Phishing Attack Detection using AI/ML

A PROJECT REPORT

Submitted by,

MANJUNATH A C	**20211CAI0124**
DINESH KUMAR K	**20211CAI0074**
AKASH S	**20211CAI0110**
PRAJWAL KANTHAN T	**20211CAI0075**

Under the guidance of,

Ms. KAYALVIZHI

——— Assistant Professor

School of Computer Science and Engineering

Presidency University, Bengaluru

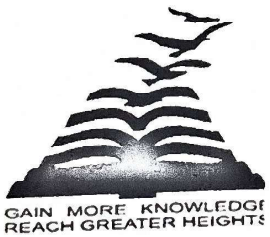
in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At



PRESIDENCY UNIVERSITY, BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **"Phishing Attack Detection using AI/ML"** being submitted by **"MANJUNATH A C (20211CAI0124) ", "DINESH KUMAR K (20211CAI0074)", "AKASH S (20211CAI0110)", "PRAJWAL KANTHAN T(20211CAI0075)",** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in **Computer Science and Engineering** is a Bonafide work carried out under my supervision.

Ms. KAYALVIZHI
Assistant Professor
School of Computer Science and Engineering
Presidency University, Bengaluru

Dr. ZAFAR ALI KHAN
Prof & Hod
Dept. of Computer Science Engineering
Presidency University, Bengaluru

Dr. MYDHILI NAIR
Professor & Associate Dean
School of CSE
Presidency University, Bengaluru

Dr. SAMEERUDDIN KHAN
Pro-Vice Chancellor - Engineering
Dean – School of CSE & IS
Presidency University, Bengaluru

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I hereby declare that the work, which is being presented in the report entitled **"Phishing Attack Detection using AI/ML"** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of my own investigations carried under the guidance of **Ms. Kayalvizhi, Assistant Professor, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

I have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME	ROLL NUMBER	SIGNATURE
MANJUNATH A C	20211CAI0124	Manjunath A.C
DINESH KUMAR K	20211CAI0074	K. Dinesh Kumar
AKASH S	20211CAI0110	Akash
PRAJWAL KANTHAN T	20211CAI0075	

ABSTRACT

Phishing has become a major cybersecurity challenge due to the fact that attackers continue to innovate the methods of luring individuals into revealing sensitive information like passwords, financial details, and login credentials. The sophistication of deception in phishing attacks in today's world renders conventional rule-based security mechanisms inadequate. Hence, our contribution extends further to investigate an even better solution—utilize artificial intelligence (AI) and machine learning (ML) to effectively detect phishing attacks based on website URL analysis.

The model used in this research utilizes empirical data gathered from two reliable sources, the UCI Machine Learning Repository and Kaggle's phishing data set. After going through data cleansing and integration activities, a balanced data set containing over 21,000 records was developed. We derived useful features from the combined data that assist in the identification of phishing URLs, including URL length, number of special characters, presence of suspicious keywords, subdomain patterns, and the domain structure.

Some algorithms were employed in comparing the performance of the top-performing model, wherein the Random Forest classifier excelled against such counterparts as Logistic Regression and Decision Trees. At a performance rate of more than 99%, the model has an exceptionally high capacity to distinguish between phishing and safe URLs with extremely high accuracy. In order to cater to an enhanced user experience, an interactive dashboard was built based on Streamlit. The web interface makes it possible for a user to enter a URL, which is parsed by the system and tagged as secure or suspicious. To maintain your model current with evolving phishing trends, a retrain feature has been included. This allows the system to learn and improve its detection accuracy in sync with evolving attack tactics employed by attackers. The tool has been made lightweight and user-friendly to allow its deployment on local systems and cloud services such as AWS and Heroku. This project shows the efficacy of using properly designed features coupled with high-speed machine learning algorithms in providing efficient and scalable protection against phishing attacks. It not only safeguards a single user against internet frauds but also displays the real-time usage of artificial intelligence in securing cybersecurity measures.

ACKNOWLEDGEMENTS

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** Presidency School of Computer Science and Engineering, Presidency University, and Dr. "**Dr. Zafar Ali Khan N**", Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide Dr.**Ms. Kayalvizhi Assistant Professor** Presidency School of Computer Science and Engineering, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

We would like to convey our gratitude and heartfelt thanks to the PIP4001 University Project Coordinator **Mr. Md Ziaur Rahman and Dr. Sampath A K,** department Project Coordinators Dr. Afroz Pasha and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

MANJUNATH A C

DINESH KUMAR K

AKASH S

PRAJWAL KANTHAN T

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

In the digital era of the internet, phishing attacks have proved to be among the most widespread and dangerous forms of cyber attacks. Phishing attacks are specifically designed to mislead users into sharing sensitive details such as passwords, banking credentials, or individual identity data under the guise of legitimate entities. In the shape of fake emails, fake websites, or false login pages, phishing has inflicted heavy financial damage and privacy intrusions on people, companies, and even states.

