

Phishing URL Detection Using Random Forest Machine Learning

Abhishek T P¹, Dr. P. Vijayakarthis²

¹Department of Computer Science and Engineering, RLJ Institute of Technology,
Visvesvaraya Technological University, Karnataka, India
Email: tpabhishek44@gmail.com

²Principal,
Department of Computer Science and Engineering, RLJ Institute of Technology,
Karnataka, India

Abstract

The increasing dependence on internet-based services has significantly improved communication, online banking, e-commerce, and digital information sharing. However, this rapid digital transformation has also led to a substantial rise in phishing attacks, where cybercriminals create deceptive websites to steal sensitive user information such as login credentials, financial data, and personal details. Conventional phishing detection techniques, including blacklist-based and rule-based approaches, often fail to identify newly emerging phishing websites, limiting their effectiveness against evolving cyber threats. To address this challenge, this paper presents a machine learning-based Phishing URL Detection system capable of accurately distinguishing between legitimate and phishing websites using URL-based features.

The proposed system employs the Random Forest classification algorithm due to its robustness, high prediction accuracy, and ability to handle complex feature relationships. A phishing URL dataset containing various lexical and domain-related attributes was preprocessed and used to train the classification model. The trained model was integrated with a Streamlit-based web application that enables users to verify URLs through an interactive graphical interface. Experimental evaluation demonstrated that the proposed system achieved an accuracy of approximately 96.97%, indicating its effectiveness in detecting phishing websites with high reliability. The developed application provides fast predictions, requires minimal computational resources, and offers a user-friendly platform suitable for real-time phishing detection. The results demonstrate that machine learning techniques can significantly improve phishing detection capabilities and contribute toward enhancing cyber-security for internet users.

Keywords: *Phishing URL Detection, Machine Learning, Random Forest, Cyber-security, URL Classification, Streamlit, Web Security.*

1. Introduction

The rapid growth of the Internet has transformed almost every aspect of modern life, including communication, education, healthcare, banking, e-commerce, and government services. Individuals and organizations increasingly rely on online platforms for performing daily activities due to their convenience, accessibility, and efficiency. Along with these advancements, however, cyber threats have become more sophisticated, creating significant challenges for information security. Among various cyber-attacks, phishing remains one of the most prevalent and damaging threats because it exploits human trust rather than technical vulnerabilities. Attackers design fraudulent websites that closely resemble legitimate websites to deceive users into revealing confidential information such as usernames, passwords, credit card details, banking credentials, and other sensitive personal information.

Traditional phishing detection mechanisms primarily rely on blacklists, manually defined rules, or signature-based techniques. Although these methods are effective against previously identified phishing websites, they often fail to detect newly created or modified phishing URLs. As phishing attacks continue to evolve rapidly, conventional

detection techniques struggle to provide timely protection, highlighting the need for intelligent and adaptive detection mechanisms capable of identifying malicious websites based on their underlying characteristics rather than relying solely on known attack patterns.

Machine learning has emerged as a promising solution for addressing these challenges due to its ability to learn patterns from historical data and accurately classify previously unseen instances. By analyzing various URL characteristics, machine learning algorithms can identify subtle differences between legitimate and phishing websites without requiring continuous manual updates. This capability makes machine learning particularly suitable for real-time phishing detection, where rapid and accurate decision-making is essential for preventing cyber-attacks.

In this work, a *Phishing URL Detection* system based on the Random Forest machine learning algorithm is proposed. The system utilizes a phishing URL dataset containing multiple URL-related features that are processed to train a classification model capable of distinguishing phishing websites from legitimate ones. The trained model is integrated with a Streamlit-based web application, allowing users to enter URL feature values through a graphical interface and obtain immediate prediction results. The Random Forest algorithm was selected because of its ability to improve classification accuracy through ensemble learning while reducing over-fitting commonly associated with individual decision trees.

The experimental evaluation of the proposed system demonstrates promising performance, achieving an overall prediction accuracy of approximately 96.97%. The developed application provides an efficient, reliable, and user-friendly solution for *Phishing URL Detection* and illustrates the practical application of machine learning techniques in modern cyber-security. By combining accurate classification with an intuitive web interface, the proposed system contributes toward improving user awareness and reducing the risks associated with phishing attacks.

