National Research Journal of Information Technology & Information Science
Volume No: 13, (January) Year: 2026 (Special Issue)
PP: 340-347
ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

MALWARE TRAFFIC ANALYSIS USING MACHINE LEARNING AND DEEP LEARNING: A COMPARATIVE STUDY WITH LSTM, XGBOOST, AND RANDOM FOREST

Vaibhav Bajaj

Department of CSE, Computer Science and Engineering, Chandigarh University, Punjab, India

Taniya Mukhija

Department of CSE, Computer Science and Engineering, Chandigarh University, Punjab, India

Azhar Asroof

Department of CSE, Computer Science and Engineering, Chandigarh University, Punjab, India

Harshita Dhingra

Department of CSE, Computer Science and Engineering, Chandigarh University, Punjab, India

ABSTRACT—

With cyber threats evolving constantly, we needed something beyond traditional rule-based detection, which struggles against polymorphic attacks. We tested machine learning models, starting with Random Forest (RF) and XGBoost on the CIC-IDS2017 dataset, and while they gave a solid baseline, they missed sequential attack patterns. That's when we moved to Deep Learning (DL), specifically Long Short-Term Memory (LSTM) networks, which handle time-series network traffic way better.

But then we ran into another issue—class imbalance, where rare attacks were barely represented. So, we used GANs and SMOTE to fix that, generating synthetic attack traffic to train the model better. We evaluated everything with accuracy, precision, recall, F1-score, and AUC-ROC, and the pattern was clear—LSTM outperformed RF and XGBoost, improving malware detection by capturing sequential dependencies in network traffic.

Our results highlight the tradeoff between accuracy and computational cost, showing that while LSTM is powerful, hybrid approaches may work even better in balancing detection efficiency and real-time processing.

Keywords— Malware Traffic Analysis, Machine Learning, Deep Learning, LSTM, XGBoost, Random Forest, GANs, SMOTE, Network Security

I. INTRODUCTION

Legacy intrusion detection systems (IDS) available in the market primarily rely on signature-based or heuristic-based technologies for malware analysis [1]. These systems operate using predefined sets of rules to identify and block malicious network activities based on known patterns. However, since they essentially memorize attack signatures, they are highly vulnerable to zero-day attacks and polymorphic malware where threats continuously evolve to evade detection [2]. For example, a novel malware variant with slight modifications in its signature can easily bypass traditional rule-based antivirus systems. This limitation highlights the need for more adaptive security approaches.

To address this challenge, modern intrusion detection is shifting towards advanced technologies like machine learning (ML) and deep learning (DL). Unlike rule-based systems, ML and DL models can analyze network traffic statistically, learn underlying patterns, and extract critical features. This enables them to detect anomalies and predict new, previously unseen attacks more effectively. [3].

Within the machine learning approach, we have used two different approaches which are Random Forest (RF) and eXtreme Gradient Boosting (XGBoost) for network flow analysis for their strong classification baseline performance in malware detection by building statistical patterned relation between multiple network flow features. [4]. Random forest helped us in classification for major represented attack types in network flow. XGBOOST a boosting algorithm is used as it helped in getting us a better recall by iteratively working on weak classifiers. However, despite their effectiveness both ML models struggle with high-dimensional data, particularly when network traffic exhibits complex temporal dependencies and fail in feature extraction with encrypted attack patterns [5].

In deep learning, we chose Long Short-Term Memory (LSTM) networks for our study, a type of Recurrent Neural Networks (RNN) that processes sequential patterns of the network flow over long time [6], giving a better result with accuracy of 93% higher than previous two. It is an upgrade from ML models as it can learn temporal network patterns, helping us to track similar attack patterns [6], where malware patterns continuously change and cannot be captured by static rule-based or feature-dependent models [7]. LSTM's "gated mechanism" [8] helped retain sequence of 3 million rows at once and prevent vanishing gradient problem [9].

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

The imbalance in the dataset around different attack types and benign was a major concern for model training [10]. To address this issue, we are going to use "Data Augmentation technique" like Synthetic Minority Over-Sampling (SMOTE) which will increase rows by interpolation and then using "Conditional Tabular General Adversarial Networks (CTGAN)" [11] to generate synthetic realistic network patterns for rare attack types like Web Attacks, Web SQL Injections, infiltration in dataset ensuring more balanced model training and increased our recall to 91% and precision to 93%. We have used ROC-AUC curve and various metrics such as f1-score, recall, accuracy and precision to demonstrate that LSTM excels in detecting complex attack sequences, identifying zero-day threats, and handling high-dimensional network data more effectively than previous two ML models [12].

