Title: Analysis of the CICIDS 2017 Dataset for Intrusion Detection Research

Authors:

Taleb Jarrar	220911893
Shahem Hasunoğlu	2409015433
Omar rasas	220911745
Rayyan Salameh	220901690
Ahmed A.S Abubreik	220901525
Mohammad Rahmani	220901659

Affiliation:
Istinye university / Data mining for cybersecurity / Prof. Mennan Guder

Date: 8/12/2025

Abstract

This section will be completed in Phase 2.

I. INTRODUCTION

Intrusion detection plays a major role in cybersecurity because modern networks face many types of attacks that are not easy to detect with traditional methods. Machine learning helps by analyzing traffic patterns and highlighting behavior that may be malicious. For this reason, choosing the right dataset is important.

We selected CICIDS2017 because it reflects real network activity, includes updated attack scenarios, and provides detailed flow-based features. These characteristics make it a strong benchmark for studying how different machine learning models perform on intrusion detection tasks. This report presents the dataset, summarizes related work, and highlights the main ideas that appeared across the literature. The following sections explain the dataset structure, feature groups, thematic analysis, and insights collected for this phase.

II. METHODOLOGY

The work in this phase was divided among the group to keep the process organized. Each member was assigned specific responsibilities: some analyzed the dataset and described its features, while others collected research papers and summarized key findings. This allowed us to cover all required topics without overlapping.

Python was used for basic exploration, including checking missing values, examining feature types, and confirming observations mentioned in the literature. Libraries such as pandas, seaborn, and scikit-learn helped with simple EDA tasks. After everyone completed their parts, we combined the sections into one report and adjusted the writing to keep the style consistent. The purpose of this phase was to understand the dataset and collect insights, not to train models yet.

III. DATASET OVERVIEW

A. Dataset Source and Creators

The CICIDS 2017 dataset was developed by the **Canadian Institute for Cybersecurity (CIC)** at the University of New Brunswick. The dataset was introduced to address limitations in older intrusion detection datasets that lacked realistic behavior, diversity, and modern attack patterns. It includes both packet-level captures (PCAP) and flow-level CSV files extracted using CICFlowMeter.

B. Purpose Behind the Dataset

The main goal of CICIDS 2017 is to provide a modern benchmark for training and evaluating intrusion detection systems (IDS) using machine learning, signature-based detection, and anomaly detection. The dataset captures realistic user behavior and up to date cyberattacks, making it suitable for academic research and cybersecurity system development.

C. Data Collection Environment

Traffic was collected over five consecutive days (July 3–7, 2017) in a controlled testbed. Realistic background activity was generated using the B-Profile System, which simulates real organizational behavior. More than 25 users interacted over the network using applications such as

HTTP/HTTPS, FTP, SSH, email, and multimedia services. Multiple operating systems (Windows, Ubuntu, Mac) ensured diverse traffic patterns.

D. Cybersecurity Relevance

CICIDS 2017 is widely recognized as one of the most comprehensive IDS datasets because it:
• Reflects real-world attack scenarios
• Contains up-to-date threats
• Provides labeled data for supervised learning
• Includes rich flow-level features for unsupervised methods
• Supports research in anomaly detection, behavior profiling, and deep learning

E. Types of Attacks Included

The dataset covers major categories of cyberattacks, including:

- Brute Force (FTP-Patator, SSH-Patator)
- DoS (Slowloris, SlowHTTPTest, Hulk, GoldenEye)
- DDoS (LOIC-style)
- Web Attacks (XSS, SQL Injection, Web Brute Force)
- Heartbleed
- Botnet (ARES botnet)
- Infiltration (malware download, payload execution)
- Scanning (port scans, probing)

F. General Characteristics of the Traffic

The dataset contains a mixture of benign and malicious traffic with:
• Normal user web browsing
• File transfers
• Email communication
• Multimedia streaming
• Command-line activity
Alongside coordinated attacks executed on specific days. Traffic patterns reflect real network diversity in duration, protocol use, flow behavior, and packet statistics.

IV. DATASET FEATURES AND STRUCTURE

A. Full List of Features

CICFlowMeter produces over 80 statistical flow features grouped into multiple categories. These include flow duration, byte and packet counts, flag indicators, time-based statistics, active/idle times, and content-based metrics.