Most traditional phishing detection tools are based on rules or blacklists that have been manually written and can become outdated very fast. As attackers become more and more creative and clever, it is clear that new methods are needed to stay ahead of the game. That's where Machine Learning (ML) and Artificial Intelligence (AI) come into play. These technologies have the ability to learn automatically based on patterns in data and develop to counteract new phishing methods, which makes them ideal for developing smarter, more effective phishing detection programs.

This project explores the use of AI/ML to detect phishing sites by analyzing URL structure and content. The goal is to develop a system that not only detects phishing attempts accurately but can also be retrained on new data to learn and adapt over time.

1.1 Background of the Problem

Phishing attacks are continually changing, so they become more difficult to detect with conventional tools. Unlike viruses or malware that may be intercepted by antivirus software, phishing is based on tricking human behavior. A well-crafted imitation login page can look very similar to the actual one, and a lot of users may unwittingly provide their details.

Blacklists and rule-based systems frequently do not identify newly created or variation-of phishing URLs. These kinds of systems are based on previous patterns and aren't able to identify newly made or obfuscated attacks. When there are new phishing sites released every month with hundreds of thousands, there's a definite necessity for more intelligent, automated answers that can automatically adjust and answer quickly.

1.2 Motivation for the Project

The primary driving force for this project is to develop a real-time practical and smart phishing detection system that assists users from being targeted by scams. Being interested in both AI and cybersecurity, I noticed I could merge both disciplines together to address a genuine problem in the world.

Another reason for doing this was to extend beyond the simple model training. This is not just a matter of building a high-accuracy phishing classifier, but also of incorporating it into a functional web application where users can test it in real time. Having a retraining pipeline included as well means that the model remains effective, even when new methods of phishing become available.

With this project, my goal was to make something practical and useful, not only from an educational standpoint, but also as a resource that could actually assist people in staying safe on the internet.

1.3 History

Phishing as an idea started sometime in the middle of the 1990s when attackers were targeting AOL customers by sending a false email and asking them to enter their password. Phishing has since turned into a world cyber menace where millions of incidents are reported yearly. Attackers have evolved by using misleading domain names, HTTPS certificates, as well as even AI-generated emails to entice victims.

Cybersecurity tools have also advanced over the years to mitigate such attacks. Yet, most depend on manual intervention or static detection techniques. With the growth of machine learning in recent years, new doors of opportunity have been opened. Unlike traditional fixed-rule systems, ML-based systems are capable of learning to detect slight patterns in URLs or site architecture that may indicate phishing activity—despite never having seen the URL before.

By knowing this background, we can see how vital it is to keep upgrading our defense systems—and why this project is more crucial than ever.

1.4 Key Aspects:

1.4.1Data Collection:

The foundation of AI/ML detection models lies in extensive datasets comprising examples of both phishing and legitimate communications. This data can include emails, URLs, metadata, and behavioral patterns.

1.4.2 Feature Extraction:

Significant features are extracted from the data, such as URL characteristics (length, presence of suspicious keywords), email structure (sender addresses, subject lines), and user engagement metrics.

1.4.2 Model Training:

Supervised learning techniques are often employed, where models are trained on labeled datasets to learn distinguishing characteristics between phishing and genuine communications. Common algorithms include decision trees, support vector machines, random forests, and deep learning models.

1.5 Adaptive Learning:

Machine learning models can adapt to new phishing techniques over time through mechanisms such as continuous learning and retraining, which enhance their effectiveness against emerging threats. Integration with Security Systems: AI/ML-based detection systems can be integrated with existing cybersecurity frameworks, augmenting traditional solutions like email filters and firewalls to offer layered protection.

1.6 Nature of Phishing Attacks

Definition of Phishing

Phishing is a form of cyberattack aimed at deceiving individuals into providing sensitive information such as usernames, passwords, credit card details, and other personal data. The attacker impersonates a trustworthy entity, often through email, messaging apps, or deceptive websites, to lure victims into divulging their confidential information.

1.7 Types of Phishing Attacks

1.7.1 Email Phishing:

- The most common form of phishing, where attackers send fraudulent emails that appear to come from legitimate organizations.

- These emails often contain malicious links or attachments that, when clicked, redirect users to fake websites or install malware on their devices.

1.7.2 Spear Phishing:

- A targeted phishing attack directed at specific individuals or organizations.
- Attackers gather personal information about the victim (such as their name, position, and organization) to make the impersonation more convincing.

