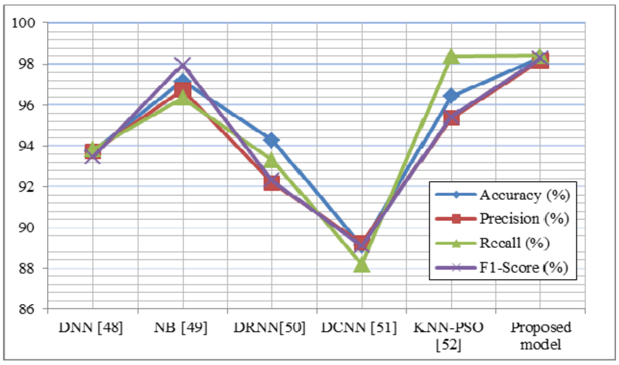

Fig. 13. Performance metrics comparison of different models for intrusion detection.

VI. Novelty of the Study

The originality of this work lies in the integration of GIWRF and Genetic Algorithm (GA) for optimized feature selection, combined with a hybrid CatBoost-LSTM model for intrusion detection in IoT-based WSNs. Unlike conventional IDS approaches that rely solely on ML or DL techniques, this study leverages the complementary strengths of both methods—CatBoost for handling categorical data and LSTM for capturing temporal relationships in network traffic. Performance is further improved by selecting the top 27 features from the UNSW-NB15 dataset while keeping computational costs low. Extensive evaluation across diverse intrusion environments addresses model stability issues and generates more robust results compared to traditional approaches. The hybrid model also offers enhanced IDS flexibility, making it deployable in dynamic IoT networks.

VII. Limitations and Future Work

While the proposed hybrid IDS framework demonstrates promising performance, certain limitations remain. First, the study primarily focused on binary classification and did not address multiclass intrusion detection scenarios, which are crucial for real-world IoT and WSN applications. Second, the time complexity and computational overhead of the framework were not analyzed, which is an important factor for deployment on resource-constrained devices. Third, although the UNSW-NB15 dataset provides a diverse range of attack categories, the generalizability of the results may be limited, as experiments were not extended to multiple datasets. Additionally, the current evaluation does not incorporate statistical reliability tests such as confidence intervals, error bars, or multiple-trial averages, which would further strengthen the robustness of the findings.

Future research will extend this work by developing multiclass IDS frameworks, incorporating time complexity analysis, validating performance on additional IoT and WSN datasets, and exploring lightweight models suitable for deployment on low-power sensor nodes. Moreover, future experiments will include statistical reliability measures to enhance the credibility of the reported results.

VIII. Discussion and Conclusion

Comparing the findings of this study on Intrusion Detection & Prevention (ID&P) in IoT-based WSNs using ML and Big Data Analytics (BDA) with previous research reveals several important insights. Efficient data collection strategies facilitated the management of complex IoT data and the construction of effective training datasets. The integration of BDA for feature extraction, combined with GA and GIWRF techniques, proved effective in enhancing model accuracy and operational performance. The system maximizes feature selection based on relevant patterns, and ML models such as RF, DT, and the Hybrid model improve node-level detection in WSNs, aligning with prior studies emphasizing model selection based on application requirements.

This research advances the current knowledge by providing updated insights into hybrid ML approaches for IoT-based WSNs, focusing on improving network reliability and security. The study trained and evaluated DT, RF, CatBoost, and Hybrid models for binary classification in ML-based IDS. Feature selection was performed on imbalanced datasets, with GA applied to the UNSW-NB15 dataset, and the GIWRF approach adopted for feature evaluation. Decision-making strategies reduced dataset dimensionality, and models were assessed based on accuracy and loss. Initially, single ML methods were evaluated, followed by all four models individually. Results indicated that the hybrid model, combined with feature selection, exhibited superior performance for the UNSW-NB15 dataset.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Rajesh: Conceptualization, Methodology, and Writing – original draft; Mridul Chawla: Supervision, Editing, and Review; both authors had approved the final version.

References

[1] K. Sharfuddin, E. Sivaraman, and P. B. Honnavalli, "Performance evaluation of advanced machine learning algorithms for network intrusion detection system," in *Proc. International Conference on IoT Inclusive Life (ICIIL 2019),* 2020, pp. 51−59.