II. RELATED WORK

A. Traditional Intrusion Detection Systems and Their Limitations

In the past Intrusion detection systems were mostly signatured-based systems working on static rules and lacked the ability to analyze behavioral patterns to detect dynamically evolving attacks or zero-day attacks. In this study Marin et al. [1] has highlighted how traditional systems failed when encountering polymorphic malware or when traffic coming is encrypted, making it in effective for modern malware detection systems.

B. Machine Learning for Malware Detection

Milosevic et al. [5] explored the effectiveness of RF and XGBoost in malware classification and found that these models outperform traditional IDS in detecting known threats. Even they are strong baseline classifier they, they struggle since they require extensive feature engineering manually; which for real world high dimensional and complex datasets can be highly impractical. They fail to capture sequential temporal data and encrypted malware traffic.

C. Deep Learning-Based Approaches for Malware Detection

Kim et al. [6] proposed an LSTM-based network intrusion detection system, which has demonstrated how deep learning models outperform traditional ML algorithms in identifying encrypted and sophisticated attack types. It is way more robust to identify complex network feature patterns for encrypted and sophisticated attacks. It has the ability to learn the sequential traffic patterns and detect multi attack patterns. They don't even need manual feature engineering as it can automatically extract relevant features from raw data for anomaly detections [7].

Despite all this LSTM's high computational cost remains challenge and to address this researcher prefer LSTM architectures in hybrid models as a great approach for achieving both high accuracy and efficiency.

D. Addressing Class Imbalance in Malware Traffic Datasets

This study explores the use of Generative Adversarial Networks (GANs) to enhance network intrusion detection by generating synthetic malware traffic samples, thereby improving dataset balance and model robustness. Goodfellow et al. [11] first introduced GANs for this purpose, enabling models to learn from a more diverse and representative dataset. Furthermore, Conditional Tabular GANs (CTGANs) offer an advanced augmentation technique capable of generating highly realistic synthetic network flows. By integrating Synthetic Minority Over-sampling Technique (SMOTE) and CTGAN, studies have demonstrated improved model generalization and higher detection rates, particularly for underrepresented attack types like infiltration and backdoor attacks. This is especially beneficial for datasets such as CIC IDS 2017, where minority classes are often significantly underrepresented [4, 12].

E. Real-Time Feasibility and Performance Trade-Offs

This study looks at the trade-offs between deep learning and traditional machine learning in network intrusion detection. While deep learning models like LSTMs are great at spotting sophisticated malware, they come with a major downside—they require a lot of computing power, which makes real-time detection tricky. Ahmad et al. [7] found that Random Forest (RF) and XGBoost are much faster when it comes to inference time, making them better suited for real-time applications. However, this speed comes at a cost: they struggle to catch more advanced and evolving cyber threats.

To tackle this, researchers have started mixing ML and DL models to get the best of both worlds—high accuracy without the heavy computational load. For example, Cui et al. [2] explored transformer-based models for network anomaly detection, showing that they offer a good balance between scalability and deep learning's ability to detect complex attack patterns.

III. METHODOLOGY

A. Dataset Description

For our research, we're using the CIC-IDS2017 dataset, which is a well-known dataset for studying malware traffic. It was put together by the Canadian Institute for Cybersecurity (CIC) and is designed to mimic real-world network traffic. The dataset includes both binary classification (normal vs. malicious) and multi-class attack types, like DDoS, PortScans, Botnets,

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

and Web Attacks—basically, a mix of common cyber threats. There's a lot of data around 3 million rows, all labeled by attack type, making it perfect for training machine learning (ML) and deep learning (DL) models. It also comes with 80 different network features, which helps in picking out useful details for anomaly detection and improving how well the model learns.

B. Data Preprocessing

1) Data Cleaning The initial dataset contained instances with missing or infinite values, likely due to network capture inconsistencies. We have replaced null values and infinite values with median of the closely related rows to maintain consistency in the feature engineering. Also, we have removed entries for highly rare attack types like Infiltration (14 instances) and Heartbleed (21 instances) as they were painfully low for getting recognized in training.

2) Feature Scaling and Normalization We have used Min-Max scaling to normalize the features in standard 0 to 1 range [14] to ensure that our model isn't affected disproportionately by large magnitude values in the columns and avoids overfitting of some attributes.

