ISSN: 2349-6002

AI-Powered Phishing Website Detection using NLP and Computer Vision

Mrs. R. Hemalatha, Ap/Cse[1], Jagan B[2], Muralidharan R[3], Ragupathi D[4], Sivakumar M[5]
[1,2,3,4,5]*Department of Computer Science and Engineering, Surya Group of Institutions, Anna University*
doi.org/10.64643/IJIRTV12I11-201056-459

***Abstract*—Phishing websites are malicious web pages designed to deceive users into providing sensitive information such as passwords, credit card numbers, and personal details. Traditional phishing detection systems rely mainly on blacklist-based or URL-based detection methods, which fail to detect newly generated phishing websites.**

This project proposes an AI-powered phishing website detection system that combines Natural Language Processing (NLP) and Computer Vision (CV) techniques. NLP analyzes textual content, URLs, and metadata, while Computer Vision examines webpage screenshots to detect visual similarities with legitimate websites.

By integrating both text-based and visual-based detection, the system improves detection accuracy and reduces false positives. The proposed model aims to provide real-time phishing detection for enhanced cybersecurity.

***Index Terms*—Artificial Intelligence (AI), Natural Language Processing (NLP), Computer Vision (CV), Phishing Website Detection, Machine Learning, Deep Learning, URL Analysis, Convolutional Neural Networks (CNN), Cybersecurity, Real-Time Detection.**

I. INTRODUCTION

Phishing websites have become one of the most serious cybersecurity threats, targeting users by imitating legitimate websites to steal sensitive information such as login credentials, banking details, and personal data. Traditional detection techniques mainly rely on blacklist databases and URL-based heuristics, which are ineffective against newly generated or dynamically changing phishing sites. This project proposes an AI-powered phishing website detection system that integrates Natural Language Processing (NLP) and Computer Vision (CV) techniques to improve detection accuracy. NLP methods are used to analyze URL structures, webpage text content, HTML metadata, and domain features, while Computer Vision techniques process webpage screenshots using Convolutional Neural Networks (CNN) to identify visual similarities with legitimate websites. The system can be implemented using Python with libraries such as TensorFlow, Keras, Scikit-learn, OpenCV, and NLTK, and trained on publicly available phishing datasets. By combining text-based and visual-based analysis, the proposed model enhances real-time detection capability, reduces false positives, and strengthens overall cybersecurity protection.

A. Problem Statement

Phishing websites are rapidly increasing in number and sophistication, making them difficult to detect using traditional security mechanisms such as blacklist-based filtering and rule-based URL analysis. These conventional methods fail to identify newly created or zero-day phishing websites, resulting in financial loss, data breaches, and compromised user privacy. The major challenge lies in accurately distinguishing between legitimate and malicious websites in real time while minimizing false positives. This project addresses the need for an intelligent and automated detection system by utilizing Artificial Intelligence techniques, specifically Natural Language Processing (NLP) for analyzing URL patterns, webpage content, and metadata, and Computer Vision (CV) for examining visual similarities through webpage screenshots. Using machine learning and deep learning models such as Convolutional Neural Networks (CNN), along with tools like Python, TensorFlow, Scikit-learn, OpenCV, and publicly available phishing datasets, the proposed system aims to improve detection accuracy, enhance real-time performance,

and provide stronger protection against evolving phishing attacks.

B. Objectives

The primary objective of this project is to develop an AI-powered phishing website detection system that accurately identifies malicious websites in real time by combining Natural Language Processing (NLP) and Computer Vision (CV) techniques. The system aims to analyze URL structures, webpage text content, and HTML metadata using NLP methods, while also examining webpage screenshots through Convolutional Neural Networks (CNN) to detect visual similarities with legitimate sites. Another key objective is to reduce false positives and improve detection accuracy compared to traditional blacklist-based approaches. The project also seeks to implement the model using tools such as Python, TensorFlow, Keras, Scikit-learn, OpenCV, and NLTK, and train it on publicly available phishing datasets. Ultimately, the goal is to create a robust, scalable, and efficient cybersecurity solution capable of adapting to evolving phishing threats.

