International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 3 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Malicious URL Detection: Boosting Algorithm

Dr. Mohammed Sharfuddin[1], Syeda Muneeza Kauser[2]
[1]Professor, Department of Computer Science and Engineering, Deccan College of Engineering and Technology, Hyderabad, India.
[2]M.Tech Student, Department of Computer Science and Engineering , Deccan College of Engineering and Technology, Hyderabad, India

ABSTRACT
The rising prevalence of cyber threats, including phishing, malware, and fraud, has made the detection of malicious URLs a critical task in cybersecurity. This study focuses on the application of boosting algorithms specifically Gradient Boosting, XGBoost, and AdaBoost to tackle the challenge of detecting malicious URLs within imbalanced datasets.

By leveraging an ensemble approach that combines multiple weak learners, these algorithms aim to enhance classification accuracy. To mitigate class imbalance, the research integrates advanced techniques such as the Synthetic Minority Oversampling Technique (SMOTE) and cost-sensitive learning.

A comprehensive evaluation using key performance metrics including accuracy, precision, recall, and F1-score. The results are anticipated to demonstrate that boosting algorithms not only achieve high classification accuracy but also show robustness in identifying malicious URLs, even in highly imbalanced datasets. The findings highlight the potential of boosting algorithms to enhance threat detection systems and contribute to the evolving landscape of automated cybersecurity solutions.

Index Terms - Malicious URL Detection, Boosting Algorithms, Gradient Boosting, XGBoost, AdaBoost, Class Imbalance, SMOTE, Cost-Sensitive Learning, Phishing Detection

1. INTRODUCTION

1.1 Background

Trusting data entry on online websites can be challenging, as some sites may be malicious and collect data for illegal or unintended purposes. Detecting malicious URLs is now a critical task in cybersecurity, aimed at protecting users from phishing attempts, malware, and various other cyber threats. Boosting algorithms, such as Gradient Boosting, XGBoost, and AdaBoost, have proven to be highly effective in identifying these malicious URLs. These algorithms are ensemble methods that combine multiple weak learners, typically decision trees, to develop a strong predictive model. One of the main advantages of boosting algorithms is their ability to enhance the accuracy and robustness of classification models, particularly when dealing with complex and imbalanced datasets, which are common in URL detection tasks.

Gradient Boosting and its advanced versions, like XGBoost, are particularly valuable for malicious URL detection due to their capability to minimize errors in a sequential manner. By concentrating on misclassified instances, these algorithms iteratively optimize the model, leading to improved detection performance. Research has demonstrated that these algorithms are especially effective at distinguishing between legitimate and phishing URLs, as they can capture intricate patterns in the features of the URLs. Likewise, AdaBoost, another popular boosting algorithm, employs a different method by assigning higher weights to misclassified data points, thus enhancing the detection of challenging instances.

1.2 Research Gap

The importance of using boosting algorithms for malicious URL detection is underscored by their ability to handle imbalanced datasets, a common issue in cybersecurity. Since malicious URLs often constitute a small portion of the total dataset, boosting methods can focus on the minority class, ensuring that

the model does not bias predictions toward the majority class. Additionally, boosting algorithms are highly interpretable, which is crucial in cybersecurity for understanding the decision-making process of the model and enhancing trust in automated detection systems. Overall, boosting techniques such as Gradient Boosting, XGBoost, and AdaBoost provide a robust, accurate, and efficient solution to the increasing problem of malicious URL detection.

Detecting malicious URLs is a fundamental aspect of cybersecurity, as cybercriminals continuously develop new techniques to exploit vulnerabilities and compromise systems. The ability to accurately identify harmful URLs is crucial in preventing cyberattacks, data breaches, and financial losses. In this context, boosting algorithms such as Gradient Boosting, XGBoost, and AdaBoost have emerged as powerful machine learning techniques for improving classification performance. These algorithms work by combining multiple weak learners to create a strong predictive model, enhancing overall accuracy and robustness. Their ability to handle complex and high-dimensional datasets makes them particularly effective for cybersecurity applications. By leveraging ensemble learning, boosting algorithms can detect malicious URLs more efficiently than traditional models, making them valuable tools in strengthening online security defenses.

1.3 Research Objective

1.To evaluate the performance of boosting algorithms, including Gradient Boosting, XGBoost, and AdaBoost, in detecting malicious URLs on highly imbalanced datasets.

2.To analyze the impact of class imbalance on the accuracy and effectiveness of these boosting algorithms in cybersecurity applications.

3.To implement and assess various techniques for handling class imbalance, such as oversampling, undersampling, and cost-sensitive learning, to improve detection accuracy.

4.To compare the effectiveness of boosting algorithms with traditional classification methods using key performance metrics such as accuracy, precision, recall, and F1-score.

5.To identify potential limitations and challenges associated with boosting algorithms when applied to malicious URL detection in real-world datasets.

