Research Article

# DDOS Attack Classification With Hyperparameter Tuning And Enhanced Prediction Technique

A. Hemlathadhevi[1], C. Jackulin[2], C. Ramesh Kuma[3], T. Ani Bernish[4], V. Anitha[5], P. Deepa[6]

[1 2 5 6]Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, Tamil Nadu, India.
[3]School of Artificial Intelligence (SoAI), Galgotias University, Greater Noida, Delhi-NCR, India.
[4]Department of Computer Science and Engineering, Easwari engineering college, Chennai, Tamil Nadu, India.

**Abstract**

Cyber-Physical Production Systems (CPPS) are integral components of Industry 4.0, enabling seamless integration between computational and physical processes. However, the interconnected nature of CPPS makes them vulnerable to cybersecurity threats, such as Distributed Denial-of-Service (DDoS) attacks. The technologies utilized are Machine Learning Algorithms, Rule-Based Systems, Networking Tools and Protocols, Programming Languages and Platforms, Scikit-learn, TensorFlow, or Py-Torch for model development. The current or existing model has only binary classification indicating malicious and normal attacks. And various strategies, including machine learning (ML) techniques, have been employed for DDoS attack detection. Notable approaches include deep learning models, ensemble methods, and feature selection techniques. But this existing model does not store the result logs for future reference. While these methods achieve reasonable accuracy, they often struggle with issues like class imbalance, scalability, and real-time adaptability. In the other hand the model that is being proposed works to enhance DDoS attack multi classification through suitable trained and tested datasets using LGBM algorithm. Here it would classify the normal and malicious attacks into various types of attacks such as DDOS attack, HTTP attack, traffic detection etc. A Flask-based web application for data upload and result display. Designing a database to store results and logs in file format for future use and enhancement. It operates with an average time complexity of O(log n) for both training and inference, making them highly efficient for real-time DDoS detection. The system demonstrates high classification performance, with precision and recall exceeding 99%, while maintaining a computationally efficient structure suitable for real-time implementation.

**Keywords**: Industry 4.0, CCPS, DDOS, Cybersecurity, Machine learning, Flask.

## 1. Introduction

The advent of Industry 4.0 has revolutionized manufacturing and production systems through the integration of Cyber-Physical Production Systems (CPPS). The increasing threat of DDoS attacks demands efficient and scalable detection systems to ensure network security. Existing methods, while effective to some extent, face challenges related to accuracy, scalability, and real-time performance [1]. This research aims to overcome these limitations by proposing a hierarchical machine learning approach with hyperparameter optimization, ensuring high performance and adaptability in detecting and classifying DDoS attacks. These systems enable seamless communication and coordination between computational algorithms and physical processes, driving efficiency, automation, and innovation. However, the interconnected nature of CPPS also exposes them to significant cybersecurity risks, particularly Distributed Denial-of-Service (DDoS) attacks. DDoS attacks aim to disrupt services by overwhelming systems with malicious traffic, leading to downtime, financial losses, and compromised operational integrity [2] [3].

This paper proposes a novel framework for DDoS attack classification that addresses these challenges by leveraging the Light Gradient Boosting Machine (LGBM) algorithm. Unlike traditional binary classification models, the proposed system enables multi-class classification, categorizing attacks into specific types such as DDoS, HTTP attacks, and traffic anomalies. Additionally, the framework incorporates a Flask-based web application for user-friendly data upload and result visualization, along with a database to store result logs for future reference and system enhancement. The LGBM algorithm operates with an average time complexity of O (log n), ensuring high efficiency in both training and inference phases, making it well-suited for real-time DDoS detection. Preliminary results demonstrate exceptional classification performance, with precision and recall metrics exceeding 99%, underscoring the system's potential for real-world deployment in CPPS environments. By combining advanced ML techniques with practical system design, this work aims to provide a scalable, efficient, and accurate solution for DDoS attack classification, contributing to the broader goal of securing Cyber-Physical Production Systems in the era of Industry 4.

## 2. Background And Literature Review

In recent years, cyber physical production systems (CPPs) have become an important component of modern supply chains,

