Comprehensive Analysis of Botnet Behavior in IoT Networks:

Patterns, Impacts, and Detection of type of attack Using Machine Learning

Team Members

- Ankur Kaushal
- Dyuti Dasmahapatra
- Yash Verma
- Priyanshu Yadav

1. Discovery

Business Understanding

IoT Networks: Networks of devices that can connect to the internet and talk to each other.

Botnets: Groups of infected devices controlled by hackers without the owners knowing.

Threats from Botnets: Botnets can cause a lot of trouble in IoT networks.

Types of Attack

Denial of Service (DoS) Attacks

The botnet floods a target device or network with an overwhelming amount of traffic, rendering it unable to respond to legitimate requests.

Distributed Denial of Service (DDoS) Attacks

Similar to DoS, involve multiple devices in the botnet collectively targeting a single victim.

Data Theft and Privacy Breaches:

Used to steal sensitive information from IoT devices, such as personal data, login credentials, or financial information.

Pain points

- Vulnerabilities in IoT Devices
- Large Attack Surface
- Lack of Security Standards
- Limited Resources for Security Updates
- Privacy Concerns
- Potential for Large-Scale Disruptions

Problem Statement

Despite advancements in IoT security, botnets continue to pose significant threats to IoT networks, compromising data integrity and system performance.

This study aims to comprehensively analyze botnet behavior in IoT networks, including attack patterns, impacts, and detection mechanisms using machine learning.

Analytic problem type

Classification problem: Identifying the type of botnet attacks within IoT network traffic data.

Objectives

Gain insights into botnet behavior and characteristics within IoT networks through an in-depth analysis of attack patterns, features, and impacts.

Develop accurate detection mechanisms using machine learning to identify various types of botnet attacks in IoT networks.

Success Criteria

Successfully identify and analyze botnet attack patterns in IoT network traffic data.

Develop machine learning models with high detection accuracy for different types of botnet attacks.

Demonstrate the effectiveness of the detection mechanisms by reducing the false positive rate and response time

Key Risks

Limited availability of comprehensive and diverse datasets for training machine learning models

Ensuring that machine learning models generalize well to new and unseen botnet attack instances.

Limited understanding of IoT network architecture, protocols, and security vulnerabilities

Initial Hypothesis

Hypothesis 1

Botnet attacks in IoT networks exhibit distinct patterns in network traffic data, including abnormal packet and byte counts, unusual traffic rates, and atypical protocol usage.

Hypothesis 2

Machine learning models trained on diverse datasets containing features indicative of botnet activities can accurately detect and classify different types of botnet attacks in IoT networks.

Use of the 3 V’s

Volume

Analyzing large volumes of IoT network traffic data to identify patterns and anomalies indicative of botnet attacks.

Variety

Handling diverse types of network traffic features, including source and destination IP addresses, ports, protocols, packet counts, and traffic rates.

Velocity

Detecting botnet attacks in real-time or near real-time to enable proactive defense measures and minimize the impact on IoT network infrastructure.

Resource Requirements

Data: Access to comprehensive and labeled datasets containing IoT network traffic data with instances of botnet attacks.

Computing Resources: Sufficient computational power and storage capacity to preprocess, analyze, and train machine learning models on large datasets.

Expertise: Domain knowledge in IoT security, data preprocessing, machine learning, and cybersecurity to effectively conduct the analysis and develop detection mechanisms.

Software Tools: Utilization of programming languages (e.g., Python), libraries (e.g., scikit-learn, TensorFlow), and data visualization tools for data analysis and model development.

