Botnet Command and Control Detection Using Network Flow Analysis

1st Usama Altaf
Faculty of Engineering and Technology
Islamia University of Bahawalpur
Bahawalpur, Pakistan
usamaaltaf3098@gmail.com

2nd Muhammad Affaq
Faculty of Engineering and Technology
Islamia University of Bahawalpur
Bahawalpur, Pakistan
maffaq12345@gmail.com

3rd Aoun Muhammad
Faculty of Engineering and Technology
Islamia University of Bahawalpur
Bahawalpur, Pakistan
aoun.muhammad@iub.edu.pk

4th Umar Fayyaz
Faculty of Engineering and Technology
Islamia University of Bahawalpur
Bahawalpur, Pakistan
umar.fayyaz@iub.edu.pk

5th Sehrish Raza
Faculty of Engineering and Technology
Islamia University of Bahawalpur
Bahawalpur, Pakistan
sehrishraza6@gmail.com

***Abstract*—This paper presents a large-scale empirical comparison of machine learning approaches for detecting botnet Command-and-Control (C2) traffic using network flow analysis. We evaluate three distinct methods on the CTU-13 dataset comprising 13.46 million network flows: Random Forest, XGBoost (supervised learning), and an LSTM Autoencoder (anomaly detection). Our comparative analysis demonstrates that Random Forest achieves the highest performance with 99.21% accuracy, 94.31% recall, and 0.9968 ROC-AUC, while XGBoost attains 98.93% accuracy and 90.54% recall. The LSTM Autoencoder shows limited effectiveness with 0.679 ROC-AUC. These findings provide empirical evidence that flow-based supervised learning can effectively detect encrypted C2 traffic without payload inspection, achieving false positive rates below 1%, making it practical for real-time botnet detection in operational networks.**

***Index Terms*—Botnet Detection, Command and Control, Network Flow Analysis, Machine Learning, Intrusion Detection**

I. Introduction

Botnets that is, when a group of hacked devices are under the control of bad guys, remain a gigantic issue in the world of cybersecurity. They are operating Command-and-Control (C2) systems to execute synchronized assaults, including DDoS, information theft, spam, and ransomware [1]. Modern botnets employ sophisticated evasion techniques such as domain generation algorithms (DGAs), encrypted communications, peer-to-peer networks, and Fast Flux DNS to remain undetected [2]. Classical signature-based detection techniques cannot keep pace with evolving botnet patterns and encrypted traffic. Network flow analysis offers a promising alternative by examining metadata such as packet counts, timing patterns, and session behaviors rather than payload content [3].

This work contributes a comprehensive empirical evaluation of machine learning techniques for flow-based botnet C2 detection. We systematically compare supervised ensemble methods (Random Forest, XGBoost) against unsupervised anomaly detection (LSTM Autoencoder) on the large-scale CTU-13 benchmark. Our objectives are to: (1) achieve detection accuracy exceeding 90% with false positives below 5%, (2) quantitatively compare supervised versus anomaly-based approaches at scale, and (3) demonstrate practical feasibility for real-time deployment. Our findings provide actionable insights for practitioners selecting detection methods for operational networks.

II. Related Work

Early botnet detection relied on signature-based systems like Snort, which are ineffective against zero-day attacks [4]. Recent research has shifted toward behavioral analysis using flow metadata. García et al. [5] demonstrated that NetFlow features can expose botnets even when traffic is encrypted. Supervised classification using Random Forests and SVMs [6], and anomaly detection with LSTMs or autoencoders [7] have shown promise. The CTU-13 dataset [5] has become a standard benchmark containing real botnet traffic from multiple malware families with background noise. However, challenges remain: high false positives, real-time scalability, and severe class imbalance.

Recent advances have further explored deep learning architectures for botnet detection. Vinayakumar et al. [16] applied deep neural networks to network traffic analysis, achieving improved detection of malicious activities. Ferrag et al. [17] provided a comprehensive survey of deep learning methods for intrusion detection, highlighting the evolution toward hybrid models. Alauthman et al. [18] developed intelligent detection systems using machine learning for IoT botnet traffic, addressing the growing threat of IoT-based botnets. Kumar et al. [19] proposed ensemble learning techniques combining multiple classifiers to improve detection robustness against evolving botnet behaviors. Asadi et al. [20] investigated Graph Neural Networks (GNNs) for detecting botnet C2 communications by modeling network traffic as graphs. These recent works demonstrate the field's progression toward more sophisti-

DOI: https://doi.org/10.67535/tsp.000002.023
Published under the ISAISS 2026 Proceedings

cated architectures, yet systematic empirical comparisons on standardized datasets remain valuable for practitioners. Our work addresses this gap by providing controlled comparisons on CTU-13, enabling direct performance assessment across traditional and modern approaches.