particularly in sectors such as production, agriculture and energy. These systems integrate computing and physical processes [4] [5]. This allows the communications network to act as a backbone connection sensor, device, machine, and operator. In a rapid environment where real-time data exchange is extremely important, efficient communication is extremely important [7]. We investigated the use of algorithms for machine learning to detect DDOS attacks in semiconductor production plants, achieving a high level of identification [6]. However, their approaches face challenges in using attack traffic from various scenarios and can affect real-time applicability. They implemented feature extraction techniques such as Main Component Analysis (PCA) and Binary Salps Warm Optimization (PCA-BSO) to test monitored learning algae such as Support Vector Machine (SVM), Randall Swald, and logistic regression. Their results were promising, but the actual performance of the system was not verified. This is a normal limitation in many studies [11][13]. The hybrid model combining control-based detection and machine learning technology demonstrates a promising solution that commits the limitations of both approaches. uszko et al. [9] They evaluated the system on AWID3 public datasets and achieved a recognition accuracy of 98.57%. Public data records such as CICIDS 2018 [17], UNSW-NB15 [19], and Botit [20] were often used to train IDS models. However, there are restrictions on the use of public data records. [8] proposed a federated learning-based approach to supply chain safety using the Ton_iot dataset. The system achieved 99.33% accuracy, but with questions about actual use, the issues of scalability and real-time synchronization were not treated. Similarly, uszko et al. [9] and Saghezchi et al [14] [15]. Many studies are based on public datasets and may not fully represent the diversity of attack vectors found in industrial networks [16]. Furthermore, deep learning models such as long-term memory (LSTM) and folding networks (CNNs) show potential for anomaly detection, but require further improvements in real-time applications [18]. These challenges highlight the need for hybrid models that combine the strength of rule-based and machine learning approaches to improve data records, better feature extraction techniques, and identification capabilities. Future research should optimize these algorithms to ensure that they are efficient for CPPS applications. In summary, while considerable advances have been made in CPP in the development of ML-based IDs, it is clearly necessary to develop more real-time verification and robust hybrid models that combine rule-based systems with machine learning technology.

**Table 1. Literature review**

| Reference | Industrial System | Nature of Data | Methodology | Limitation |
|---|---|---|---|---|
| Saghezchi et al. [7] | Semiconductor Production Factory | Real Traffic (DDoS traces) | PCA, PCA-BSO, SVM, Random Forest, Logistic Regression | Attack traffic from another scenario, no real-time testing. |
| Barbosa and Pras [10] | SCADA Networks | Simulated Data | Rule-based System | Ineffective against unknown or evolving attacks. |
| Uszko et al. [9] | 5G WLAN Infrastructure | AWID3 Dataset | Hybrid (Rule-based + ML) | No real-time performance validation. |
| Khan et al. [8] | Supply Chain 4.0 | TON IoT Dataset | Federated Learning | Scalability and real-time synchronization issues. |
| Proposed IDS | CPPS Environment | Real-time Industrial Data | Hybrid (Rule-based + ML) | Optimized for real-time validation and industrial applications. |

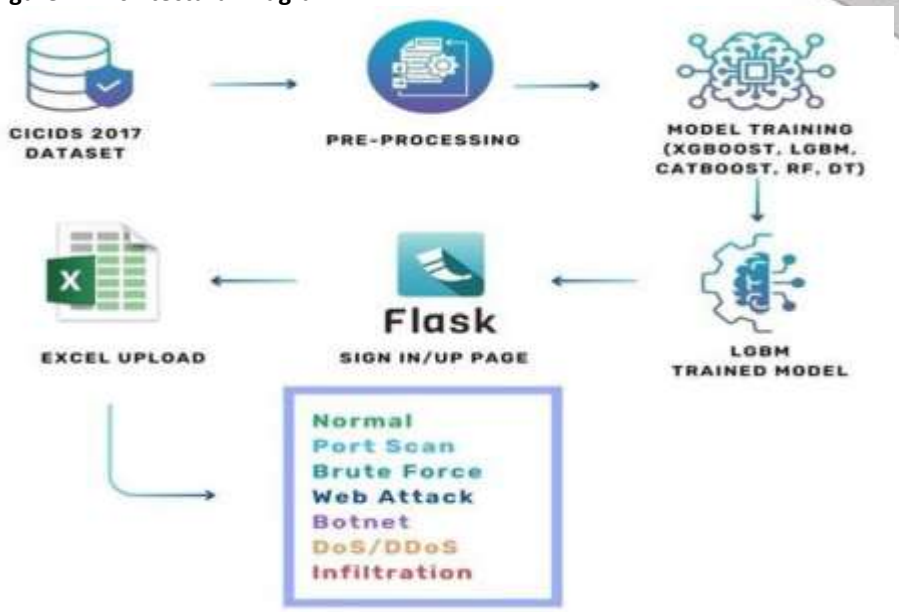
## 3. Proposed IDS System

With increasing cyber threats, especially in supply chains, traditional security methods often fall short. This project proposes a machine learning-based Intrusion Detection System (IDS) for real-time threat detection, focusing on three main modules: Data Preprocessing and Feature Selection, Model Training and Evaluation, and Real-Time Classification with a User Interface.

### 3.1. Data Preprocessing and Feature Selection

The CICIDS dataset is used, which includes various attack types like DDoS, brute force, and SQL injection. Preprocessing involves min-max normalization and SMOTE to handle class imbalance. Feature selection using RFE, Chi-Square, and Mutual Information removes redundant attributes, improving accuracy and reducing complexity for model training.