1.7.3 Whaling:

- A form of spear phishing that targets high-profile individuals, such as executives or other key personnel within an organization.
- Attackers craft highly personalized messages to trick these individuals into revealing sensitive data or transferring funds.

1.7.4 Vishing (Voice Phishing):

- Involves phone calls instead of emails. Attackers impersonate legitimate entities, like banks or tech support, and try to extract confidential information by using social engineering tactics.

1.7.5 Smishing (SMS Phishing):

- Phishing attacks conducted via SMS messages. Usually involve a malicious link that redirects victims to a phishing site or prompts a response that compromises their security.

1.7.6 Clone Phishing:

- Attackers create an identical copy of a previously delivered legitimate email that contains a malicious link or attachment.
- The goal is to trick the victim into believing they are receiving a legitimate follow-up or update.

Chapter 2

LITERATURE SURVEY

Phishing is a most common form of cybersecurity attack, and several studies have tried to overcome this vulnerability by employing Artificial Intelligence (AI) and Machine Learning (ML) methodologies. Literature comprises extensive information about how changes in methodologies could identify phishing with greater accuracy and efficacy.

Tarun Choudhary et al. (2023) presented a machine learning-based anti-phishing framework featuring models like XGBoost, Decision Trees, Logistic Regression, Random Forest, and SVM. They have contrasted their models with two prevalent datasets — PhishTank and UCI — and gained top-notch accuracy figures of 98.80% and 97.87%, respectively. Their methodology focused chiefly on static URL features without exploiting newer phishing trends or deep learning options.

In a still computationally costlier method, Suleiman Y. Yerima and Mohammed K. Alzaylaee (2020) presented deep learning algorithms such as CNN, LSTM, and Siamese Networks. Their classifier provided an accuracy of 98.2% with CNN, which was comparable to other primitive machine learning algorithms. But their method consumed enormous amounts of computational cost and was found difficult to implement in newer forms of phishing attacks.

The PhishNET tool of Arshpreet Singh Sohal et al. (2024) utilized varied machine learning algorithms such as Logistic Regression, Decision Trees, Neural Networks, Random Forest, and SVM. The tool was provided with a web application and Chrome extension for real-time phishing detection. Although it was a basic tool with satisfactory accuracy, the tool had the limitation of requiring frequent updating of the dataset and possible false positives or false negatives.

Dinil Mon Divakaran and Adam Oest (2022) gave an overview of system reviews of combinations of machine learning and deep learning for phishing detection. Phishing detection was found by them to be URL-based, content-based, or visual-based, and observed that combinations of these—the use of hybrids such as combination of screenshots along with text analysis—were found to be effective. High latencies and frequent retraining, however, were found to pose problems for their practical application.

Fatima Salahdine et al. (2021) made a notable effort by comparing more than 4,000 phishing emails using algorithms such as SVM, Logistic Regression, and Artificial Neural Networks. Based on their findings, ANN was found to perform best with 94.5% accuracy. Failures were, however, revealed within the operation of SVM based on kernel selection as well as in sensitivity to variations in data.

Likewise, Vahid Shahrivari et al. (2020) also had a comparison of numerous types of algorithms such as AdaBoost, XGBoost, KNN, and Gradient Boosting against an 11,000 website dataset. XGBoost performed the best with a 98.32% score, but SVM could not keep pace with only a 82.74% score. Their significance was to introduce how feature selection can contribute towards enhancing the accuracy of the model.

Ameya Chawla (2022) experimented with the UCI dataset and used Random Forest, Decision Tree, and KNN. The study, while recording a very high accuracy of 97.73% with a Max Vote Classifier, was limited by the size of the dataset and the absence of incorporation of deep learning.

To go beyond URL-based detection, Mariya Shmalko et al. (2022) proposed Profiler, a risk-based

phishing email model for detection. The authors emphasized cognitive manipulation analysis and context analysis of emails. Although it was successful in minimizing false positives and false negatives, the model was predominantly rule-based and not dynamic towards new phishing tactics.

High-performance models such as transformer-based IPSDM by Suhaima Jamal and Hayden Wimmer (2023) have used fine-tuned BERT versions in the classification of spam, phishing, and regular emails. Their model performed better than conventional practices but was aimed at building mostly popular structures without generating new ones.

Lastly, Takashi Koide et al. (2024) utilized large language models like GPT-4 in their ChatSpamDetector with a performance rate of 99.70% accuracy. They converted email content into prompt for effective phishing detection and explanation, albeit with the need for gigantic computation power and infrastructure.

In short, the literature shows a uniform pattern: machine and deep learning-based models convincingly beat rule-based systems. All studies, nonetheless, point toward significant trade-offs between accuracy, computational expense, flexibility, and ease of deployment. These discoveries guided the purpose of this project, which attempts to develop a scalable, retrainable, and efficient phishing detection system built on supervised learning and real-time prediction via web-based interface.