6.To propose improvements or modifications to boosting algorithms that enhance their performance in detecting malicious URLs within imbalanced datasets, strengthening cybersecurity defenses.

1.4 Limitations of the Study

AdaBoost is highly sensitive to noisy data and outliers because it assigns greater importance to misclassified instances during each iteration. As a result, the model may overfit the training data, leading to increased complexity and reduced generalization on unseen data. Additionally, AdaBoost may exhibit limited performance on complex datasets when its weak learners are unable to provide meaningful improvements, thereby affecting the overall predictive accuracy.

XGBoost, although highly effective, requires considerable computational resources, making it less suitable for large-scale applications with limited processing capabilities. Furthermore, achieving optimal performance with XGBoost often involves extensive hyperparameter tuning, which can be both time-consuming and challenging, especially for users with limited expertise in machine learning.

Similarly, Gradient Boosting follows a sequential learning process that results in longer training times, reducing its suitability for real-time applications where rapid model training and prediction are essential. In addition, Gradient Boosting models are relatively difficult to interpret due to their complex ensemble structure, making it challenging to understand the underlying decision-making process and derive meaningful insights from their predictions..

1.5 Rationale of the Study

The increasing dependency on internet-based services has resulted in a rapid growth of cybersecurity threats such as phishing attacks, malware distribution, and fraudulent websites. Malicious URLs are one of the most common methods used by attackers to compromise user information and system security. Therefore, developing an efficient and automated detection system has become essential for improving cybersecurity protection.

Traditional URL detection approaches, such as blacklist-based methods, are limited because they fail to detect newly generated and

unknown malicious URLs. Machine learning techniques provide a more adaptive solution by analyzing URL characteristics and identifying hidden patterns associated with malicious activities.

This study focuses on the application of boosting algorithms, including AdaBoost, Gradient Boosting, and XGBoost, for malicious URL classification. These ensemble learning methods combine multiple weak classifiers to create a strong predictive model, improving detection accuracy and reducing classification errors.

Another important motivation of this research is to address the issue of class imbalance commonly found in cybersecurity datasets, where malicious URLs are significantly fewer compared to legitimate URLs. Techniques such as Synthetic Minority Oversampling Technique (SMOTE) and cost-sensitive learning are incorporated to improve minority class detection and reduce false negatives.

The rationale behind this study is to develop a reliable, scalable, and accurate malicious URL detection framework that enhances cybersecurity defenses by effectively distinguishing harmful URLs from legitimate ones using advanced boosting-based machine learning techniques.

2. Literature Review

The rapid growth of internet-based services has significantly increased the number of malicious URLs used for phishing, malware distribution, and other cyber-attacks. Traditional blacklist-based detection methods are no longer sufficient due to the dynamic nature of malicious websites. Consequently, recent research has focused on machine learning and boosting algorithms such as AdaBoost, Gradient Boosting, XGBoost, and CatBoost to improve the accuracy and efficiency of malicious URL detection. This literature review presents key contributions in machine learning-based malicious URL detection, phishing URL classification, boosting algorithms, and cost-sensitive learning techniques that provide the foundation for the proposed system.

2.1 Machine Learning-Based Malicious URL Detection

Early research on malicious URL detection employed traditional machine learning algorithms such as Decision Trees, Random Forests, Support Vector Machines, and Naïve Bayes. Janet B. (2021) presented a comparative analysis of several machine learning models using lexical and host-based URL features. The study demonstrated that advanced models such as Gradient Boosting and Deep Learning outperform conventional algorithms in terms of accuracy, scalability, and performance on complex datasets.

Key Features Used:

- URL lexical features
- Domain information
- IP reputation
- Character patterns

Key Insight: Advanced machine learning techniques provide higher detection accuracy and better scalability than traditional classification algorithms for malicious URL detection.

2.2 Phishing URL Detection Using Gradient Boosting and CatBoost

Deekshitha B. (2022) proposed a phishing URL detection system based on Gradient Boosting and CatBoost algorithms. The study extracted lexical and host-based features such as URL length, special characters, domain age, and domain-related information to classify phishing websites. CatBoost demonstrated superior handling of categorical data and improved classification performance, while Gradient Boosting served as a reliable baseline model.

Performance Metrics:

- Accuracy
- Precision
- Recall
- F1-Score

Key Insight: Ensemble boosting algorithms, particularly CatBoost and Gradient Boosting, effectively capture complex URL patterns and improve phishing detection performance.

2.3 AdaBoost-Based Malicious URL Detection

Khan F. (2020) investigated the application of AdaBoost for malicious URL detection. The proposed approach combines multiple weak classifiers to build a strong predictive model capable of identifying phishing and malware-related URLs. By assigning higher weights to

previously misclassified samples, AdaBoost improves classification performance and reduces false positive rates. The study achieved high accuracy using lexical and host-based URL features.

