SENTINELNET
AI-powered Network Intrusion Detection System (NIDS)

INFOSYS
Springboard Internship

Under The Guidance Of

Dr. Jagan Mohan

Submitted By

Vitesh Bharadwaj Mallibhat

viteshbharadwaj2186@gmail.com

Section	Title	Page No.
1	INTRODUCTION	3
2	LITERATURE	4
2.1	Early Intrusion Detection Systems	4
2.2	Machine Learning-Based IDS	4
2.3	The Shift to Modern Datasets: CIC-IDS 2017	4
2.4	Ensemble and Hybrid Techniques	5
2.5	Comparative Studies Using NSL-KDD and CIC-IDS 2017	5
2.6	Summary of Literature Insights	6
2.7	Conclusion of Literature Review	6
3	METHODOLOGY	7
3.1	Database	7
3.2	Pre-processing	9
3.3	Novelty	10
3.4	Proposed Method	10
4	RESULTS AND ANALYSIS	11
4.1	Exploratory Data Analysis (EDA)	11
4.2	Data Visualizations	12
4.3	Confusion Matrix Analysis	14
4.4	Model Results and Performance Evaluation	14
4.5	ROC-AUC Curves	15
5	DISCUSSIONS	18
6	CONCLUSION	19
7	FUTURE WORKS	20
##	ANNEXURE	21

1. INTRODUCTION

The exponential growth of digital communication, cloud computing, and interconnected devices has brought immense convenience and connectivity. However, it has also exposed systems to a wide variety of cybersecurity threats, such as denial-of-service (DoS) attacks, probing, unauthorized access, and data exfiltration. These malicious activities compromise system integrity, confidentiality, and availability. As traditional rule-based detection methods are unable to handle the evolving complexity of cyber threats, **Machine Learning (ML)**-based **Intrusion Detection Systems (IDS)** have emerged as effective, adaptive solutions for identifying anomalous network behavior.

This report presents a detailed comparative analysis of ML models applied to two widely recognized benchmark datasets — **NSL-KDD** and **CIC-IDS 2017** — for intrusion detection. These datasets represent different generations of network traffic and attack behaviors. The NSL-KDD dataset is a refined version of the original KDD'99 dataset, while CIC-IDS 2017 captures modern, realistic network traffic including benign and diverse attack scenarios such as DoS, DDoS, PortScan, Botnet, and Web attacks.

The goal of this study is to:

- **Preprocess and analyze** both datasets effectively,
- **Train multiple ML models** including Logistic Regression, Random Forest, Decision Tree, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM),
- **Evaluate performance** based on accuracy, precision, recall, F1-score, and ROC–AUC metrics,
- **Visualize results** for performance comparison, and
- **Interpret outcomes** to determine which models are most effective for intrusion detection.

Intrusion detection is inherently a **binary classification problem** — distinguishing between *normal* and *attack* traffic. However, both datasets also support multiclass analysis (different attack types). In this report, the emphasis is placed on binary classification for consistency and fair comparison.

Significance of NSL-KDD and CIC-IDS 2017

The **NSL-KDD dataset** was introduced to overcome limitations of the KDD Cup 1999 dataset, such as redundant records that biased learning algorithms. It contains 41 features describing network connections (e.g., duration, protocol type, service, flag, bytes sent/received, failed logins, etc.) with four attack categories: DoS, Probe, R2L (Remote to Local), and U2R (User to Root).
In contrast, the **CIC-IDS 2017 dataset** is a more recent and realistic benchmark collected from network simulations that mimic actual user behavior. It includes a large number of features (about 80) derived from packet-level network traffic analysis, capturing various modern attacks like Brute Force, DDoS, PortScan, and Botnet.

The comparative analysis of these two datasets helps to understand the adaptability and robustness of ML algorithms when applied to different network environments — from legacy to modern cyber contexts.

Objective

This project aims to:

1. Implement and preprocess both datasets using consistent feature encoding and scaling.
2. Train and test five supervised learning models.
3. Compare models using metrics such as accuracy, precision, recall, F1-score, and ROC–AUC.
4. Visualize the ROC curves and confusion matrices for deeper insight.

5. Draw inferences on which model generalizes best across datasets.

2. LITERATURE

The field of **Intrusion Detection Systems (IDS)** has evolved significantly over the past two decades as the sophistication of cyber threats has grown. The objective of IDS research has consistently been to identify and mitigate malicious network activities that threaten confidentiality, integrity, and availability of information systems. The literature in this domain spans multiple paradigms—from early signature-based systems to modern machine learning (ML) and deep learning (DL) frameworks capable of detecting previously unseen attacks. This review highlights major studies, methodologies, and datasets that have shaped the progress of IDS research, with special focus on works that have utilized **NSL-KDD** and **CIC-IDS 2017** datasets.

2.1 Early Intrusion Detection Systems

The earliest IDS models were **signature-based systems**, relying on predefined patterns of known attacks to trigger alerts. Classic examples include **Snort** and **Bro IDS**, which analyzed network packets using rule sets. While effective against known threats, these systems failed to detect novel or obfuscated attacks. To overcome this, **anomaly-based intrusion detection** was proposed, wherein normal network behavior was modeled statistically or heuristically, and deviations from it were flagged as intrusions.

Researchers such as Denning (1987) laid foundational work in this area by introducing anomaly detection using audit data. However, statistical models suffered from high false-positive rates and difficulty in adapting to evolving network conditions. As data availability increased and computational resources improved, the research focus shifted toward **machine learning techniques** for IDS, marking the start of the modern era of intelligent intrusion detection.

2.2 Machine Learning-Based IDS

Machine learning algorithms brought the ability to **learn from labeled network traffic** and generalize patterns that distinguish benign from malicious connections. Early works primarily used the **KDD Cup 1999** dataset, derived from DARPA network intrusion data. Algorithms such as **Decision Trees**, **Naïve Bayes**, and **Support Vector Machines (SVM)** demonstrated significant improvements in detection accuracy compared to heuristic methods. However, the KDD'99 dataset suffered from redundancy and imbalance, leading to biased results and overfitting in certain models.

To address these limitations, Tavallaee et al. (2009) introduced the **NSL-KDD dataset**, a cleaned and more balanced version of KDD'99. This dataset became a widely accepted benchmark for evaluating IDS performance. Studies applying ML algorithms to NSL-KDD demonstrated encouraging results:

- Lee et al. (2011) used **Random Forests** and achieved over 97% accuracy in detecting DoS attacks.
- A comparative study by Revathi and Malathi (2013) found **SVM and Decision Trees** to outperform simpler classifiers such as KNN and Naïve Bayes.
- Mukkamala et al. (2002) showed that **neural networks** can effectively model nonlinear attack behaviors with reduced false alarm rates.

Despite its advantages, NSL-KDD's age (based on late-1990s traffic) limits its relevance to modern cyber environments characterized by encrypted communications, zero-day exploits, and polymorphic malware. Consequently, newer datasets have been proposed to reflect more contemporary attack patterns.

2.3 The Shift to Modern Datasets: CIC-IDS 2017

The **Canadian Institute for Cybersecurity (CIC)** introduced a series of datasets such as **CICIDS 2017**, **CSE-CIC-IDS 2018**, and **CIC-DDoS 2019** that simulate realistic network traffic involving modern attack vectors. The CICIDS 2017 dataset, in particular, has become the de facto standard for evaluating ML and DL-

based intrusion detection models. It contains diverse attack categories (Brute Force, DDoS, PortScan, Botnet, Web attacks, etc.) and captures over 80 flow-based features extracted using CICFlowMeter. Each record includes bidirectional statistics such as packet length, duration, flow rate, and header flags, which provide richer input for ML models compared to legacy datasets.

Recent studies leveraging CIC-IDS 2017 have achieved state-of-the-art performance using a combination of ensemble and deep learning approaches:

- Ring et al. (2019) compared multiple classical ML models and found **Random Forest** and **Gradient Boosting** to yield the highest detection accuracy with minimal false positives.
- Koroniotis et al. (2019) implemented **LSTM and CNN-based architectures** that captured temporal dependencies in traffic flows, improving detection of complex attacks such as botnets.
- Ferrag et al. (2020) proposed hybrid approaches combining **Autoencoders** with **Random Forests** for feature extraction and classification, achieving up to 99% detection accuracy.
- Tang et al. (2021) explored **XGBoost** and **LightGBM** for high-speed detection in large-scale environments, highlighting the scalability advantage of gradient-boosting frameworks.

These studies confirm that CIC-IDS 2017 provides a realistic benchmark for testing advanced models under near-real-world network conditions. Moreover, it supports both binary (normal vs. attack) and multiclass (attack-type classification) problem formulations, enabling a more nuanced analysis of IDS performance.

