A Machine Learning-Integrated Dashboard for DNS Traffic Monitoring and Anomaly Detection

1st Sebastian Skubisz
Department of Computer Science
New Jersey Institute of Technology
Newark, New Jersey, USA
ss365@njit.edu

***Abstract*—This paper presents a comprehensive network dashboard designed to monitor Domain Name System traffic and detect anomalies using machine learning models. The objective is to provide real-time visibility into DNS query patterns, highlight suspicious domains, and facilitate proactive threat mitigation. The methodology integrates a web-based dashboard for traffic visualization and anomaly alerts, coupled with multiple supervised (Random Forest, K-Nearest Neighbors, Neural Network, XGBoost) and unsupervised (Isolation Forest, Local Outlier Factor, Autoencoder) learning models. Results indicate that supervised models trained on balanced datasets (via SMOTE) achieved high accuracy in classifying malicious and benign domains, while unsupervised models offered complementary anomaly detection insights. The significance of this study lies in demonstrating an end-to-end pipeline—from data ingestion and feature engineering to model evaluation and interactive visualization—which enables robust network monitoring and improving cyber defense strategies.(*Abstract*)**

Keywords—Anomaly Detection, Machine Learning, Network Traffic, Dashboard Visualization, Benign and Malicious Domains, DNS (Domain Name System)

I. Introduction

Evolving cyber threats are increasingly challenging network security. They exploit DNS with malicious activities such as phishing, malware distribution, and command-and-control communication. The traditional methods of monitoring DNS traffic are not sophisticated enough to combat modern cyber attacks. Using static rule sets and manual analysis are not sustainable against the fast evolving landscape of cyber attacks. The motive for this research is to develop an automated and data driven solution that integrates visualization and ML-based anomaly detection to enhance situational awareness and timely threat response.

The primary challenge for this was differentiating benign and malicious DNS queries, which are often hidden within vast volumes of network data. Other challenges that affect anomaly detection imbalanced datasets, evolving malicious domain patterns, and the need for low latency analysis.

This research aims to create a user-friendly dashboard that presents DNS traffic patterns and flagged anomalies; design a data pipeline to ingest and process both benign and malicious domain datasets; train and compare multiple supervised and unsupervised models for robust and accurate anomaly detection; and feature engineering for improved classification accuracy.

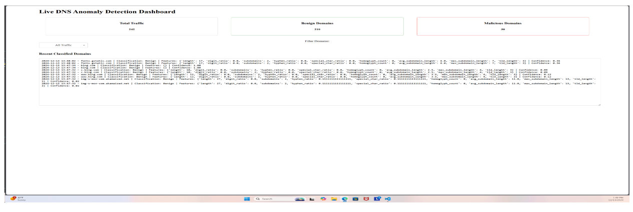
The paper is structured as follows: Section II introduces the system architecture and methodologies, including dashboard design, data processing, and model training. Section III presents results, including model performance metrics and insights from visualizations. Section IV concludes with a summary of findings and suggestions for future work.

II. System Architecture and Methodology

A. The implemented dashboard, developed using Python's Dash library, provides an intuitive interface for monitoring DNS traffic and classifying domains in real-time. It features counters that display the total number of DNS queries captured, alongside separate counts for domains classified as "Benign" or "Malicious" based on predictions from a pre-trained machine learning model. Users can filter the displayed data through a dropdown menu, selecting options to view all traffic, only benign domains, or malicious domains. The filtered results are shown in a text area, which includes essential details such as the query timestamp, domain name, classification (Benign or Malicious), the features used for classification, and a confidence score derived from the model.

The dashboard is designed to refresh periodically, ensuring that users always see the most recent metrics and domain classifications as new data is captured. This functionality is supported by the dcc.Interval component, which triggers automatic updates to the displayed information. DNS traffic is captured using the Scapy library, and each query is processed in real-time. Classification results are stored in an SQLite database, from which the dashboard retrieves and displays data.