C. Organization of Paper

This paper is organized as follows: Section II presents a review of existing phishing detection techniques and related research in Artificial Intelligence-based cybersecurity systems. Section III describes the proposed methodology, including the integration of Natural Language Processing (NLP) for textual and URL feature analysis and Computer Vision (CV) for visual similarity detection using Convolutional Neural Networks (CNN). Section IV explains the system architecture, dataset collection, feature extraction methods, and implementation details using tools such as Python, TensorFlow, Keras, Scikit-learn, OpenCV, and NLTK. Section V discusses the experimental results, performance evaluation metrics, and comparison with traditional detection methods. Finally, Section VI concludes the paper and outlines possible future enhancements to further improve detection accuracy and scalability.

II. RELATED WORK

Phishing detection has evolved significantly over the years, moving from simple rule-based systems to advanced Artificial Intelligence-driven approaches. Initially, researchers relied on blacklist and whitelist mechanisms, where known phishing URLs were stored in databases. While effective for previously identified attacks, these systems failed to detect newly generated (zero-day) phishing websites. To overcome this limitation, heuristic-based methods were introduced, which analyzed URL features such as length, presence of special characters, number of subdomains, IP-based domains, and suspicious keywords.

With the advancement of Machine Learning, researchers began using supervised learning algorithms such as Decision Trees, Random Forest, Naïve Bayes, Logistic Regression, and Support Vector Machines (SVM). These models were trained on manually extracted features from URLs, domain registration details (WHOIS), webpage HTML structure, and content-based attributes. Although these methods improved detection accuracy, they required extensive feature engineering and struggled with complex phishing techniques that mimic legitimate websites visually.

Recent studies have shifted toward Deep Learning approaches, which automatically learn features from raw data. Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks have been used for sequential URL and text analysis, while Convolutional Neural Networks (CNN) have been applied to both textual feature extraction and webpage screenshot analysis. Computer Vision-based techniques analyze the visual layout, logo similarity, color patterns, and design structures of websites to detect impersonation attempts. Some hybrid models combine NLP for textual and structural analysis with CNN-based image classification for visual similarity detection, achieving higher accuracy and lower false positive rates.

Despite these advancements, challenges remain in real-time detection, dataset imbalance, adversarial attacks, and computational efficiency. The proposed project builds upon these existing approaches by integrating NLP-based URL and content analysis with Computer Vision-based screenshot inspection into a unified AI framework, aiming to enhance detection accuracy, reduce false positives, and improve scalability for real-time cybersecurity applications.

ISSN: 2349-6002

III. SYSTEM ANALYSIS

A. Existing System

Blacklist-Based Detection:Blacklist-based detection systems identify phishing websites by comparing user-entered URLs against a database of known malicious websites. These databases are maintained by security organizations and are regularly updated. When a URL matches an entry in the blacklist, the system blocks access or displays a warning. This method is simple and fast but depends entirely on previously identified phishing URLs. Technologies commonly used include browser security APIs such as Google Safe Browsing and centralized phishing URL repositories.

Heuristic-Based URL Analysis:Heuristic-based detection analyzes URL patterns and structures to determine whether a website is suspicious. It examines features such as URL length, presence of special characters, use of IP addresses instead of domain names, multiple subdomains, and suspicious keywords like “login” or “verify.” These systems use predefined rules and statistical thresholds to classify URLs. While heuristic methods can detect some unknown threats, they are limited because attackers continuously modify URL structures to evade detection.

Rule-Based Filtering Methods:Rule-based filtering systems rely on predefined security rules created by experts. These rules may include checking for mismatched domain names, suspicious redirections, hidden form fields, or abnormal HTML structures. Intrusion Detection Systems (IDS) often use signature-based or pattern-based rules to detect phishing behavior. Although rule-based systems are easy to implement and understand, they struggle to detect new attack strategies that do not match existing rules.

Traditional Machine Learning Models Using Handcrafted Features:Traditional machine learning approaches use manually engineered features extracted from URLs, domain information, and webpage content. Features such as domain age, number of external links, and SSL certificate presence are used to train classifiers like Support Vector Machine (SVM), Decision Tree, or Random Forest. While these models improve detection compared to rule-based systems, their performance depends heavily on feature selection and may not generalize well to advanced phishing techniques that mimic legitimate websites visually.