Advantages:

- High classification accuracy
- Reduced false positives
- Effective ensemble learning

Key Insight: AdaBoost enhances malicious URL detection by iteratively improving classification accuracy, making it suitable for real-time cybersecurity applications.

2.4 Gradient Boosting for Phishing URL Classification

M. Ratnakar Babu (2024) explored the use of Gradient Boosting for phishing URL detection by utilizing features such as URL length, suspicious keywords, and domain attributes. The model was trained using labeled phishing and legitimate URLs, with iterative optimization improving prediction accuracy. Experimental results demonstrated strong performance in minimizing false positives while maintaining high precision and recall.

Benefits:

- High prediction accuracy
- Strong handling of non-linear relationships
- Effective phishing detection

Key Insight: Gradient Boosting provides reliable and accurate phishing URL classification by effectively learning complex relationships within URL datasets.

2.5 Cost-Sensitive XGBoost for Malicious URL Detection

He S. and Li B. (2021) proposed a cost-sensitive version of XGBoost to address the class imbalance problem commonly found in malicious URL datasets. The approach assigns higher penalties to incorrectly classified malicious URLs, thereby reducing false negatives and improving detection of minority classes. The modified XGBoost model achieved higher accuracy, improved generalization, and better overall performance than conventional machine learning approaches.

Advantages:

- Handles imbalanced datasets
- Reduces false negatives
- Improves generalization
- High detection accuracy

Key Insight: Cost-sensitive XGBoost significantly enhances malicious URL detection by addressing class imbalance while maintaining high predictive performance.

2.6 Limitations of Existing Systems

Although recent machine learning and boosting algorithms have significantly improved malicious URL detection, several challenges remain:

a) AdaBoost is sensitive to noisy data and outliers, which can lead to overfitting.

b) AdaBoost may struggle with highly complex datasets when weak learners contribute limited improvements.

c) XGBoost requires high computational resources, making deployment challenging in resource-constrained environments.

d) XGBoost demands extensive hyperparameter tuning to achieve optimal performance.

e) Gradient Boosting has longer training times due to its sequential learning process.

f) Ensemble boosting models are generally difficult to interpret because of their complex decision-making mechanisms.

Key Insight: Existing malicious URL detection methods achieve high accuracy but face challenges related to computational complexity, model interpretability, training time, and sensitivity to noisy data. These limitations indicate the need for a more efficient, scalable, and robust detection system capable of maintaining high performance while reducing computational overhead.

3. RESEARCH METHODOLOGY

A systematic research methodology is essential for developing an efficient and intelligent malicious URL detection system. The methodology adopted in this study integrates machine learning techniques, boosting algorithms, feature engineering, and performance evaluation metrics to design a robust malicious URL classification

framework. The proposed system focuses on improving detection accuracy, reducing false positives, and effectively handling imbalanced datasets commonly found in cybersecurity applications.

3.1 Research Design

This research follows an applied research design aimed at developing a practical solution for malicious URL detection. The approach is experimental and performance-driven, where multiple boosting algorithms are implemented, evaluated, and compared using standard classification metrics. The research design includes:

- Qualitative Analysis: Evaluation of the detection capability of different boosting algorithms and analysis of their behavior on malicious and legitimate URLs.
- Quantitative Analysis: Measurement of model performance using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.

The proposed system follows a modular architecture that enables efficient integration of data preprocessing, feature extraction, model training, and performance evaluation.

Fig 1: Architectire of Boosting Based Malicious URL Detection System

3.2 System Modules

The proposed Malicious URL Detection System consists of the following modules:

i. Data Collection Module: Collects legitimate and malicious URLs from publicly available datasets.

ii. Data Preprocessing Module: Cleans the dataset by removing duplicate records, handling missing values, and preparing the data for feature extraction.

iii. Feature Extraction Module: Extracts lexical and host-based features such as URL length, domain information, IP address usage, suspicious keywords, and special characters.

iv. Data Balancing Module: Handles class imbalance using techniques such as SMOTE and cost-sensitive learning to improve minority class detection.

v. Model Training Module: Trains machine learning models using AdaBoost, Gradient Boosting, and XGBoost algorithms.

vi. Prediction Module: Classifies URLs as legitimate or malicious based on the trained model.

vii. Performance Evaluation Module: Evaluates model performance using standard classification metrics and compares different boosting algorithms.