The major contributions of this paper are summarized as follows:

- Development of a machine learning-based *Phishing URL Detection* system using the Random Forest classifier.
- Implementation of a preprocessing pipeline for preparing URL-based features for classification.
- Integration of the trained model with a Streamlit-based web application for real-time phishing detection.
- Experimental evaluation demonstrating a prediction accuracy of approximately 96.97%.
- Presentation of a practical and user-friendly solution that can assist users in identifying potentially malicious websites before accessing them.

The remainder of this paper is organized as follows. Section II discusses the related work carried out in *Phishing URL Detection*. Section III presents the proposed methodology and system architecture. Section IV describes the experimental setup and implementation details. Section V presents the experimental results and discussion. Finally, Section VI concludes the paper and outlines possible directions for future research.

2. Related Work

Phishing has emerged as one of the most persistent cybersecurity threats due to its ability to deceive users into revealing sensitive information through fraudulent websites and deceptive URLs. Over the past decade, researchers have proposed numerous techniques for phishing detection, ranging from traditional blacklist-based approaches to advanced machine learning and deep learning methods. The increasing sophistication of phishing attacks has motivated the development of intelligent detection systems capable of identifying malicious websites in real time.

Early phishing detection techniques primarily relied on blacklist and white list mechanisms, where known phishing URLs were stored in centralized databases and compared against user requests. Although these approaches are

simple to implement and computationally efficient, they are ineffective against newly generated phishing websites because such URLs are not immediately included in the blacklist databases. Consequently, these methods provide limited protection against zero-day phishing attacks and rapidly evolving malicious domains [1].

To overcome the limitations of blacklist-based systems, researchers introduced machine learning techniques for phishing URL classification. These approaches extract lexical, host-based, and domain-related features from URLs and employ supervised learning algorithms to distinguish between legitimate and phishing websites. Decision Tree, Support Vector Machine (SVM), Naïve Bayes, k-Nearest Neighbors (KNN), and Random Forest have been widely investigated due to their ability to classify URLs based on learned patterns rather than predefined rules [2]. Among these algorithms, Random Forest has demonstrated strong performance because it combines multiple decision trees to improve prediction accuracy and reduce over-fitting.

Recent studies have explored ensemble learning methods to further enhance phishing detection accuracy. Random Forest and Gradient Boosting-based models have shown promising results by combining the predictions of multiple classifiers to improve robustness and reliability. Ensemble models generally outperform individual classifiers because they reduce classification errors and provide better generalization when dealing with large and diverse phishing datasets [3].

With the advancement of artificial intelligence, deep learning techniques such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks have also been applied to phishing detection. These models automatically learn complex feature representations from URLs and webpage content without requiring extensive manual feature engineering. While deep learning approaches often achieve high prediction accuracy, they require larger datasets, increased computational resources, and longer training times compared to conventional machine learning algorithms [4].

Several researchers have integrated phishing detection models into browser extensions and web-based applications to enable real-time identification of malicious websites. These systems allow users to verify URLs before accessing them, thereby reducing the likelihood of credential theft and financial fraud. The integration of machine learning models with lightweight web frameworks has significantly improved the usability and accessibility of phishing detection systems for both technical and non-technical users [5].

Despite the considerable progress achieved in phishing detection research, several challenges remain. Many existing systems depend heavily on static datasets and require frequent retraining to maintain performance as phishing techniques continue to evolve. Additionally, some approaches focus solely on URL analysis while ignoring webpage content, SSL certificate information, domain registration details, and user behavior, which can further improve detection accuracy [6]. These limitations indicate the need for efficient, adaptive, and scalable phishing detection systems capable of providing accurate predictions with minimal computational overhead.

In this paper, a machine learning-based *Phishing URL Detection* system is developed using the Random Forest classifier. The proposed system utilizes URL-based features extracted from a phishing dataset to classify websites as legitimate or phishing. A Streamlit-based web interface is integrated with the trained model to provide a user-friendly platform for real-time phishing detection. The experimental results demonstrate that the proposed approach achieves high prediction accuracy while maintaining simplicity and computational efficiency, making it suitable for practical cyber-security applications.

