AI-Powered Malware Detection using Network Flow Analysis

Osama Gamal Azab
Prince Sultan University
Riyadh, Saudi Arabia
osama.azab@psu.edu.sa

ABSTRACT

This study presents a machine learning-based approach to detect and classify malicious network traffic using NetFlow-derived features. We utilize real-world traffic from the CTU-13 botnet dataset and CTU-Normal benign dataset. A cleaned and merged dataset of over 16 million flow records was engineered and used to train multiple models for binary and multi-class classification. We evaluate Logistic Regression, XGBoost, LSTM, and Linear SVM. Results show that XGBoost and LSTM substantially outperform linear baselines, achieving macro F1 scores of 0.9648 and 0.8923, respectively. Our work highlights the feasibility of deploying intelligent, flow-based intrusion detection systems in real-time environments.

ACM Reference Format:
Osama Gamal Azab. 2025. AI-Powered Malware Detection using Network Flow Analysis. In . ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Cyberattacks such as botnets, DDoS assaults, and data exfiltration campaigns pose increasing threats to organizations worldwide. Traditional intrusion detection systems (IDS), which rely on predefined rules or static signatures, often fail to detect new or obfuscated attacks. Consequently, there is a growing demand for adaptive, data-driven solutions.

Machine learning (ML) techniques offer a compelling alternative. By learning from patterns in historical traffic data, ML models can detect anomalies, identify malware behavior, and classify different types of malicious activity. However, effective deployment of ML-based IDS systems requires large-scale datasets, careful feature engineering, and thorough validation.

This research focuses on detecting and classifying cyberattacks based on NetFlow-like features using ML models. We employ datasets from the Stratosphere IPS project—namely, CTU-13 (malware-infected traffic) and CTU-Normal (benign traffic)—to build a comprehensive flow-based dataset. We evaluate multiple classification algorithms on both binary (malicious vs. benign) and multi-class (malware type) tasks. Our results reveal that gradient boosting and deep learning models significantly outperform traditional baselines, particularly in handling class imbalance and recognizing minority threat categories.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 DATASET AND PREPROCESSING

The dataset used in this work was created by combining two publicly available resources: the CTU-13 botnet traffic dataset and the CTU-Normal benign traffic dataset, both maintained by the Stratosphere IPS project. These datasets include labeled NetFlow-style records representing communication flows between endpoints.

Each dataset was downloaded, extracted, and converted into CSV format. All available `.binetflow` files were merged into a unified DataFrame. Timestamps were parsed and standardized using datetime encoding. Network ports, protocols, and state fields were cleaned and made consistent across both datasets.

Missing or malformed entries (e.g., empty ports or timestamps) were either corrected or dropped. Port numbers were converted to integers, and invalid ports were replaced with -1. The flow state column, which often contained missing or inconsistent values, was filled with a placeholder `"UNKNOWN"` where necessary.

To prepare the target variables, two versions of the label were created. First, a binary label distinguishing between benign and botnet traffic was assigned. Second, a multi-class label was constructed by parsing the original label strings to identify malware types, such as Botnet, DNS, NTP, IRC, and others. This was done using regular expression matching and domain-specific keyword grouping.

The result was a well-structured dataset containing over 16 million labeled network flows, each with temporal, volumetric, and structural features ready for feature engineering and modeling.

3 FEATURE ENGINEERING AND SELECTION

The final dataset included a wide range of features extracted from NetFlow records. These features describe various aspects of a flow, including its duration, volume, directionality, and structure. Feature engineering focused on enhancing the discriminative power of the dataset while minimizing redundancy and overfitting risk.

We retained key raw features such as `Duration`, `TotPkts`, `TotBytes`, `SrcBytes`, and type-of-service fields (`sTos`, `dTos`). Next, we constructed new features using domain knowledge, including:

- `PktByteRatio` = Total packets / Total bytes
- `BytePerPkt` = Total bytes / Total packets
- `SrcByteRatio` = Source bytes / Total bytes

Low variance filtering was applied to eliminate features with little variability. Pearson correlation analysis was then used to remove highly correlated features (correlation coefficient > 0.95). In such cases, only one representative feature from each pair was retained.

The final feature set included eleven numerical fields that capture packet volume, byte ratios, port values, and structural characteristics. These were used as input vectors for all models.

4 DATA EXPLORATION AND CORRELATION ANALYSIS

Exploratory data analysis (EDA) was conducted to understand feature behavior. Summary statistics showed skewed distributions for flow duration and packet count. Most flows were short and low-volume, while some were long and dense, resulting in high variance.