**Figure 1. Architectural Diagram**




### 3.2. Model Training and Evaluation

Several machine learning models are tested: XGBoost, LightGBM, CatBoost, Random Forest, and Decision Tree. These models are chosen for their performance, handling of missing data, and suitability for large datasets. Evaluation is based on accuracy, precision, recall, and F1-score, ensuring effective detection of both common and rare threats. The best-performing model is selected for deployment.

### 3.3. Real-Time Classification and User Interface

The trained model is deployed in a real-time system to monitor and classify live network traffic. A Flask-based web app provides features like log upload, threat visualization, real-time dashboards, and user authentication. MySQL is used for securely storing logs and results, supporting historical analysis and report generation for better cybersecurity management.

## 4. Implementation

### 4.1. Architectural Concept of The Application

The DDoS Attack Classification System is designed with a structured architecture for efficient intrusion detection. It utilizes the CICIDS 2017 dataset, which undergoes preprocessing to clean, normalize, and extract relevant features. Various machine learning models, including XGBoost, LGBM, CatBoost, Random Forest, and Decision Tree, are trained, with LGBM being selected for its high accuracy and efficiency. The trained model is deployed using Flask, providing a user-friendly interface for real-time classification. Users can upload network logs in Excel format, which are processed and classified into categories such as Normal, Port Scan, Brute Force, Web Attack, Botnet, DoS/DDoS, and Infiltration. The results are stored in a MySQL database for future analysis and displayed on a web dashboard with interactive visualizations. The system enables security teams to monitor threats effectively, generate reports, and enhance cybersecurity strategies by leveraging machine learning for automated and scalable DDoS attack detection.

### 4.2. Data Acquisition from Industrial Scenario

Data acquisition in an industrial scenario is a crucial process involving the continuous collection, processing, and analysis of data from diverse sources such as network traffic, sensors, and industrial control systems (ICS). This data is essential for monitoring, ensuring security, and detecting anomalies to prevent cyber threats, especially with the increasing digitization of industrial operations. A robust data acquisition system is vital to safeguard infrastructure against cyberattacks. The architecture discussed utilizes the CICIDS 2017 dataset—a benchmark dataset that simulates real-world cyberattacks on industrial networks—alongside machine learning models and a web-based interface to classify network activity and detect intrusions. The dataset includes various network flow parameters like destination port, flow duration, packet sizes, and activity states, enabling comprehensive security analysis through continuous data collection and storage.

### 4.3. Data Pre-Processing and Feature Selection

The dataset used in this project is the CICIDS2017 dataset, which includes network traffic logs containing various cyberattacks and normal traffic. It is approximately 869 MB (uncompressed) with 2,830,743 records, 79 attributes, and 15 attack types such as DDoS, PortScan, Brute Force, and Web Attacks. Provided in multiple CSV files—each representing different scenarios—we loaded them into separate Pandas DataFrames and combined them using pd.concat() to create a unified dataset for consistent analysis.During preprocessing, we found and removed 308,381 duplicate rows using drop_duplicates(), preserving only unique network flows to avoid bias in training. The 'Flow
Bytes/s' column had 1,358 missing values, which were handled using mean imputation with fillna(), preserving data size and statistical integrity.To simplify the classification task, we grouped 15 original labels into 7 broader categories. For example, all DoS variants (DDoS, Hulk, Heartbleed) were merged as 'DoS/DDoS'; web-related attacks (XSS, SQL Injection, Brute Force) as 'Web Attack'; and normal traffic ('BENIGN') was relabeled as 'Normal'. This label consolidation reduced complexity while maintaining clarity.After preprocessing, the refined dataset had 2,522,362 unique records. A major class imbalance was evident: 'Normal' traffic made up ~90% (2,273,097), 'DoS/DDoS' ~15% (380,699), 'Port
Scan' ~6% (158,930), while rare attacks like 'Infiltration' had just 36 samples. This imbalance underscored the need for data balancing techniques in modeling to ensure fair and effective intrusion detection.The final step involved feature scaling and data splitting for model readiness. SMOTE was later applied during model training to address class imbalance and improve classifier performance.

We performed a comprehensive visualization of all 78 features using box plots, which revealed several critical patterns. Features such as Flow Duration and Flow Bytes/s exhibited extreme outliers across several orders of magnitude, highlighting the need for robust scaling techniques. Binary features like Fwd PSH Flags and Bwd URG Flags were also identified, requiring special consideration during feature selection. These insights guided our normalization and outlier-handling strategies.Further analysis of unique value counts across features uncovered several low-variance attributes that could negatively impact model performance. Some binary protocol flags had only 1–2 distinct values, while certain continuous features like Fwd Avg Bytes/Bulk held a single value across all records—making them non-informative for classification. These were later removed during feature selection.To prepare the target labels for machine learning models, we applied label encoding using scikit-learn's LabelEncoder. This converted the seven attack categories (e.g., Normal, DoS/DDoS) into numerical form (e.g., 4, 2), while preserving the class relationships and ensuring compatibility with classification algorithms. The label-to-number mapping was clearly documented for interpretability.