3. Proposed Methodology

The proposed *Phishing URL Detection* system employs a supervised machine learning approach to classify websites as either legitimate or phishing based on URL-related characteristics. The methodology consists of five major stages: dataset collection, data preprocessing, model training, model evaluation, and real-time prediction through a web-based interface. The overall workflow is designed to provide an efficient and reliable mechanism for identifying malicious websites while maintaining simplicity and ease of deployment.

3.1 Overview of the Proposed System

The proposed system begins by collecting a phishing URL dataset containing multiple URL-based features extracted from legitimate and phishing websites. These features are preprocessed and used to train a Random Forest classifier. Once the training process is completed, the trained model is saved and integrated into a Streamlit-based web application. Users provide the required feature values through the graphical interface, and the trained model predicts whether the given website is legitimate or phishing. The prediction result is displayed instantly, allowing users to identify potentially malicious websites before accessing them.



Figure 1. Overall Workflow of the Proposed Phishing URL Detection System

3.2 Dataset Collection

The performance of any machine learning model depends significantly on the quality of the dataset used during training. In this work, a phishing URL dataset containing both legitimate and phishing website records was used. The dataset consists of multiple URL-based attributes representing various lexical and structural characteristics commonly associated with phishing attacks. Each record is assigned a class label indicating whether the corresponding website is legitimate or phishing.

3.3 Data Preprocessing

Before training the machine learning model, the dataset was preprocessed to improve data quality and ensure compatibility with the learning algorithm. The preprocessing stage involved loading the dataset using the Pandas library, verifying data consistency, handling missing values if present, and separating the input features from the

target class labels.

The dataset was then divided into training and testing subsets. The training dataset was used to train the Random Forest classifier, while the testing dataset was utilized to evaluate the predictive performance of the trained model.

3.4 Random Forest Classification

The Random Forest algorithm was selected as the classification model due to its high accuracy, robustness, and ability to reduce over-fitting. Random Forest is an ensemble learning technique that constructs multiple decision trees using randomly selected subsets of the training data and features. Each decision tree independently predicts the class label of an input sample, and the final prediction is determined through majority voting among all generated trees.

Compared to individual decision trees, Random Forest provides better generalization, improved prediction stability, and higher classification accuracy, making it suitable for *Phishing URL Detection* problems where feature interactions are often complex.

3.5 Model Training

The prepared dataset was used to train the Random Forest classifier using the Scikit-learn machine learning library. During training, the model learned the relationships between URL-based features and their corresponding class labels. Once training was completed, the model parameters were stored using the Joblib library, allowing the trained model to be reused without repeating the training process whenever predictions are required.

3.6 Model Evaluation

The trained model was evaluated using the testing dataset to measure its classification performance. Accuracy was used as the primary evaluation metric for assessing the effectiveness of the proposed system. The experimental results demonstrated that the Random Forest classifier achieved an overall prediction accuracy of approximately 96.97%, indicating its capability to distinguish phishing websites from legitimate websites with high reliability.

3.7 Real-Time Prediction Using Streamlit

To improve usability, the trained machine learning model was integrated into a Streamlit-based web application. The graphical interface allows users to enter the required URL feature values through simple input fields. After the user clicks the "Check Website" button, the application processes the entered values, loads the trained Random Forest model, and predicts whether the website is legitimate or phishing.

The prediction result is displayed immediately on the interface, providing users with a fast and convenient phishing detection mechanism. This integration demonstrates the practical applicability of machine learning techniques in real-world cyber-security systems.

3.8 Methodology Summary

The proposed methodology combines data preprocessing, supervised machine learning, and web application development to create an efficient *Phishing URL Detection*. The use of the Random Forest classifier enables accurate classification of phishing websites, while the Streamlit interface provides an interactive platform for real-time prediction. The methodology is computationally efficient, easy to deploy, and capable of assisting users in identifying potentially malicious websites before they are accessed.