III. Methodology

A. Dataset

We sampled seven scenarios from the CTU-13 botnet dataset representing five botnet families (Neris, Rbot, Menti, NsisAy, Virut), totaling 13.46 million network flows. The dataset exhibits realistic class imbalance: 98.20% normal traffic and 1.80% botnet traffic.

B. Preprocessing Pipeline

Data Cleaning: We consolidated scenarios and performed: (1) missing value imputation (median for numeric features), (2) duplicate removal, (3) exclusion of non-predictive variables (timestamps, IP addresses, ports), (4) binary label encoding (Normal=0, Botnet=1), and (5) infinite value handling. Memory optimization through downcasting reduced usage by 24%.

Feature Engineering: We processed three categorical features (Proto, Dir, State) and six numeric features (Dur, sTos, dTos, TotPkts, TotBytes, SrcBytes). For the State feature (432 distinct values), we retained states occurring in 0.1% of samples (27 states) and grouped rare states into ”other”. Proto (19 values) and Dir (7 values) were fully one-hot encoded. StandardScaler normalized numeric features, yielding 57 final features.

C. Data Balancing

We applied stratified 80/20 train-test split, then used RandomUnderSampler on training data to achieve 1:5 minority-to-majority ratio, producing a balanced training set of 1.16 million samples (83.33% normal, 16.67% botnet) while preserving all 193,502 minority samples.

D. Models

Random Forest: Configured with 100 trees, unlimited depth, bootstrap sampling, and parallel processing.

XGBoost: Parameters: 100 boosting rounds, learning rate 0.05, max depth 10, subsample 0.9, colsample_bytree 0.7.

LSTM Autoencoder: Architecture: Encoder (LSTM(64)→LSTM(32)), Decoder (RepeatVector→LSTM(32)→LSTM(64)→Dense(1)). Trained on normal traffic for 20 epochs using MSE loss. Detection threshold set at 95th percentile of reconstruction error.

E. Evaluation Metrics

We employed accuracy, precision, recall, F1-score, ROC-AUC, and Matthews Correlation Coefficient (MCC) for comprehensive performance assessment.

IV. Results

A. Overall Performance

Table I presents performance metrics for all models on the 2.69 million sample test set:

TABLE I
Model Performance Comparison

Model	Random Forest	XGBoost	LSTM AE
Accuracy	99.21%	98.93%	93.32%
Precision	71.07%	64.44%	0.05%
Recall	94.31%	90.54%	0.14%
F1-Score	81.06%	75.30%	0.07%
ROC-AUC	0.9968	0.9955	0.6789
MCC	0.8150	0.7590	-0.0298

B. Random Forest Analysis

Random Forest achieved the best overall performance with 99.21% accuracy and 94.31% recall, capturing 94% of all botnet flows. Confusion matrix: True Negatives 2,625,599, False Positives 18,572, False Negatives 2,753, True Positives 45,622. The false positive rate of 0.70% is substantially below the 5% target. High MCC (0.8150) and ROC-AUC (0.9968) indicate excellent class separation.

C. XGBoost Analysis

XGBoost achieved 98.93% accuracy and 90.54% recall, detecting 90% of botnet flows. Confusion matrix: True Negatives 2,620,004, False Positives 24,165, False Negatives 4,575, True Positives 43,800. False positive rate of 0.91% remains below target. ROC-AUC of 0.9955 demonstrates strong discrimination. Feature importance analysis revealed Feature 52, Feature 8, and Feature 21 as most significant, likely corresponding to timing patterns and packet counts characteristic of botnet behavior. While precision (64.44%) is lower than Random Forest, XGBoost offers faster training and inference.

D. LSTM Autoencoder Analysis

The autoencoder underperformed despite 93.32% nominal accuracy (inflated by class imbalance). Precision (0.05%) and recall (0.14%) were near-zero with negative MCC (-0.0298). Confusion matrix: True Negatives 2,512,520, False Positives 131,651, False Negatives 48,308, True Positives 67. Only 67 of 48,375 actual botnet flows were detected. ROC-AUC of 0.6789 indicates poor discrimination. Contributing factors include: (1) suboptimal threshold selection, (2) overlapping reconstruction errors between normal and botnet traffic, (3) limited temporal dependencies in flow-level features, and (4) training exclusively on normal traffic without exposure to botnet patterns.