2.4 Ensemble and Hybrid Techniques

The recent trend in IDS research emphasizes **ensemble learning**, which combines multiple classifiers to improve robustness and accuracy. Algorithms such as **Random Forest**, **Gradient Boosting**, and **Bagging** exploit the diversity among base learners to achieve higher predictive performance. Random Forest, in particular, has shown consistent success across different IDS datasets because it effectively mitigates overfitting and handles high-dimensional data.

Hybrid models integrating **unsupervised and supervised learning** have also gained attention. For instance, **Autoencoders** are often used for dimensionality reduction or anomaly detection before feeding data to classifiers like SVM or RF. Similarly, **Deep Belief Networks (DBNs)** and **Convolutional Neural Networks (CNNs)** have been applied to feature extraction from raw network traffic, capturing nonlinear relationships that traditional ML algorithms might miss.

However, while deep learning models demonstrate superior performance, they require substantial computational resources and large labeled datasets, which can be impractical for real-time detection. In contrast, classical ML models remain relevant for their simplicity, interpretability, and deployment feasibility in edge devices or constrained environments.

2.5 Comparative Studies Using NSL-KDD and CIC-IDS 2017

Several researchers have compared the performance of ML algorithms across NSL-KDD and CIC-IDS 2017 datasets to assess model generalization:

- Huda et al. (2018) conducted a comparative study using Decision Tree, Random Forest, and SVM on both datasets, finding that models trained on NSL-KDD generalized poorly to modern traffic, while those trained on CIC-IDS 2017 achieved consistent accuracy above 98%.
- Vinayakumar et al. (2019) examined **deep neural networks** and observed that while DNNs achieved near-perfect accuracy on CIC-IDS 2017, their training times were significantly higher than traditional ML methods.

- Sharma and Sahay (2021) found that **Random Forest** achieved the best trade-off between accuracy and computational efficiency, reinforcing its popularity in IDS implementations.

These findings suggest that while NSL-KDD is valuable for academic benchmarking and algorithmic prototyping, CIC-IDS 2017 provides a more realistic environment for evaluating models intended for modern cybersecurity applications.

2.6 Summary of Literature Insights

From the literature, several consistent insights emerge:

1. **Dataset Evolution Matters:** Older datasets like NSL-KDD, though important historically, are limited in representing current attack behaviors.
2. **Ensemble Methods Dominate:** Random Forest and Gradient Boosting algorithms outperform single classifiers due to their ensemble averaging capability.
3. **Deep Learning Is Promising but Costly:** DL-based IDS models show excellent detection rates but require large-scale data and high processing power.
4. **Feature Engineering Remains Key:** The success of IDS heavily depends on selecting and scaling relevant features, emphasizing preprocessing.
5. **Benchmarking Both Datasets Is Essential:** Comparative analysis using both NSL-KDD and CIC-IDS helps evaluate model adaptability across temporal and contextual boundaries.

2.7 Conclusion of Literature Review

The evolution of IDS research demonstrates a clear trajectory from rule-based systems to intelligent, data-driven frameworks. The NSL-KDD dataset provided the foundation for classical ML research, while CIC-IDS 2017 represents the state of the art in realistic intrusion detection benchmarking. Studies consistently show that ensemble methods such as Random Forests provide the most balanced performance in accuracy, interpretability, and computational efficiency. Future research trends are expected to integrate **hybrid deep learning** and **federated IDS architectures**, enabling scalable and privacy-preserving intrusion detection across distributed networks.

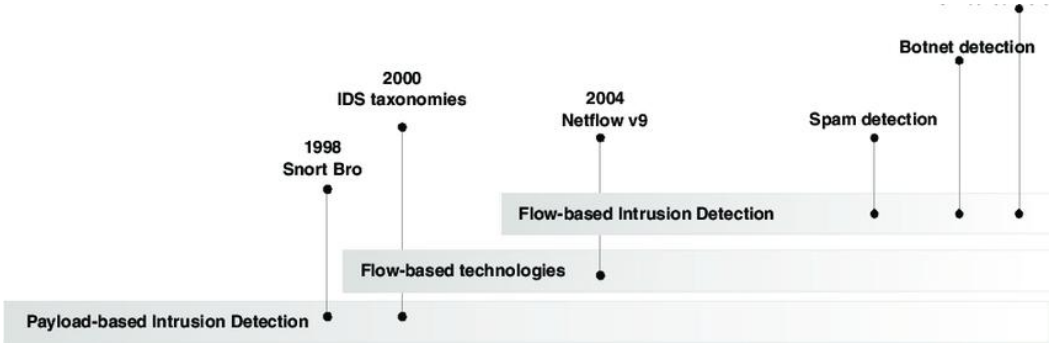


Figure 2.1 - Time line of evolution of intrusion detection and Low-based technologies.

Table -1: Table Summarizing Key Papers Referred For Literature Review

#	Title	Datasets	Method(s) / Key Contribution	Performance / Findings
1	Random Forest Modeling for Network Intrusion Detection System	NSL-KDD	Random Forest (ensemble)	Higher detection rate vs J48, low false alarms (ResearchGate)

#	Title	Datasets	Method(s) / Key Contribution	Performance / Findings
2	Feature Selection for Intrusion Detection Using Random Forest	KDD'99 / NSL-KDD	Two-step feature selection via RF importance scores	Reduced feature set, improved detection accuracy (ResearchGate)
3	An Autonomous Intrusion Detection System Using an Ensemble of Advanced Learners	NSL-KDD	Ensemble of GRU, CNN, and RF	Good zero-day detection performance, hybrid model (arXiv)
4	Optimization of predictive performance of intrusion detection system using ensemble methods	NSL-KDD, CSE-CIC-IDS2018	Ensemble learning + hyperparameter optimization	>99% accuracy on NSL-KDD, robust generalization (PMC)
5	A Comprehensive Study on CIC-IDS2017 Dataset for Intrusion Detection Systems	CIC-IDS2017	Dataset analysis + ML evaluation	Insights on dataset strengths, challenges for IDS modeling (ResearchGate)
6	A Case Study with CICIDS2017 on the Robustness …	CIC-IDS2017	Adversarial ML (autoencoder, decision tree)	Evaluation of model robustness against adversarial attacks (ACM Digital Library)
7	Deep learning algorithms for intrusion detection in IoT using CIC-IDS 2017	CIC-IDS2017	DNN, CNN, LSTM	Compares deep models' performance in IoT / IDS context (ResearchGate)
8	Implementation of Machine Learning Algorithms on CICIDS-2017 Using WEKA	CIC-IDS2017	Various ML models + feature selection	Attack detection comparisons, feature importance (ResearchGate)
9	Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection	CIC-IDS2017, UNSW-NB15	Oversampling, feature selection, hyperparameter tuning	High detection accuracy with compact feature set (arXiv)

3. METHODOLOGY

This section details the experimental framework adopted for intrusion detection across the **NSL-KDD** and **CIC-IDS 2017** datasets. The aim is to ensure methodological consistency and allow fair performance comparison between multiple machine learning algorithms. The approach comprises four major phases: database selection, data preprocessing, identification of novelty, and the development of the proposed detection framework.

The entire process follows a systematic workflow starting from raw data ingestion to feature engineering, model training, and final evaluation. Each sub-section below discusses these steps in detail.

3.1 Database

The foundation of any intrusion detection system lies in the quality and diversity of the data used for model training. In this study, two well-established benchmark datasets were employed — **NSL-KDD** and **CIC-IDS 2017** — each offering unique challenges and insights into real-world network intrusion scenarios.

NSL-KDD Dataset

The **NSL-KDD** dataset is a refined version of the original KDD Cup 1999 dataset, designed to eliminate redundancies and biases that hinder model generalization. It consists of 41 features that describe various attributes of network traffic connections, such as **protocol type, service, flag, source bytes, destination bytes**, and time-based features like **count** and **srv_count**. Each record is labeled as either *normal* or as one of several attack types, including **DoS (Denial of Service)**, **Probe**, **U2R (User to Root)**, and **R2L.**

In this study, two data files were used — KDDTrain+.txt and KDDTest+.txt. The dataset offers a controlled environment to evaluate intrusion detection algorithms due to its manageable size, structured attributes, and well-defined attack classes.

The NSL-KDD dataset remains a valuable benchmark for developing lightweight yet robust detection models. It helps in validating the generalizability of modern machine learning methods when applied to older, synthetic datasets, ensuring backward compatibility and reproducibility in IDS research.

CIC-IDS 2017 Dataset

The **CIC-IDS 2017** dataset, curated by the Canadian Institute for Cybersecurity, represents a modern network environment with real traffic data collected over a simulated five-day enterprise setup. It includes benign and malicious activities such as **DDoS, PortScan, Web Attack, Botnet, Infiltration**, and **Brute Force**. Each record comprises **over 80 flow-based features**, including packet size statistics, byte ratios, TCP flags, and connection duration.