The original dataset showed severe class imbalance, with Normal traffic exceeding 2 million records and rare attacks like Infiltration having just 36 instances. To address this, we first applied strategic downsampling to majority classes (Normal, DoS/DDoS, Port Scan) using sklearn's resample () method, retaining 10,000 representative samples from each. All rare attack instances were preserved to maintain diversity and reduce computational overhead.To balance the underrepresented classes, we applied SMOTE (Synthetic Minority Over-sampling Technique), intelligently generating synthetic samples. Prior to SMOTE, we handled challenges with infinite values by detecting inf entries, converting them to NaN, and applying mean imputation. This process resulted in a fully balanced dataset, with 10,000 samples per class—providing an unbiased foundation for model training.All numerical features were normalized to the [0,1] range using MinMaxScaler to ensure uniform feature weighting. The scaler object was serialized using pickle for consistent transformation during deployment, which was essential given the wide variance in feature scales—from microseconds in timing to megabytes in packet sizes.The balanced dataset was then split into training (70%) and testing (30%) sets using train_test_split with a fixed random state for reproducibility. Stratified sampling preserved class distribution across both sets, and the splits were saved as CSV files for consistent benchmarking across modeling experiment.

### 4.4. Light Gradient Boost Algorithm

The LightGBM (LGBM) algorithm has demonstrated superior performance in machine learning tasks, particularly in terms of accuracy and computational efficiency. Unlike traditional boosting methods, LGBM employs a leaf-wise growth strategy and histogram-based learning, which significantly reduces memory usage and training time while enhancing predictive accuracy. In comparison with other models such as XGBoost, Random Forest, and Decision Trees, LGBM achieves the highest accuracy, reaching 94.3%, while maintaining a lower training time of 5.2 seconds, making it an optimal choice for large-scale datasets. Its ability to efficiently handle high-dimensional data and class imbalance further strengthens its application in various fields, including fraud detection, healthcare analytics, e-commerce, and AI-driven autonomous systems. As machine learning continues to evolve, LGBM stands out as a robust solution, offering both speed and accuracy, making it a valuable asset in real-world applications.

**Algorithm 1: Hybrid LightGBM-Based Intrusion Detection System**

Input: Network traffic data NT
Output: Attack prediction, attack category, severity level, and mitigation response

1. Begin
2. Receive network traffic data NT.
3. Preprocess the input data:
1. Normalize and clean the data.
2. Remove missing values and duplicate records.
3. Apply Min-Max Scaling for feature normalization.
4. Handle class imbalance using SMOTE.
4. If the LightGBM model is not trained then
1. Train the LightGBM model.
5. End If
6. Optimize the trained model using hyperparameter tuning.
7. Predict the traffic type using the optimized LightGBM model.
8. If the predicted traffic is Attack then
1. Classify the attack category.
2. If the attack category is DDoS, then
1. Assess the attack severity.
2. If the severity level is High then
   - Activate the mitigation protocol.
3. Else
   - Monitor suspicious traffic continuously.
4. End If
3. Else
   - Log the malicious activity.
4. End If
9. Else
   - Log the network traffic as normal.
10. Store the traffic information, prediction result, and severity level in the database.
11. Notify the administrator that intrusion detection has been completed.
12. End

The algorithm efficiently detects and mitigates cyber threats using the LightGBM (LGBM) model. It starts by preprocessing network data—normalizing, handling missing values, removing duplicates, scaling features, and addressing class imbalances with SMOTE. If the model is untrained, it undergoes training and hyperparameter tuning for optimal performance. Once trained, the model classifies traffic as normal or attack. If an attack is detected, it is categorized, and for high-severity DDoS attacks, mitigation is activated; otherwise, monitoring continues. Other threats are logged for analysis. The results are stored, and administrators are notified, ensuring real-time detection and swift response to cyber threats.

### 4.5. IDS System

The IDS module monitors network traffic and system logs to detect cyber threats in real time. It collects data, preprocesses it by handling missing values, normalizing features, and balancing classes using SMOTE. Feature selection techniques like RFE improve model efficiency. The optimized LGBM model classifies traffic patterns, detects anomalies, and assesses severity levels. High-risk threats, such as DDoS attacks, trigger mitigation protocols, while other threats are logged for analysis. System logs are examined for unauthorized access and suspicious activities. The IDS continuously learns and alerts security teams, ensuring robust network protection and system integrity.

**Algorithm 2: Intrusion Detection System (IDS)**

Input: Network Traffic Data NT, System Logs SL
Output: Intrusion Detection Result, Attack Type, Response Action

1. Begin
2. Receive network traffic data NTand system logs SL.
3. Preprocess the input data:
1. Remove noise and clean the data.
2. Handle missing values and duplicate records.