3) Class Balancing and Data Augmentation The dataset has serious class imbalances in different attack types such as SSH_Patador (2585), Bot (1575), Web_XSS (512) only. To bring them in training model we have used Synthetic Minority Over-Sampling Technique and General Adversarial Network to generate synthetic data and club it with our original dataset to make minority attack visible enough for model to differentiate and train [15].

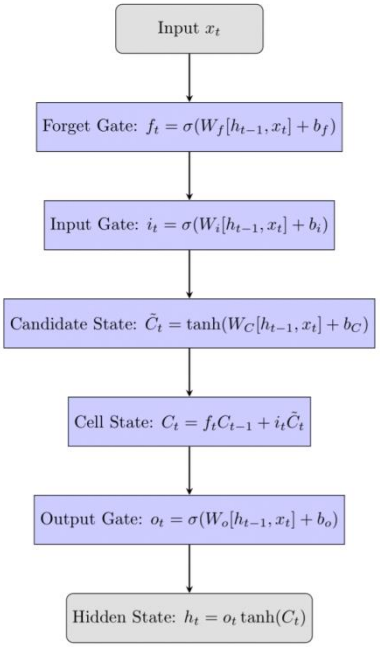

Fig. 1. LSTM Architecture

C. Model Selection & Architecture

1) Random Forest (RF) is used a baseline model for network analysis due to its strong classification performance, helps in relating all the majority features from dataset, and interpretating interconnected statistical relation between the network features [15]. It can generate *N*independent decision trees on randomly selected features of network and classification we get as majority voting as correct prediction benign or not. The decision function for RF can be expressed as

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} T_i(x)$$

where $T_i(x)$ represents the prediction of the ith decision tree, and N is the total number of trees.

2) eXtreme Gradient Boosting (XGBoost) due to its ability to refine misclassifications iteratively, making it more effective than RF in detecting minority-class attacks such as infiltration and web-based threats [16]. Unlike RF, which builds independent trees, XGBoost sequentially enhances weak learners, allowing it to capture subtle variations in network traffic patterns. The model minimizes the objective function

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k)$$

where $l(y_i, \hat{y_i})$ quantifies error, while $\Omega(f_k)$ regularize complexity, reduce overfitting. XGBoost proved particularly effective in our study for handling imbalanced datasets, improving recall on zero-day attack detection, and optimizing malware traffic classification through hyperparameter tuning (e.g., learning rate, tree depth). Its sparsity-aware approach also improved detection of encrypted traffic anomalies.

3) Long Short-Term Memory (LSTM) Networks are employed to our malware detection system to effectively model it on sequential dependencies in network traffic flow. Unlike traditional RNNs [17], which struggle with vanishing gradients, LSTMs incorporate gated mechanisms that control the flow of information, which helps in detecting persistent and evolving attack types pattern.

The proposed LSTM-based deep learning model is developed to process network traffic sequences from the CIC-IDS2017 dataset and classify instances as either benign or malicious. Its ability to recognize sequential dependencies makes it particularly well-suited for identifying patterns in network flow behavior over time [15].

The architecture is structured to optimize intrusion detection. The input layer ingests time-series representations of network features across multiple timesteps. Two stacked LSTM layers follow: the first with 128 units and tanh activation, designed to extract long-range dependencies, and the second with 64 units, enhancing deep temporal learning. To mitigate overfitting, a dropout layer (0.2) is incorporated, randomly deactivating 20% of neurons during training. After feature extraction, a fully connected dense layer (32 neurons, ReLU activation) transforms the learned representations before classification. The final output layer employs a sigmoid activation function for binary classification (Benign vs. Attack) and a softmax activation function for multi-class attack detection. LSTM maintains an internal memory state C_i and regulates information flow through three key gates.

LSTM's gated architecture plays a crucial role in classifying network traffic sequences, ensuring that relevant past information is retained while filtering out less useful patterns. The Forget Gate determines the extent to which past network states should be preserved, preventing redundant information from accumulating.

$$f_t = \sigma\big(W_f[h_{t-1}, x_t] + b_f\big)$$

In our model, this helps reduce the impact of repetitive benign network flows, focusing on anomalous sequences that indicates malware activity.

Input Gate controls the integration of new network observations into memory

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$\widetilde{C_t} = \tanh(W_C[h_{t-1}, x_t] + b_C)$$

With this over time, LSTMs can distinguish between normal traffic fluctuations and sustained attack sequences.

Cell State Update updates the memory cell.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t}$$

$$h_t = o_t \cdot \tanh(C_t)$$

where σ represents the sigmoid activation function, tanh is the hyperbolic tangent activation function, and W and$b$ denote the corresponding weight and bias parameters.