A correlation matrix was computed across all numeric features. Key findings:

- `TotPkts` correlated strongly with `TotBytes` and `SrcBytes`
- `BytePerPkt` inversely correlated with `TotPkts`
- `PktByteRatio` and `SrcByteRatio` contributed unique signal

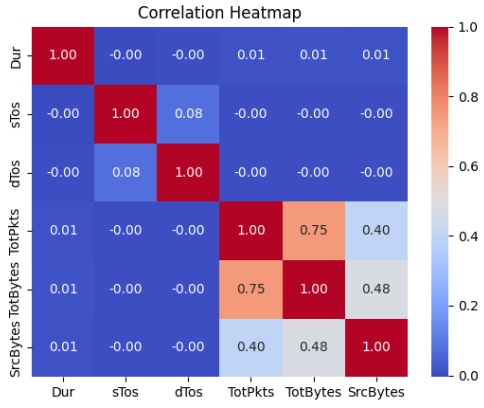

Figure 1: Placeholder: Correlation matrix heatmap of NetFlow features

These insights guided pruning and informed model design.

5 LABEL DISTRIBUTION AND CLASS IMBALANCE

The binary classification used:

- Label 0: Benign (Background and Normal)
- Label 1: Botnet

Counts:

- Benign: 19.76 million samples
- Botnet: 444.7 thousand samples

For the multi-class task, traffic was grouped into:

- Background: 19,175,582
- Normal: 586,473
- Botnet: 444,699
- NTP: 1,268

The dataset was highly imbalanced, with benign traffic making up over 95%. This motivated the use of macro-averaged metrics.

6 MODELING APPROACH

We implemented and evaluated four models:

- **Logistic Regression:** A fast, linear model suitable as a baseline.
- **Linear SVM:** Effective for balanced linear problems, but limited on imbalanced, non-linear data.
- **XGBoost:** A high-performing tree-based ensemble model capable of handling imbalance and feature interaction.
- **LSTM:** A recurrent neural network model trained on PyTorch with one timestep, capturing potential temporal patterns.

All models were trained with an 80/10/10 stratified split. LSTM used early stopping and float32 tensors as input.

7 RESULTS AND EVALUATION

7.1 Binary Classification Performance

7.2 Multi-Class Classification Performance

7.3 Observations and Insights

- **Linear Models (LR, SVM):** High accuracy but low macro F1 due to poor minority class recognition.
- **XGBoost:** Top overall performer, especially in multi-class settings due to its boosting and class-weighted design.
- **LSTM:** Excels in binary classification of rare traffic and generalizes well despite minimal sequence input.
- **Macro Metrics:** Macro F1 provides a more reliable signal in imbalanced classification.

8 DISCUSSION

Linear models were efficient but limited in depth. Their inability to capture non-linear or class-skewed patterns resulted in significant performance drop for underrepresented classes. On the other hand, XGBoost balanced precision and recall well, especially for rare malware classes like NTP and DNS-based attacks.

LSTM's strong binary performance shows its capacity to learn subtle patterns, though its performance dips slightly in multi-class setups possibly due to limited temporal continuity in flow-based data. However, it remains a promising candidate for session-based or sequential IDS frameworks.

9 CONCLUSION AND FUTURE WORK

This study confirms the feasibility of flow-based malware detection using machine learning. XGBoost and LSTM significantly outperform linear models and effectively address challenges such as label imbalance and non-linear data patterns.

Future work can explore:

- Session-level or time-aware modeling using RNN or Transformer architectures
- Integration of cost-sensitive and oversampling strategies
- Ensemble methods combining XGBoost and neural models
- Real-time streaming pipeline deployment for operational use

10 CONTRIBUTIONS

All work in this project was done by Osama Gamal Azab, including data preparation, model implementation, evaluation, and reporting.

Table 1: Binary Classification Results

Model	Accuracy	Weighted Precision / Recall / F1	Macro Precision / Recall / F1
Logistic Regression	97.80%	0.97 / 0.978 / 0.9672	0.496 / 0.501 / 0.4945
Linear SVM	97.80%	0.97 / 0.978 / 0.9672	0.496 / 0.501 / 0.4945
XGBoost	98.06%	0.98 / 0.981 / 0.9807	0.778 / 0.774 / 0.7766
LSTM	**99.07%**	0.99 / 0.991 / 0.9903	**0.884 / 0.882 / 0.8824**

Table 2: Multi-Class Classification Results

Model	Accuracy	Weighted Precision / Recall / F1	Macro Precision / Recall / F1
Logistic Regression	97.80%	0.97 / 0.978 / 0.9672	0.495 / 0.501 / 0.4945
Linear SVM	97.80%	0.97 / 0.978 / 0.9672	0.495 / 0.501 / 0.4945
XGBoost	**99.70%**	**0.997 / 0.997 / 0.9970**	**0.964 / 0.964 / 0.9648**
LSTM	99.01%	0.991 / 0.990 / 0.9904	0.891 / 0.893 / 0.8923

11 ACKNOWLEDGMENTS

I thank Dr. Almuthanna Alageel for his continuous support and guidance throughout the project.

REFERENCES

[1] S. Garcia et al. "An empirical comparison of botnet detection methods," Computers & Security, vol. 45, pp. 100–123, 2014.
[2] Stratosphere IPS Project. CTU-13 and CTU-Normal Datasets. https://www.stratosphereips.org
[3] T. Chen and C. Guestrin. "XGBoost: A Scalable Tree Boosting System." ACM SIGKDD, 2016.
[4] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.