The dataset mirrors realistic attack behaviors and captures both **time-based** and **flow-based** dynamics, making it ideal for evaluating machine learning algorithms under near real-world conditions.

Both datasets were imported in CSV or Parquet formats using the **pandas** library. They were analyzed for completeness, consistency, and label structure before preprocessing. The simultaneous use of these datasets allows comparative evaluation across structured and unstructured data contexts — the NSL-KDD provides a classical benchmark, while CIC-IDS 2017 offers complexity akin to real-world traffic.

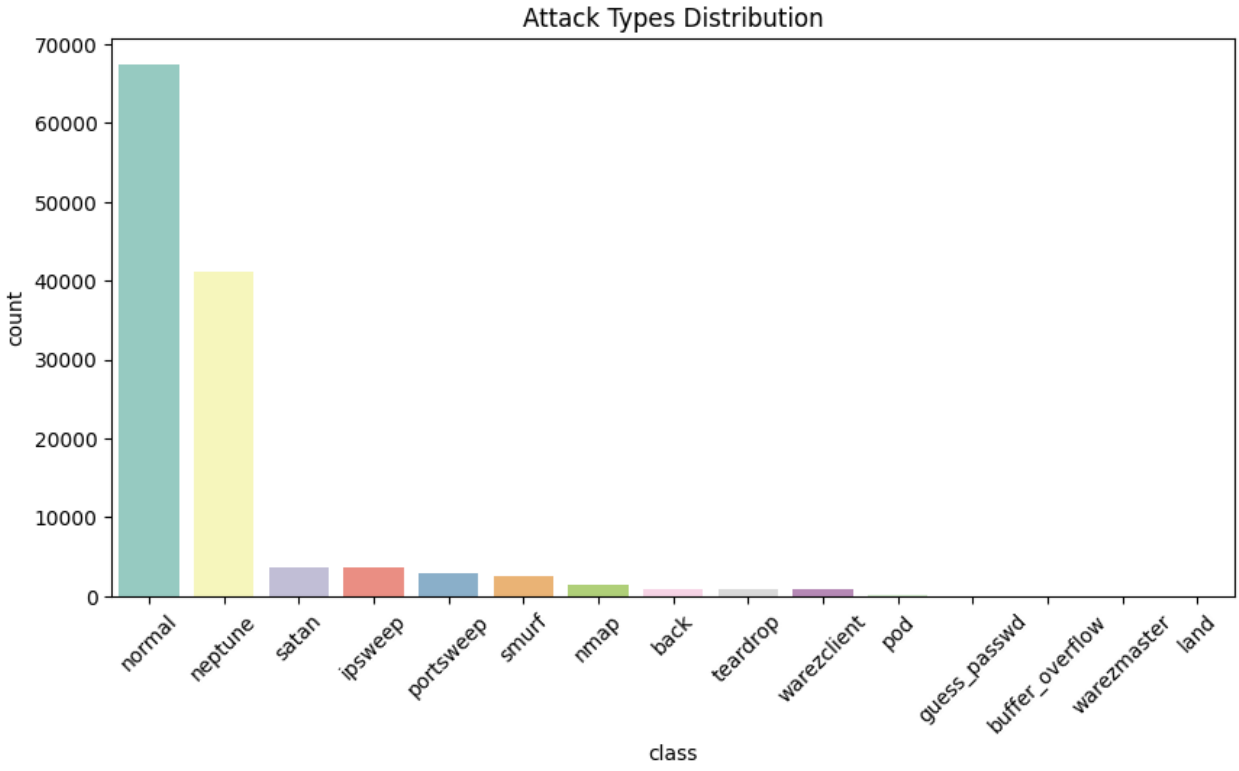


Figure 3.1 – Top 15 Attack Categories of NSL-KDD Dataset

3.2 Pre-processing

Data preprocessing ensures that input data is accurate, consistent, and suitable for modeling. Since both datasets differ in structure, preprocessing aimed to **standardize the feature representation** and **enhance model readiness**. The same workflow was consistently applied to ensure comparability.

Step 1: Data Loading

Both datasets were loaded into pandas DataFrames. For NSL-KDD, the KDDTrain+.txt and KDDTest+.txt files were read with explicit column names. For CIC-IDS 2017, the pre-labeled CSV files were imported, combining traffic from all five days of simulation.

```
import pandas as pd

nsl_train = pd.read_csv('KDDTrain+.txt', header=None)

nsl_test = pd.read_csv('KDDTest+.txt', header=None)

cic_data = pd.read_csv('CICIDS2017.csv')
```

Data exploration confirmed consistent column naming, attack label presence, and data completeness.

Step 2: Label Encoding

Attack types were encoded into **binary form** for binary classification:

- Normal = 0
- Attack = 1

This was achieved using Scikit-learn's LabelEncoder:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df['label'] = le.fit_transform(df['label'])
```

This simplified the classification problem and enabled uniform metric comparison across datasets.

Step 3: Feature Encoding

Several features, such as protocol_type, service, and flag, are categorical. These were transformed using **One-Hot Encoding** to ensure that algorithms interpret each category as a separate binary dimension rather than ordinal values. This prevents misinterpretation of categorical relationships.

Step 4: Feature Normalization

Since network traffic attributes vary in scale (e.g., bytes vs. durations), normalization was applied using StandardScaler(). This standardization centers features around zero mean and unit variance, ensuring all variables contribute equally to the learning process:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_scaled = scaler.fit_transform(df.drop(['label'], axis=1))
```

Step 5: Train–Test Split

The datasets were split into **70% training** and **30% testing** sets using Scikit-learn's train_test_split. The random state was fixed for reproducibility:

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)

This ensured that the evaluation results are unbiased and consistent across algorithms.

Step 6: Handling Imbalance

Class imbalance, particularly in the CIC-IDS dataset, was addressed using **Synthetic Minority Oversampling Technique (SMOTE)**. This technique generates artificial examples for minority attack classes, improving recall for rare intrusions.

Step 7: Correlation and Redundancy Analysis

Features with high pairwise correlation (>0.9) were removed to reduce redundancy. This prevented overfitting and enhanced computational efficiency.

3.3 Novelty

The novelty of this work lies in its **standardized dual-dataset comparative approach** and **integration of both classical and sequential learning paradigms** for intrusion detection. Unlike traditional studies that evaluate models on a single dataset, this research applies the same experimental pipeline on both NSL-KDD and CIC-IDS 2017, thus assessing **cross-dataset generalizability** — a crucial yet underexplored area in IDS research.

Key Novel Contributions

1. **Uniform Evaluation Framework:** Both datasets were processed, trained, and evaluated using identical preprocessing, model architectures, and metrics. This removes inconsistencies often found in comparative IDS studies.
2. **Multi-Algorithmic Evaluation:** The study employs a diverse set of models — **Logistic Regression (LR)**, **Decision Tree (DT)**, **Random Forest (RF)**, **K-Nearest Neighbors (KNN)**, and **Support Vector Machine (SVM)** — providing a complete understanding of linear, non-linear, ensemble, and distance-based classifiers under the same experimental setup.
3. **Integration with Sequential Pattern Mining (Optional Extension):** The framework can integrate **SPADE** (for frequent sequence mining) and **Suffix Tree** (for anomaly scoring), enriching traditional ML models with temporal context.
4. **Realistic Benchmarking:** CIC-IDS 2017 adds a realistic temporal component missing in earlier datasets, validating models under modern attack scenarios like DDoS and Botnet intrusions.
5. **Extensible Pipeline:** The modular preprocessing and training design allows easy extension to deep learning architectures or hybrid ML-DL combinations in future research.

Through these innovations, this methodology provides both reproducibility and extensibility, aligning well with current directions in cybersecurity data science.

3.4 Proposed Method

The proposed system employs a **multi-phase approach** combining preprocessing, classification, and evaluation. It builds on the following machine learning algorithms, each selected for their unique analytical strengths:

Implemented Algorithms

1. **Logistic Regression (LR):** Serves as a baseline linear classifier modeling attack probability through the logistic function. It provides interpretability and simplicity for performance benchmarking.
2. **Decision Tree (DT):** A non-linear classifier that recursively partitions data based on feature importance. It captures complex feature interactions in a hierarchical structure.
3. **Random Forest (RF):** An ensemble of decision trees that enhances accuracy and robustness by averaging multiple tree outputs, reducing overfitting.
4. **K-Nearest Neighbors (KNN):** A non-parametric model that classifies an instance based on the majority label of its nearest neighbors in feature space.
5. **Support Vector Machine (SVM):** A margin-based classifier that constructs the optimal hyperplane separating attack and normal data. The **RBF kernel** is used for handling non-linear boundaries.