[2] R. J. Zhao, G. Guan, X. Zhi, Y. Jie, O. Tomoaki, A. Bamidele, and G. Haris, "A novel intrusion detection method based on lightweight neural network for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 12, 2021.

[3] L. Yang, A. Moubayed, A. Shami, P. Heidari, A. Boukhtouta, A. Larabi, R. Brunner, S. Preda, and D. Migault, "Multi-perspective content delivery networks security framework using optimized unsupervised anomaly detection," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, 2021.

[4] M. N. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Machine learning towards intelligent systems: Applications, challenges, and opportunities," *Artificial Intelligence Review*, vol. 54, no. 5, 2021.

[5] L. Yang, and A. Shami, "IoT data analytics in dynamic environments: From an automated machine learning perspective," *Engineering Applications of Artificial Intelligence*, vol. 116, 2022.

[6] W. M. Zuo, D. Zhang, and K. Q. Wang. "On kernel difference-weighted k-nearest neighbor classification," *Pattern Analysis and Applications*, vol. 11, 2008.

[7] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, 1991.

[8] R. A. Khalil, N. Saeed, M. Masood, Y. M. Fard, M. S. Alouini, and T. Y. A. Naffouri, "Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications," *IEEE Internet of Things Journal*, vol. 8, no. 14, 2021.

[9] K. Alsabti, S. Ranka, and V. Singh, "An efficient k-means clustering algorithm," 1997.

[10] L. H. Li, R. J. Hansman, R. Palacios, and R. Welsch, "Anomaly detection via a gaussian mixture model for flight operation and safety monitoring," *Transportation Research Part C: Emerging Technologies*, vol. 64, pp. 45−57, 2016.

[11] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *Proc 2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413−422.

[12] S. M. H. Bamakan, B. Amiri, M. Mirzabagheri, and Y. Shi, "A new intrusion detection approach using PSO based multiple criteria linear programming," *Procedia Computer Science*, vol. 55, pp. 231−237, 2015.

[13] S. X. N. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Applied Soft Computing*, vol. 10, no. 1, pp. 1−35, 2010.

[14] H. J. Liao, C. H. R. Lin, Y. C. Lin, and K. Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[15] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 4, pp. 70–73, 2014.

[16] J. Zhang and M. Zulkernine, "Anomaly-based network intrusion detection with unsupervised outlier detection," in *Proc. 2006 IEEE International Conference on Communications*, 2006, vol. 5, pp. 2388–2393.

[17] A. R. A. Ibrahem, M. Aljanabi, A. H. Ali, M. A. Mohammed, and T. Sutikno, "Optimized machine learning algorithm for intrusion detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, pp. 590–599, 2021.

[18] N. A. A. Shamis, M. Alsajri, and H. R. Ibraheem, "Rao-SVM machine learning algorithm for the intrusion detection system," *Iraqi Journal for Computer Science and Mathematics*, vol. 1, no. 1, pp. 23–27, 2020.

[19] G. Y. Liu, H. Q. Zhao, F. Fan, G. Liu, Q. Xu, and S. Nazir, "An enhanced intrusion detection model based on improved kNN in WSNs," *Sensors*, vol. 22, no. 4, 2022

[20] C. Liu, J. Yang, and J. Q. Wu, "Web intrusion detection system combined with feature analysis and SVM optimization," *EURASIP Journal on Wireless Communications and Networking*, no. 1, no. 33, 2020.

[21] M. A. Janabi and M. A. Ismail, "Improved intrusion detection algorithm based on TLBO and GA algorithms," *Int. Arab J. Inf. Technol.*, vol. 18, no. 2, pp. 170−179, 2021.

[22] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA, and GA algorithms," *Symmetry*, vol. 12, no. 6, 2020.

[23] S. Bhattacharya, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu, M. Alazab, and U. Tariq, "A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU," *Electronics*, vol. 9, no. 2, 2020.

[24] N. Kumar and S. Sanjeev, "A hybrid modified deep learning architecture for intrusion detection system with optimal feature selection," *Electronics*, vol. 12, no. 19, 2023.

[25] A. Nazir and K. A. Rizwan, "A novel combinatorial optimization-based feature selection method for network intrusion detection," *Computers and Security*, vol. 102, 2021.

[26] S. Hanif, T. Ilyas, and M. Zeeshan, "Intrusion detection in IoT using artificial neural networks on UNSW-15 dataset," in *Proc. 2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT and IoT and AI*, 2019, pp. 152−156.