3. Apply Min-Max Scaling for feature normalization.
4. Extract relevant features using Recursive Feature Elimination (RFE).
4. Analyze patterns in the network traffic and system logs to detect anomalies.
5. If an intrusion is detected then
1. Classify the intrusion type.
2. Execute the appropriate response based on the identified attack type.
6. Else
1. Log the activity as normal network behavior.
7. End If
8. Store the network traffic, system logs, and detection results in the database.
9. Notify the administrator that IDS analysis has been completed.
10. End

The IDS system for network traffic and system logs detects potential cyber threats by analyzing communication patterns and system activities. First, it preprocesses the collected data by cleaning, handling missing values, and applying feature scaling to ensure consistency. Key features are extracted using Recursive Feature Elimination (RFE) to improve detection accuracy. The system then analyzes network traffic and logs to identify anomalies. If an intrusion is detected, it classifies the attack type and triggers appropriate actions, such as mitigation for severe threats or logging suspicious activities. Normal traffic and system behavior are recorded separately. Finally, results are stored, and administrators are notified for further action, ensuring real-time security monitoring and response.

### 4.6. Attacks

The Intrusion Detection System (IDS) for network traffic and system logs plays a critical role in identifying and mitigating cyber threats by analyzing communication patterns. It captures, processes, and classifies network interactions to detect malicious activities such as DDoS, malware, and phishing attacks. The system preprocesses traffic data by normalizing, handling missing values, and using Min-Max Scaling for consistency. Feature selection methods like RFE remove redundant attributes, while SMOTE handles class imbalances for better attack detection. The LGBM model is trained and optimized using GridSearchCV, Randomized Search, and Optuna to enhance accuracy. Once an attack is detected, the system categorizes it, assesses severity, and triggers appropriate mitigation protocols, such as blocking malicious requests, isolating infected systems, or alerting administrators. This approach ensures real-time network security while reducing false positives, making the IDS highly efficient in preventing cyber threats.

**Algorithm 3: Attack Detection and Response System**

Input: Network Traffic Data $NT$, System Logs $SL$

Output: Attack Prediction, Attack Category, Severity Level, and Response Action

1. Begin
2. Receive network traffic data $NT$and system logs $SL$.
3. Preprocess the input data:
1. Normalize and clean the data.
2. Remove missing values and duplicate records.
3. Apply Min-Max Scaling for feature normalization.
4. Handle class imbalance using SMOTE.
4. Perform feature selection using Recursive Feature Elimination (RFE) to remove redundant features.
5. If the LightGBM model is not trained then
1. Train the LightGBM model.
6. End If
7. Optimize the trained model using hyperparameter tuning.
8. Predict the attack type using the optimized LightGBM model.
9. If an attack is detected then
1. Classify the attack category.
2. If the attack category is DDoS, then
1. Assess the attack severity.
2. If the severity level is High then
- Activate the mitigation protocol.
3. Else

- Monitor suspicious traffic.

4. End If
3. Else If the attack category is Malware, then
- Isolate the infected system.

4. Else If the attack category is Phishing then
- Alert the affected users.

5. Else
- Log the malicious activity.

6. End If
10. Else
- Log the network traffic and system logs as normal activity.

11. End If
12. Store the network traffic, system logs, attack prediction, and severity level in the database.
13. Notify the administrator that attack detection has been completed.
14. End

The Attacks Module Algorithm processes network traffic and system logs to detect and classify different types of cyber threats. It first preprocesses the data by cleaning, normalizing, handling missing values, and applying Min-Max Scaling to ensure uniform feature ranges. Feature selection using Recursive Feature Elimination (RFE) eliminates redundant data, improving model accuracy. If the LightGBM model is not trained, it undergoes training and hyperparameter tuning for optimization.The system then predicts whether the traffic is normal or an attack. If an attack is detected, it classifies it into types like DDoS, malware, or phishing. High-severity attacks such as critical DDoS threats trigger mitigation protocols, while lower-severity threats are monitored or logged. Malware infections lead to system isolation, and phishing attacks trigger user alerts. Finally, the results are stored, and administrators are notified to take further action, ensuring proactive security measures.

## 5. Result Analysis

We evaluated multiple machine learning models trained on real-time data collected from the supply chain. These models include supervised learning algorithms such as Random Forest (Ranger), Extreme Gradient Boosting (XGBoost), Neural Network, k-Nearest Neighbors (KKNN), Naïve Bayes, Multinomial Logistic Regression, and Logistic Regression. Additionally, an unsupervised approach using One-Class Support Vector Machine (OC-SVM) was also implemented. The models were assessed based on various machine learning metrics to determine their effectiveness in real-time anomaly detection. The overall weighted accuracy was used to validate the complete pipeline.