E. ROC and Precision-Recall Analysis

ROC curves show Random Forest and XGBoost approaching the top-left corner with high true positive rates across all thresholds. Precision-recall curves maintain precision ¿0.90 across nearly all recall values (Average Precision: RF=0.923, XGB=0.905). LSTM curve remains near the random baseline, confirming limited discriminative capability.

V. Discussion

A. Supervised Learning Success

Random Forest and XGBoost exceeded the 90% accuracy target with false positives well below 5%. Success factors include: comprehensive feature engineering capturing behavioral patterns, effective class balancing, ensemble methods reducing overfitting, and ability to learn complex non-linear decision boundaries. Random Forest's superior precision (71% vs 64%) and recall (94% vs 91%) make it the preferred choice for production deployment, while XGBoost's computational efficiency offers a viable alternative.

B. Feature Importance Insight

XGBoost feature importance analysis highlights timing-related features, flow duration, and packet-rate distributions as key discriminators. This confirms that botnet C2 traffic exhibits distinctive behavioral signatures detectable without payload inspection, enabling detection of encrypted communications.

C. Autoencoder Limitations

The autoencoder's failure highlights challenges in unsupervised anomaly detection: (1) insufficient separation between normal and anomalous reconstruction errors, (2) difficulty in optimal threshold calibration, (3) weak temporal dependencies in flow features, and (4) lack of exposure to attack patterns during training. Future work should explore hybrid semi-supervised approaches, alternative architectures, and adaptive thresholding techniques.

D. Practical Implications

Results demonstrate practical viability for production deployment. High detection rates ensure most C2 traffic is intercepted, while low false positives minimize alert fatigue. Flow metadata-based detection supports encrypted traffic analysis. Successful generalization on CTU-13 scenarios suggests robustness across different botnet families. Production deployment requires: periodic model retraining, efficient real-time feature extraction, and integration with existing network monitoring infrastructure.

E. Comparison with Literature

Our Random Forest results (99.21% accuracy, 0.9968 ROC-AUC) exceed typical CTU-13 benchmark performance (95-98% accuracy) [5], [6]. Recall (94.31%) surpasses common ranges (85-90%), and our 0.7% false positive rate improves upon typical 2-5% rates. The flow-based approach's key advantage is applicability to encrypted C2 traffic through metadata analysis, maintaining effectiveness regardless of encryption.

VI. Threats to Validity

Internal validity and bias: The CTU-13 dataset originates from 2011 and represents five specific botnet families that may not reflect modern botnet behaviors. Contemporary botnets employ more sophisticated evasion techniques including adaptive traffic patterns, low-and-slow communication strategies, and mimicry of legitimate protocols. Random undersampling may eliminate informative normal traffic patterns, potentially impacting model robustness. Feature engineering choices may introduce bias toward specific attack types present in training data.

External Validity and Generalizability: Results are specific to CTU-13 scenarios and may not generalize to: (1) different network environments with distinct traffic characteristics, (2) modern botnets with evolved C2 protocols, (3) networks with different baseline traffic distributions, and (4) enterprise versus IoT versus cloud environments. The 2011 dataset age raises concerns about applicability to 2026 threat landscapes where botnets leverage machine learning for evasion, blockchain-based C2, and encrypted DNS channels. Cross-dataset validation on contemporary benchmarks (e.g., CIC-IDS-2017, IoT-23) is essential to establish generalizability.

Temporal Validity: The 13-year gap between dataset creation and current deployment scenarios represents a significant limitation. Botnet evolution during this period includes: widespread adoption of HTTPS, increased use of legitimate cloud services for C2, deployment of AI-powered adaptive malware, and proliferation of IoT-based botnets with distinct traffic patterns. Models trained on historical data may exhibit degraded performance against modern threats without continuous retraining on contemporary traffic.

Construct Validity: Engineered features may fail to capture emerging C2 characteristics such as covert channels in legitimate protocols, extremely low-rate communications, or behavioral mimicry. We assume CTU-13 labels are ground truth, but labeling errors may exist. The binary classification (botnet vs. normal) oversimplifies the spectrum of malicious activities and does not address multi-class detection scenarios.