[27] O. Harrison, "Machine learning basics with the k-nearest neighbors algorithm," *Towards Data Science*, vol. 11, 2018.

[28] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721−82743, 2019.

[29] W. H. He, H. J. Li, and J. U. Li, "Ensemble feature selection for improving intrusion detection classification accuracy," in *Proc. 2019 International Conference on Artificial Intelligence and Computer Science*, 2019, pp. 28−33.

[30] E. Hodo, B. Xavier, A. Hamilton, P. L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in *Proc. 2016 International Symposium on Networks, Computers and Communications (ISNCC)*, 2016, pp. 1−6.

[31] I. Boukabous, M. Azizi, M. Moussaoui, O. E. F. Hakim, "Toward a deep learning-based intrusion detection system for IoT against botnet attacks," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, 2021.

[32] Y. Imrana, Y. P. Xiang, A. Liaqat, A. R. Zaharawu, and Y. C. Hu, K. Seifedine, and S. Lim, "χ 2-bidlstm: A feature driven intrusion detection system based on χ 2 statistical model and bidirectional lstm," *Sensors*, vol. 22, no. 5, 2022.

[33] A. Inamdar, "Data science," *Ensemble Learning Techniques in Machine Learning*, vol. 18, no. 9, 2021.

[34] J. Jabez and B. Muthukumar, "Intrusion Detection System (IDS): Anomaly detection using outlier," *Procedia Computer Science*, pp. 338−346, 2015.

[35] D. R. Abedin and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," *Cybersecurity*, vol. 5, no. 1, 2022.

[36] M. Shafiq, Z. H. Tian, A. K. Bashir, X. J. Du, and M. Guizani, "CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, 2020.

[37] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, 2019.

[38] J. Y. Liu, D. S. Yang, M. J. Lian, and M. S. Li, "Research on intrusion detection based on particle swarm optimization in IoT," *IEEE Access*, vol. 9, pp. 38254−38268, 2021.

[39] A. Chohra, P. Shirani, E. M. B. Karbab, and M. Debbabi, "Chameleon: Optimized feature selection using particle swarm optimization and ensemble methods for network anomaly detection," *Computers and Security*, vol. 117, 2022.

[40] M. Nour, B. Turnbull, and K. K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, 2018.

[41] L. L. Joffrey, J. Hancock, T. M. Khoshgoftaar, and J. M. Peterson, "IoT information theft prediction using ensemble feature selection," *Journal of Big Data*, vol. 9, no. 1, 2022.

[42] G. Shashank, A. S. Raghuvanshi, and S. Tiwari, "An optimized maximum correlation based feature reduction scheme for intrusion detection in data networks," *Wireless Networks*, vol. 28, no. 6, pp. 2609−2624, 2022.

[43] L. Zhou, Y. T. Zhu, R. Zong, and Y. Xiang, "A feature selection-based method for DDoS attack flow classification," *Future Generation Computer Systems*, vol. 132, 2022.

[44] A. A. Kumar, "Learning texture features from GLCM for classification of brain tumor MRI images using random forest classifier," *Trans Signal Process*, vol. 18, pp. 60−63, 2022.

[45] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustain Cities Soc*., vol. 72, 2021.

[46] M. A. Omari, M. Rawashdeh, F. Qutaishat, M. Alshira'H, and N. Ababneh, "An intelligent tree-based intrusion detection model for cyber security," *Journal of Network and Systems Management*, vol. 29, 2021.

[47] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Inf. Sci.*, pp. 340–341, 2016.

[48] A. Awajan, "A novel deep learning-based intrusion detection system for IoT networks," *Computers*, vol. 12, no. 2, 2023.

[49] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. C. Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Engineering Journal*, vol. 61, no. 12, 2022.

[50] M. Almiani, A. A. Ghazleh, A. A. Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, 2020.

[51] M. Bhargavi and Y. Pachipala, "Advancing IoT security: Integrative machine learning models for enhanced intrusion detection in wireless sensor networks," *Engineering, Technology and Applied Science Research*, vol. 14, no. 4, 2024.

[52] M. Karthikeyan, D. Manimegalai, and R. G. Karthikeyan, "Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection," *Scientific Reports*, 14, no. 1 2024.