The primary focus of the dashboard design is simplicity and efficiency, prioritizing clear and actionable insights over complex visualizations. While it does not include advanced features such as geolocation mapping or anomaly-specific visualizations, the dashboard serves as a foundational tool for administrators to quickly monitor DNS traffic and identify potentially harmful domains.



```
2024-12-13 13:48:02 - fonts.gstatic.com | Classification: Benign | Features: {'length': 17, 'digit_ratio': 0.0, 'subdomains': 2, 'hyphen_ratio': 0.0, 'special_char_ratio': 0.0, 'homoglyph_count': 0, 'avg_subdomain_length': 6.0, 'max_subdomain_length': 7, 'tld_length': 3} | Confidence: 0.36
2024-12-13 13:48:02 - fonts.gstatic.com | Classification: Benign | Features: {'length': 17, 'digit_ratio': 0.0, 'subdomains': 2, 'hyphen_ratio': 0.0, 'special_char_ratio': 0.0, 'homoglyph_count': 0, 'avg_subdomain_length': 6.0, 'max_subdomain_length': 7, 'tld_length': 3} | Confidence: 0.36
2024-12-13 13:47:36 - bing.com | Classification: Benign | Features: {} | Confidence: 1.00
2024-12-13 13:47:36 - bing.com | Classification: Benign | Features: {} | Confidence: 1.00
2024-12-13 13:47:32 - c.bing.com | Classification: Benign | Features: {'length': 10, 'digit_ratio': 0.0, 'subdomains': 2, 'hyphen_ratio': 0.0, 'special_char_ratio': 0.0, 'homoglyph_count': 0, 'avg_subdomain_length': 2.5, 'max_subdomain_length': 4, 'tld_length': 3} | Confidence: 0.09
2024-12-13 13:47:32 - c.bing.com | Classification: Benign | Features: {'length': 10, 'digit_ratio': 0.0, 'subdomains': 2, 'hyphen_ratio': 0.0, 'special_char_ratio': 0.0, 'homoglyph_count': 0, 'avg_subdomain_length': 2.5, 'max_subdomain_length': 4, 'tld_length': 3} | Confidence: 0.09
2024-12-13 13:47:32 - www.bing.com | Classification: Benign | Features: {'length': 12, 'digit_ratio': 0.0, 'subdomains': 2, 'hyphen_ratio': 0.0, 'special_char_ratio': 0.0, 'homoglyph_count': 0, 'avg_subdomain_length': 3.5, 'max_subdomain_length': 4, 'tld_length': 3} | Confidence: 0.12
2024-12-13 13:47:32 - www.bing.com | Classification: Benign | Features: {'length': 12, 'digit_ratio': 0.0, 'subdomains': 2, 'hyphen_ratio': 0.0, 'special_char_ratio': 0.0, 'homoglyph_count': 0, 'avg_subdomain_length': 3.5, 'max_subdomain_length': 4, 'tld_length': 3} | Confidence: 0.12
2024-12-13 13:47:32 - img-s-msn-com.akamaized.net | Classification: Benign | Features: {'length': 27, 'digit_ratio': 0.0, 'subdomains': 2, 'hyphen_ratio': 0.1111111111111111, 'special_char_ratio': 0.1111111111111111, 'homoglyph_count': 0, 'avg_subdomain_length': 11.0, 'max_subdomain_length': 13, 'tld_length':
3} | Confidence: 0.02
2024-12-13 13:47:32 - img-s-msn-com.akamaized.net | Classification: Benign | Features: {'length': 27, 'digit_ratio': 0.0, 'subdomains': 2, 'hyphen_ratio': 0.1111111111111111, 'special_char_ratio': 0.1111111111111111, 'homoglyph_count': 0, 'avg_subdomain_length': 11.0, 'max_subdomain_length': 13, 'tld_length':
3} | Confidence: 0.02
```

Fig. 1. *Dashboard for DNS monitoring*

B. Data Pipeline

Benign domains were sourced from publicly available top-ranked domain lists, representing legitimate websites commonly accessed by users. Examples of such sources include Alexa's top websites and similar reputable datasets. For malicious domains, data was gathered from well-known security resources, including ViriBack and PhishTank. These repositories provide lists of confirmed malicious domains and phishing websites, making them ideal for training anomaly detection models. Benign domain data was structured in CSV files, whereas malicious domains were parsed from blacklists and phishing repositories, ensuring comprehensive coverage of known threats.