Scalability Considerations: While computational efficiency was adequate for our 13.46 million flows, scalability to enterprise-scale traffic (billions of flows daily) requires validation. Real-time feature extraction latency and model inference time must be assessed under production load conditions. Memory requirements and distributed processing capabilities need evaluation for operational deployment.

VII. Future Work

Future research directions include: (1) evaluating advanced architectures (CNNs, Transformers, GNNs) on sequential flow data, (2) developing hybrid semi-supervised models combining labeled and unlabeled data, (3) implementing online learning for continuous adaptation to evolving threats, (4) engineering fine-grained temporal and graph-based features capturing network-wide behaviors, (5) developing explainable AI interfaces for security analyst decision support, (6) conducting real-time system deployment with production traffic, (7) performing cross-dataset validation on contemporary benchmarks (CIC-IDS-2017, IoT-23, Bot-IoT), and (8) investigating model robustness against adversarial evasion attempts. Addressing dataset age limitations through transfer learning from historical to contemporary traffic patterns represents a critical research priority.

VIII. Conclusion

This study presents a comprehensive empirical comparison of machine learning approaches for botnet C2 detection using network flow analysis. Random Forest achieved the best performance (99.21% accuracy, 94.31% recall, 0.70% false positive rate), exceeding our target metrics. XGBoost provided competitive results (98.93% accuracy, 90.54% recall, 0.91% false positive rate). Both supervised models substantially outperformed the LSTM Autoencoder anomaly detection approach.

Key findings: (1) flow-level features provide sufficient information for C2 detection without payload inspection, (2) supervised ensemble methods demonstrate strong generalization across multiple botnet families, (3) proper preprocessing and class balancing are critical for success, and (4) the approach remains effective for encrypted traffic. However, the 13-year dataset age and evolving botnet landscape necessitate validation on contemporary data before operational deployment.

Our framework provides actionable guidance for practitioners deploying network-based botnet detection. As botnets continue evolving with sophisticated evasion techniques, machine learning-based behavioral analysis offers advantages over traditional signature-based methods. This work contributes empirical evidence supporting the viability of flow-based detection for protecting enterprise and IoT networks against modern C2 threats, while highlighting the importance of continuous model adaptation to emerging attack patterns.

References

[1] M. Antonakakis et al., "Understanding the Mirai botnet," in *26th USENIX Security Symposium*, Vancouver, BC, 2017, pp. 1093–1110.

[2] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in *Third International Conference on Emerging Security Information, Systems and Technologies*, Athens, 2009, pp. 268–273.

[3] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, and D. Garant, "Botnet detection based on traffic behavior analysis and flow intervals," *Computers & Security*, vol. 39, pp. 2–16, 2013.

[4] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *LISA*, vol. 99, no. 1, 1999, pp. 229–238.

[5] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100–123, 2014.

[6] E. B. Beigi, H. H. Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *IEEE Conference on Communications and Network Security*, San Francisco, CA, 2014, pp. 247–255.

[7] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *European Symposium on Artificial Neural Networks*, Bruges, Belgium, 2015, pp. 89–94.

[8] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in *International Conference on Computing, Networking and Communications*, Honolulu, HI, 2014, pp. 797–801.

[9] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, 2016, pp. 785–794.

[10] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[12] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting botnet command and control channels in network traffic," in *Network and Distributed System Security Symposium*, San Diego, CA, 2008, pp. 1–18.

[13] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *International Conference on Information Systems Security and Privacy*, Funchal, Madeira, 2018, pp. 108–116.

[14] S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey," *Computer Networks*, vol. 57, no. 2, pp. 378–403, 2013.

[15] R. Perdisci, I. Corona, D. Dagon, and W. Lee, "Detecting malicious flux service networks through passive analysis of recursive DNS traces," in *Annual Computer Security Applications Conference*, Honolulu, HI, 2009, pp. 311–320.

[16] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[17] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[18] M. Alauthman, N. Aslam, L. Al-Kasassbeh, S. Khan, A. Al-Qerem, and K. R. Choo, "An efficient reinforcement learning-based botnet detection approach," *Journal of Network and Computer Applications*, vol. 150, p. 102479, 2020.

[19] P. Kumar, G. P. Gupta, and R. Tripathi, "Design of anomaly-based intrusion detection system using fog computing for IoT network," *Automated Software Engineering*, vol. 28, no. 1, pp. 1–29, 2021.

[20] M. Asadi, M. Jamzad, and H. Beigy, "A GNN-based architecture for group-level traffic classification in encrypted communications," *Computer Networks*, vol. 205, p. 108745, 2022.