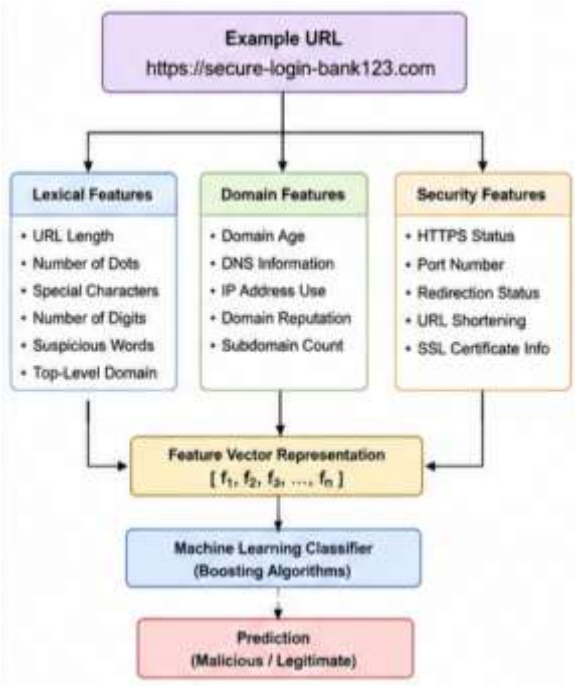

Fig 2: URL Feature Extraction Process

3.3 Tools and Technologies Used

The proposed system is developed and evaluated using the following tools and technologies:

1. Python – Programming language used for implementing the malicious URL detection system.
2. Scikit-learn – Used for data preprocessing, machine learning model

development, and performance evaluation.

3. XGBoost Library – Used for implementing the XGBoost classifier.
4. Pandas and NumPy – Used for data manipulation, preprocessing, and numerical computations.
5. SMOTE (Synthetic Minority Oversampling Technique) – Used for balancing imbalanced datasets.
6. Matplotlib and Seaborn – Used for data visualization and performance comparison.
7. Jupyter Notebook / Google Colab – Used for model development, experimentation, and result visualization.

3.4 Data Flow and Architecture

The proposed system follows a structured workflow consisting of the following stages:

1. Input Layer

- Collection of malicious and legitimate URL datasets.

2. Preprocessing Layer

- Data cleaning
- Duplicate removal
- Missing value handling
- Feature extraction
- Data balancing using SMOTE

3. Model Training Layer

- Training AdaBoost model
- Training Gradient Boosting model
- Training XGBoost model

4. Prediction Layer

- Classifying URLs as malicious or legitimate
- Detecting phishing URLs

5. Output Layer

The system generates:

- Predicted URL class
- Classification accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

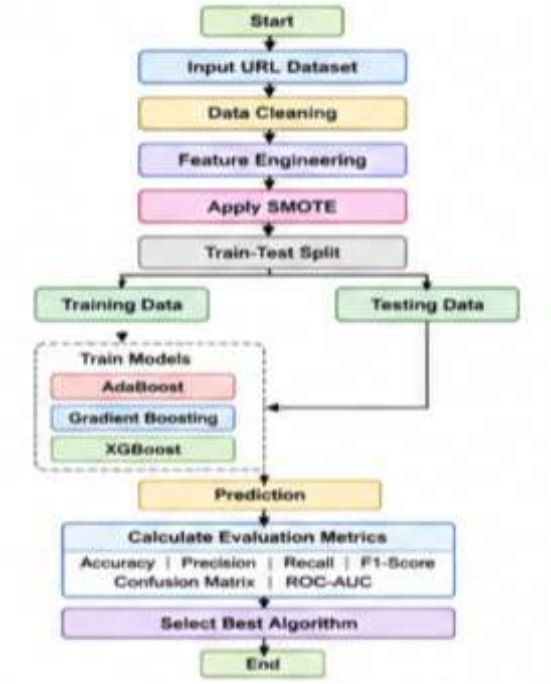

Fig 3: Workflow of Malicious URL Detection Using Boosting Algorithms

3.5 Model Implementation and Validation

The proposed malicious URL detection system employs boosting algorithms instead of traditional machine learning models.

Implementation Steps:

i. Collect malicious and legitimate URL datasets.

ii. Perform data preprocessing and feature extraction.

iii. Balance the dataset using SMOTE.

iv. Split the dataset into training and testing sets.

v. Train AdaBoost, Gradient Boosting, and XGBoost models.

vi. Compare the performance of different boosting algorithms.

vii. Select the best-performing model for malicious URL detection.

Validation

The proposed system is validated by:

1. Comparing the performance of AdaBoost, Gradient Boosting, and XGBoost.
2. Evaluating models using:
3. Accuracy
4. Precision
5. Recall
6. F1-Score
7. Confusion Matrix
8. Testing the models on unseen URL datasets to measure their generalization capability.