To classify the domains effectively, each domain was processed into a set of handcrafted features. These features capture various domain characteristics, such as the total length of the domain, the ratio of digits to the total length, the number of subdomains, the presence of special characters (e.g., hyphens or underscores), and entropy, which quantifies the randomness or unpredictability of the domain name. By extracting such features, the model is equipped to identify patterns and anomalies indicative of benign or malicious behavior. To better understand feature interrelationships and their influence on model training, a heatmap was generated to visualize the correlation among these features. This heatmap highlights key patterns, such as highly correlated features or those that independently contribute to classification accuracy.

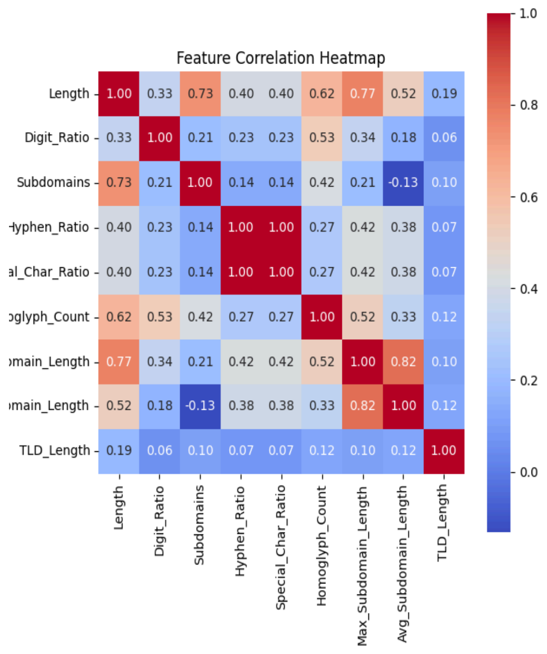


Fig. 2. *Heatmap for feature correlation*

A significant challenge in working with real-world datasets is class imbalance. Typically, there are far more benign domains than malicious ones. This imbalance can lead to biased models that favor the majority class (benign domains) at the expense of correctly identifying the minority class (malicious domains). To address this, the Synthetic Minority Over-sampling Technique (SMOTE) was employed. SMOTE generates synthetic samples for the minority class by interpolating between existing data points, effectively increasing the number of malicious domain samples. This results in a balanced dataset, enhancing the model's ability to generalize and perform effectively on unseen data.

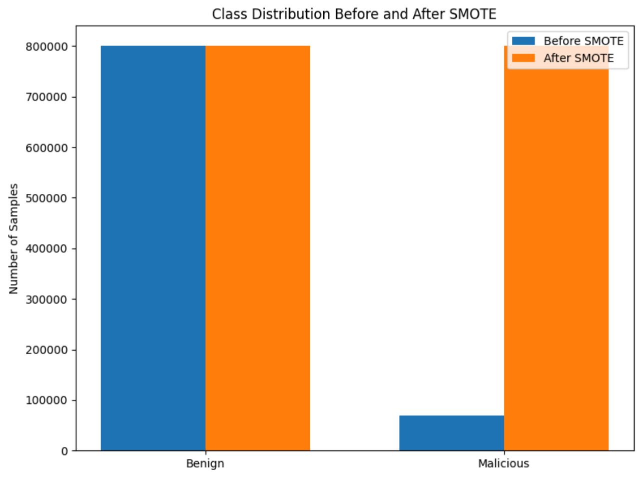


Fig. 3. *Dataset before and after SMOTE*

C. Training Models

Supervised learning models were trained using labeled datasets, where benign and malicious domains were clearly distinguished. The primary goal was to classify domains accurately and evaluate their performance based on well-defined metrics such as precision, recall, and F1-score.