4. Experimental Setup

The experimental setup was designed to evaluate the performance of the proposed machine learning-based Phishing URL Detection system under a controlled software environment. The implementation was carried out using Python programming language with various open-source libraries for data preprocessing, model development, and web application deployment. The Random Forest classifier was selected as the prediction model due to its robustness, high classification accuracy, and ability to handle complex feature relationships.

4.1 Hardware Configuration

The proposed system was developed and tested on a personal computer with the hardware configuration shown in Table 1.

Table 1. Hardware Configuration

Component	Specification
Processor	Intel Core i5 (or equivalent)
RAM	8 GB
Storage	512 GB SSD
Operating System	Windows 11
Architecture	64-bit

4.2 Software Configuration

The software environment used for implementing the proposed *Phishing URL Detection* system is presented in Table 2.

Table 2. Software Configuration

Software	Purpose
Python	Programming Language
Visual Studio Code	Integrated Development Environment
Pandas	Data loading and preprocessing
NumPy	Numerical computations
Scikit-learn	Machine learning model development
Streamlit	Web application development
Joblib	Model serialization and loading

4.3 Dataset Description

The phishing URL dataset used in this study contains multiple URL-based features extracted from legitimate and phishing websites. These features represent various lexical and structural characteristics of URLs that assist the machine learning model in distinguishing malicious websites from genuine ones. The dataset was preprocessed before training to ensure data consistency and improve model performance.

The input features were separated from the target class labels, and the dataset was divided into training and testing subsets for model development and evaluation. The training dataset was used to build the Random Forest classifier, while the testing dataset was used to evaluate its predictive performance.

4.4 Model Training Procedure

The Random Forest classifier was implemented using the Scikit-learn machine learning library. During training, the algorithm generated multiple decision trees from randomly selected subsets of the dataset. Each decision tree independently classified the input samples, and the final prediction was obtained using majority voting among all trees.

After successful training, the trained model was stored using the Joblib library, enabling the model to be loaded directly into the Streamlit application without requiring retraining during every execution.

4.5 Performance Evaluation

The performance of the proposed *Phishing URL Detection* system was evaluated using the testing dataset. Model accuracy was selected as the primary evaluation metric because it represents the percentage of correctly classified phishing and legitimate URLs.

The experimental results indicated that the proposed Random Forest classifier achieved an overall classification accuracy of 96.97%, demonstrating its effectiveness in identifying phishing websites with high reliability. The trained model generated prediction results within a short execution time, making it suitable for real-time phishing detection applications.

4.6 Experimental Workflow

The complete experimental procedure followed in this research is summarized below:

1. Collection of the phishing URL dataset.
2. Data preprocessing and feature preparation.
3. Division of the dataset into training and testing sets.
4. Training the Random Forest classifier.
5. Evaluating the trained model using the testing dataset.
6. Saving the trained model using Joblib.
7. Integrating the trained model into the Streamlit web application.
8. Performing real-time phishing URL prediction through the user interface.

The experimental setup confirms that the proposed system can efficiently classify URLs with high prediction accuracy while maintaining low computational complexity and providing an interactive user interface for practical deployment.

Table 3. Experimental Parameters

Parameter	Value
Classification Algorithm	Random Forest
Programming Language	Python
IDE	Visual Studio Code
Machine Learning Library	Scikit-learn
User Interface	Streamlit
Model Storage	Joblib
Prediction Accuracy	96.97%

5. Results and Discussion

The proposed *Phishing URL Detection* system was successfully implemented using the Python programming language and evaluated using the Random Forest machine learning algorithm. The model was trained on a phishing URL dataset containing URL-based features extracted from both legitimate and phishing websites. After the completion of the training process, the trained model was integrated with a Streamlit-based web application to facilitate real-time phishing URL prediction through an interactive graphical interface.

The effectiveness of the proposed system was evaluated using the testing dataset. The Random Forest classifier demonstrated excellent classification capability by accurately distinguishing phishing websites from legitimate websites. The experimental evaluation showed that the developed model achieved an overall prediction accuracy of 96.97%, indicating that the selected machine learning approach is highly effective for *Phishing URL Detection*.