2. Data Perparation

	pkSeqID	proto	saddr	sport	daddr	dport	seq	stddev	N_IN_Conn_P
0	792371	udp	192.168.100.150	48516	192.168.100.3	80	175094	0.226784	
1	2056418	tcp	192.168.100.148	22267	192.168.100.3	80	143024	0.451998	
2	2795650	udp	192.168.100.149	28629	192.168.100.3	80	167033	1.931553	
3	2118009	tcp	192.168.100.148	42142	192.168.100.3	80	204615	0.428798	
4	303688	tcp	192.168.100.149	1645	192.168.100.5	80	40058	2.058381	
5	420025	tcp	192.168.100.149	39733	192.168.100.5	80	156396	2.177835	
6	3008812	udp	192.168.100.147	10800	192.168.100.3	80	118034	1.368196	
7	1064106	udp	192.168.100.150	19625	192.168.100.3	80	184672	1.788	
8	3258414	udp	192.168.100.147	22692	192.168.100.3	80	105486	0.82	
9	1793063	tcp	192.168.100.148	39738	192.168.100.3	80	141822	0.03	

DATASET

Dataset Description

- We selected the Dataset from kaggle name - **Bot-Iot 2018.**
- The dataset comprises of 18 columns and around 8 million rows and after preprocessing we were left with 12 columns.
- The main columns are Category which contains the types of attack: Dos and DDos

3. EDA

4. Model Planning

Naive Bayes

- Naive Bayes is a simple probabilistic classifier based on Bayes' theorem with the assumption of independence between features.
- Assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Logistic Regression

- Statistical method for binary classification that models the relationship between the independent variables and the probability of a binary outcome using the logistic (sigmoid) function.
- Unlike Naive Bayes, Logistic Regression does not assume independence between features; instead, it estimates the impact of each feature on the log-odds of the outcome directly from the data.

XGBoost

- XGBoost (Extreme Gradient Boosting) is a scalable and efficient implementation of gradient boosting.
- Builds multiple decision trees sequentially and combines their predictions to improve accuracy.

Decision Tree (Information Gain)

- Decision trees recursively split the dataset based on feature values to make predictions.
- Information gain is used as a criterion to select the best split at each node.
- Information gain measures the reduction in entropy or impurity after a dataset is split.
- Higher information gain indicates a better feature for splitting.

Decision Tree (Gini)

- Instead of information gain, Gini impurity is used as a criterion to measure the impurity of a dataset.
- It measures how often a randomly chosen element would be incorrectly classified.

Random Forest

- Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes.
- Reduces overfitting and improves generalization by averaging multiple decision trees.