Supervised Models

- **Neural Network:** The Neural Network, implemented as a Keras model, proved to be the best-performing model among all. It was built with multiple dense layers and utilized ReLU activations along with dropout for regularization. Trained on SMOTE-balanced datasets to address class imbalance, it demonstrated robust learning capabilities. Early stopping was applied to prevent overfitting, and hyperparameters like batch size and learning rate were fine-tuned for optimal performance. The model achieved the highest accuracy (0.939413) and precision (0.672334) across all models while maintaining a competitive recall (0.467991).
- **Random Forest:** The Random Forest model combined multiple decision trees to improve classification accuracy and provided valuable insights into feature importance, identifying the most influential domain characteristics. Hyperparameters such as the number of trees and maximum depth were optimized using randomized search, ensuring robust performance. This model achieved an accuracy of 0.849602, with a high recall (0.738152), making it reliable for predictions across various scenarios.

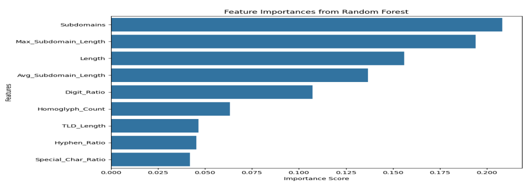


Fig. 4. *Feature importance in Random Forest*

- **K-Nearest Neighbors (KNN):** K-Nearest Neighbors used Euclidean distances for classification and served as an effective baseline model. While it performed well in terms of accuracy (0.923699) and recall (0.541361), it struggled with high-dimensional features and noisy data, which limited its performance compared to more sophisticated models like Neural Networks and Random Forest.
- **XGBoost:** XGBoost, a gradient-boosting model, was designed to handle imbalanced datasets by iteratively focusing on misclassified samples to improve predictions. It demonstrated strong performance in identifying malicious domains, achieving a recall of 0.811638 and a competitive ROC AUC of 0.807757. However, its precision (0.263864) was lower than other supervised models, reflecting its slight trade-off between sensitivity and specificity.

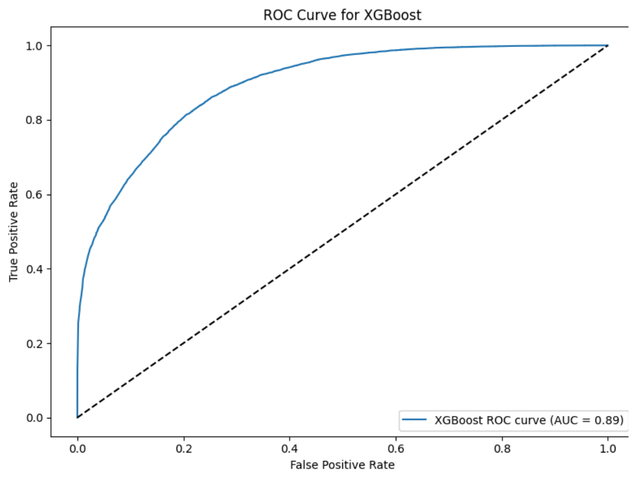
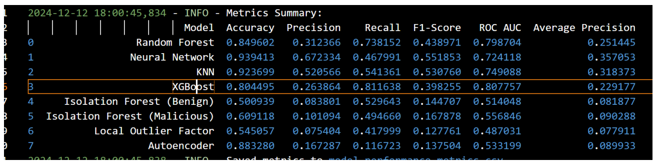


Fig. 5. *ROC Curve for XGBoost*

Unsupervised learning models were trained on both benign and malicious domains independently. This dual approach allowed the detection of anomalies and novel patterns that might not align with known labels, simulating real-world zero-day threats.