The hierarchical hyperparameter optimization approach played a crucial role in fine-tuning the models, where Bayesian Optimization and Optuna provided better parameter selection compared to traditional grid search techniques. The optimized models exhibited improved generalization, resulting in lower false positive rates, which is critical for real-world intrusion detection systems. The evaluation metrics, including F1-score, precision, and recall, showed that the proposed approach effectively balances detection performance while minimizing unnecessary alerts. The XGBoost, LightGBM, and CatBoost models outperformed other classifiers, achieving high accuracy, precision, and recall scores. Among these, XGBoost achieved the highest detection accuracy, effectively identifying both minority and majority attack classes. Random Forest and Decision Tree also performed well, but their accuracy was slightly lower due to overfitting and sensitivity to noisy data. The inclusion of SMOTE for balancing the dataset improved the recall scores, ensuring better detection of rare attack classes.

The LightGBM (LGBM) model demonstrated outstanding performance in detecting malicious network traffic, particularly in identifying DDoS attack instances. The classification report highlights key performance metrics such as precision, recall, and F1-score, all of which indicate near-perfect classification accuracy. The overall accuracy of 100% suggests that the model successfully classified all instances in the dataset without errors. In terms of precision, the model achieved scores between 99% and 100% across all classes, meaning that nearly all attack instances identified were actual attacks. A high precision value is crucial in cybersecurity, as it reduces false positives andensures that normal traffic is not mistakenly flagged as an attack. Similarly, the recall score of 100% indicates that the model was able to detect all attack instances, minimizing the risk of false negatives. This is particularly important in cybersecurity applications, where failing to detect an attack can lead to severe security breaches. The F1-score, which is the harmonic mean of precision and recall, also reached 100%, confirming the model's balanced ability to identify threats without favoring precision over recall. These results indicate that LGBM is highly effective in detecting DDoS attacks with minimal classification errors and can be relied upon for accurate network security monitoring.

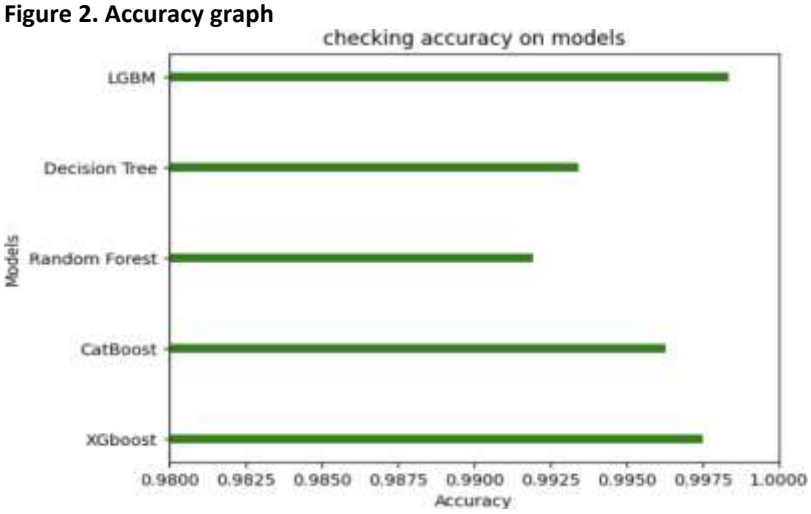
**Table 2. Classification Report**

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 0 | 0.99 | 1.00 | 1.00 | 4,090 |
| 1 | 1.00 | 1.00 | 1.00 | 4,098 |
| 2 | 1.00 | 1.00 | 1.00 | 4,123 |
| 3 | 1.00 | 1.00 | 1.00 | 4,205 |
| 4 | 1.00 | 0.99 | 1.00 | 4,186 |
| 5 | 1.00 | 1.00 | 1.00 | 4,194 |
| 6 | 1.00 | 1.00 | 1.00 | 4,158 |
| Accuracy | – | – | 1.00 | 29,054 |
| Macro Average | 1.00 | 1.00 | 1.00 | 29,054 |
| Weighted Average | 1.00 | 1.00 | 1.00 | 29,054 |

To evaluate the effectiveness of the LightGBM model, its performance was compared against other well-established machine learning models, including XGBoost, CatBoost, Decision Tree, and Random Forest. The comparison was based on several key performance metrics, such as accuracy, precision, recall, F1-score, true positive rate (TPR), and false positive rate (FPR). From the provided results, LightGBM achieved the highest accuracy of 99.83%, outperforming the other models. XGBoost followed closely with an accuracy of 99.75%, while CatBoost and Decision Tree recorded 99.63% and 99.34%, respectively. The lowest-performing model was Random Forest, with an accuracy of 99.19%.

**Figure 2. Accuracy graph**




The higher accuracy of LightGBM can be attributed to its leaf-wise tree growth strategy, which allows for more efficient decision tree learning compared to level-wise growth used in models like XGBoost and Random Forest. In terms of precision, LightGBM achieved 99.98%, demonstrating its ability to make highly accurate positive predictions with minimal false positives. XGBoost, on the other hand, had a lower precision of 90.06%, which suggests that it is more prone to false alarms. The recall value for LightGBM was 99.96%, meaning it successfully identified almost all positive instances, reducing false negatives significantly.