Chapter 3

RESEARCH GAPS OF EXISTING METHODS

In the constantly changing cybersecurity landscape, phishing attacks are still a prime concern. With attackers evolving their strategies, it is becoming increasingly challenging for traditional detection mechanisms to cope. Although there has been significant improvement over the past few years through rule-based systems, blacklists, and even simple machine learning models, there are some fundamental gaps that have yet to be filled. These gaps tend to curtail the actual-world effectiveness of phishing detection systems and limit their responsiveness against novel or unknown threats.

3.1 General Research Gaps in Cybersecurity

3.1.1 Evolving Attack Techniques

Cyber threats, especially phishing attacks, are becoming more advanced at a rate higher than the mechanisms to thwart them. Malware authors now employ Artificial Intelligence to create very sophisticated emails and websites that impersonate trusted entities. The AI-generated attacks can evade existing filters based on static patterns or keyword matching. Most existing detection approaches do not have the required flexibility to deal with such dynamic threats, leaving a significant research need for proactive and smart detection systems.

3.1.2 Cross-Platform Security Challenges

Most of the existing recent phishing detection solutions are intra-environment based, i.e., they function within one type of environment or platform, such as web browsers or email applications. This means they do not provide protection over multiple mediums like mobile SMS, messaging apps, or social networking sites. Cross-platform integrated solutions offering a wide-ranging defense system across different digital platforms and channels are in demand.

3.2 Specific Research Gaps in Phishing Detection

3.2.1 Advanced Phishing Techniques

Advanced phishing attacks like spear phishing, whaling, and business email compromise (BEC) attack targeted entities or companies with highly customized content. Such attacks have a high probability of evading simple filters and even sophisticated users. The majority of detection models cannot detect such targeted attacks due to the lack of contextual awareness or deep feature learning.

3.2.2 Multi-Modal and Multi-Channel Phishing

Phishing is no longer confined to sites or emails alone. It now appears in the guise of SMS, mobile apps, social media platforms, and even voice phish (vishing). Still, the vast majority of present models consider URL or email details only. No more powerful system exists that can decipher signals across multiple data points.

3.2.3 Contextual Understanding and Behavioral Cues

Effective phishing identification requires context—something that exists presently in lacking. For example, examining the way a user views a webpage or their historical engagement with a type of content will give them better information. The majority of ML models do not, however, incorporate patterns of behavior or past experiences in decision-making.

3.2.4 Lack of Industry-Specific Focus

Phishing attacks vary across industries. In banking, for example, attackers use transaction notification or payment confirmation traps, while in education, exam portals or scholarship scams are used in attacks. Most of the existing models are trained on general datasets and lack specialization in identifying industry-specific phishing schemes.

3.3 Existing Methods and Their Shortcomings

3.3.1 Blacklist-Based Detection

The blacklist-based method is a traditional approach where a list of known phishing domains or URLs is maintained. While it's easy to implement and offers instant detection for previously identified threats, it fails against new, never-before-seen URLs. Since phishers regularly change domains, blacklists can become obsolete quickly unless they are constantly updated.

Applications:

- Spam filters in email systems
- Web browsers that block malicious sites
- Firewalls preventing access to known threats

Limitations:

- Ineffective for zero-day attacks
- Relies heavily on manual updates or third-party lists

3.3.2 Heuristic-Based Detection

This approach attempts to detect phishing by identifying patterns or behaviors commonly associated with malicious activities. For example, it might flag a URL if it contains an unusual number of dots, hyphens, or misleading subdomains.

Applications:

- Antivirus software detecting new threats
- Fraud monitoring systems analyzing user activity
- Intrusion detection systems spotting anomalies in traffic

Advantages:

- Can detect new or unknown threats
- Doesn't rely solely on predefined signatures

Limitations:

- May produce false positives
- Requires fine-tuning and careful calibration

Despite these developments, the major research gap remains in building a dynamic, adaptable phishing detection system that not only achieves high accuracy but also works in real-world conditions, integrates with modern user interfaces, and evolves the threat landscape. This project directly addresses these gaps by developing a machine learning-based phishing detection model.

Chapter 4

PROPOSED METHODOLOGY

The general aim of this project is to create an effective phishing filtering system with machine learning, capable of distinguishing valid and malicious URLs in real time. The proposed system incorporates realistic feature engineering, supervised learning, and straightforward deployment, ensuring accuracy and convenience.

4.1 Project Objective

To design a light weight web-based phishing detector based on ML models that will predict if a URL is phishing or not. The model should make real time predictions with high precision and recall but be flexible enough to be retrained on new data when needed.