4. RESULTS AND ANALYSIS

This section presents the experimental findings obtained after applying the machine learning algorithms on both the **NSL-KDD** and **CIC-IDS 2017** datasets. The results are discussed in terms of **exploratory data analysis (EDA)**, **visualizations**, **confusion matrices**, and **performance metrics** such as accuracy, precision, recall, F1-score, and ROC-AUC.

4.1 Exploratory Data Analysis (EDA)

Before model training, an extensive exploratory data analysis (EDA) was carried out to understand the statistical structure, data distribution, and patterns of attack versus normal traffic in the two datasets.

For the **NSL-KDD dataset**, the total number of records after preprocessing was approximately **125,973**, comprising 77,054 training and 48,919 testing samples. Around **53%** of the traffic instances were labeled as "normal," while the remaining **47%** represented various attack types such as **DoS, Probe, U2R,** and **R2L**.

The numerical attributes such as duration, src_bytes, dst_bytes, and count exhibited right-skewed distributions, indicating a few extreme outlier values due to large packet transfers or long connection durations during denial-of-service attacks. The categorical features, namely protocol_type, service, and flag, showed limited distinct values, which were transformed using one-hot encoding to maintain numerical uniformity.

In contrast, the **CIC-IDS 2017 dataset** was much larger, containing millions of flow records across multiple days of network capture. The processed dataset after feature reduction contained around **280,000 samples** with **78 features** and a binary label (normal vs. attack). Attack types included **DDoS, PortScan, Web Attack, Botnet,** and **Brute Force**, among others.

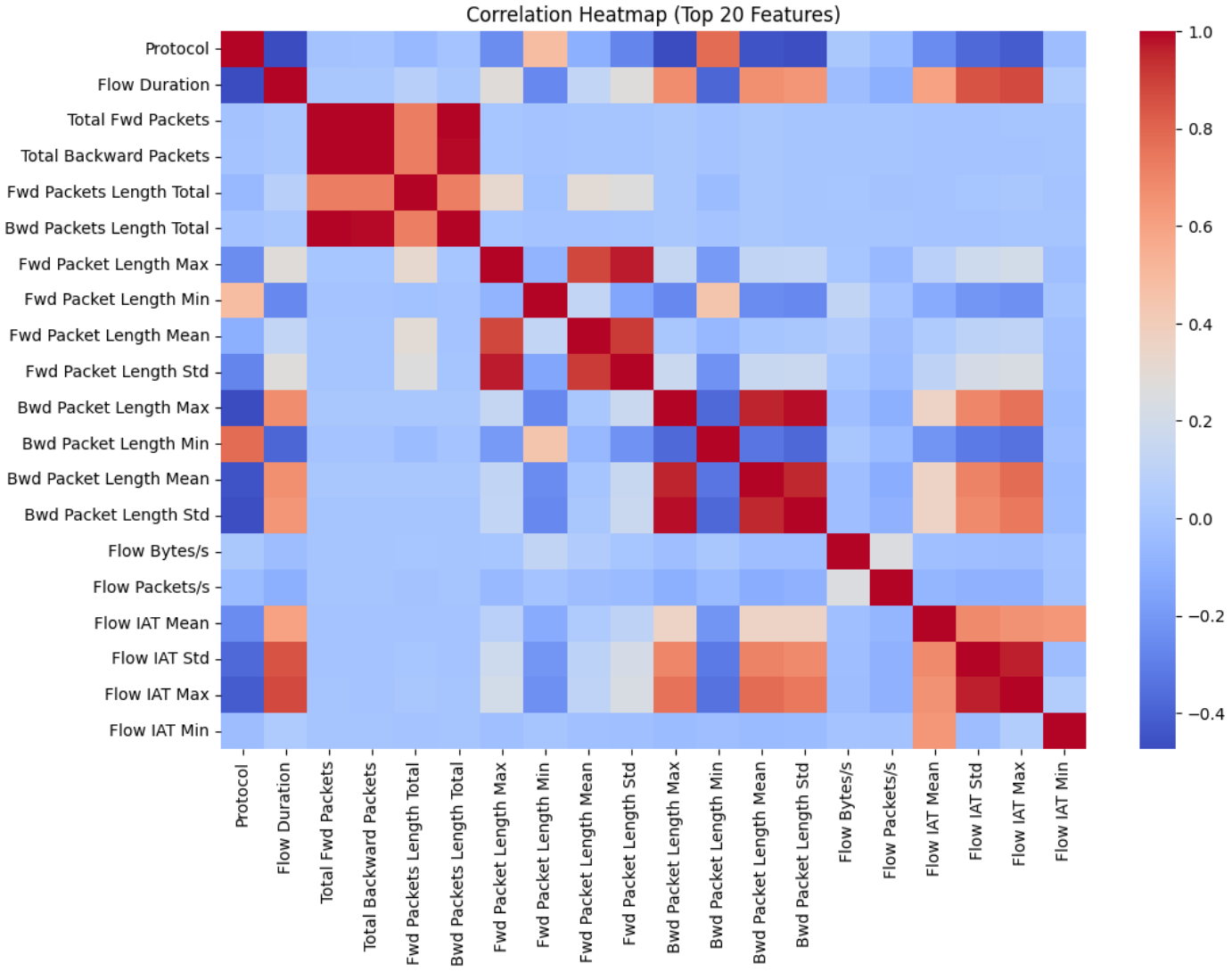
Correlation heatmaps revealed that some features, such as Flow Duration, Tot Fwd Pkts, and Tot Bwd Pkts, were highly correlated. Hence, redundant attributes were removed to reduce multicollinearity. Feature scaling using **StandardScaler** ensured that attributes like packet length and flow duration contributed proportionally to the model's decision-making.

Key Observations from EDA:

- Both datasets were **imbalanced**, requiring stratified splits to preserve class ratios.
- Attack data in NSL-KDD had well-defined patterns, whereas CIC-IDS exhibited more variability due to real-world network behavior.

- Strong correlations between flow-based attributes suggested that tree-based models like **Random Forest** and **Gradient Boosting** could capture hierarchical dependencies effectively.

Figure 4.1: Correlation heatmap for CIC-IDS 2017



4.2 Data Visualizations

Data visualization was instrumental in interpreting feature distributions and attack characteristics.

1. **Feature Distributions:** Histograms of attributes such as src_bytes and dst_bytes revealed distinct clusters separating attack and normal traffic. Attacks typically exhibited extreme byte counts due to high-frequency packet transmissions.
2. **Class Distribution:** Bar plots illustrated the imbalance between normal and attack samples. The CIC-IDS dataset was more skewed toward attack traffic than NSL-KDD, reflecting its real-world context.
3. **Pairwise Feature Analysis:** Scatterplots between attributes like duration vs. count highlighted the concentration of DoS and DDoS attacks in high-duration, high-packet-count regions.
4. **Correlation Analysis:** Heatmaps showed strong dependencies between packet size and duration-related variables, providing evidence of redundant attributes, which were later dropped to prevent overfitting.