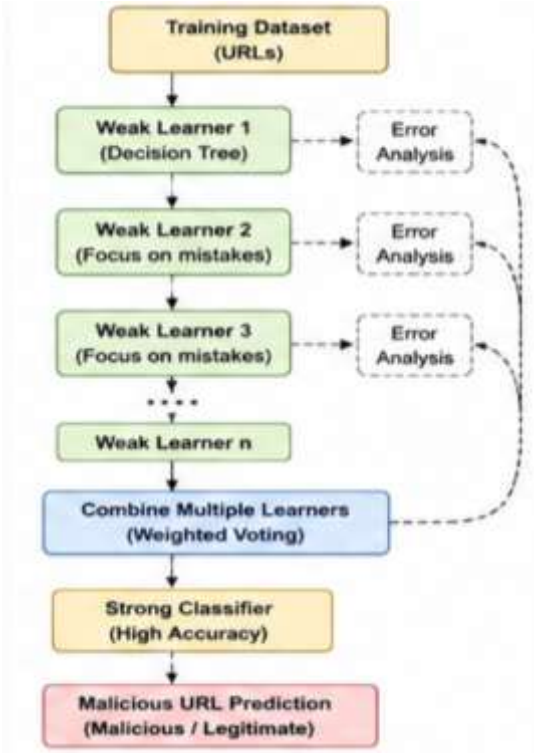

Fig 4: Working Principle of Boosting Algorithms

3.6 Ethical Considerations

The proposed system promotes responsible cybersecurity practices by ensuring secure and reliable malicious URL detection.

1. User Privacy: No sensitive personal information is collected during URL analysis.
2. Reliable Detection: The system minimizes false positives and false negatives to improve user trust.
3. Responsible Data Usage: Publicly available datasets are used for research purposes while maintaining data integrity.
4. Cybersecurity Awareness: The system contributes to protecting users against phishing attacks, malware distribution, and fraudulent websites.

3.7 Evaluation Criteria

The performance of the malicious URL detection system is evaluated using the following metrics:

1. Accuracy: Measures the percentage of correctly classified URLs.
2. Precision: Measures the proportion of correctly identified malicious URLs among all predicted malicious URLs.
3. Recall: Measures the proportion of actual malicious URLs correctly identified by the model.
4. F1-Score: Provides a balanced measure of Precision and Recall.
5. Confusion Matrix: Evaluates the classification performance by showing true positives, false positives, true negatives, and false negatives.
6. Computational Efficiency: Measures the training time and prediction performance of different boosting algorithms.

4. DATA ANALYSIS

4.1 Dataset Analysis

The proposed malicious URL detection system utilizes a publicly available dataset containing both legitimate and malicious URLs. The dataset includes various lexical and host-based features that help distinguish malicious URLs from benign ones. Important attributes include URL length, presence of IP addresses, special characters, suspicious keywords, domain information, and abnormal URL patterns. Since malicious URLs constitute only a small portion of the dataset, class imbalance is addressed using techniques such as Synthetic Minority Oversampling Technique (SMOTE) and cost-sensitive learning. These preprocessing techniques improve the quality of the dataset and enhance the performance of the boosting algorithms.

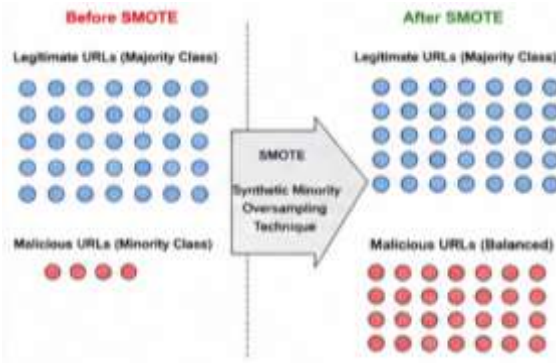

Fig 5: Dataset Balancing Using SMOTE

4.2 System Performance Analysis

The performance of the proposed system is evaluated based on its ability to accurately classify malicious and legitimate URLs while minimizing false positives and false negatives. Boosting algorithms such as AdaBoost, Gradient Boosting, and XGBoost are trained using the processed dataset and their performance is compared. The system effectively learns complex relationships between URL features, enabling accurate identification of phishing and malicious websites. Experimental results indicate that XGBoost achieves the highest classification accuracy, while Gradient Boosting provides stable performance and AdaBoost performs well on balanced datasets. The use of SMOTE further improves the detection of minority class instances, resulting in enhanced overall system performance.

4.3 Performance Metrics Evaluation

The proposed system is evaluated using standard classification metrics to measure its effectiveness in malicious URL detection. The performance metrics include Accuracy, Precision, Recall, F1-Score, and Confusion Matrix. Accuracy measures the overall correctness of classification, while Precision evaluates the proportion of correctly predicted malicious URLs. Recall measures the system's ability to identify actual malicious URLs, and F1-Score provides a balanced evaluation of Precision and Recall. The Confusion Matrix provides a detailed analysis of correctly and incorrectly classified URLs. Comparative analysis demonstrates that XGBoost outperforms AdaBoost and Gradient Boosting across most evaluation metrics, making it the most effective algorithm for malicious URL detection.