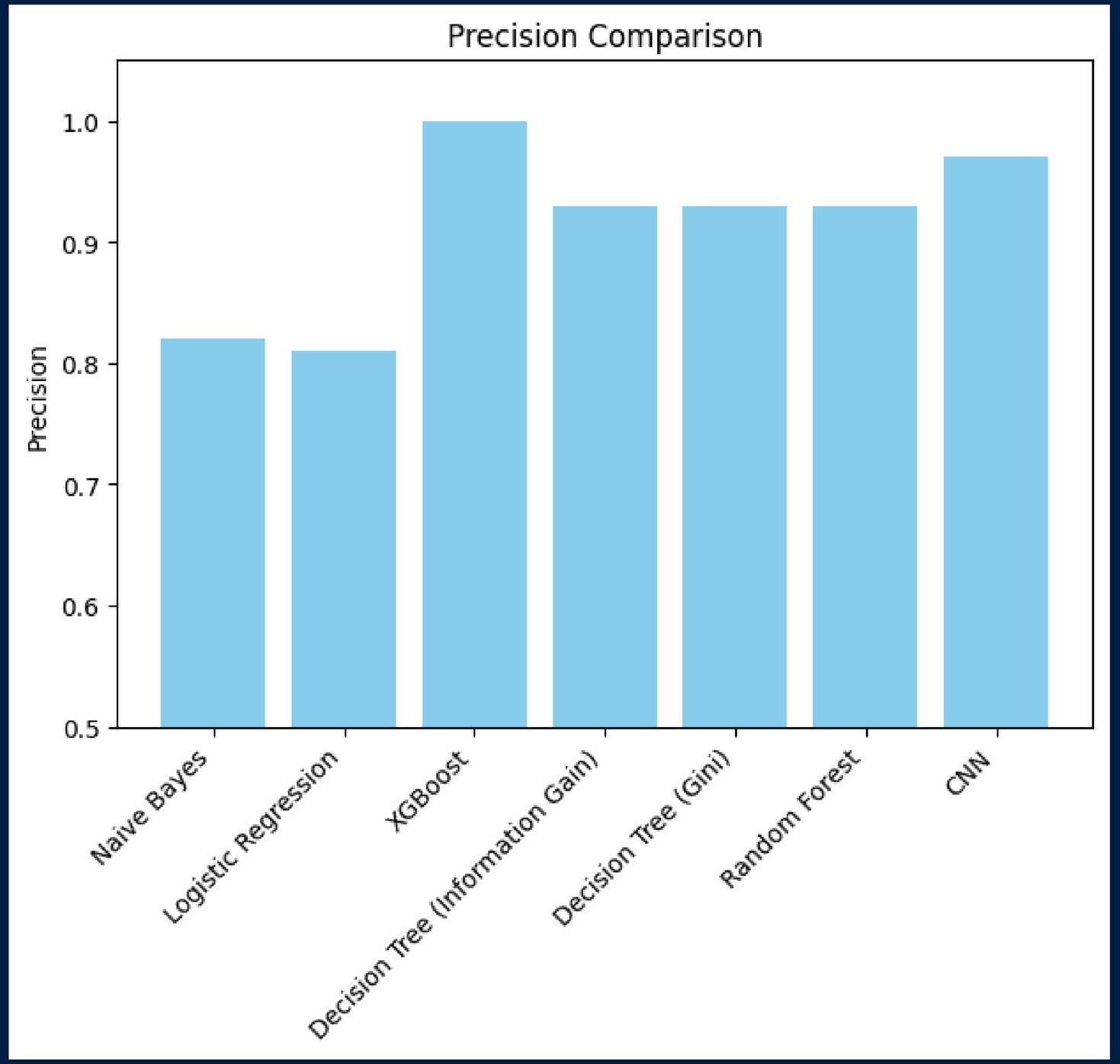
Convolutional Neural Network (CNN)