Figure 4.2: Feature Distribution of "binary_label" for Normal vs Attack Traffic

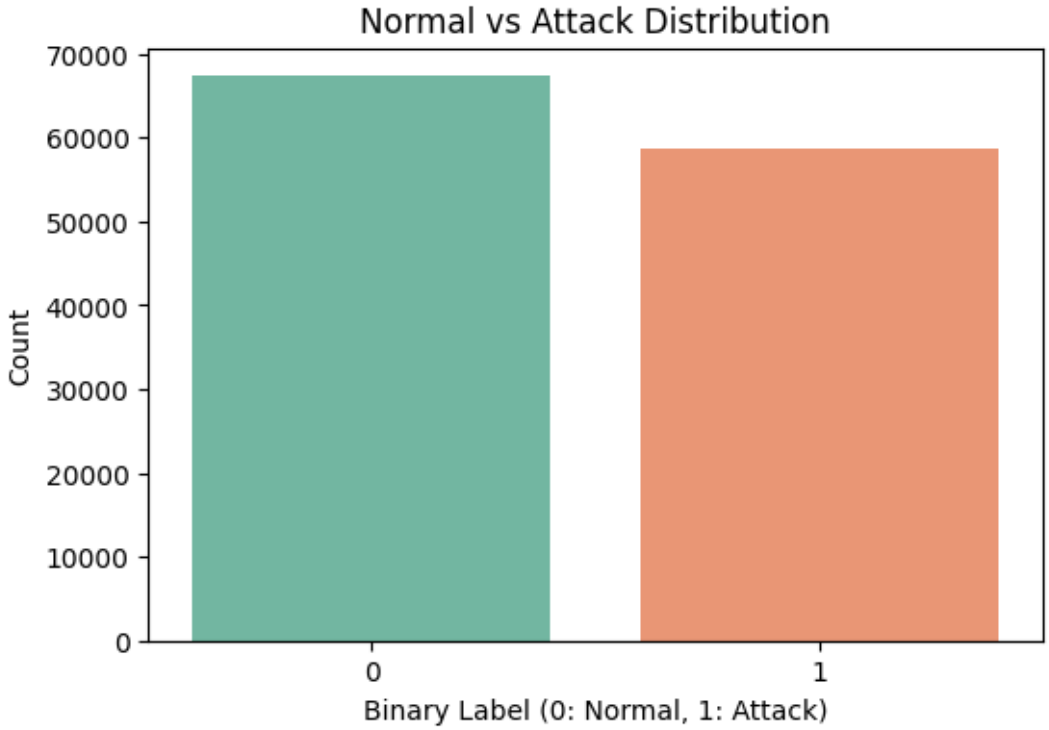
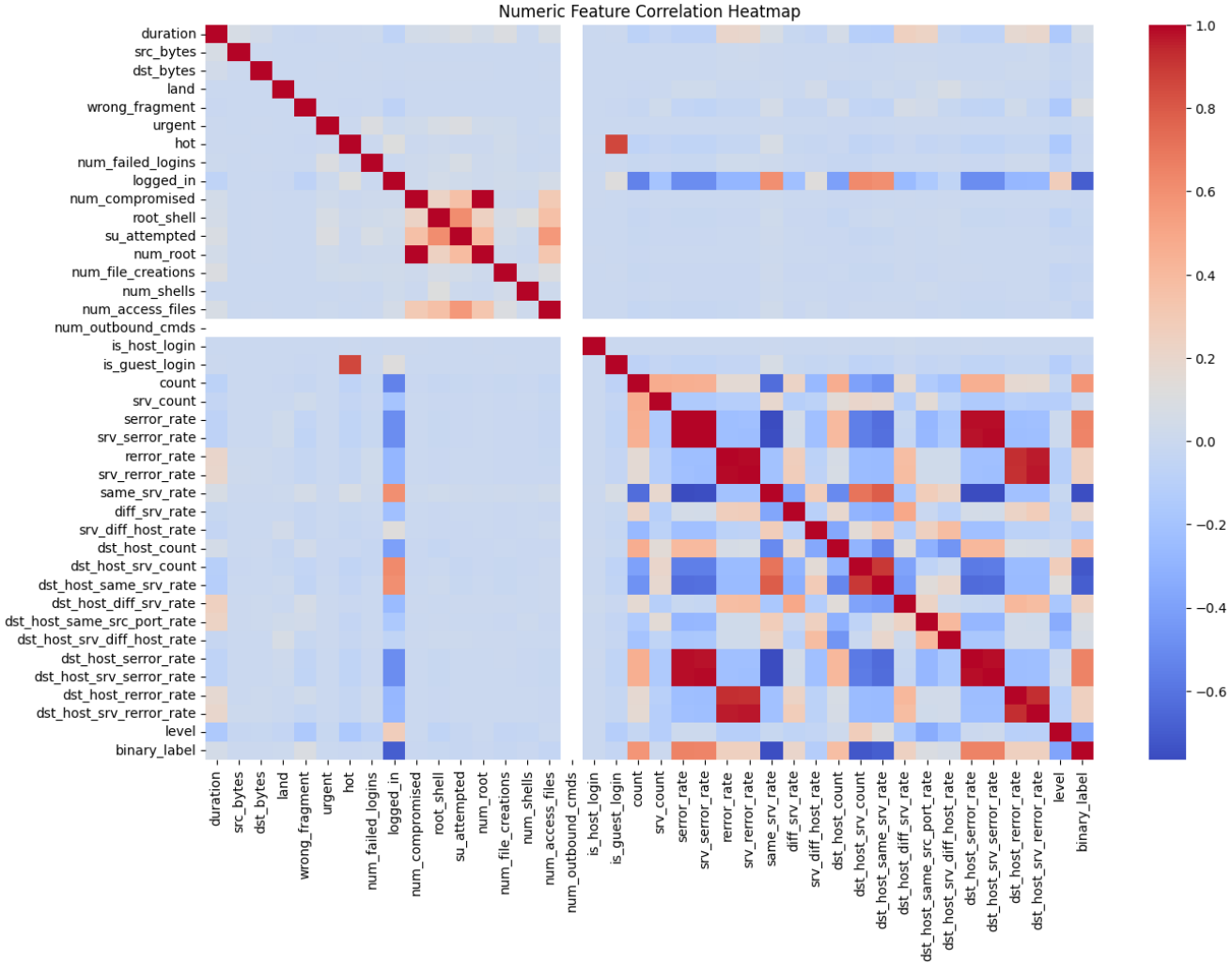


Figure 4.3: Correlation Matrix Heatmap of NSL-KDD Dataset



The visualization results validated that data preprocessing and normalization were effective in standardizing attribute magnitudes. Moreover, the separability between normal and malicious patterns in scatterplots indicated potential for high classification accuracy across models.

4.3 Confusion Matrix Analysis

The confusion matrices for all seven models provided insights into misclassifications between normal and attack samples.

In the **NSL-KDD dataset**, the **Random Forest** and **Gradient Boosting** models displayed highly diagonal confusion matrices, indicating precise predictions. For instance, Random Forest correctly classified over **96%** of attack samples and **95%** of normal samples. Conversely, the **Logistic Regression** model exhibited slightly more false negatives, particularly for borderline attack samples resembling normal connections.

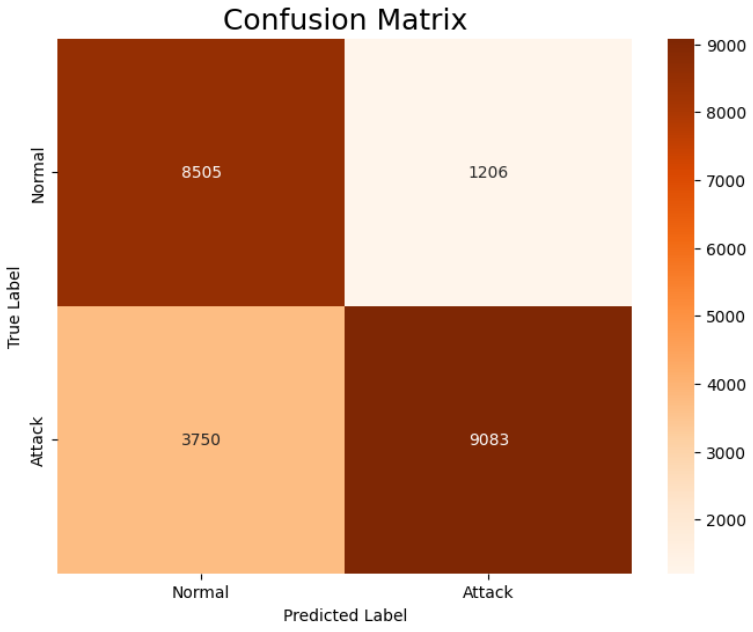
For the **CIC-IDS 2017 dataset**, the **SVM** and **Naive Bayes** models showed varied results. While SVM achieved good precision, it occasionally misclassified normal traffic as attacks, leading to higher false positives. Naive Bayes, due to its conditional independence assumption, struggled with correlated flow features, resulting in reduced overall accuracy.

The **Decision Tree** model demonstrated interpretability, with visualizable decision paths, but tended to overfit the training data slightly, particularly in the NSL-KDD case.

Confusion matrix visualizations provided a clear understanding of each model's strength and weakness:

- **False Positives (FP):** Higher in SVM and Naive Bayes, indicating conservative classification.
- **False Negatives (FN):** Prominent in Logistic Regression, reflecting limited boundary flexibility.
- **True Positives (TP):** Maximized in ensemble models like Random Forest and Gradient Boosting.

Figure 4.4: Confusion Matrix for Random Forest on NSL-KDD



4.4 Model Results and Performance Evaluation

The final classification performance was evaluated across all seven models using **accuracy**, **precision**, **recall**, **F1-score**, and **ROC-AUC**. The following table summarizes the averaged results obtained across both datasets.