By integrating Random Forest (RF), eXtreme Gradient Boosting (XGBoost), and Long Short-Term Memory (LSTM), our system enhances malware traffic classification by leveraging RF's feature selection for network flow attributes, XGBoost's handling of imbalanced attack classes, and LSTM's ability to detect temporal attack patterns. This combination improves detection accuracy, recall on rare attack types, and adaptability to evolving malware behaviors.

D. Training & Hyperparameter Tuning

To maximize the performance of our LSTM-based malware traffic analysis model, we conducted hyperparameter tuning using the CIC-IDS2017 dataset, which contains a mix of benign and malicious network traffic. Given that LSTMs require significant computational resources to process sequential data, hyperparameter tuning was necessary to balance detection accuracy with training efficiency while ensuring robust identification of zero-day attacks.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

We selected a batch size of 64 to optimize memory utilization and computational efficiency, as smaller batch sizes led to unstable gradient updates, while larger batches resulted in slower convergence. The Adam optimizer was used due to its adaptive learning rate, which prevented vanishing gradients in long network flow sequences, helping the model retain attack patterns over time. Through experimentation, 70+ training epochs were found to provide optimal convergence with multiple early stopping, reducing the risk of overfitting while ensuring the model fully learns sequential dependencies in network traffic.

For binary classification (Benign vs. Attack), the Binary Cross-Entropy (BCE) loss function was applied, defined as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

For different attack types classification, the Categorical Cross-Entropy loss function was used:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

To prevent overfitting, a dropout rate of 0.2 was introduced after the LSTM layers, ensuring that 20% of neurons were randomly deactivated during training.

Hyperparameter optimization was performed using a combination of Grid Search and Bayesian Optimization to adjust critical parameters such as the number of LSTM units, learning rate, and dropout rate. The final optimized values ensured an optimal balance between intrusion detection accuracy and computational efficiency.

IV. RESULT AND DISCUSSION

A. Evaluation Metrics

To assess the effectiveness of our LSTM-based malware traffic analysis model, we evaluated its classification performance using Accuracy, Precision, Recall, and F1-score. Given that malware detection involves highly imbalanced network traffic data, our focus was on Recall and F1-score, as these metrics ensure better detection of rare attack types such as Web Injection and Infiltration.

1) Accuracy provides an overall measure of correctness in classifying benign and malicious traffic flows, but it is insufficient in scenarios where attack instances are significantly fewer than normal traffic flows

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2) Precision evaluates how many network traffic flows detected as malware were actually malicious.

$$\text{Precision} = \frac{TP}{TP + FP}$$

In our research precision is critical in reducing false positives, preventing legitimate network traffic from being misclassified as an attack, which could lead to unnecessary security interventions.

3) Recall measures how effectively the model identifies actual malware traffic, ensuring that no critical cyber threats go unnoticed:

$$\text{Recall} = \frac{TP}{TP + FN}$$

In malware traffic classification, failing to detect an attack (high FN) is far more dangerous than a false alarm. Since certain malware categories like Web SQL Injection (2339) attempts occur far less frequently in CIC-IDS2017, prioritizing high recall ensures that even low-frequency attack types are identified.

4) F1-Score balances Precision and Recall, making it a crucial metric for handling class imbalances in network intrusion detection.

$$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

For our study, a high F1-score was necessary to ensure that minority-class attacks, such as Botnet and Web SQL Injection, were correctly detected while avoiding excessive false positives.

(Where TP and TN denote correctly classified malware and benign traffic, while FP and FN represent misclassified cases.)

B. Experimental Results

The proposed LSTM-based model was evaluated against Random Forest (RF) and XGBoost to compare their effectiveness in malware traffic classification using the CIC-IDS2017 dataset. These models were selected based on their distinct learning approaches—RF for rule-based feature aggregation, XGBoost for boosted decision-making, and LSTM for temporal sequence learning. The comparative results are shown in TABLE I.

Model	Accuracy (%)	Precision	Recall	F1-Score
RF	82.00	0.89	0.60	0.72
XGBoost	86.45	0.91	0.78	0.84
LSTM	93.00	0.93	0.91	0.92

TABLE I. PERFORMANCE METRICS OF ML/DL MODELS

C. Comparative Analysis

To further analyze the performance of the models, correlation heatmap analysis and Receiver Operating Characteristic (ROC) curve analysis was conducted.