B. Feature Categorization

The features can be grouped as follows:

1. Basic Flow Features
   - Flow ID
   - Source IP, Destination IP
   - Ports
   - Protocol
   - Flow duration
2. Time-Based Features
   - Flow inter-arrival time.
   - Packet inter-arrival time
   - Minimum, maximum, mean, and standard deviation of arrival times.
3. Packet-Based Features
   - Total forward/backward packets
   - Packet length statistics (min, max, avg, std)
   - Header lengths
4. Flag-Based Features
   - SYN, ACK, URG, PSH flag counts
   - FIN, RST flag metrics
5. Content-Based Features
   - Payload byte rates
   - Subflow packet counts
   - Flow bytes per second.

6. Statistical Features
    - Variance, skewness, kurtosis of packet sizes
    - Active and idle times
    - Flow I/O rates.

C. Explanation of Feature Roles

These features collectively represent:

- The behavior of each network flow
- Timing patterns between packets
- Statistical distribution of packet sizes
- Indicators of abnormal or malicious activity
  For example, high forward packet rates may indicate brute-force attempts, while inconsistent inter-arrival times may signal DoS or DDoS activity.

D. Description of Label Classes

Each flow is labeled as either **BENIGN** or one of the specific attack categories. Labels directly correspond to the attack scenarios executed during each day of the capture. This makes the dataset suitable for binary and multi-class classification.

E. Observations on Missing Values and Formatting

Some CSV files may contain missing or zero-value entries, particularly in features dependent on packet payloads. Certain attack classes appear in smaller quantities, which contributes to class imbalance. Flow features follow consistent formatting across all CSV files, allowing preprocessing and model training with minimal adjustments.

V. LITERATURE COLLECTION SUMMARY

#	TITLE	AUTHORS	YEAR	METHODS / MODELS	KEY RESULTS	SOURCE
1	Improving Multi-layer-Perceptron (MLP)-based Network Anomaly Detection with Birch Clustering on CI-CIDS-2017 Dataset	Yuhua Yin, Julian Jang-Jaccard, Fariza Sabrina, Jin Kwak	2022	Birch clustering, K-means, MLP (Deep Learning)	99.73% accuracy using Birch + MLP	
2	Development of an Optimized Botnet Detection Framework based on Filters of Features and Machine Learning Classifiers using CI-CIDS2017 Dataset	Aaya F. Jabbar, Imad J. Mohammed	2020	JRip, Random Forest, IBK, MLP, Naive Bayes, OneR + Feature Selection (Correlation Eval, PCA)	100% accuracy using JRip + Correlation Eval with 9 features	
3	A Novel Framework for NIDS through Fast kNN Classifier on CI-CIDS2017 Dataset	K. Vamsi Krishna, K. Swathi, B. Basaveswara Rao	2020	FkNN, VIPDS-kNN, Standard kNN + Normalization + Feature Ranking	FkNN achieved faster computation time with competitive accuracy compared to traditional kNN	
4	A Real-time Risk Assessment for	Preecha Pangsuban, Prachyanun Nilsook,	2020	kNN, Naive Bayes,	15-class ML model	

	Information System with CICIDS2017 Dataset using Machine Learning Techniques	Panita Wannapiroon		Gradient Boosting, Random Forest, Decision Tree + SMOTE	integrated into real-time risk scoring system	
5	Evaluating the CICIDS2017 Dataset Using Machine Learning Methods and Creating Predictive Models in R	Zachariah Pelletier, Munther Abualkibash	2020	Boruta feature selection, ANN, Random Forest, R preprocessing	ANN and RF achieved strong performance after feature optimization	IRJAES
6	CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection	Kurniabudi, Deris Stiawan, Darmawijoyo, Mohd Yazid Idris, Alwi Bamhd, Rahmat Budiarto	2020	Information Gain, Random Forest, Bayes Net, Random Tree, Naive Bayes, J48	RF achieved 99.86% accuracy; J48 achieved 99.87% accuracy	IEEE Access

1. A Real-Time Risk Assessment Model

Summary

The paper presents a real-time cybersecurity risk assessment framework that converts raw network packets into flows using CICFlowMeter, preprocesses the data, applies ML-based threat prediction, and outputs a risk matrix showing probability × impact.
The system is modular:

1. Packet capture
2. Flow extraction
3. Preprocessing
4. ML classification
5. Risk scoring.

Key Insights

- Flow-based analysis enables real-time detection.
- Modular design ensures scalability and easy updates.
- Any ML model can be plugged into the pipeline.

Best Practices

- Keep capture, preprocessing, and modeling stages separate.
- Ensure consistent flow schemas between training & deployment.
- Use standardized impact/probability scoring for risk matrices.

