Machine Learning for Network Anomaly Detection

Problem Statement:

Rising cyberattacks demand robust detection methods. Signature-based systems fail against zero-day attacks.

Motivation:

Anomaly-based detection excels with encrypted traffic. Machine learning offers adaptability to evolving threats.

Goals:

Evaluate ML algorithms for network anomaly detection. Achieve high accuracy with the CICIDS2017 dataset.

Key Challenges in Network Security

Attack Types

Common attacks include DoS, DDoS, PortScan, Brute Force, and Botnets. Protecting against these is critical.

Traditional Limitations

Traditional methods rely on known attack signatures. They are ineffective against encrypted traffic.

Why Machine Learning?

ML detects unknown patterns and handles large datasets. This makes it ideal for modern threats.

Dataset Overview: CICIDS2017

1 Features

Real-world network traffic with labeled attacks. It has 85 features.

2 Advantages

Includes up-to-date attack diversity with HTTPS traffic.

3 Structure

Five days of network traffic include benign and attack types.

Advantages and Disadvantages of Dataset

Real-World Data

Captured from a testbed with Windows, Mac, and Linux computers. It reflects diverse OS environments.

Up-to-Date Attacks

Attack types are based on the 2016 McAfee Security Report. It supports protocols like HTTPS and SSH.

Large File Sizes

Raw data is 47.9 GB, processed is 1.1 GB. This presents storage and compute challenges.

No Test Data

Requires manual partitioning (e.g., 80-20 split via train_test_split)

Potential Minor Errors

Newer dataset; lacks iterative refinement seen in DARPA98 → KDD99 → NSL-KDD.

Limited Benchmarking

Few prior studies for direct performance comparison

Machine Learning Algorithms

1

Supervised Learning

Used Naive Bayes, QDA, Random Forest, ID3, AdaBoost, MLP, and KNN algorithms.

2

Selection Criteria

Diversity in methodology was key. Balance between accuracy and efficiency.

Methodology Steps

1 Data Preprocessing

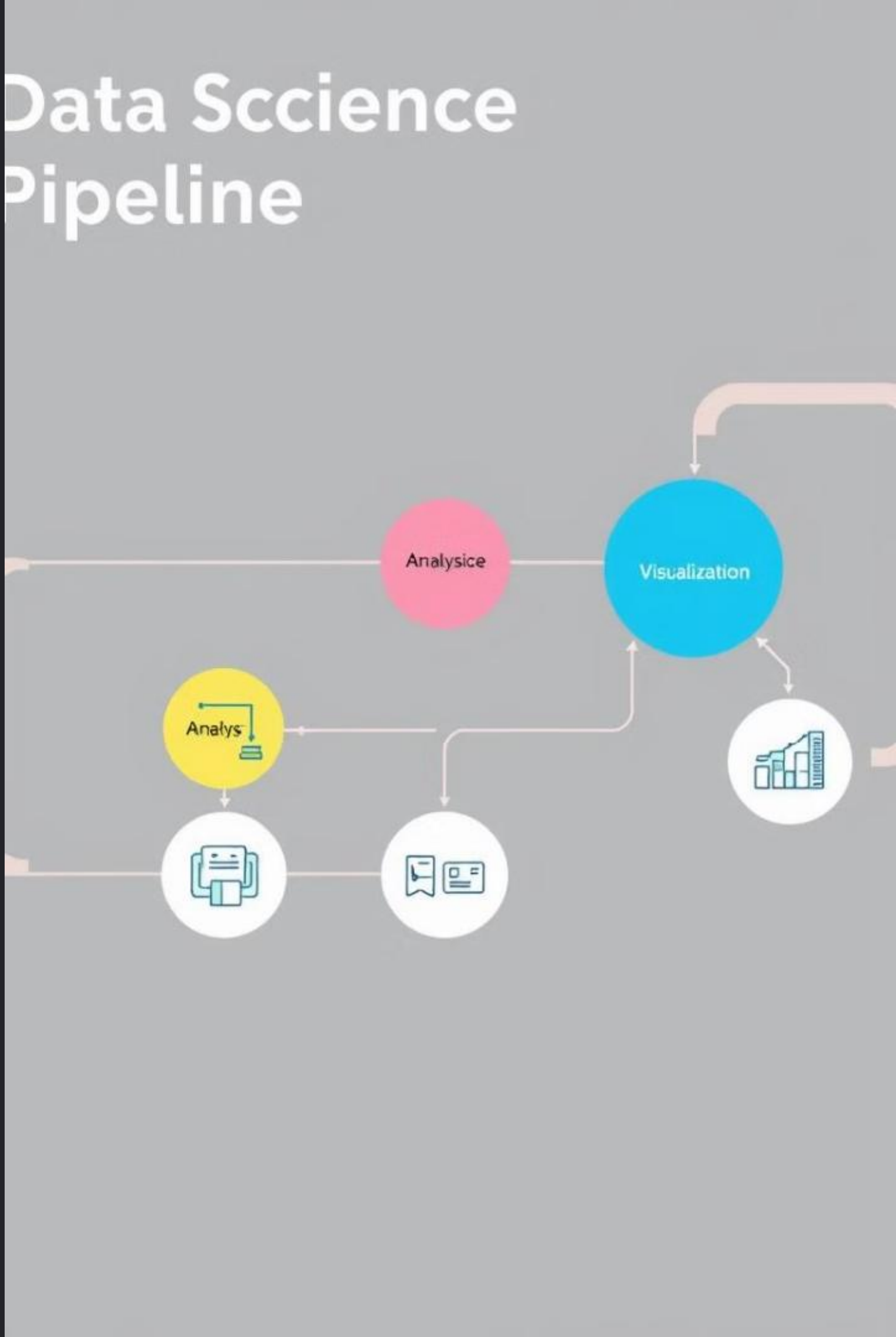
Cleaned duplicates, handled missing values, and encoded data.