Unsupervised Models

- **Isolation Forest (Benign):** The Isolation Forest model trained exclusively on benign data, effectively detecting anomalies by isolating outliers in the feature space. While it performed well in identifying irregularities, its higher false-positive rate reduced overall reliability compared to supervised models. It achieved an accuracy of 0.500939 and a precision of 0.083801, showcasing its focus on benign traffic patterns.
- **Isolation Forest (Malicious):** The malicious-focused variant of the Isolation Forest was trained to detect deviations from known threats, providing complementary insights to its benign counterpart. Despite its utility in flagging edge-case anomalies, the model's accuracy (0.609118) and precision (0.101094) remained lower, indicating its limitations in identifying complex malicious patterns.
- **Local Outlier Factor (LOF):** The Local Outlier Factor model identified anomalies by comparing the density of a data point to its neighbors. This method required careful tuning of sensitivity and specificity parameters to balance detection accuracy. Despite its potential, LOF struggled to achieve competitive performance, reflected in its low accuracy (0.545057) and precision (0.075404).
- **Autoencoder:** The Autoencoder model, designed to reconstruct input data, flagged inputs with high reconstruction errors as anomalies. It proved effective in identifying subtle irregularities within benign traffic, achieving an accuracy of 0.883280. However, its precision (0.167287) and recall (0.116723) were significantly lower, highlighting limitations in anomaly detection for more complex or malicious patterns.



```
2024-12-12 18:00:45,834 - INFO - Metrics Summary:
│    │    │    │    │    │    │ Model  Accuracy  Precision    Recall  F1-Score   ROC AUC  Average Precision
0                  Random Forest  0.849602   0.312366  0.738152  0.438971  0.798704           0.251445
1                 Neural Network  0.939413   0.672334  0.467991  0.551853  0.724118           0.357053
2                            KNN  0.923699   0.520566  0.541361  0.530760  0.749088           0.318373
3                        XGBoost  0.804495   0.263864  0.811638  0.398255  0.807757           0.229177
4       Isolation Forest (Benign)  0.500939   0.083801  0.529643  0.144707  0.514048           0.081877
5    Isolation Forest (Malicious)  0.609118   0.101094  0.494660  0.167878  0.556846           0.090288
6            Local Outlier Factor  0.545057   0.075404  0.417999  0.127761  0.487031           0.077911
7                    Autoencoder  0.883280   0.167287  0.116723  0.137504  0.533199           0.089933
```

Fig. 6. *Metric Summary*

D. Observations

Supervised and unsupervised models were trained on both benign and malicious subsets to better understand the impact of data selection on model behavior. Models trained on benign data excelled at detecting irregularities, effectively isolating outliers within the feature space. Conversely, models trained on malicious data focused on identifying deviations from known threat patterns, proving particularly useful for edge-case anomaly detection. This comparative approach provided actionable insights into how different training strategies influence model performance, highlighting the importance of tailoring datasets to specific detection goals.

Training the models posed significant challenges due to the inherent class imbalance in DNS datasets, where benign domains vastly outnumber malicious ones. This imbalance risked skewing model performance toward benign classifications, undermining the ability to detect malicious domains. To address this issue, SMOTE (Synthetic Minority Oversampling Technique) was employed, as detailed in Section II. By generating synthetic samples for the minority class, SMOTE enhanced dataset balance, enabling more effective training for both supervised and unsupervised models. The application of SMOTE not only improved classifier performance in supervised models but also reduced bias in anomaly detection methods, ensuring a more comprehensive and unbiased evaluation of network traffic.