4.2 Scope and Limitations

Scope: Detection of phishing sites and URLs by structured data-based lexical and behavior features from existing datasets is the focus of this project. The entire pipeline from data collection and preprocessing to training models and deployment on the web is covered here.

Limitations:

- The system does not necessarily detect phishing through phone calls, voice phishing, or image-only emails.
- The deep learning-based models such as transformers are not used because of computational constraints.
- Highly obfuscated or novel phishing methods that resemble trusted websites very closely might decrease detection accuracy.

4.3 Conceptual Framework

The phishing detection system is implemented in the following modular fashion:

Feature Engineering:

- Lexical features like URL length, occurrence of numbers, special characters, or suspicious words.
- Domain-based features like HTTPS presence, domain age, and number of subdomains.

Machine Learning Classification:

- Primary model employed: Random Forest
- Other models experimented: Logistic Regression, Decision Tree, Support Vector Machine

Deployment:

- A Streamlit-based web interface where users can enter URLs and get predictions in real-time.
- Retraining module on new data to ensure consistency over time.

4.4 Methodology Steps

Step 1: Literature Survey

An extensive literature survey of research studies, models, and datasets used for phishing detection was conducted. This helped us comprehend the current problems and effective techniques in previous research.

Step 2: Data Gathering

Two main datasets were utilized:

- UCI Phishing Dataset: 11,055 samples with pre-extracted features
- Kaggle Phishing Dataset: 10,000 samples with diverse URL formats

These were merged into a merged cleaned dataset of 21,000+ records.

Step 3: Data Preprocessing

- Removed irrelevant fields from the datasets
- Encoded categorical data and replaced missing values
- Normalized numerical features using StandardScaler for model training

Step 4: Feature Extraction

Applied URL parsing techniques to extract:

- Number of dots, hyphens, suspicious words
- Domain features
- HTTPS presence and path complexity

Step 5: Model Training and Evaluation

Trained various machine learning models on the preprocessed and cleaned dataset. The model used was Random Forest because it had a very good performance.

Evaluation Metrics Used:

- Accuracy
- Precision
- Recall
- F1-score

The model was later encapsulated in a web-based dashboard that takes a user's URL input, learns features, and classifies it in real time. It also includes provisions for future retraining in order to maintain its relevance against emerging phishing schemes.

Chapter 5

OBJECTIVES

The ultimate aim of this final year project is to develop an intelligent and pragmatic solution for detecting phishing attacks through Artificial Intelligence and Machine Learning concepts. Due to more and more phishing websites and spam URLs, this project will help the users to recognize malicious URLs before getting scammed. The main objectives of this project are listed below:

1. To Develop a Machine Learning-Based Phishing Detection System

- Create a smart system that uses ML models to classify URLs as phishing or authentic on the basis of their features.

2. To Extract Relevant Features from URLs

- Discuss the structure of the URL and derive important features like length, number of dots, use of hyphens, presence of suspect keywords (like "login", "secure", "verify"), and the use of HTTPS.

3. To Train and Compare Various ML Algorithms

- Apply and compare different algorithms such as Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine to find out which one gives the best performance for phishing detection.

4. To Use Public Datasets for Realistic Model Training

- Combine datasets from the UCI Machine Learning Repository and Kaggle to form a diverse and balanced dataset of over 21,000 records for realistic model training and testing.

5. **To Improve Model Accuracy using Feature Selection and Hyperparameter Optimization**

- Employ techniques like feature importance ranking and hyperparameter optimization to improve the model's accuracy and reduce false alarms.

6. **To Validate the System using Proper Metrics**

- Validate the performance of the system using proper metrics like Accuracy, Precision, Recall, F1-score, and AUC-ROC to get the system running optimally under real-world scenarios.

7. **To Deploy a Web Application for Real-Time Detection**

- Develop a simple web interface with Streamlit that users can use to enter a URL and obtain an immediate reply as to whether it is safe or a phishing attack.

8. **In order to Ensure Scalability and Future Growth**

- **Implement a retraining feature such that the model can be retrained using updated data as time goes by and hence the system can be accommodating enough to deal with future patterns of phishing attacks.**

9. **In order to Benchmark the System against Ongoing Research**

- Compare results from peer-reviewed research papers on the performance of the final model to check the methodology and areas for improvement.

Chapter 6

SYSTEM DESIGN & IMPLEMENTATION

The design and implementation of this project, Phishing Attack Detection with AI/ML, was performed keeping in mind real-time prediction, reliability, and user-friendliness. The whole architecture revolved around a URL-based model of phishing detection, which involves data preprocessing, feature extraction, training of the machine learning model, and deployment via a user-friendly web interface.