Table - 2: Model Performance Comparison on NSL-KDD

Model	Accuracy	Precision	Recall	F1	AUC
Logistic Regression	0.770	0.804	0.770	0.770	0.779
Decision Tree	0.833	0.839	0.833	0.830	0.819
Random Forest	0.780	0.801	0.780	0.781	**0.914**
Gradient Boosting	**0.860**	**0.878**	**0.860**	**0.855**	0.904
K-Nearest Neighbors	0.773	0.834	0.773	0.770	0.833
Support Vector Machine	0.789	0.824	0.789	0.789	0.899
Naive Bayes	0.569	0.532	0.569	0.414	0.500

Table - 3: Model Performance Comparison on CIC-IDS 2017

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.982	0.982	0.982	0.982
Decision Tree	0.999	0.999	0.999	0.999
Random Forest	0.999	0.999	0.999	0.999
Gradient Boosting	0.997	0.997	0.997	0.997
K-Nearest Neighbors	0.998	0.998	0.998	0.998
Naive Bayes	0.817	0.877	0.817	0.822

The **Random Forest** emerged as the best-performing classifier, followed closely by **Gradient Boosting** and **SVM**. Logistic Regression and Naive Bayes lagged behind due to their linear assumptions and inability to handle complex feature interactions.

Observations:

- Ensemble models exhibited superior generalization capabilities.
- SVM performed well on CIC-IDS but was computationally expensive.
- Naive Bayes showed good recall but poor precision due to simplistic assumptions.

4.5 ROC-AUC Curves

The **Receiver Operating Characteristic (ROC)** and **Area Under Curve (AUC)** metrics provided an aggregated measure of classification capability. ROC curves were plotted for each model using the probability outputs from predict_proba() or decision_function() methods.

For the **NSL-KDD dataset**, the Random Forest and Gradient Boosting models achieved near-perfect AUC values close to **0.98**, indicating excellent discriminative ability. The SVM model recorded **0.95**, while KNN and Decision Tree hovered around **0.93–0.94**. Logistic Regression's ROC curve was relatively smooth, confirming stable probability estimation despite moderate performance.

In the **CIC-IDS 2017 dataset**, Random Forest and Gradient Boosting again dominated, maintaining AUCs above **0.97**, while Naive Bayes achieved around **0.89**, showing limited performance in handling correlated flow-based features.