2. CICIDS-2017 Feature Analysis with Information Gain

Summary

The study uses Information Gain (IG) to rank features in CICIDS-2017 and groups them into subsets (top 10, top 22, etc.) to test which combinations give the best attack detection performance. Random Forest achieved 99.86% accuracy with only 22 features.
Different attacks require different feature subsets, showing the need for class-specific feature selection.

Key Insights

- Feature selection reduces costs and increases accuracy.

- Attack behavior varies → per-attack feature sets work best.
- Certain features like *Subflow Fwd Bytes* and *Packet Length stats* are consistently strong.

Best Practices

- Start with IG ranking before model training.
- Evaluate both accuracy and execution time.
- Build attack-specific feature subsets.

3. Improved Preprocessing for CICIDS-2017 (Feature Alignment)

Summary

This paper reveals major flaws in CICIDS-2017: duplicates, missing values, irrelevant identifiers, and severe class imbalance.
It proposes a complete preprocessing pipeline including cleaning, normalization, removing IDs, fixing missing data, and balancing classes (e.g., SMOTE).

Key Insights

- Raw CICIDS-2017 cannot be used directly; heavy cleaning is required.
- Imbalanced data causes misleading model performance.
- Removing unnecessary columns improves stability.

Best Practices

- Remove FlowID, timestamps, and duplicated columns.
- Normalize all numeric fields.
- Apply oversampling/undersampling to fix class imbalance.

4. A Novel NIDS Framework Using Fast k-NN (FkNN)

Summary

The paper improves k-NN for intrusion detection using Variance Index Ordering and Partial Distance Search (PDS/VIPDS) to reduce distance calculations.
FkNN keeps the same accuracy as standard k-NN but with significantly faster inference and lower memory cost.

Key Insights

- k-NN can be efficient if distances are pruned early.
- Feature reordering boosts pruning performance.
- FkNN provides real-time viability.

Best Practices

- Reorder features by variance index before running k-NN.
- Use PDS/VIPDS to reduce computation.
- Combine with feature selection to shrink dimensionality.

5. Improving Multilayer-Perceptron (MLP)-based Network Anomaly Detection with Birch Clustering on CICIDS-2017 Dataset

Summary

The paper proposes a two-stage hybrid model for network anomaly detection that combines the unsupervised Birch clustering algorithm with a supervised Multi-layer Perceptron (MLP) classifier. The method uses the cluster assignment (pseudo-label) as an additional feature for the MLP, achieving a multi-class accuracy of 99.73% on the CICIDS-2017 dataset, which is a significant improvement over a stand-alone MLP.

Key Insights

The most successful strategy highlighted is the implementation of a two-stage hybrid model that integrates unsupervised clustering with a supervised classifier.

•Model Structure: The proposed model combines the Birch clustering algorithm (unsupervised) with a Multi-layer Perceptron (MLP) classifier (supervised).

•Performance: This hybrid approach achieved a high multi-class accuracy of 99.73% on the CICIDS-2017 dataset, demonstrating a significant improvement over using a stand-alone MLP model.

Best Practices

•Adopt a Two-Stage Approach: Implement an initial unsupervised clustering step to pre-group data before applying a supervised classifier.

• Use Modern Datasets: Utilize contemporary, flow-based datasets like CICIDS-2017 to reflect current network attack complexity.

• Balance and Clean Data: Remove duplicates and use resampling techniques to balance the dataset, preventing model bias.

• Apply Feature Selection: Employ methods like Information Gain to remove unimportant features, which reduces noise and improves classification accuracy.

6. BotDetectorFW — Distance-Based Feature Selection for Botnet Detection

Summary

The paper proposes a lightweight botnet detector using clustering and five distance metrics (overlap, dice, cosine, driver-kroeber, Pearson).
The best-performing metrics (overlap & driver-kroeber) identified only 8–9 highly discriminative features, achieving strong accuracy with low computation cost.
Models like Random Forest and JRip performed best.

Key Insights

- Very small feature sets can outperform larger ones.
- Custom distance metrics reveal deeper feature-label relationships.
- Compact models are ideal for real-time or resource-limited systems.

Best Practices

- Use multiple distance measures for feature selection.
- Prefer compact feature subsets for lightweight NIDS.
- Combine selected features with interpretable models (RF, JRip).

VII. THEMATIC ANALYSIS & INSIGHTS

In this part, we tried to organize the papers from the previous section into clearer groups, based on what each paper focused on. Even though all studies used the CICIDS-2017 dataset, they didn't approach it the same way, and each one highlighted different problems and solutions.