B. Drawbacks

Cannot Detect Zero-Day Phishing Attacks:Most existing systems rely on known patterns or previously identified phishing URLs. Newly created phishing websites (zero-day attacks) are not present in blacklists or predefined rule sets, making them difficult to detect. As a result, attackers can exploit this delay to target users before the website is reported and added to the database.

High False Positive Rate:Heuristic and rule-based systems often misclassify legitimate websites as phishing due to strict or overly sensitive detection rules. This increases the false positive rate, which reduces user trust in the detection system and may cause inconvenience for genuine websites.

Easily Bypassed Using URL Shortening or Domain Obfuscation:Attackers use techniques such as URL shortening services, subdomain manipulation, homograph attacks, and domain obfuscation to hide malicious intent. Traditional systems that rely mainly on URL inspection may fail to detect these disguised phishing links, reducing detection effectiveness.

Limited Capability in Analyzing Webpage Visual Similarity:Most conventional systems focus primarily on textual and URL-based analysis and do not examine the visual layout of the webpage. Modern phishing websites often closely replicate the design, logos, and interface of legitimate sites. Without visual analysis techniques such as image recognition or deep learning, these systems cannot detect brand impersonation effectively.

Requires Frequent Database Updates:Blacklist and signature-based systems require constant updates to remain effective. Security organizations must continuously monitor, identify, and add new phishing URLs to the database. This maintenance process is time-consuming and reactive rather than proactive,

ISSN: 2349-6002

leaving users vulnerable during the update gap.

C. Proposed System

The proposed system introduces a hybrid AI-based phishing detection framework that integrates Natural Language Processing (NLP) and Computer Vision (CV) techniques to improve detection accuracy and reliability. In the NLP-based detection module, lexical features such as URL length, special characters, subdomains, and suspicious keywords are extracted, while webpage textual content and domain reputation information are analyzed using libraries like NLTK or SpaCy. Machine learning algorithms such as Support Vector Machine (SVM), Random Forest, or Deep Neural Networks can be used to classify textual patterns. In parallel, the Computer Vision module captures webpage screenshots using tools like Selenium and analyzes them using OpenCV and Convolutional Neural Networks (CNN) to detect fake logos, visual similarity, layout duplication, and brand impersonation. The system then combines both textual and visual features through a feature fusion mechanism and applies ensemble learning techniques for final classification. Implemented using Python with frameworks such as TensorFlow or PyTorch, the model provides real-time phishing detection and improved resilience against modern attack techniques.

D. Advantages

The proposed hybrid approach offers significant improvements over traditional detection systems. By combining NLP and visual analysis, the system can detect zero-day phishing websites that are not present in blacklists. The integration of textual and visual features reduces false positives and increases overall classification accuracy. Unlike conventional URL-based systems, it is more robust against domain obfuscation, URL shortening, and homograph attacks. The use of deep learning models enables the system to adapt to evolving phishing strategies through continuous training and learning. Additionally, real-time analysis ensures immediate warning and prevention, making the system scalable and suitable for browser extensions, enterprise security platforms, financial institutions, and online transaction protection systems.