5.1 Model Training Results

The Random Forest classifier was trained using the preprocessed phishing URL dataset. During the training phase, the algorithm constructed multiple decision trees using randomly selected subsets of the training data. The ensemble learning mechanism enabled the classifier to improve prediction stability and reduce over-fitting, thereby increasing the overall classification performance.

The successful completion of model training confirmed that the selected feature set provided sufficient information for distinguishing phishing URLs from legitimate websites. The trained model was subsequently saved using the Joblib library, allowing it to be loaded directly by the Streamlit application without requiring retraining.

Table 4. Model Performance

Performance Metric	Value
Classification Algorithm	Random Forest
Training Status	Successfully Completed
Model Accuracy	96.97%
Model Storage	Joblib (.pkl)
Application Interface	Streamlit

5.2 Web Application Results

The trained model was integrated into a Streamlit-based graphical user interface to provide real-time phishing detection. The application allows users to enter URL feature values through an intuitive interface. Once the required information is provided, the system processes the input and predicts whether the corresponding website is legitimate or phishing.

The graphical interface simplifies the phishing detection process by eliminating the need for technical knowledge. Users receive immediate prediction results, making the application suitable for educational demonstrations and practical cyber-security awareness.

5.3 Prediction Analysis

The developed system successfully classified phishing URLs based on the learned feature patterns. During testing, the application correctly identified phishing websites and displayed an appropriate warning message to the user. The prediction process required only a short execution time, demonstrating the computational efficiency of the Random Forest classifier.


```
PROBLEMS OUTPUT DESKTOP CONSOLE TERMINAL PORTS
(.venv) PS C:\Users\tpabb\OneDrive\Desktop\Phishing_URL_Detection> python phishing_detector.py
accuracy          0.97          0.97          0.97          2211
macro avg         0.97          0.97          0.97          2211
weighted avg      0.97          0.97          0.97          2211

Model Saved Successfully
(.venv) PS C:\Users\tpabb\OneDrive\Desktop\Phishing_URL_Detection> streamlit run app.py
2026-07-18 20:16:39.362 Unicorn server started on 0.0.0.0:8501

You can now view your Streamlit app in your browser.

local URL: http://localhost:8501
network URL: http://10.220.242.1:8501
```