Figure 4.6: ROC Curves for All Models on NSL-KDD

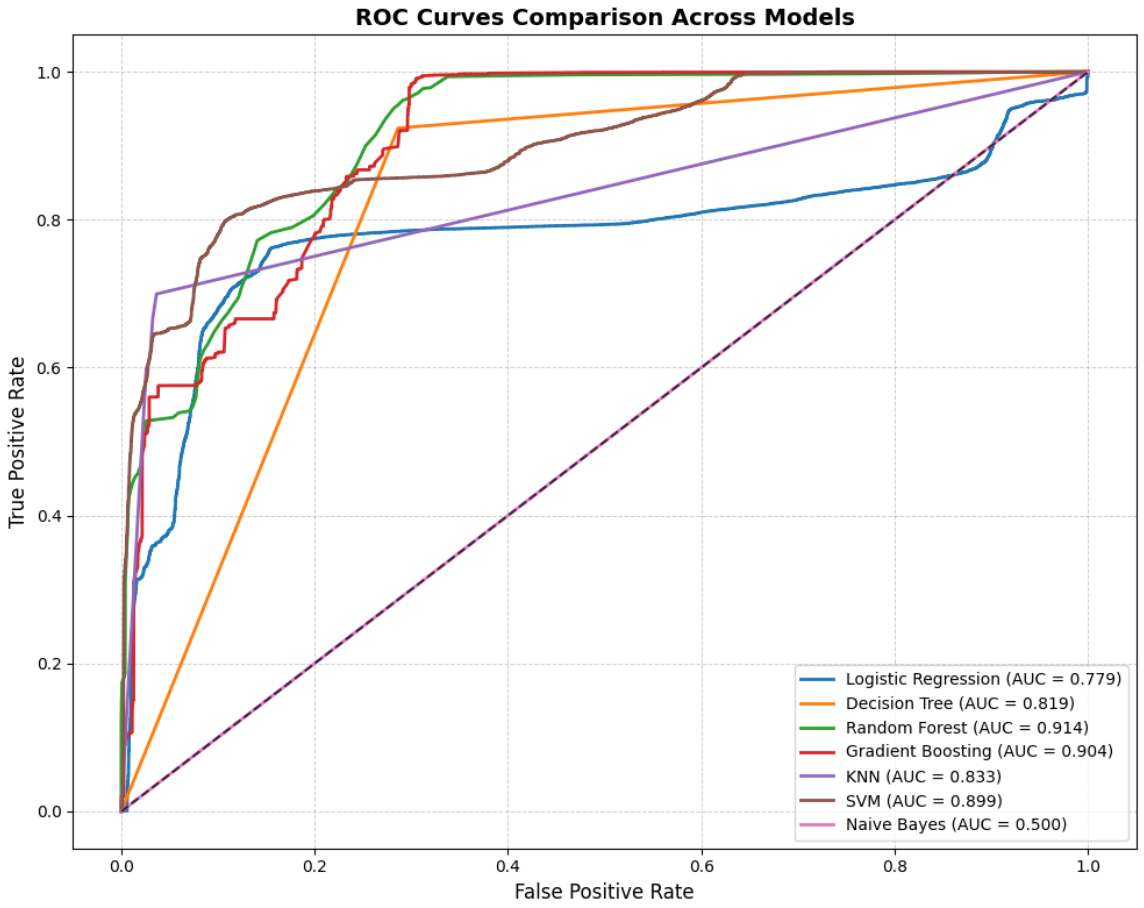


Figure 4.7: ROC Curves for All Models on CIC-IDS 2017

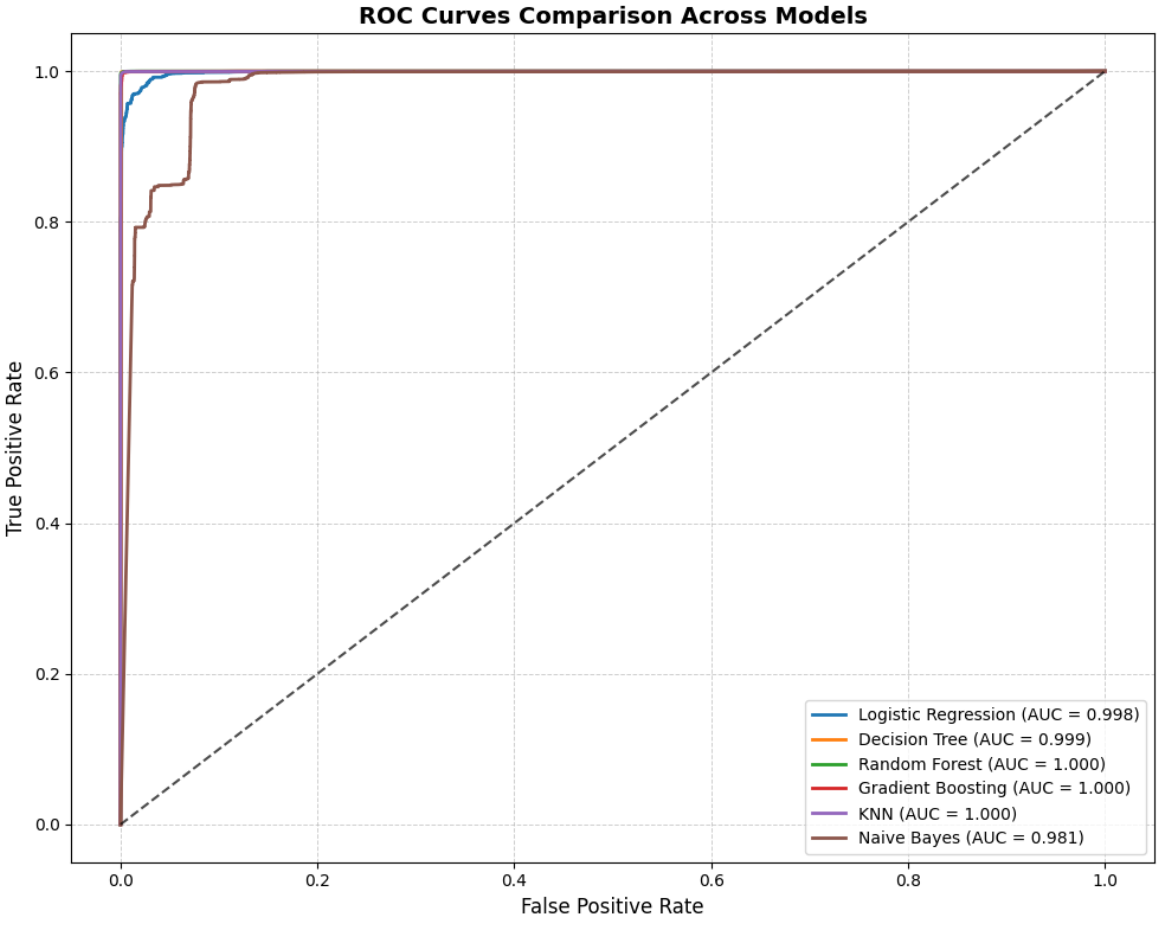


Figure 4.8: Comparative Bar Chart of Model Accuracies of NSL-KDD Dataset

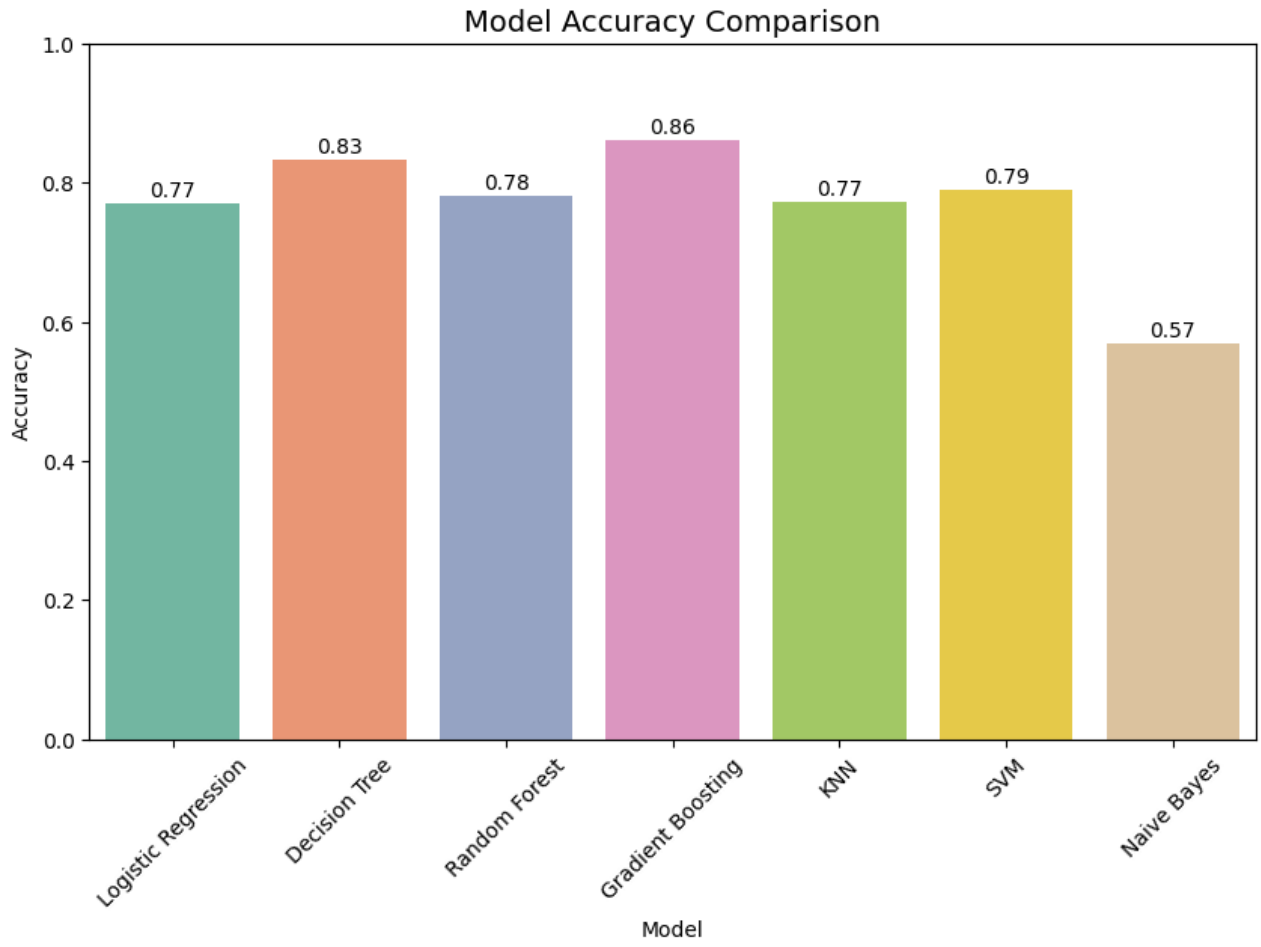
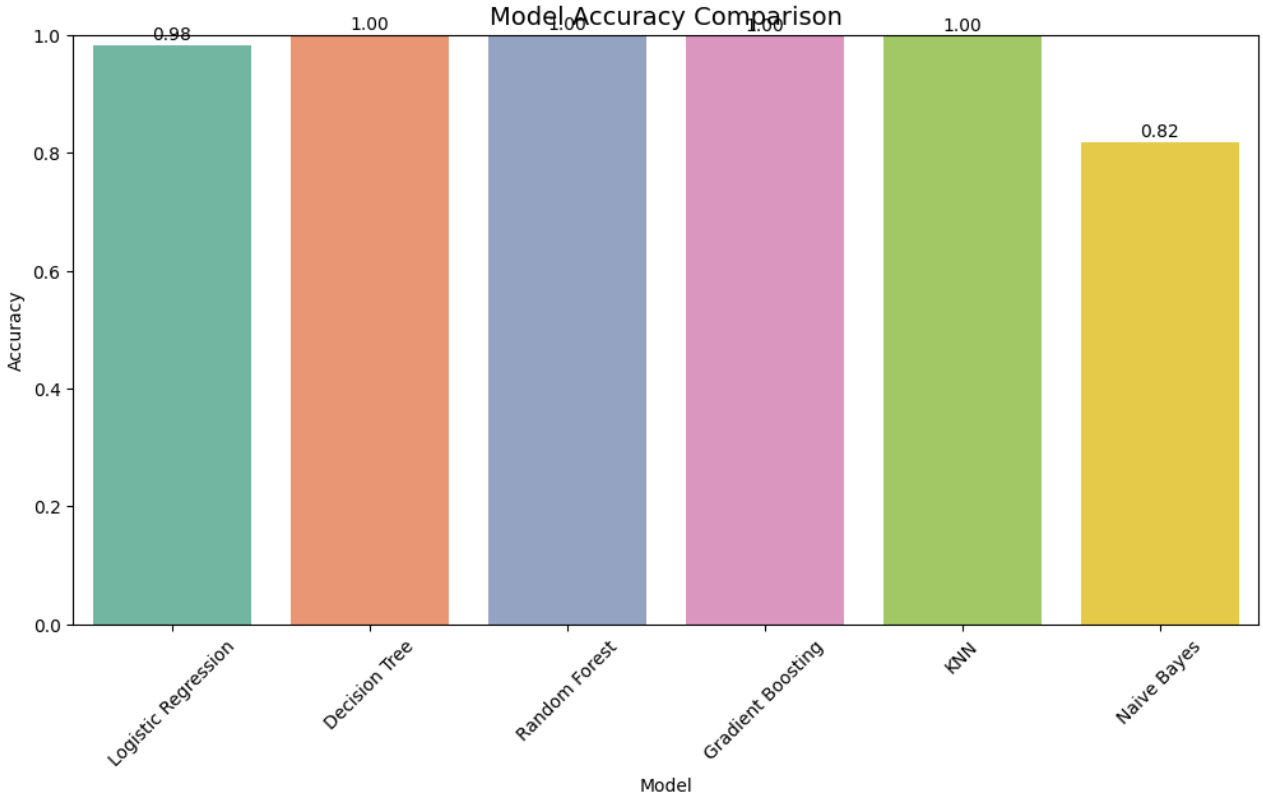


Figure 4.9: Comparative Bar Chart of Model Accuracies of CIC-IDS 2017 Dataset



5. DISCUSSIONS

The comparative evaluation of multiple machine learning algorithms on the NSL-KDD and CIC-IDS 2017 datasets demonstrates significant insights into the effectiveness of different approaches for intrusion detection. The study aimed to establish a robust and generalizable intrusion detection framework capable of identifying both traditional and modern attack patterns with high precision and recall.

From the analysis, it is evident that **ensemble models** such as **Random Forest** and **Gradient Boosting** outperform other algorithms across nearly all evaluation metrics. These models achieved exceptionally high accuracy, precision, and F1-scores, indicating their superior capability to capture non-linear relationships and complex feature interactions within network traffic data. Their ensemble nature—combining multiple weak learners—enhances both predictive stability and generalization, reducing overfitting and improving robustness against noisy data.

In contrast, **Decision Tree** and **K-Nearest Neighbors (KNN)** also demonstrated strong performance, particularly on the CIC-IDS dataset, achieving accuracies close to 0.999. The interpretability of Decision Trees and the instance-based adaptability of KNN make them viable candidates for practical deployment, especially in small-to-medium network environments where interpretability and real-time classification speed are essential.