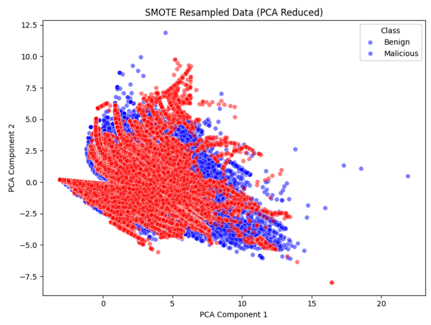


Fig. 7. *SMOTE Resampled Data*

E. Performance Metrics

The models were evaluated using key classification metrics such as accuracy, precision, recall, and the F1-score. Accuracy measured the overall correctness of predictions, while precision focused on the percentage of correctly identified malicious domains among those flagged. Recall assessed the model's ability to detect actual malicious domains, and the F1-score provided a balanced metric combining precision and recall.

Advanced metrics like ROC AUC (Receiver Operating Characteristic Area Under the Curve) and average precision were also used to analyze model performance across varying thresholds. For unsupervised models, outliers were mapped to binary classifications (malicious or benign) to ensure consistent evaluation with supervised models.

Visualization tools enhanced the understanding of model results. Recursive Feature Elimination (RFE) was employed to identify the most significant features, ensuring models were trained with optimal inputs. Confusion matrices highlighted classification errors and successes, while ROC curves and precision-recall curves illustrated trade-offs in performance, particularly when addressing the challenges of imbalanced datasets.

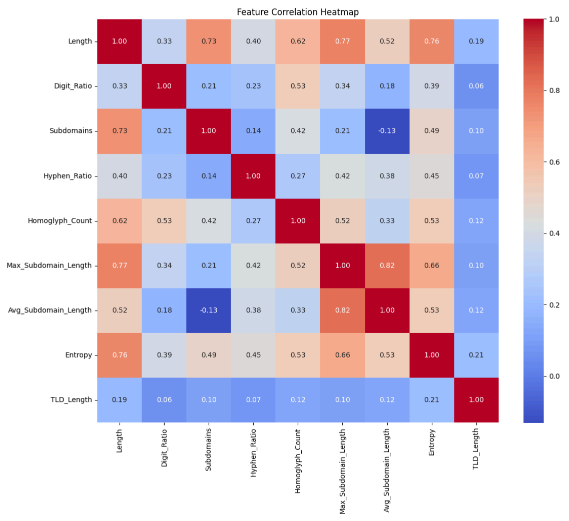


Fig. 8. *Reduced Feature set after RFE*

III. Results and Discussion

A. Training Performance

The training phase for supervised models demonstrated varying levels of efficiency. Both Random Forest and XGBoost classifiers trained relatively quickly, especially after hyperparameter tuning using randomized search methods. Their short training durations highlighted their computational efficiency, even with a large feature set. On the other hand, the neural network required longer training times due to the iterative nature of its epochs. However, early stopping was employed to stabilize the training process and prevent overfitting, ensuring a balance between performance and training time. The application of SMOTE increased training durations, as the dataset size expanded significantly due to synthetic minority samples. Despite this, SMOTE played a crucial role in enhancing the balance and overall generalizability of the models.

B. Evaluation Results

The evaluation phase highlighted the varying strengths of supervised and unsupervised models. Among the supervised classifiers, the Neural Network emerged as one of the most effective models. Trained with SMOTE-balanced datasets and benefiting from Recursive Feature Elimination (RFE) for optimal feature selection, the neural network demonstrated high F1-scores and precision-recall metrics. Fine-tuned thresholds further enhanced its classification accuracy, placing it on par with ensemble models like Random Forest and XGBoost. The tuned Random Forest and XGBoost classifiers also performed exceptionally well, providing balanced and reliable results for distinguishing malicious from benign domains. In contrast, K-Nearest Neighbors (KNN), though simpler and less effective in high-dimensional spaces, offered valuable baseline insights for comparative analysis.