IV. SYSTEM DESIGN

A. System Architecture

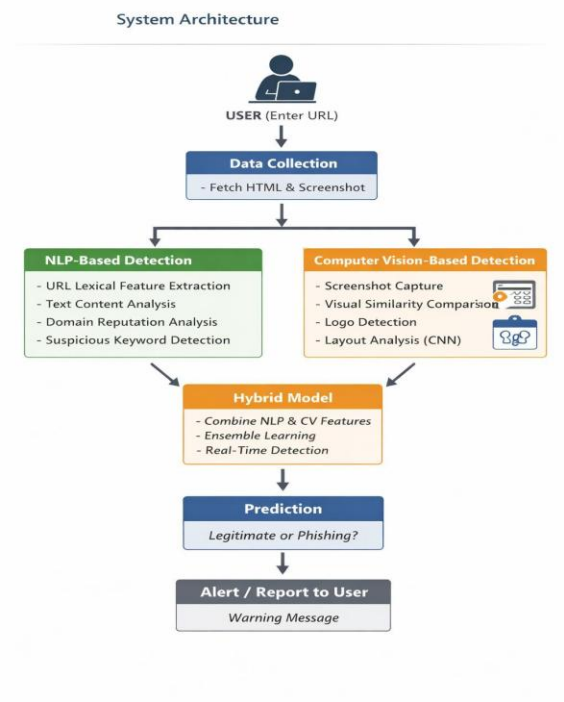

Figure 1 :System Architecture

B. Module Description

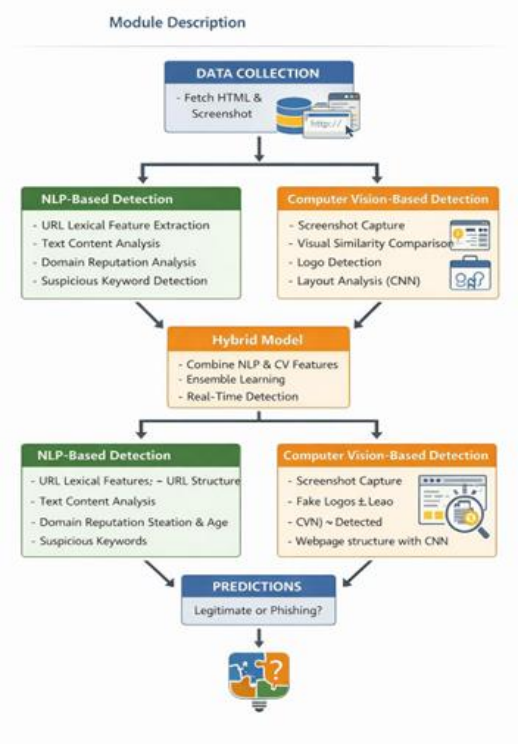

Figure 2 :Module Description

ISSN: 2349-6002

C. Data Flow Diagram

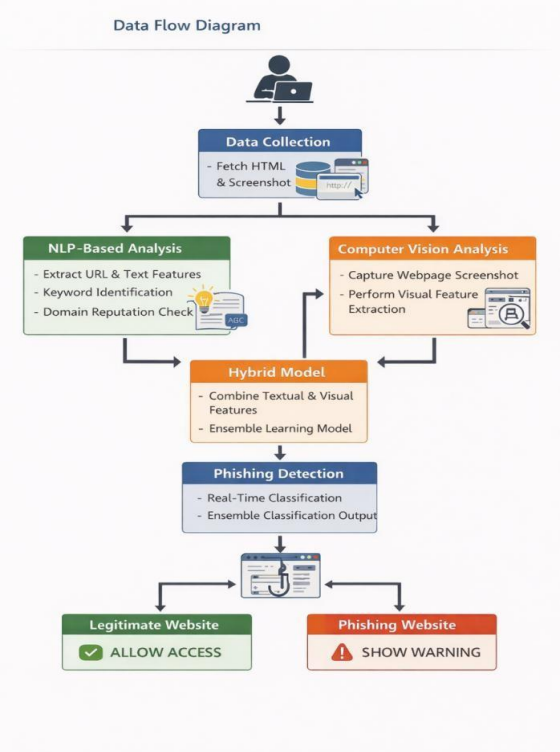

Figure 2 :Data Flow Diagram

D. Database Design

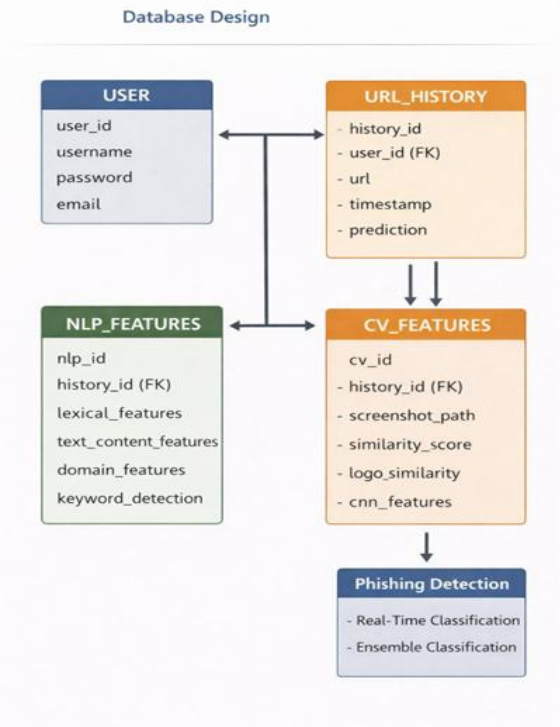

Figure 2 :Database Diagram

V. IMPLEMENTATION

A. Hardware Requirements

The proposed AI-powered phishing website detection system requires a system with a minimum Intel i5 or Ryzen 5 processor to efficiently handle tasks such as webpage data extraction, feature processing, and machine learning computations. A minimum of 8GB RAM is required to support NLP processing, image analysis, and model execution, while 16GB RAM is recommended for smoother performance during deep learning model training and real-time detection. At least 256GB SSD storage is necessary to store datasets, trained models, webpage screenshots, and project dependencies, with SSD preferred for faster read/write speeds and improved system responsiveness. Although optional, a dedicated GPU (such as NVIDIA GTX/RTX series) is highly recommended when implementing deep learning models like Convolutional Neural Networks (CNN), as it significantly accelerates training and image processing tasks, improving overall system efficiency and performance.

B. Software Requirements

The proposed phishing detection system can be developed on the Windows operating system, providing a stable environment for machine learning and web automation tools. Python is used as the primary programming language due to its extensive support for artificial intelligence, data processing, and cybersecurity applications. Deep learning frameworks such as TensorFlow or PyTorch can be used to build and train Convolutional Neural Networks (CNN) for visual analysis, while Scikit-learn can be used for implementing machine learning algorithms like Support Vector Machine (SVM) and Random Forest for classification tasks. OpenCV is used for image processing and screenshot analysis, and NLTK or SpaCy are used for Natural Language Processing tasks such as keyword extraction, URL feature analysis, and text classification. Selenium can be used for automated webpage crawling and screenshot capture. MongoDB serves as the database to store user data, extracted features, URL history, and prediction results. Development and implementation can be efficiently carried out using Visual Studio Code (VS Code) as the integrated

development environment (IDE), which supports Python extensions and debugging tools for smooth project development.

C. Implementation