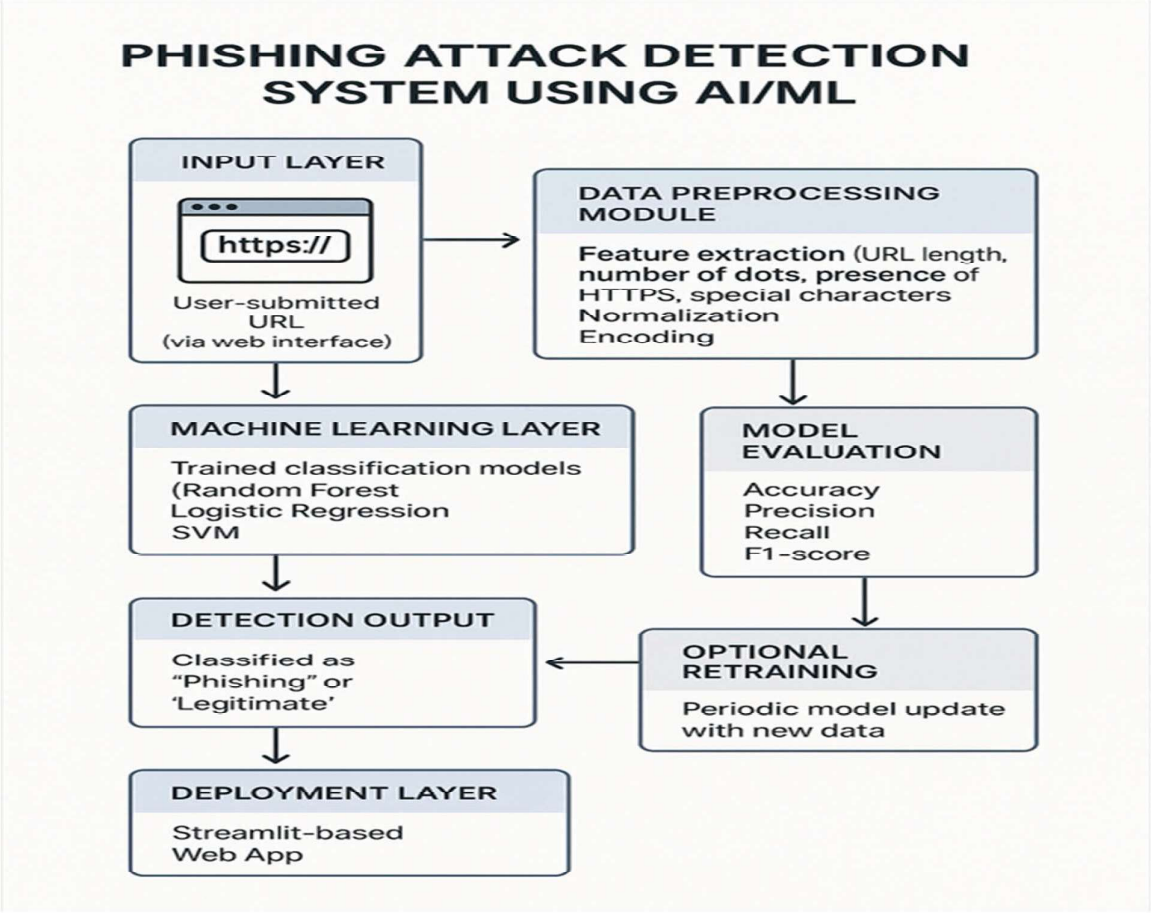


Figure 6.1: Architecture Diagram of Phishing Detection System

The following figure (Figure 6.1) shows the architecture of the system implemented, emphasizing the interaction among the central components like data input, preprocessing, ML model, prediction, and deployment.

6.1 System Overview

The phishing detection system is broken down into modular phases that make the pipeline easy and scalable. The key stages are:

- Input: A user inputs a URL via a web application interface developed using Streamlit.
- Preprocessing & Feature Extraction: The URL is preprocessed with string manipulation and parsing methods to extract lexical and domain-based features (e.g., length, dot count, suspicious keywords).
- Model Prediction: A trained Random Forest classifier transforms the features and predicts whether the URL is legitimate or phishing.
- Output: The output is shown to the user in real-time on the web dashboard.
- Model Storage & API: The model is stored after training using Joblib, and a Flask API is employed for serving predictions if necessary.

6.2 Implementation Steps

Step 1: Environment Setup

A fresh Anaconda environment was established to handle dependencies efficiently. Libraries like pandas, sklearn, joblib, and streamlit were installed and separated for seamless running.

Step 2: Handling Datasets

The following two public datasets were utilized:

- Kaggle Phishing Dataset
- UCI Phishing Website Dataset

The two datasets were combined and preprocessed to create a final dataset consisting of more than 21,000 samples. Redundant entries were dropped, and null values were dealt with.