Unsupervised models played a pivotal role in anomaly detection, revealing complementary strengths. The Isolation Forest (Benign) effectively identified outliers in benign traffic but incurred a higher false positive rate compared to supervised models. The Isolation Forest (Malicious), trained exclusively on malicious domains, offered a distinct advantage by capturing deviations from established threat patterns. The Local Outlier Factor (LOF) was particularly adept at identifying novelty cases, while the Autoencoder, leveraging reconstruction errors, excelled at detecting subtle anomalies. However, the Autoencoder's performance was highly sensitive to threshold settings. Training unsupervised models on either benign or malicious data sets significantly influenced their focus, with benign-trained models excelling at detecting unusual patterns and malicious-trained models capturing deviations from known threats.

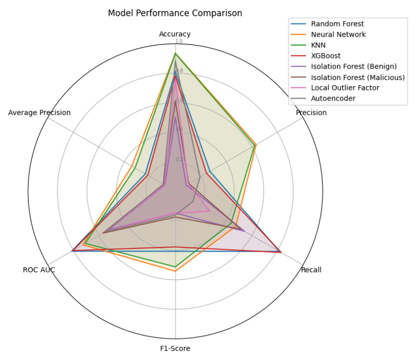


Fig. 9. *Visualization of performance for each model*

The radar graph provides a comparative visualization of the models' performance across metrics such as accuracy, precision, recall, F1-score, ROC AUC, and average precision. It underscores the strengths of the Neural Network, particularly its top performance in F1-score and overall metrics, as well as ensemble models like Random Forest and XGBoost in supervised learning tasks. Meanwhile, the unsupervised models demonstrated trade-offs, excelling in specific anomaly detection scenarios but falling short in terms of precision and F1-score compared to their supervised counterparts.

C. Insights from Dashboard Visualization

The integrated dashboard provided a simple yet effective interface for monitoring DNS traffic and anomaly detection. Key metrics such as total traffic, total benign queries, and total malicious queries were prominently displayed, offering users a clear overview of network activity. The logging feature allows administrators or users to filter logs based on traffic type—viewing either all traffic or isolating benign or malicious queries—making it easier to analyze specific patterns. While the dashboard focused on these essential functionalities, its streamlined design ensured real-time visibility into network events and provided a solid foundation for proactive threat analysis. Future iterations could build upon this framework by introducing additional visualizations and advanced analytics.

D. Limitations and Future Work

Despite its strengths, the system has notable limitations. Real-time processing is constrained by the periodic update mechanism of the dashboard, which may not be sufficient for high-volume, real-time network environments. The feature set, though comprehensive, is limited to handcrafted features, potentially missing insights offered by advanced automated feature extraction techniques, such as lexical analysis and WHOIS-based features. Additionally, the use of static thresholds in anomaly detection models restricts their adaptability to dynamic traffic patterns.

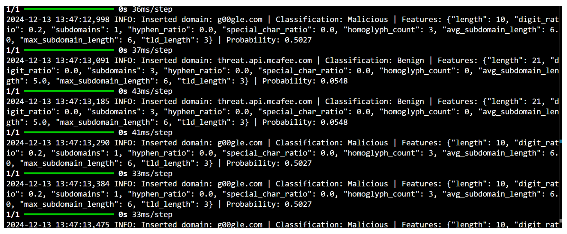
Future improvements will address these limitations through the incorporation of real-time streaming architectures, enabling instant anomaly detection. Enhanced feature engineering will expand the scope of detection, integrating advanced datasets and methods to improve classification accuracy. Dynamic thresholding mechanisms will replace static values, enabling models to adapt to evolving traffic baselines. Finally, continuous learning approaches will be explored to ensure the system evolves alongside emerging cyber threats, maintaining its effectiveness in a constantly changing security landscape.

IV. Conclusion

Anomaly detection is vital in identifying irregularities across various fields, from cybersecurity to healthcare. This study explored the application of supervised and unsupervised machine learning models to classify DNS traffic and detect malicious activities effectively. Techniques like Recursive Feature Elimination (RFE) and SMOTE improved feature selection and addressed dataset imbalances, enhancing model accuracy and reliability.