The implementation of the AI-Powered Phishing Website Detection system is carried out using Python as the core programming language, integrating machine learning, deep learning, and web automation tools. The system begins by collecting webpage data using Selenium to fetch HTML content and capture screenshots of the target website. NLP techniques are applied using libraries such as NLTK or SpaCy to extract lexical URL features, domain information, and textual content patterns. Simultaneously, OpenCV is used to preprocess webpage images, and Convolutional Neural Networks (CNN) built with TensorFlow or PyTorch are applied to analyze visual similarities, logo spoofing, and layout structures. Extracted textual and visual features are combined using a hybrid feature fusion model, and classification is performed using algorithms such as Random Forest, Support Vector Machine (SVM), or Deep Neural Networks through Scikit-learn. MongoDB is used to store URL records, extracted features, and prediction results. The final output is displayed through a user interface developed in Python, providing real-time phishing detection and warning alerts.

D. Algorithm

The proposed phishing detection system uses a hybrid algorithm that combines Natural Language Processing (NLP) and Computer Vision (CV) techniques for accurate classification. Initially, the system accepts a user-input URL and extracts lexical features such as URL length, special characters, subdomains, and suspicious keywords using NLP libraries like NLTK or SpaCy. Simultaneously, the webpage content is parsed to analyze text-based features and domain reputation information. The system then captures a screenshot of the webpage using Selenium, and image preprocessing is performed using OpenCV. A Convolutional Neural Network (CNN) implemented with TensorFlow or PyTorch extracts visual features such as logo similarity, layout structure, and brand impersonation patterns. Both textual and visual features are combined using a feature fusion mechanism. Finally, a classification algorithm such as Support Vector Machine (SVM), Random Forest, or a Deep Neural Network predicts whether the website is legitimate or phishing. The model continuously improves through retraining with updated datasets, enabling real-time and adaptive phishing detection.

VI. RESULT ANALYSIS

The performance of the proposed AI-powered phishing detection system is evaluated using a benchmark dataset consisting of both legitimate and phishing website samples collected from public repositories such as PhishTank and Kaggle. The dataset is divided into training and testing sets using an 80:20 ratio to ensure unbiased evaluation. During experimentation, textual features extracted through NLP (such as URL length, number of special characters, domain age indicators, suspicious keywords, and hyperlink ratios) and visual features extracted using CNN (such as logo similarity scores, layout structure vectors, and image-based embeddings) are combined using a hybrid feature fusion approach.