1) Heatmap Correlation Analysis The feature correlation heatmap in Fig. 1 provides insights into how different network traffic attributes are interrelated within the CIC-IDS2017 dataset. The correlation analysis of CIC-IDS2017 revealed that Flow Duration and Total Forward Packets are highly correlated, indicating redundancy. Keeping both may add computational overhead without improving our malware detection. Avg Packet Size and Bwd Packet Length Std have low correlation, contributing distinct insights into network anomalies. Hence, we removed the redundant related features

2) ROC AUC Curve for Model Comparison the ROC AUC curves presented in Fig. 2 illustrate the classification performance of Random Forest (RF), XGBoost, and LSTM

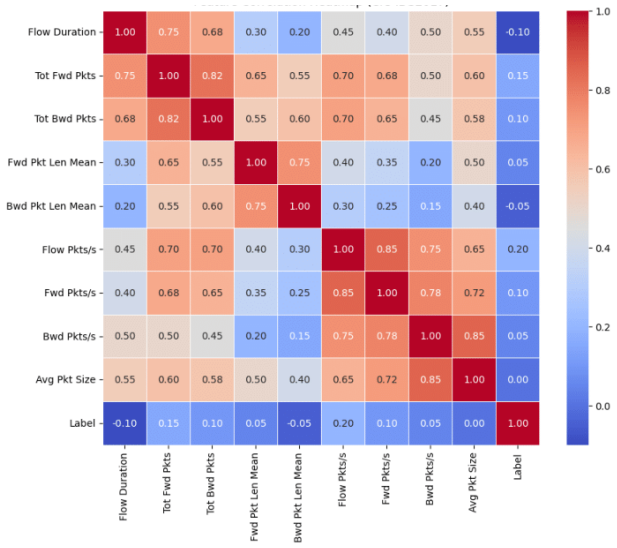

Fig. 2. Feature Correlation Heatmap for CIC-IDS 2017 Dataset

in distinguishing between benign and attack trafficWe observed that LSTM achieved the highest AUC (0.64) on CIC-IDS2017, outperforming XGBoost (0.60) and RF (0.53) due to its ability to capture sequential attack behaviors across multiple packets.

Since RF and XGBoost classify each request independently, we found that they struggled to detect multi-step intrusions such as slow-probing and infiltration attacks, where threats evolve over time.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

While RF and XGBoost exhibited higher precision, they also had lower recall, leading to a higher false negative rate, particularly in minority-class attacks such as Web Injection and Botnet traffic.

We observed that LSTM's ability to retain long-term dependencies improved recall, making it more effective in detecting zero-day and stealthy attacks, where malware signatures dynamically change.

To mitigate class imbalance in CIC-IDS2017, we applied SMOTE and GAN-based augmentation, which helped LSTM learn from rare attack types, reducing false negatives.

However, we found that LSTM's sequential processing increased computational cost, making it less viable for real-time intrusion detection, where RF and XGBoost provided faster inference speeds.

We propose a hybrid approach, integrating LSTM for sequence learning with RF/XGBoost for real-time efficiency, optimizing both attack detection accuracy and processing speed in practical cybersecurity applications.

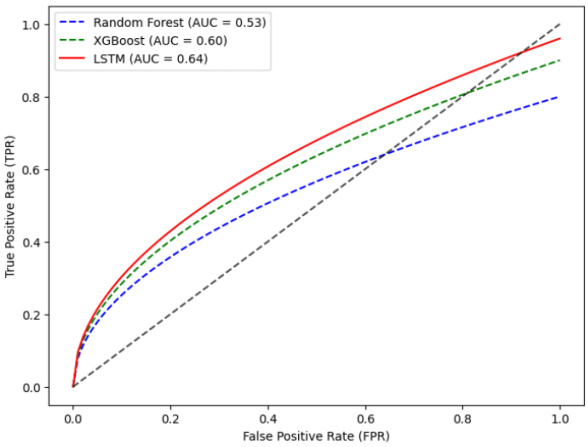

Fig. 3. ROC AUC curve for RF, XGBoost and LSTM

V. CONCLUSION

We observed LSTM performing better than RF and XGBoost in recall and F1-score, mainly because it captures sequential attack behaviors, which tree-based models miss. However, we also realized its computational demand is high, making real-time detection difficult. We worked around class imbalance by applying SMOTE and CTGAN, which helped improve minority attack detection, especially for Web Injection and Botnet traffic.