The research highlights the strengths of different anomaly detection approaches, such as statistical methods, density-based techniques, and reconstruction-based models like Autoencoders. While these methods offer advantages like early problem detection and enhanced security, challenges such as imbalanced data, false positives, and scalability require ongoing improvements.

Future work should focus on integrating deep learning, enhancing unsupervised methods for adaptability, and emphasizing Explainable AI for better transparency. Real-time processing through edge computing and hybrid approaches will further improve detection capabilities. This study demonstrates the importance of anomaly detection in managing modern data challenges, providing a foundation for future advancements in this critical area.



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
2024-12-13 13:47:12,998 INFO: Inserted domain: g00gle.com | Classification: Malicious | Features: {"length": 10, "digit_rat
io": 0.2, "subdomains": 1, "hyphen_ratio": 0.0, "special_char_ratio": 0.0, "homoglyph_count": 3, "avg_subdomain_length": 6.
0, "max_subdomain_length": 6, "tld_length": 3} | Probability: 0.5027
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
2024-12-13 13:47:13,091 INFO: Inserted domain: threat.api.mcafee.com | Classification: Benign | Features: {"length": 21, "d
igit_ratio": 0.0, "subdomains": 3, "hyphen_ratio": 0.0, "special_char_ratio": 0.0, "homoglyph_count": 0, "avg_subdomain_len
gth": 5.0, "max_subdomain_length": 6, "tld_length": 3} | Probability: 0.0548
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2024-12-13 13:47:13,185 INFO: Inserted domain: threat.api.mcafee.com | Classification: Benign | Features: {"length": 21, "d
igit_ratio": 0.0, "subdomains": 3, "hyphen_ratio": 0.0, "special_char_ratio": 0.0, "homoglyph_count": 0, "avg_subdomain_len
gth": 5.0, "max_subdomain_length": 6, "tld_length": 3} | Probability: 0.0548
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
2024-12-13 13:47:13,290 INFO: Inserted domain: g00gle.com | Classification: Malicious | Features: {"length": 10, "digit_rat
io": 0.2, "subdomains": 1, "hyphen_ratio": 0.0, "special_char_ratio": 0.0, "homoglyph_count": 3, "avg_subdomain_length": 6.
0, "max_subdomain_length": 6, "tld_length": 3} | Probability: 0.5027
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
2024-12-13 13:47:13,384 INFO: Inserted domain: g00gle.com | Classification: Malicious | Features: {"length": 10, "digit_rat
io": 0.2, "subdomains": 1, "hyphen_ratio": 0.0, "special_char_ratio": 0.0, "homoglyph_count": 3, "avg_subdomain_length": 6.
0, "max_subdomain_length": 6, "tld_length": 3} | Probability: 0.5027
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
2024-12-13 13:47:13,475 INFO: Inserted domain: g00gle.com | Classification: Malicious | Features: {"length": 10, "digit_rat
```

Fig. 10. *Model working on live domain names*

References

[1] DataCamp, "An Introduction to Anomaly Detection," [Online]. Available: https://www.datacamp.com/tutorial/introduction-to-anomaly-detection. [Accessed: Dec. 13, 2024].

[2] GeeksforGeeks, "Anomaly Detection using R," [Online]. Available: https://www.geeksforgeeks.org/anomaly-detection-using-r/. [Accessed: Dec. 13, 2024].

[3] Y. Jin, K. Kakoi, N. Yamai, N. Kitagawa, and M. Tomoishi, "A Client Based Anomaly Traffic Detection and Blocking Mechanism by Monitoring DNS Name Resolution with User Alerting Feature," *2018 International Conference on Cyberworlds (CW)*, Singapore, 2018, pp. 351-356, doi: 10.1109/CW.2018.00070.

[4] Z. Wang and M. Zhang, "The Research of DNS Anomaly Detection Based on the Method of Similarity and Entropy," *2010 International Conference on Intelligent Computation Technology and Automation*, Changsha, China, 2010, pp. 905-909, doi: 10.1109/ICICTA.2010.442.