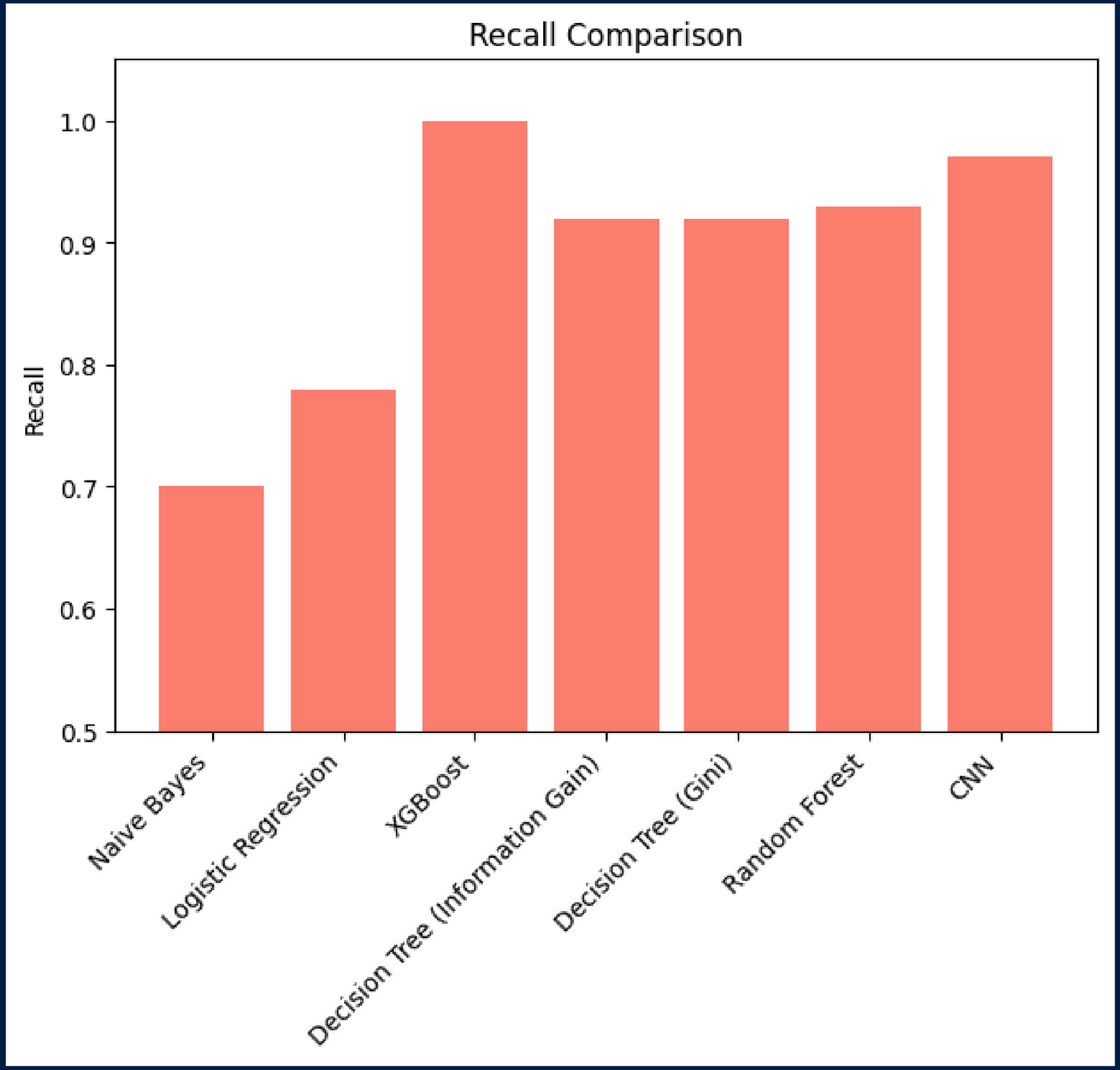
- Instead of information gain, Gini impurity is used as a criterion to measure the impurity of a dataset.
- It measures how often a randomly chosen element would be incorrectly classified.
- Deep learning models are adept at processing raw spatial data by automatically learning hierarchical features through multiple layers of non-linear transformations.
- Leveraging convolutional layers and specialized architectures, these models excel in tasks like image recognition and object detection, outperforming traditional machine learning methods.