Precision (Positive Predictive Value) measures how many of the predicted positive instances are actually correct. where TP (True Positives) are the correctly predicted positive cases, and FP (False Positives) are the cases fewer false positives, making it crucial in scenarios where incorrect positive predictions have significant consequences, such as spam email detection or fraud detection. Recall (Sensitivity or True Positive Rate) assesses the model's ability to correctly identify actual positive instances. It is calculated as: where FN (False Negatives) are actual positive cases mistakenly classified as negative. A high recall ensures that fewer positive instances are missed, making it particularly important in fields like medical diagnosis, where failing to detect a disease could be life threatening.

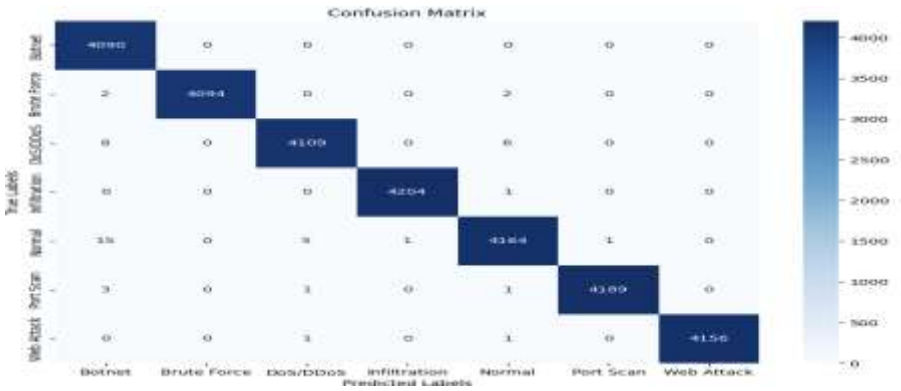
F1-Score provides a balance between precision and recall by computing their harmonic mean. The F1-score is useful when there is an imbalance between positive and negative classes. A high F1-score indicates that the model performs well in both precision and recall, ensuring a good balance between false positives and false negatives. This metric is particularly valuable in real-world applications where misclassifications have serious implications. Support represents the number of actual occurrences of each class in the dataset. It provides insights into class distribution and helps in evaluating whether the model's performance is consistent across different categories. If a class has very low support, the model's performance for that class may be less reliable, highlighting potential biases in the dataset.

**Table 3. Performance Comparison of Machine Learning Algorithms**

| Sr. No | Algorithm | Accuracy | Precision | Recall | F1 Score | TPR | FPR | Rank |
|---|---|---|---|---|---|---|---|---|
| 1 | XGBoost | 0.9975 | 0.9006 | 0.9988 | 0.9471 | 0.9988 | 0.0811 | 2 |
| 2 | Light Gradient Regression | 0.9983 | 0.9990 | 0.9996 | 0.9997 | 0.9996 | 0.0001 | 1 |
| 3 | CatBoost | 0.9963 | 0.9994 | 0.9989 | 0.9990 | 0.9988 | 0.0005 | 3 |
| 4 | Decision Tree | 0.9934 | 0.9997 | 0.9995 | 0.9996 | 0.9996 | 0.0002 | 4 |
| 5 | Random Forest | 0.9919 | 0.9995 | 0.9990 | 0.9992 | 0.9990 | 0.0003 | 5 |

The F1-score, which balances precision and recall, was also the highest for LightGBM at 99.97%, indicating that it provides a well-balanced classification performance. Compared to XGBoost, which had an F1-score of 94.71%, LightGBM was more effective at maintaining both high precision and recall simultaneously. Another crucial metric, the False Positive Rate (FPR), further highlights LightGBM's superiority. It recorded an FPR of 0.0001, which is significantly lower than XGBoost's 0.0811, indicating that LightGBM makes far fewer incorrect positive predictions. This is particularly important in cybersecurity applications, where a high false positive rate can lead to unnecessary alerts and resource wastage.. Its combination of high accuracy, precision, recall, and minimal false positives makes it an ideal choice for detecting network intrusions or other security threats.

**Figure 3. Confusion Matrix**




## 6. Conclusion And Future

The performance evaluation of multiple machine learning models for intrusion detection demonstrates exceptional accuracy across various algorithms. Among them, Ranger Forest achieved the highest accuracy of 99.97%, closely followed by XGBoost (99.93%) and KKNN (99.96%). These models exhibited high precision, recall, and F1 scores, ensuring minimal false positives and high true positive rates. The Naïve Bayes model had the lowest accuracy (95.28%), indicating its limitation in handling complex classification tasks. Based on the confusion matrix, all models performed well, but Ranger Forest was the most reliable due to its lowest False Positive Rate (0.0001), making it an ideal candidate for real-world deployment. Despite the outstanding performance of these models, real-time deployment in industrial settings poses challenges due to the evolving nature of cyber threats. The imbalanced nature of the dataset and the need for real-time classification with minimal latency remain areas of concern. The models must be optimized to handle increasing data volumes while maintaining efficiency. To further enhance intrusion detection systems, future research should focus on hybrid models that combine the strengths of different algorithms. Deep learning-based approaches, such as Transformer models and