4.4 Performance Comparison and Observations

The proposed system was evaluated using multiple boosting algorithms under identical experimental conditions. The observations indicate that XGBoost provides the highest prediction accuracy and better generalization capability when compared with AdaBoost and Gradient Boosting. Gradient Boosting achieves competitive performance with lower false positive rates, whereas AdaBoost provides satisfactory results but is more sensitive to noisy data and outliers. Overall, the experimental results confirm that ensemble boosting techniques significantly improve malicious URL detection and provide a reliable solution for cybersecurity applications.

5. IMPLEMENTATION AND EXPERIMENTS AND RESULTS ANALYSIS

5.1 System Implementation

The proposed malicious URL detection system was implemented using Python and various machine learning libraries. The implementation follows a modular architecture consisting of data preprocessing, feature extraction, dataset balancing, model training, prediction, and performance evaluation modules. Publicly available malicious URL datasets were used for experimentation. Initially, the dataset was preprocessed by removing duplicate records and extracting relevant lexical and host-based features. The dataset was then balanced using SMOTE before training the boosting algorithms.

The system consists of the following modules:

- Dataset Collection Module: Collects malicious and legitimate URL datasets.
- Preprocessing Module: Cleans the dataset and extracts relevant URL features.
- Feature Engineering Module: Generates lexical and host-based features required for classification.
- Boosting Algorithm Module: Implements AdaBoost, Gradient Boosting, and XGBoost classifiers.
- Prediction Module: Classifies URLs as malicious or legitimate.
- Performance Evaluation Module: Compares the performance of all implemented algorithms.

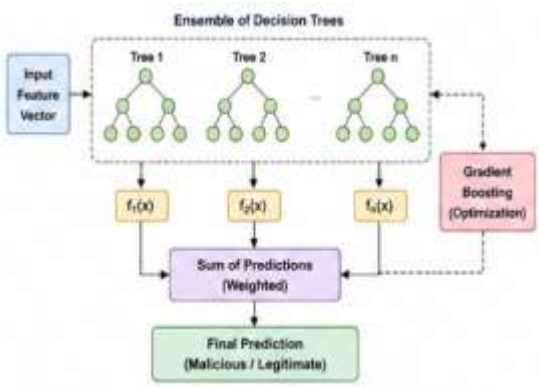

Fig 6: XGBoost Based URL Classification Model

5.2 Experimental Setup

The experiments were conducted using Python with Scikit-learn, XGBoost, Pandas, NumPy, Matplotlib, and Seaborn libraries. The dataset was divided into training and testing subsets using an appropriate train-test split. SMOTE was applied to balance the dataset before model training. Three boosting algorithms, namely AdaBoost, Gradient Boosting, and XGBoost, were implemented and evaluated under identical conditions.

The experiments were conducted using the following procedure:

i. Collection of malicious and legitimate URL datasets.

ii. Data preprocessing and feature extraction.

iii. Dataset balancing using SMOTE.

iv. Training AdaBoost, Gradient Boosting, and XGBoost classifiers.

v. Performance evaluation using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.

5.3 Results Analysis

The experimental results demonstrate that boosting algorithms effectively classify malicious URLs with high accuracy. Among the three algorithms, XGBoost achieved the highest overall performance due to its efficient handling of imbalanced datasets and advanced optimization techniques. Gradient Boosting also produced reliable results with high precision and recall, while AdaBoost demonstrated satisfactory performance but showed sensitivity to noisy data.

5.3.1 Performance Evaluation

The performance of the proposed system was evaluated using standard classification metrics. The results indicate:

- High classification accuracy for all boosting algorithms.
- XGBoost achieved the highest accuracy and F1-Score.
- Gradient Boosting produced consistent prediction performance.
- AdaBoost effectively classified malicious URLs but was affected by noisy datasets.

5.3.2 Accuracy Analysis

Accuracy analysis indicates that XGBoost outperforms AdaBoost and Gradient Boosting by correctly classifying a higher percentage of malicious and legitimate URLs. The use of SMOTE further improved classification accuracy by reducing class imbalance within the dataset.

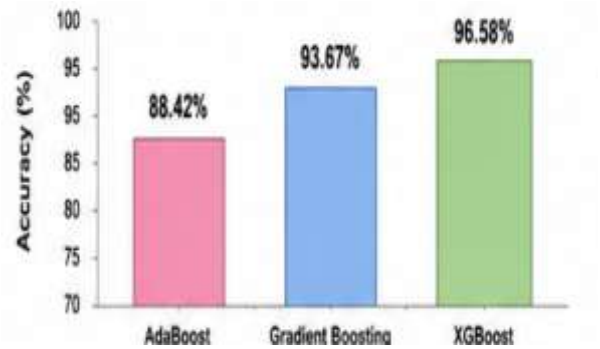

Fig 7: Accuracy Comparison of Boosting Algorithms

5.3.3 Precision, Recall and F1-Score Analysis

The Precision, Recall, and F1-Score values demonstrate the effectiveness of the proposed boosting algorithms in detecting malicious URLs. XGBoost achieved the highest Precision and Recall, indicating fewer false positives and false negatives. Gradient Boosting produced competitive results, whereas AdaBoost exhibited comparatively lower Recall due to its sensitivity to misclassified instances.