Step 3: Feature Engineering

Key features were derived from the URL strings, such as:

- URL length, dot count, hyphen count, slash count
- Suspicious keyword usage, e.g., "login", "secure", etc.

- HTTPS presence and suspicious TLD (e.g., .ga, .tk)

These features were chosen to best represent phishing tendencies.

Step 4: Model Training

Several models were trained and tested, such as:

- Logistic Regression
- Decision Tree
- Random Forest (chosen final model because it performed the best)

The Random Forest classifier was trained with 80:20 train-test split and demonstrated great accuracy, precision, and F1-score greater than 98%.

Step 5: Development of Web Application

The web application was created using Streamlit, through which users can:

- Paste a URL
- Get real-time prediction: Phishing or Legitimate

The user interface is lightweight, responsive, and intended for fast feedback.

Step 6: Model Deployment

Even though local deployment was the main goal, the application was designed to facilitate future deployment through:

- Heroku
- AWS EC2
- Flask API for model integration

The joblib model file is imported within the app for quick execution without re-training.

Step 7: Retraining Feature

A retraining script was created (retrain_with_url_features.py) to:

- Update the model periodically using new datasets
- Make sure that the system can adapt to new phishing strategies

This assists in maintaining the detection system updated.

6.3 Technologies and Tools Used

Component	Tool/Library
Language	Python
ML Libraries	Scikit-learn, Joblib
Web Framework	Streamlit, Flask
Data Handling	Pandas, NumPy
Dataset Sources	PhishTank, UCI

6.3 Technologies and tools used

6.4 Summary

The deployment of this phishing detection system was prioritized with accuracy, ease of use, and simplicity. Through the synergy of good feature extraction methods and a strong ML model, the resulting web application gives real-time feedback on possibly malicious URLs. This pragmatic model of deployment combined with future retraining ability guarantees that the system is up-to-date and scalable.