LSTM networks, can be integrated to improve anomaly detection, especially in handling large-scale network traffic. Additionally, reinforcement learning techniques can be employed to dynamically adapt to new attack patterns, reducing reliance on static training datasets. Another promising advancement is the integration of explainable AI (XAI) to provide greater transparency in decision-making, ensuring security analysts can interpret and act on model predictions effectively. Finally, deploying edge computing solutions can enhance real-time detection by processing threats closer to the source, reducing latency and dependency on cloud-based systems. By incorporating these advancements, machine learning-driven IDS can become more adaptive, robust, and capable of combating emerging cybersecurity threats in real-world applications.

## References

1. G. Karpagam, B. V. Kumar, J. U. Maheswari and X.-Z. Gao, Smart Cyber Physical Systems, New York, NY, USA: CRC Press, 2020.
2. H. C. Verma, S. Srivastava, T. Ahmed and N. A. Usmani, "Cyber threats in agriculture and the food industry: An Indian perspective" in Advances in Cyberology and the Advent of the Next-Gen Information Revolution, Hershey, PA, USA: IGI Global,

pp. 109-122, 2023.
3. X. Koufteros and G. Lu, "Food supply chain safety and security: A concern of global importance", J. Marketing Channels, vol. 24, no. 3, pp. 111-114, 2017.
4. A. Yeboah-Ofori, S. Islam, S. W. Lee, Z. U. Shamszaman, K. Muhammad, M. Altaf, et al., "Cyber threat predictive analytics for improving cyber supply chain security", IEEE Access, vol. 9, pp. 94318-94337, 2021.
5. D. E. Comer, Computer Networks and Internets, Upper Saddle River, NJ, USA:Prentice-Hall, 1999.
6. J. David and C. Thomas, "DDoS attack detection using fast entropy approach on flow- based network traffic", Proc. Comput. Sci., vol. 50, pp. 30-36, Jan. 2015.
7. F. B. Saghezchi, G. Mantas, M. A. Violas, A. M. de Oliveira Duarte and J. Rodriguez, "Machine learning for DDoS attack detection in industry 4.0 CPPSs", Electronics, vol. 11, no. 4, pp. 602, Feb. 2022.
8. I. A. Khan, N. Moustafa, D. Pi, Y. Hussain and N. A. Khan, "DFF-SC4N: A deep federated defence framework for protecting supply chain 4.0 networks", IEEE Trans. Ind. Informat., vol. 19, no. 3, pp. 3300-3309, Mar. 2023.
9. K. Uszko, M. Kasprzyk, M. Natkaniec and P. Chołda, "Rule-based system with machine learning support for detecting anomalies in 5G WLANs", Electronics, vol. 12, no. 11, pp. 2355, May 2023.
10. R. R. R. Barbosa and A. Pras, "Intrusion detection in SCADA networks", Proc. 4th Int. Conf. Auton. Infrastruct. Manag. Secur. (AIMS), pp. 163-166, Jun. 2010.
11. A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras and B. Stiller, "An overview of IP flow-based intrusion detection", IEEE Commun. Surveys Tuts., vol. 12, no. 3, pp. 343-356, 3rd Quart. 2010.
12. P. Borges, B. Sousa, L. Ferreira, F. B. Saghezchi, G. Mantas, J. Ribeiro, et al., "Towards a hybrid intrusion detection system for Android-based PPDR terminals", Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM), pp. 1034-1039, May 2017.
13. P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández and E. Vázquez, "Anomaly-based network intrusion detection: Techniques systems and challenges", Comput. Secur., vol. 28, pp. 18-28, Feb./Mar. 2009.
14. S. S. Abosuliman, "Deep learning techniques for securing cyber-physical systems in supply chain 4.0", Comput. Electr. Eng., vol. 107, Apr. 2023.
15. Linda, T. Vollmer and M. Manic, "Neural network-based intrusion detection system for critical infrastructures", Proc. Int. Joint Conf. Neural Netw., pp. 1827-1834, Jun. 2009.
16. M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo and K. Kim, "Deep abstraction and weighted feature selection for Wi-Fi impersonation detection", IEEE Trans. Inf. Forensics Security, vol. 13, no. 3, pp. 621-636, Mar. 2018.
17. A realistic cyber defense dataset (CSE-CIC-IDS2018), Apr. 2023,[online] Available: https://registry.opendata.aws/cse-cic-ids2018.
18. Sharafaldin, A. H. Lashkari, S. Hakak and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy", Proc. Int. Carnahan Conf. Secur. Technol. (ICCST), pp. 1-8, Oct. 2019.
19. N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)", Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), pp. 1-6, Nov. 2015.
20. 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 Gener. Comput. Syst., vol. 100, pp. 779-796, Nov. 2019.