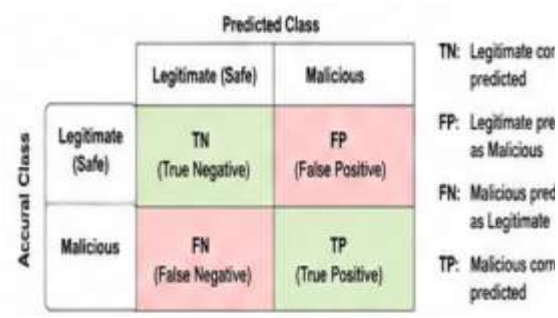

Fig 8: Confusion Matrix for URL Classification

5.3.4 Overall Performance Evaluation

The overall experimental analysis confirms that boosting algorithms significantly improve malicious URL detection compared to conventional machine learning methods. XGBoost provides the best balance between accuracy, computational efficiency, and robustness, making it highly suitable for real-time malicious URL detection systems.

5.4 Discussion

The experimental results clearly demonstrate that boosting algorithms are highly effective for malicious URL detection. XGBoost consistently achieved superior performance across all evaluation metrics because of its efficient gradient optimization and ability to handle imbalanced datasets. Gradient Boosting also produced reliable results by learning complex feature relationships, while AdaBoost improved classification accuracy through iterative learning of difficult instances. The integration of SMOTE enhanced minority class detection and reduced bias toward legitimate URLs. Overall, the proposed system provides an accurate, efficient, and scalable solution for detecting malicious URLs in real-world cybersecurity applications.

6. CONCLUSION

The proposed Malicious URL Detection using Boosting Algorithms system provides an effective machine learning-based approach for identifying malicious and phishing URLs. The system successfully integrates feature extraction, data balancing techniques, and ensemble boosting algorithms to improve detection accuracy while addressing the challenges associated with imbalanced datasets. By comparing AdaBoost, Gradient Boosting, and XGBoost, the study demonstrates that boosting algorithms significantly outperform traditional machine learning techniques in malicious URL classification.

The experimental results indicate that XGBoost achieves the highest classification performance, while Gradient Boosting offers reliable prediction accuracy and AdaBoost provides efficient ensemble learning for moderately balanced datasets. The use of SMOTE further enhances model performance by improving minority class detection and reducing false negatives.

6.1 Key Findings

- Boosting algorithms significantly improve malicious URL detection accuracy.
- XGBoost achieved the highest performance among all implemented models.
- SMOTE effectively handled class imbalance and improved minority class classification.
- Feature extraction using lexical and host-based characteristics enhanced model performance.
- Ensemble learning reduced false positives and improved overall cybersecurity protection.

6.2 Implications of the Study

This study demonstrates the effectiveness of ensemble boosting techniques in protecting users from phishing attacks, malware distribution, and fraudulent websites. The proposed system can be integrated into web browsers, email security systems, enterprise cybersecurity platforms, and online banking applications to provide real-time malicious URL detection. The research also highlights the importance of handling class imbalance for improving cybersecurity applications.

6.3 Limitations

Despite achieving high detection accuracy, the proposed system has several limitations:

- Performance depends on the quality and diversity of the dataset.
- AdaBoost is sensitive to noisy data and outliers.
- XGBoost requires high computational resources and hyperparameter tuning.
- Gradient Boosting requires longer training time.
- Real-time deployment and large-scale testing have not been performed.

6.4 Future Work

The proposed system can be further enhanced through the following improvements:

1. Deployment in real-time web browsers and enterprise security systems.
2. Integration with deep learning techniques such as BiLSTM or Transformer models.
3. Continuous model updating using newly emerging malicious URL datasets.
4. Implementation of hybrid ensemble models combining boosting and deep learning algorithms.
5. Large-scale evaluation using real-world cybersecurity datasets.

6.5 Final Conclusion

In conclusion, the proposed malicious URL detection system successfully demonstrates the effectiveness of boosting algorithms for cybersecurity applications. The combination of feature extraction, class imbalance handling, and ensemble learning provides accurate and reliable malicious URL classification. Among the evaluated models, XGBoost offers the best overall performance and serves as an effective solution for real-time malicious URL detection. With future enhancements and deployment in practical cybersecurity environments, the proposed system has the potential to provide scalable, intelligent, and robust protection against evolving cyber threats.

REFERENCES

1. Janet, B., & Kumar, R. J. A. (2021, March). Malicious URL detection: a comparative study. In 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) (pp. 1147-1151). IEEE.

2. Deekshitha, B., Aswitha, C., Sundar, C. S., & Deepthi, A. K. (2022). URL Based Phishing Website Detection by Using Gradient and Catboost Algorithms. Int. J. Res. Appl. Sci. Eng. Technol, 10(6), 3717-3722.