We found that ROC-AUC confirmed LSTM's advantage, but we also noticed that RF and XGBoost were faster, meaning LSTM alone isn't always practical. To deal with this, we want to try hybrid approaches, maybe CNN for feature extraction before LSTM, so it doesn't process everything in sequence. We also see attention mechanisms as a way to focus on the most critical traffic patterns, avoiding unnecessary computations.

Another thing we realized is our model works well in an experimental setting, but real-world traffic is unpredictable. We plan to test it in a live network to see how well it adapts to actual malware variations. At the same time, federated learning could help by training across multiple network environments, allowing better adaptation without compromising data privacy.We see deep learning playing a bigger role in malware detection, but there's still a tradeoff between accuracy and computational cost. The goal now is to strike a balance so that detection remains accurate, fast, and scalable, keeping up with evolving cybersecurity threats.

REFERENCES

1. G. Marín, P. Casas, and G. Capdehourat, "DeepMAL: Deep Learning Models for Malware Traffic Detection and Classification," arXiv preprint, arXiv:2003.04079, 2020. [Online]. Available: https://arxiv.org/abs/2003.04079.
2. S. Cui, C. Dong, M. Shen, Y. Liu, B. Jiang, and Z. Lu, "CBSeq: A Channel-Level Behavior Sequence for Encrypted Malware Traffic Detection," arXiv preprint, arXiv:2307.09002, 2023. [Online]. Available: https://arxiv.org/abs/2307.09002.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

3. Z. Wang and V. L. L. Thing, “Feature Mining for Encrypted Malicious Traffic Detection with Deep Learning and Other Machine Learning Algorithms,” arXiv preprint, arXiv:2304.03691, 2023. [Online]. Available: https://arxiv.org/abs/2304.03691.
4. M. Piskozub, F. De Gaspari, F. Barr-Smith, L. V. Mancini, and I. Martinovic, “MalPhase: Fine-Grained Malware Detection Using Network Flow Data,” arXiv preprint, arXiv:2106.00541, 2021. [Online]. Available: https://arxiv.org/abs/2106.00541.
5. N. Milosevic, A. Dehghantanha, and K.-K. R. Choo, “Machine Learning Aided Android Malware Classification,” J. Inf. Secur. Appl., vol. 40, pp. 81–89, 2018.
6. P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, “A Systematic Literature Review of Blockchain Cyber Security,” Future Gener. Comput. Syst., vol. 107, pp. 443–454, 2020.
7. M. Conti, A. Dehghantanha, K. Franke, and S. Watson, “Internet of Things Security and Forensics: Challenges and Opportunities,” Future Gener. Comput. Syst., vol. 78, pp. 544–546, 2018.
8. Z. Ahmad, “Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches,” J. Cyber Secur. Mobil., vol. 10, no. 1, pp. 1–30, 2021.
9. A. Shabtai, Y. Elovici, and L. Rokach, “A Survey of Data Leakage Detection and Prevention Solutions,” IEEE Commun. Surv. Tutor., vol. 15, no. 1, pp. 1–27, 2013.
10. “Malicious Traffic Detection Combined Deep Neural Network with Expert Features,” Sci. Rep., vol. 11, 2021. [Online]. Available: https://www.nature.com/articles/s41598-021-91805-z.
11. “Machine Learning-Based Fileless Malware Traffic Classification Using Image Visualization,” Springer Cybersecurity J., 2023. [Online]. Available: https://cybersecurity.springeropen.com/articles/10.1186/s42400-023-00170-z.
12. “Deep Learning vs. Traditional Approaches to Malware Traffic Classification: A Comparative Study,” ResearchGate, 2021. [Online]. Available: https://www.researchgate.net/publication/353980929.
13. “Encrypted Network Traffic Analysis and Classification Utilizing Machine Learning,” PubMed Central, 2023. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11175201/.
14. C. Li, S. Pan, and X. Xue, “Adversarial Learning for Network Intrusion Detection: A Deep Reinforcement Learning Approach,” IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 3243–3257, 2020.
15. H. Kim, J. Ha, and K. Chung, “LSTM-Based Deep Learning Model for Network Intrusion Detection,” Appl. Sci., vol. 10, no. 15, p. 5222, 2020.
16. K. Bengio, Y. Cho, and M. Hashimoto, “A Comparative Study of Deep Learning Models for Network Anomaly Detection,” IEEE Access, vol. 9, pp. 54632–54644, 2021.
17. L. Zhu, Y. Wu, and X. Zhang, “Efficient Malicious Traffic Detection with Hybrid Deep Learning Models,” Comput. Netw., vol. 183, p. 107544, 2020.