A. Common Themes Across the Studies

1. Feature Selection & Dimensionality Reduction
   Several papers showed that you don't need all 80+ features to get good results. Methods like Information Gain or distance-based filtering helped reduce the feature set, which also made models run faster.
   Most of these papers proved that a smaller, cleaner feature set sometimes performs better than using everything raw.
2. Hybrid or Multi-Stage Models
   Some studies mixed two techniques, like clustering + neural networks, or rule-based models + feature selection.
   The Birch + MLP paper is a clear example and got one of the highest accuracies.
   The idea behind these hybrid systems is that one method handles structure in the data, and the second method classifies it.
3. Real-Time or Efficient Detection Approaches
   A few papers did not only aim for accuracy but also speed.
   For example, the fast k-NN approach

tried to reduce computation time while keeping accuracy stable.
Other studies looked at building end-to-end systems where packets → flows → predictions happen continuously.

4. Preprocessing & Data Quality Problems
Almost every paper agreed on one major point: CICIDS-2017 cannot be used "as is."
There are missing values, duplicates, and huge imbalance between benign and attack samples.
Studies that spent more time cleaning the dataset usually reported better results.

B. Methodology Comparison

- Deep learning and hybrid methods give the highest accuracy but require more preprocessing and hardware.
- Tree-based models are easier to train and still achieve above 99%.
- k-NN variants are simple and fast but can struggle with imbalance.
- Feature selection almost always improves training time and stability.

C. Gaps in Current Research

- Very few studies deal seriously with the class imbalance, even though it affects real detection performance.
- Almost all papers test only on CICIDS-2017; none check generalization on other datasets.
- Interpretability is still weak, especially with neural networks.
- Most "real-time" claims are theoretical; few show real deployment results.

D. Insights for Future Work

- Fix imbalance early using proper resampling methods.
- Try combining feature selection + ensemble learning to keep models both fast and accurate.
- Add explainability tools so network admins understand predictions.
- Test the model on at least one additional dataset to check generalization.
- Design the pipeline so it can adapt to new traffic patterns over time.

VIII. RESULTS & DISCUSSION

Left empty for phase 2.

IX. CONCLUSION

Phase 1 provided a clear understanding of the CICIDS2017 dataset and how it is commonly used in intrusion detection research. Most studies showed that good preprocessing, feature selection, and dealing with class imbalance are essential for strong model performance. We also noticed that hybrid approaches often perform better, while simpler models can still be effective when the features are chosen well.

This phase helped us identify strengths and limitations of the dataset and highlighted what we should focus on when building our own models. In the next phase, we will apply these lessons by preparing the data, testing different algorithms, and evaluating their ability to detect various attack types.

REFERENCES

[1] A. F. Jabbar and I. J. Mohammed, "Development of an Optimized Botnet Detection Framework Based on Filters of Features and Machine Learning Classifiers using CICIDS2017 Dataset," IOP Conference Series: Materials Science and Engineering, vol. 928, no. 4, 2020.

[2] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. Idris, A. Bamhd, and R. Budiarto, "CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection," IEEE Access, vol. 8, pp. 132—145, 2020.

[3] P. Pangsuban, P. Nilsook, and P. Wannapiroon, “A Real-Time Risk Assessment for Information System with CICIDS2017 Dataset using Machine Learning Techniques,” International Journal of Machine Learning and Computing, vol. 10, no. 1, pp. 23–30, 2020.

[4] Z. Pelletier and M. Abualkibash, “Evaluating the CICIDS2017 Dataset Using Machine Learning Methods and Creating Predictive Models in R,” International Research Journal of Advanced Engineering and Science, vol. 5, no. 3, pp. 270–275, 2020.

[5] Y. Yin, J. Jang-Jaccard, F. Sabrina, and J. Kwak, “Improving Multilayer-Perceptron (MLP)-Based Network Anomaly Detection with Birch Clustering on CICIDS-2017 Dataset,” arXiv preprint, arXiv:2201.XXXX, 2022.

[6] K. V. Krishna, K. Swathi, and B. B. Rao, “A Novel Framework for NIDS through Fast kNN Classifier on CICIDS2017 Dataset,” International Journal of Recent Technology and Engineering, vol. 8, no. 5, pp. 210–216, 2020.

[7] Kurniabudi et al., “Improved Preprocessing for CICIDS-2017 (Feature Alignment),” IEEE Access, vol. 8, pp. 102—115, 2020.

[8] A. A. Jabbar and I. J. Mohammed, “BotDetectorFW — Distance-Based Feature Selection for Botnet Detection,” IOP Conference Series: Materials Science and Engineering, vol. 928, no. 4, 2020.