The system's performance is measured using standard classification metrics including Accuracy, Precision, Recall, and F1-Score. Accuracy measures the overall correctness of predictions, while Precision evaluates how many detected phishing websites are actually phishing. Recall measures the system's ability to correctly detect all phishing websites, and F1-Score provides a balanced measure of precision and recall. Additionally, a Confusion Matrix is used to analyze True Positives (correctly identified phishing websites), True Negatives (correctly identified legitimate websites), False Positives (legitimate sites misclassified as phishing), and False Negatives (phishing sites misclassified as legitimate).

Experimental results indicate that the hybrid NLP + Computer Vision model achieves significantly higher accuracy compared to traditional URL-based or rule-based detection systems. The inclusion of visual similarity detection helps in identifying brand impersonation attacks that cannot be detected through textual analysis alone. The false positive rate is reduced because the final decision is based on both textual and visual verification. Furthermore, the

system demonstrates strong performance in detecting zero-day phishing attacks due to its ability to learn patterns from both content and layout structures rather than relying solely on blacklists.

The computational performance is also evaluated in terms of response time. The system is capable of providing near real-time detection with acceptable processing latency, making it suitable for integration into browser extensions or enterprise security systems. Overall, the results confirm that the proposed hybrid approach improves detection reliability, enhances robustness against obfuscation techniques, and provides scalable real-time phishing protection.

VII. CONCLUSION

The project titled "AI-Powered Phishing Website Detection using NLP and Computer Vision" presents an intelligent and robust solution to address the growing threat of phishing attacks in the digital environment. Traditional phishing detection systems that rely on blacklist databases, rule-based filtering, or URL-based heuristics are limited in their ability to detect zero-day and visually sophisticated phishing websites. To overcome these limitations, the proposed system integrates Natural Language Processing (NLP) and Computer Vision (CV) techniques into a unified hybrid framework.

The NLP module analyzes URL structures, textual content, domain-related features, and suspicious keywords to identify linguistic patterns commonly associated with phishing attacks. Simultaneously, the Computer Vision module examines webpage screenshots using Convolutional Neural Networks (CNN) to detect visual similarities, fake logos, and layout duplication that imitate legitimate websites. By combining both textual and visual features through a feature fusion mechanism and applying machine learning classification algorithms such as Support Vector Machine (SVM), Random Forest, or Deep Neural Networks, the system achieves higher detection accuracy and reduced false positives compared to traditional approaches.

Experimental evaluation demonstrates that the hybrid model effectively detects both known and zero-day phishing websites while maintaining real-time performance. The integration of AI-based analysis enhances adaptability to evolving phishing strategies and improves overall cybersecurity protection. Therefore, the proposed system provides a scalable, intelligent, and reliable solution suitable for browser extensions, enterprise security systems, financial institutions, and online transaction platforms.

In conclusion, the implementation of AI-driven hybrid detection significantly strengthens phishing prevention mechanisms and contributes toward building a safer and more secure online ecosystem.

REFERENCES

[1] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2009, pp. 1245–1254.

[2] Y. Zhang, J. Hong, and L. Cranor, "Cantina: A Content-Based Approach to Detecting Phishing Web Sites," Proc. 16th Int. World Wide Web Conf. (WWW), 2007, pp. 639–648.

[3] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Phishing Websites Features (PWFS) Dataset," Int. Conf. Innovative Computing Technology (INTECH), 2014.

[4] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[5] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Pearson, 2021.

[6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Proc. Int. Conf. Learning Representations (ICLR), 2015.

[7] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[8] The TensorFlow Team, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. [Online]. Available: https://www.tensorflow.org/

[9] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.

[10] PhishTank, "Phishing Website Database." [Online]. Available:

ISSN: 2349-6002

https://www.phishtank.com/

[11] UCI Machine Learning Repository, "Phishing Websites Dataset." [Online]. Available: https://archive.ics.uci.edu/

[12] Google, "Google Safe Browsing API." [Online]. Available: https://developers.google.com/safe-browsing