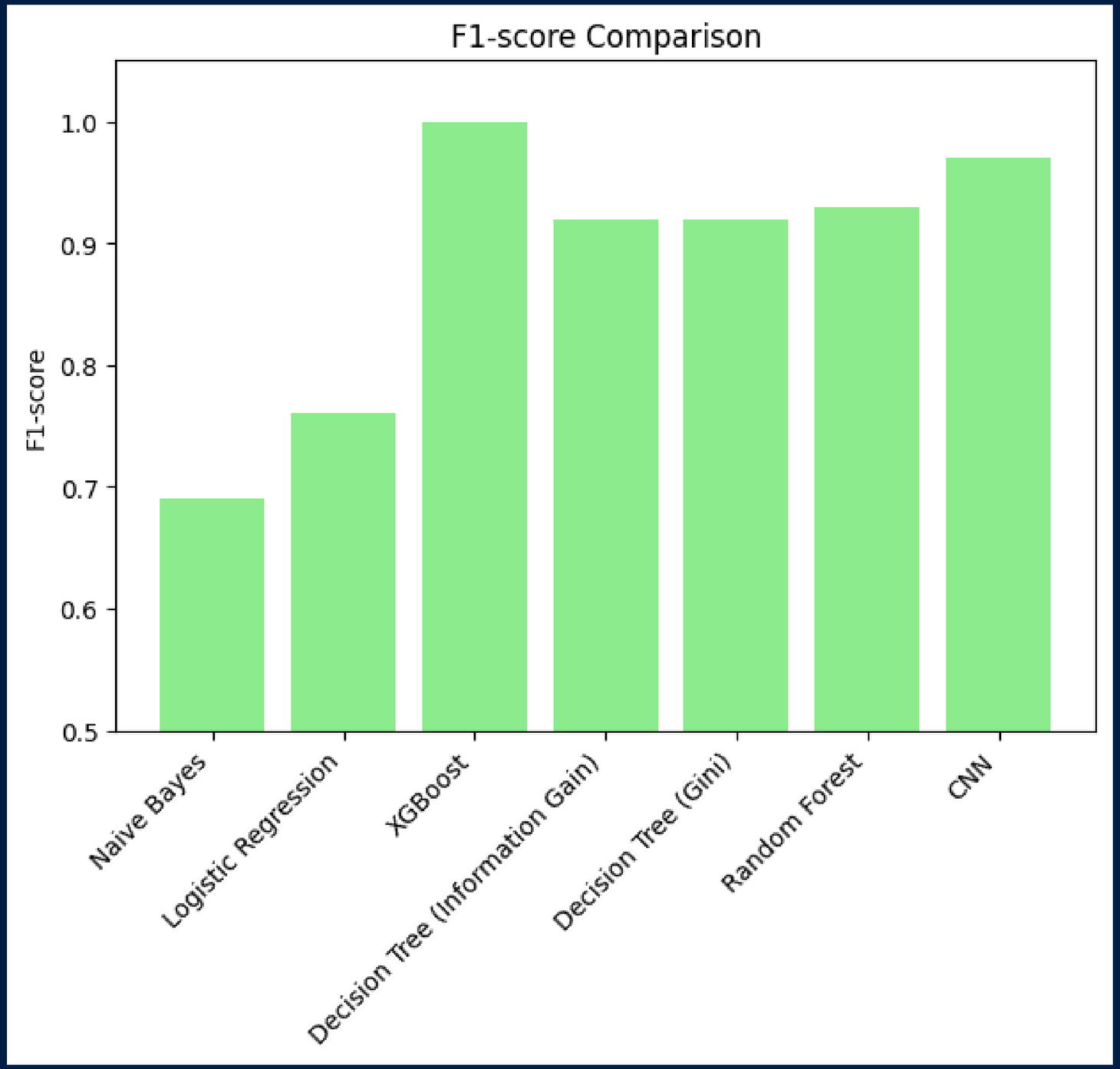
5. Data Evaluation

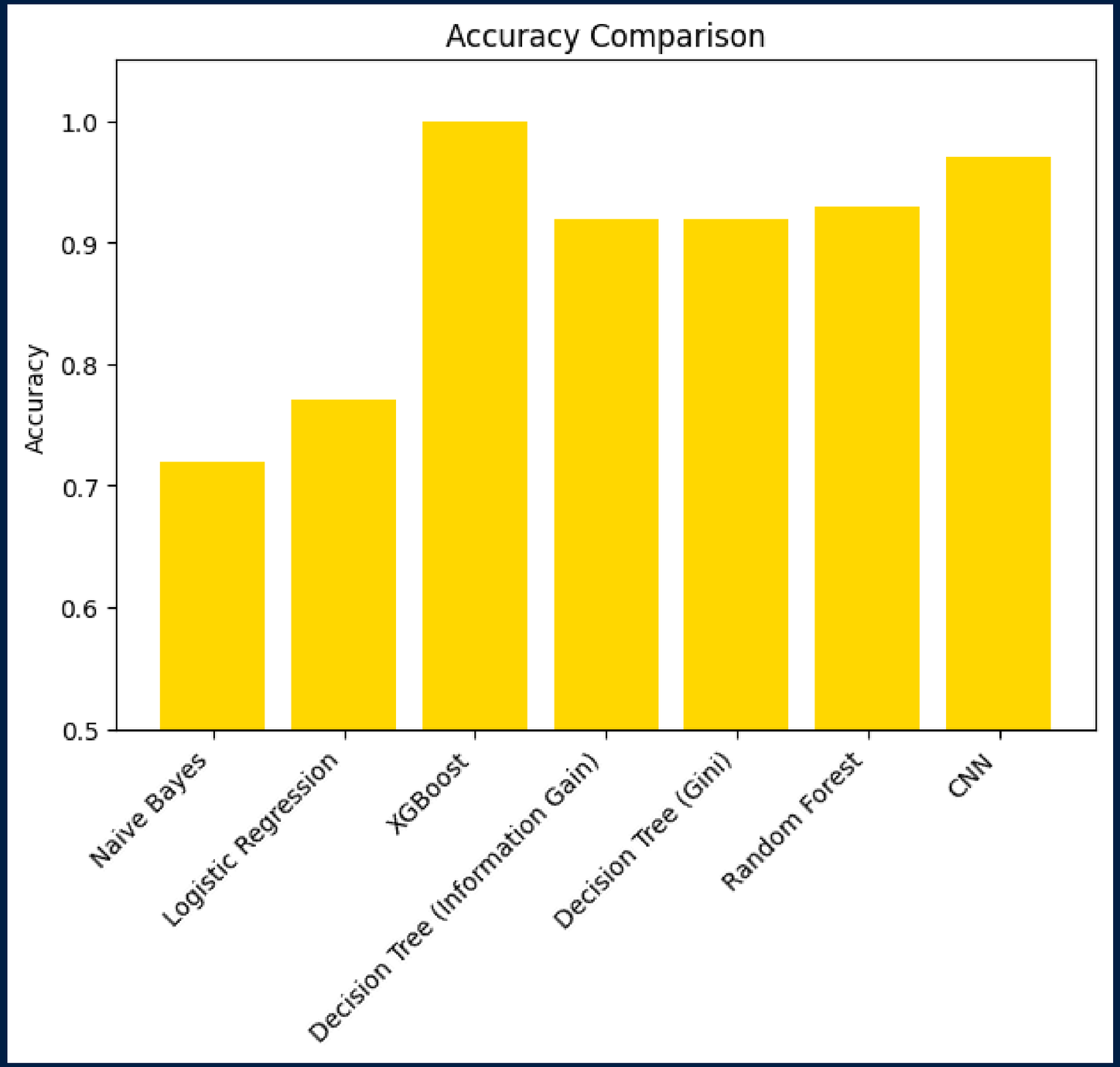
Performance Metrics

Model	Precision	Recall	F1-score	Accuracy
Naive Bayes	0.82	0.70	0.69	0.72
XGBoost	1.00	1.00	1.00	1.00
Decision Tree (Information Gain)	0.93	0.92	0.92	0.92
Decision Tree (Gini)	0.93	0.92	0.92	0.92
Random Forest	0.93	0.92	0.92	0.93
Convolutional Neural Network	0.97	0.97	0.97	0.97


Precision Comparison
Precision
1.0
0.9
0.8
0.7
0.6
0.5
Naive Bayes
Logistic Regression
XGBoost
Decision Tree (Information Gain)
Decision Tree (Gini)
Random Forest
CNN


Recall Comparison
Recall
1.0
0.9
0.8
0.7
0.6
0.5
Naive Bayes
Logistic Regression
XGBoost
Decision Tree (Information Gain)
Decision Tree (Gini)
Random Forest
CNN


F1-score Comparison
F1-score
1.0
0.9
0.8
0.7
0.6
0.5
Naive Bayes
Logistic Regression
XGBoost
Decision Tree (Information Gain)
Decision Tree (Gini)
Random Forest
CNN


Accuracy Comparison
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
Naive Bayes
Logistic Regression
XGBoost
Decision Tree (Information Gain)
Decision Tree (Gini)
Random Forest
CNN

Conclusion

Successfully analyzed botnet behavior in IoT networks and developed a machine learning-based detection mechanism with 97% accuracy. This achievement significantly enhances cybersecurity for IoT ecosystems, providing valuable insights and practical solutions for mitigating botnet threats.


Thank You