Figure 5.2: Random Forest Model Training Output Showing 96.97% Accuracy

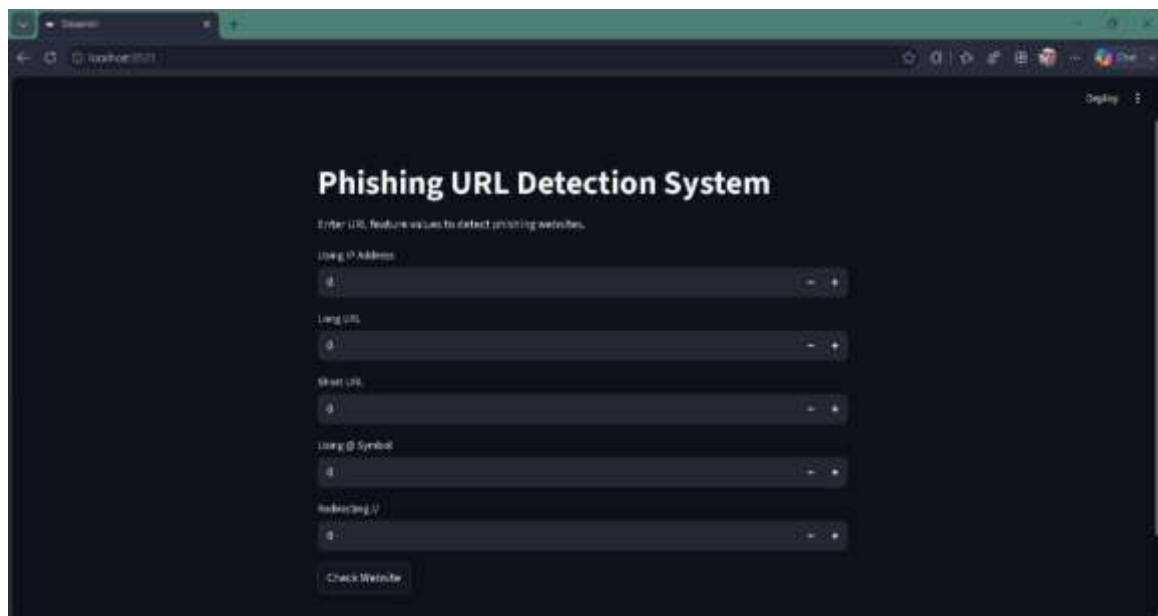


Figure 5.3: Home Page of the Streamlit-Based Phishing URL Detection System



Figure 5.4: Phishing Website Detection Result Displayed by the Application

6. Conclusion

Phishing attacks continue to pose a significant threat to individuals and organizations by exploiting user trust to obtain sensitive information through fraudulent websites. As traditional blacklist-based detection methods often struggle to identify newly emerging phishing URLs, there is an increasing need for intelligent and adaptive detection techniques. This study presented a machine learning-based *Phishing URL Detection* system that utilizes the Random Forest classification algorithm to accurately distinguish between legitimate and phishing websites using URL-based features.

The proposed system was successfully implemented using Python and integrated with a Streamlit-based web application to provide a simple and interactive platform for real-time phishing URL prediction. Experimental evaluation demonstrated that the Random Forest classifier achieved an overall prediction accuracy of 96.97%, indicating its effectiveness in identifying phishing websites with high reliability. The integration of machine learning with a user-friendly interface enables users to verify suspicious URLs quickly and efficiently without requiring extensive technical knowledge.

The results obtained in this research confirm that URL-based feature analysis combined with ensemble learning techniques can provide an effective solution for phishing detection. Compared with conventional rule-based approaches, the proposed system offers improved adaptability and the ability to classify previously unseen phishing URLs based on learned patterns. Furthermore, the developed application demonstrates the practical applicability of machine learning in enhancing cyber-security and improving user awareness against phishing attacks.

Although the proposed system achieved promising performance, certain limitations remain. The current implementation focuses primarily on URL-based features and does not incorporate webpage content analysis, domain registration information, SSL certificate verification, or visual similarity analysis. Including these

additional features may further improve detection accuracy and strengthen the system against more sophisticated phishing techniques.

Future research can focus on evaluating additional machine learning and deep learning algorithms, such as Gradient Boosting, XGBoost, LightGBM, and Transformer-based models, to further enhance classification performance. The system can also be extended by integrating real-time threat intelligence services, browser extensions, and cloud-based deployment to support large-scale phishing detection in practical environments.

In conclusion, the proposed *Phishing URL Detection Using Machine Learning* system successfully achieved its objectives by providing a reliable, accurate, and user-friendly phishing detection solution. The achieved accuracy of 96.97% demonstrates the effectiveness of the Random Forest algorithm for phishing URL classification. The developed system contributes to improving online security by assisting users in identifying potentially malicious websites before they are accessed, thereby reducing the risks associated with phishing attacks and supporting safer internet usage.

References

- [1] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, vol. 25, no. 2, pp. 443–458, Aug. 2014.
- [2] F. Toolan and J. Carthy, "Feature Selection for Spam and Phishing Detection," *Proceedings of the eCrime Researchers Summit (eCrime)*, Dallas, TX, USA, 2010, pp. 1–12.
- [3] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent Phishing Website Detection System Using Fuzzy Techniques," *Proceedings of the International Conference on Information Networking and Automation*, Kuala Lumpur, Malaysia, 2010, pp. 345–350.
- [4] C. Xiang, P. C. Yong, and L. S. Meng, "Detecting Phishing Websites Using Machine Learning Algorithms," *International Journal of Computer Science and Information Security*, vol. 16, no. 4, pp. 200–206, 2018.
- [5] F. Thabtah and D. Peebles, "A New Machine Learning Model Based on Induction of Rules for Phishing Websites Classification," *Applied Soft Computing*, vol. 68, pp. 329–342, Jul. 2018.
- [6] S. Marchal, J. François, 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, Dec. 2014.
- [7] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [8] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [9] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [10] W. McKinney, *Python for Data Analysis*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
- [11] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.
- [12] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024–8035, 2019.