2 Feature Selection

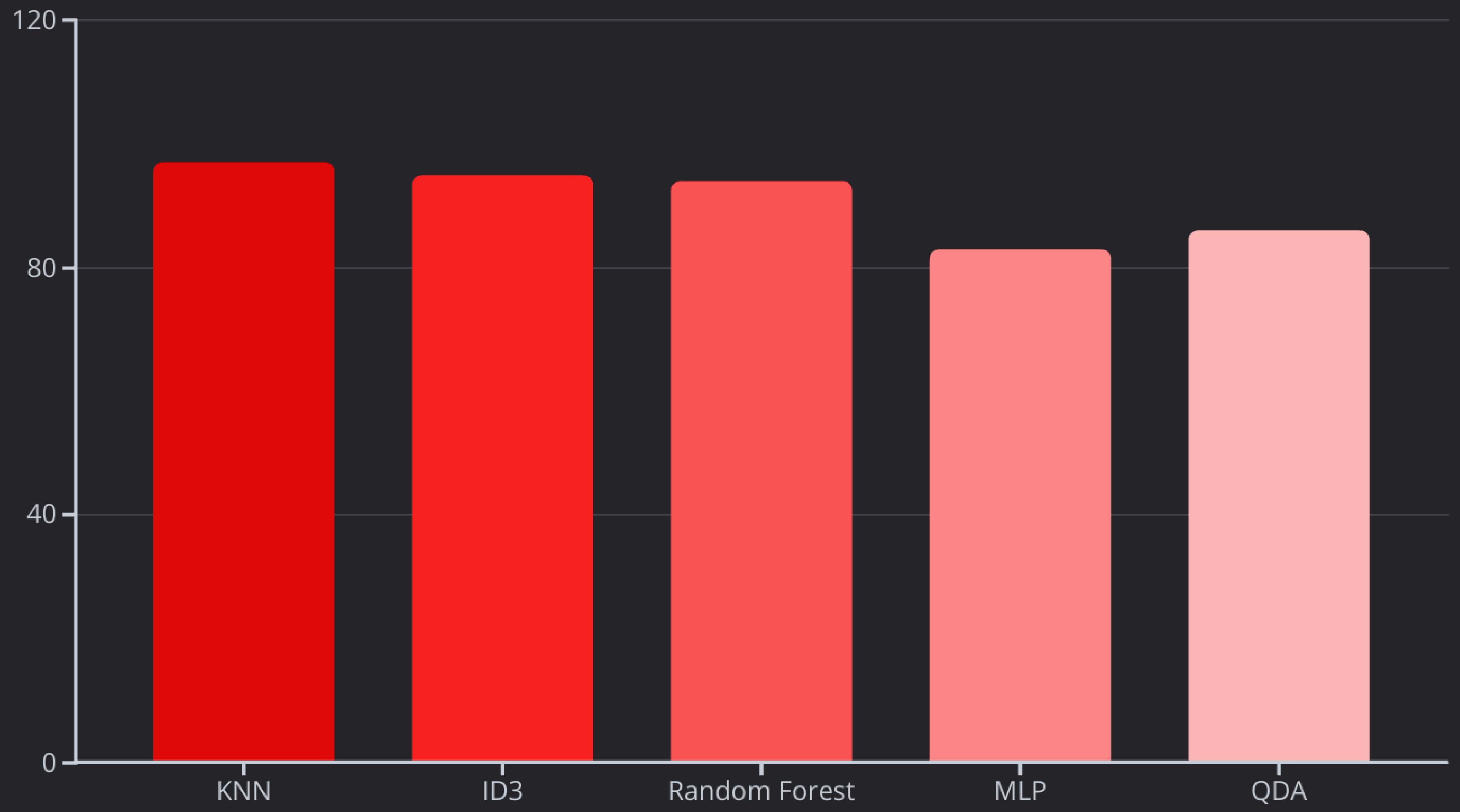
Used Random Forest Regressor to identify top features.

3 Model Training & Testing

80% training, 20% testing with 10-fold cross-validation.



Key Results: Algorithm Performance



KNN achieved 97% accuracy, and ID3 achieved 95%. Random Forest also performed well, at 94% accuracy.

Challenges and Lessons Learned

Data Imbalance

Rare attacks led to overfitting. This was a significant challenge.

Feature Selection

Critical for model efficiency. Reduced features from 85 to 7.

Algorithm Trade-offs

KNN was accurate but slow. Naive Bayes was fast but less accurate.

[redacted]

Future Work

Real-Time Detection

Integrate live traffic analysis modules.

Hybrid Models

Combine fast classifiers (Naive Bayes) with high-accuracy models (KNN).

Expand Datasets

Include IoT and cloud-based traffic.

[redacted]

Conclusion

1

Key Achievement

Demonstrated KNN and ID3 as top performers for anomaly detection.

2

Impact

Provides a framework for real-world ML-driven cybersecurity solutions.