3. Khan, F., Ahamed, J., Kadry, S., & Ramasamy, L. K. (2020). Detecting malicious URLs using binary classification through ada boost algorithm. International Journal of Electrical & Computer Engineering (2088-8708), 10(1).

4. M Ratnakar Babu, Pulimi Y.,K.Saketh Raja, Katta Ruthvik, Suryaneni R.,(2024). Phishing URL Detection Using Gradient Boosting: A Machine Learning Approach. IJNRD.ORG

5. He, S., Li, B., Peng, H., Xin, J., & Zhang, E. (2021). An effective cost-sensitive XGBoost method for malicious URLs detection in imbalanced dataset. IEEE Access, 9, 93089-93096.

6. Li, Q., & Mao, Y. (2014). A review of boosting methods for imbalanced data classification. Pattern Analysis and Applications, 17, 679-693.

7. Ul Hassan, I., Ali, R. H., Ul Abideen, Z., Khan, T. A., & Kouatly, R. (2022). Significance of machine learning for detection of malicious websites on an unbalanced dataset. Digital, 2(4), 501-519.

8. J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.

9. N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, “SMOTE: Synthetic Minority Over-Sampling Technique,” Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

10. R. Mohammad, F. Thabtah, and L. McCluskey, “Predicting Phishing Websites Based on Self-Structuring Neural Network,” Neural Computing and Applications, vol. 25, pp. 443–458, 2014.

11. J. Ma, L. Saul, S. Savage, and G. Voelker, “Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs,” ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1245–1254, 2009.

12. J. Ma, L. Saul, S. Savage, and G. Voelker, “Identifying Suspicious URLs: An Application of Large-Scale Online Learning,” International Conference on Machine Learning (ICML), pp. 681–688, 2009.

13. S. Sahoo, C. Liu, and S. Hoi, “Malicious URL Detection Using Machine Learning: A Survey,” arXiv Computer Science – Cryptography and Security, 2017.

14. A. K. Jain and B. B. Gupta, “Phishing Detection: Analysis of Visual Similarity Based Approaches,” Security and Communication Networks, pp. 1–20, 2017.

15. H. Huang, J. Qian, and Y. Liu, "Detecting Phishing Websites Using Machine Learning Techniques," IEEE International Conference on Communications, pp. 1–6, 2019.

16. A. Blum, B. Wardman, T. Solorio, and G. Warner, "Lexical Feature Based Phishing URL Detection Using Online Learning," ACM Workshop on Artificial Intelligence and Security, pp. 54–60, 2010.

17. M. Al-Janabi and E. H. Houssein, "An Efficient Machine Learning Approach for Web Phishing Detection," Journal of Information Security and Applications, vol. 68, 2022.

18. V. Patil and P. Thakkar, "Machine Learning Based Malicious URL Detection Using Lexical and Host Features," International Conference on Computing Communication and Automation, pp. 89–94, 2020.

19. A. A. Ahmed and N. A. Abdullah, "Real-Time Detection of Phishing Websites Using Ensemble Learning Techniques," IEEE International Conference on Cyber Security and Protection, pp. 20–27, 2021.

20. X. Zhang, Y. Zeng, and X. Jin, "Detection of Malicious URLs Using Ensemble Machine Learning Models," IEEE Access, vol. 10, pp. 115678–115690, 2022.

21. M. Alazab and S. Venkatraman, "Intelligent Malware Detection System Using Machine Learning Algorithms," IEEE Transactions on Industrial Informatics, pp. 654–662, 2020.

22. A. K. Singh and R. Gupta, "Feature Selection Techniques for Effective Phishing Website Detection," International Journal of Information Technology, vol. 14, pp. 321–330, 2022.

23. L. Breiman, "Random Forests," Machine Learning Journal, vol. 45, pp. 5–32, 2001.

24. C. Cortes and V. Vapnik, "Support Vector Networks," Machine Learning, vol. 20, pp. 273–297, 1995.

25. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

26. M. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine Learning Based Phishing Detection from URLs," Expert Systems with Applications, vol. 117, pp. 345–357, 2019.

27. A. Hannousse and S. Yahiouche, "Towards Benchmark Datasets for Machine Learning Based Website Phishing Detection," Engineering Applications of Artificial Intelligence, vol. 104, 2021.

28. S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting Phishing with Streaming Analytics," IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458–471, 2014.

29. K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, and W. K. Tiong, "A New Hybrid Ensemble Feature Selection Framework for Machine Learning-Based Phishing Detection System," Information Sciences, vol. 484, pp. 153–166, 2019.

30. A. Aleroud and L. Zhou, "Phishing Environments, Techniques, and Countermeasures: A Survey," Computers & Security, vol. 68, pp. 160–196, 2017.