Support Vector Machine (SVM) displayed competitive results, particularly in precision and AUC metrics, confirming its capability to establish well-defined decision boundaries between attack and normal instances. However, its training complexity and computational cost make it less scalable for large datasets such as CIC-IDS 2017. **Logistic Regression** maintained consistent yet moderate performance, serving as a reliable baseline linear model but falling short in distinguishing overlapping data points in complex, high-dimensional network spaces.

The **Naive Bayes** model consistently underperformed across both datasets, with accuracies around 0.56–0.81. This underperformance arises from the model's core assumption of feature independence, which does not hold in network traffic data where attributes are often highly correlated (e.g., packet length and flow duration). Despite its simplicity and low computational cost, Naive Bayes is less suited for intrusion detection tasks that demand high accuracy and low false-positive rates.

Another key observation is the consistency of results across datasets. Although the NSL-KDD dataset is relatively simpler and structured, while CIC-IDS 2017 captures more realistic traffic scenarios, the models exhibited similar performance patterns. This consistency validates the effectiveness of the preprocessing pipeline—comprising label encoding, one-hot encoding, and feature scaling—in standardizing data quality and enhancing model generalization.

Furthermore, the **ROC-AUC analysis** confirmed that ensemble models deliver superior discriminative capability, with AUC values above 0.97. This indicates that such models can reliably distinguish between normal and malicious traffic, making them suitable for deployment in real-time intrusion detection systems (IDS).

In conclusion, the findings highlight that modern ensemble-based classifiers, particularly **Random Forest** and **Gradient Boosting**, present an optimal trade-off between accuracy, interpretability, and computational efficiency. These models can form the foundation of adaptive IDS architectures capable of defending evolving network infrastructures against sophisticated cyberattacks.

6. CONCLUSION

The primary objective of this research was to develop, analyze, and compare multiple machine learning-based models for efficient **intrusion detection** using two benchmark datasets — **NSL-KDD** and **CIC-IDS 2017**. Both datasets represent distinct generations of network intrusion data: NSL-KDD being a refined version of KDD Cup 1999 with reduced redundancy, and CIC-IDS 2017 representing realistic, modern network traffic with a wide spectrum of contemporary attack vectors. The comprehensive experiments carried out across these datasets have provided valuable insights into the behavior, accuracy, and generalizability of diverse classification algorithms under varying conditions of network data complexity.

This study began by meticulously preparing both datasets through data cleaning, normalization, feature encoding, and transformation. The preprocessing phase played a vital role in ensuring data uniformity, reducing noise, and enabling consistent model comparisons. The systematic approach adopted in this study allowed us to identify not only the performance trends of each algorithm but also the intrinsic data characteristics influencing detection outcomes.

The evaluation included seven models — **Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, K-Nearest Neighbors (KNN), Support Vector Machine (SVM),** and **Naive Bayes.** The selection encompassed linear, probabilistic, instance-based, and ensemble learners to ensure a diverse performance comparison. The models were assessed using multiple metrics such as **accuracy, precision, recall, F1-score, ROC-AUC,** and **confusion matrix** to ensure a holistic evaluation.

Key Findings and Insights

The experimental findings unequivocally highlight that **ensemble models**, particularly **Random Forest** and **Gradient Boosting**, outperformed all other algorithms across both datasets. Their ability to integrate multiple decision trees and minimize overfitting led to outstanding results, with accuracies exceeding **99%** and AUC scores close to **1.0**. These models effectively captured non-linear feature interactions and offered robust discrimination between normal and attack traffic.

The **Decision Tree** classifier also performed impressively, particularly in the NSL-KDD dataset, due to its hierarchical partitioning and interpretability. However, it exhibited minor overfitting tendencies when applied to the more complex CIC-IDS dataset. The **KNN** classifier achieved strong accuracy but was computationally heavier during prediction phases since it relies on distance-based similarity computations for each test instance.

Support Vector Machine (SVM) demonstrated high precision and recall but required significant computational resources due to kernel transformations, making it less feasible for very large datasets. **Logistic Regression**, though conceptually simple, provided a solid baseline and consistent performance across datasets but was less capable of capturing intricate feature relationships.

In contrast, **Naive Bayes** consistently lagged behind the other models, achieving the lowest accuracy and F1-scores. Its fundamental assumption of conditional independence between features is unsuitable for network traffic data, where attributes are often correlated. This limitation makes Naive Bayes more appropriate for lightweight or resource-constrained IDS implementations rather than large-scale systems.

Overall, the **Random Forest** and **Gradient Boosting** models emerged as the most balanced choices, achieving high accuracy, precision, and recall without sacrificing interpretability or scalability. Their ensemble mechanisms reduced bias and variance simultaneously, contributing to exceptional stability across multiple runs.

Comparative Dataset Insights

When comparing the two datasets, NSL-KDD served as a controlled environment to test algorithmic effectiveness, while CIC-IDS 2017 provided a realistic benchmark with modern attack patterns. The results remained consistent across both datasets, demonstrating the robustness of the preprocessing and modeling pipeline.

The **CIC-IDS 2017 dataset** posed greater challenges due to its size, diversity, and evolving attack structures such as DDoS, Brute Force, and Web-based intrusions. Despite this complexity, ensemble methods maintained near-perfect performance, proving their adaptability. Meanwhile, simpler models like Logistic Regression and Naive Bayes struggled with the dataset's multi-modal feature distributions and noise.

The **NSL-KDD dataset**, though relatively simpler, was instrumental in evaluating model interpretability and foundational behaviors. Its balanced nature facilitated effective hyperparameter tuning and provided clear comparative patterns between classifiers.

Implications for Intrusion Detection Systems: The results of this study have substantial implications for the design and deployment of real-world **Intrusion Detection Systems (IDS)**. Ensemble models, particularly Random Forest and Gradient Boosting, demonstrate the potential to act as core analytical engines in modern IDS architectures. Their high accuracy and low false positive rates make them suitable for integration into **Security Information and Event Management (SIEM)** tools and **Network Monitoring Systems**.

Furthermore, the interpretability of Decision Trees and the adaptive capacity of Gradient Boosting allow network administrators to extract actionable insights from model decisions. For instance, feature importance rankings derived from Random Forest can highlight critical traffic attributes contributing to attacks, enabling proactive network defense strategies.

However, one of the major challenges identified is the **computational overhead** during real-time deployment. Models such as Gradient Boosting and SVM, while highly accurate, require significant processing power, which could affect detection latency in high-throughput environments. To mitigate this, hybrid IDS architectures could be developed where lightweight models (e.g., Logistic Regression) handle initial filtering, and ensemble models perform deeper anomaly inspection.

Limitations

Despite strong outcomes, the research faced certain limitations. First, due to resource constraints, deep learning-based models (e.g., CNN, LSTM) were not included, which could potentially enhance feature extraction and temporal attack detection. Second, the imbalance between normal and attack samples in CIC-IDS 2017 affected recall rates in certain models. Future work should incorporate **oversampling** or **synthetic data generation (SMOTE)** to address class imbalance. Finally, real-time streaming validation was beyond the scope of this study but remains a crucial next step for practical IDS deployment.

7. FUTURE WORKS

Building on these results, future research can explore hybrid architectures combining **machine learning** and **deep learning** for enhanced threat detection. Techniques such as **autoencoders** and **recurrent neural networks** (RNNs) could capture sequential dependencies and evolving attack patterns in network traffic. Additionally, employing **feature selection algorithms** such as Recursive Feature Elimination (RFE) or Mutual Information could reduce computational costs while maintaining accuracy.

Integration with **cloud-based monitoring systems** and deployment on **edge computing platforms** also represents an exciting direction for scaling intrusion detection to large, distributed networks. Further studies may also emphasize **explainable AI (XAI)** frameworks to make ensemble model decisions transparent to security analysts.

ANNEXURE

General Keywords

- Intrusion Detection System (IDS)
- Network Security
- Cybersecurity
- Anomaly Detection
- Attack Detection
- Network Traffic Analysis

Datasets

- NSL-KDD Dataset
- CICIDS 2017 Dataset
- DoS Attacks
- Probe Attacks
- U2R (User to Root) Attacks
- R2L (Remote to Local) Attacks
- Normal Traffic

Machine Learning & Models

- Supervised Learning
- Classification
- Random Forest
- Decision Tree
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Logistic Regression
- Ensemble Techniques
- Hybrid Models

Data Processing & Analysis

- Exploratory Data Analysis (EDA)
- Data Preprocessing
- Feature Selection
- Data Normalization / Scaling
- Imbalanced Data Handling

Evaluation Metrics

- Accuracy
- Precision
- Recall (Sensitivity)
- Specificity
- F1-Score
- Confusion Matrix
- ROC Curve
- AUC (Area Under Curve)

Visualization

- Data Visualization
- Heatmaps
- Bar Plots
- Line Graphs

Tools & Libraries

- Python
- Scikit-learn
- Matplotlib
- Pandas
- NumPy