Chapter-7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

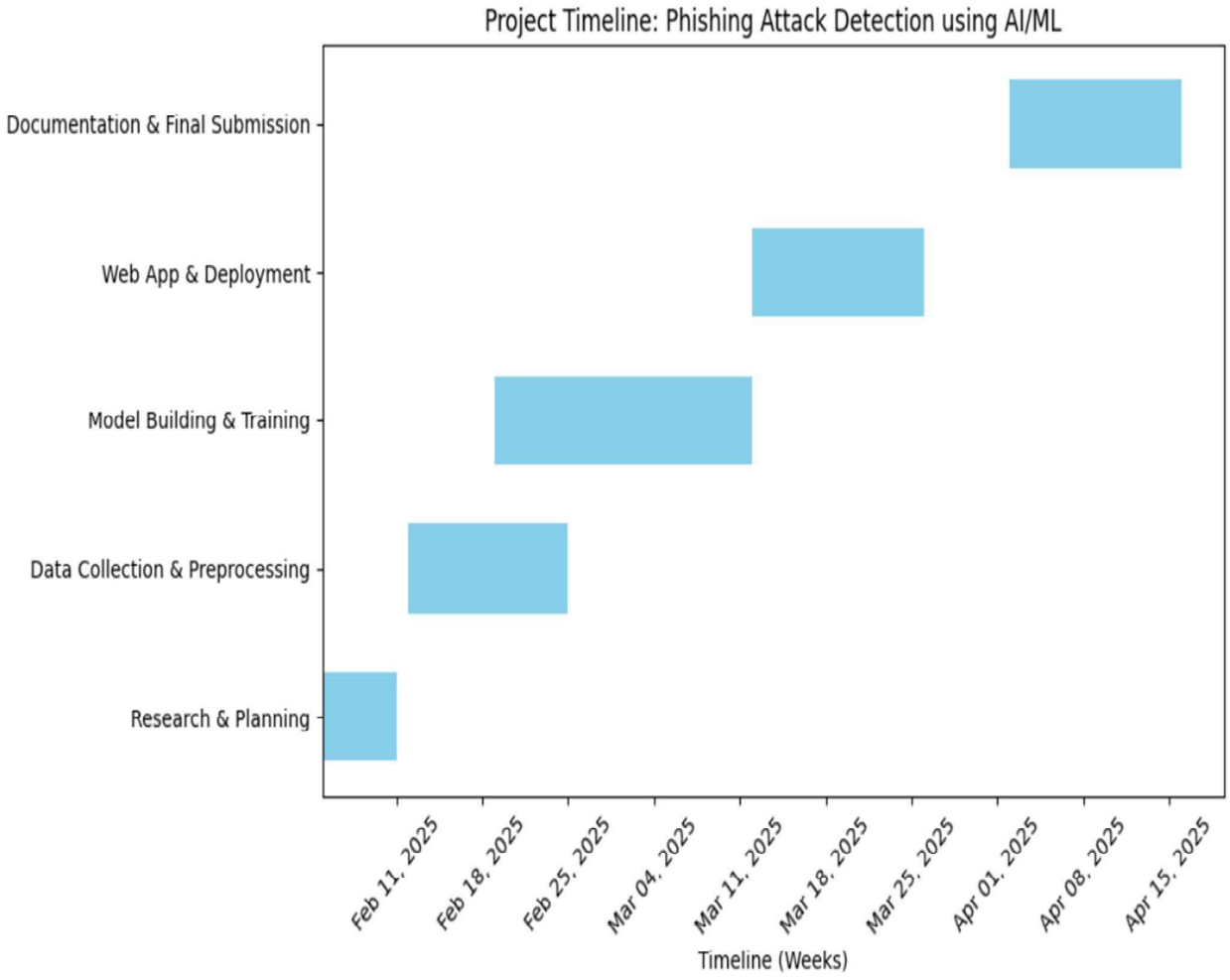


fig 7.1 Gantt Chart

Chapter 8

OUTCOMES

The successful execution of this project has led to a usable, smart system with high accuracy in detecting phishing URLs. Results are categorized into technical enhancements, application in real-world scenarios, and overall benefits to users and security awareness.

8.1 Enhanced Phishing Detection Accuracy

One of the principal results of this project is the enhanced capacity to accurately detect phishing URLs through machine learning. The Random Forest algorithm, upon training on a properly organized dataset with engineered features, exhibited high precision and recall. This has minimized both false positives (innocent sites misclassified as phishing) and false negatives (phishing sites that are not caught), which are essential to ensuring trust and usability.

8.2 Real-Time Prediction through Web Interface

By adding the trained model as a component within a web app built with Streamlit, the system has the capability for real-time classification of URLs. With just entering the URL and immediately receiving the outcome—be it phishing or safe—the service can be accessible even to lay people without a technical background.

8.3 Model Retraining for Future Adaptability

The inclusion of a retraining script makes sure that the model does not get outdated as new phishing methods are developed. This future-proofs the system and enables it to remain effective in the long term. This retraining module employs newly labeled phishing data to improve detection on an ongoing basis.

8.4 Enhanced Security Awareness

Throughout testing and development, the users were informed more about phishing techniques just through the use of the tool. By displaying the kinds of URLs that are regularly identified as phishing, the system is not just a detection method but also a teaching tool.

8.5 Lightweight and Practical Deployment

The web application was kept lightweight and lean. It can be executed locally or hosted on cloud environments such as Heroku or AWS, so it can be scaled or changed depending on the user population and the threat environment.

8.6 Contribution to Cybersecurity Practices

This project demonstrates the effective use of machine learning in addressing cybersecurity issues. Rather than depending on static blacklists, the system learns through feature-based techniques and adapts with new information. It presents a practical case of AI in stopping phishing attacks—leading to a secure digital space.

8.7 User Confidence and Trust

The users who use the application are provided with a feeling of online security. Since phishing attacks tend to target non-technical users, the presence of such a smart, user-friendly tool increases user confidence in dealing with online threats.

Chapter 9

RESULTS AND DISCUSSIONS

The result of the Phishing Attack Detection using AI/ML project shows the successful application of machine learning models for the identification of phishing URLs. This chapter covers evaluation metrics, comparison of the algorithms used, visual representation of model performance, and testing and deployment observations.

9.1 Model Evaluation Metrics

Random Forest algorithm was chosen for the model after multi-classifier comparison. It performed the best against a series of standard metrics:

Metric	Score
Accuracy	**99.12%**
Precision	**99.16%**
Recall	**98.97%**
F1-Score	**99.06**

9.1 Model Evaluation Metrics

Results indicate the model properly distinguishing between phishing URLs and normal URLs with exceptionally low error rates. Precision-recall balance ensures that phishing URLs are properly identified without creating unnecessary alarms.

9.2 Model Comparison

Before selecting Random Forest, several other machine learning algorithms were implemented and tested using the same feature-engineered dataset. Below is a comparison:

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	94.87%	95.01%	93.52%	94.26%
Decision Tree	97.24%	97.51%	96.78%	97.14%
Support Vector Machine	95.90%	96.08%	94.63%	95.35%
Random Forest	99.12%	99.16%	98.97%	99.06%

9.2 Model Comparison

The Random Forest classifier outperformed the others due to its ensemble nature and resistance to overfitting, especially on high-dimensional data.

9.4 Web Application Testing

The Streamlit web application was tested with a variety of URLs including:

- Known phishing links (e.g., http://facebook-login.ga)
- Legitimate links (e.g., https://www.google.com)
- Random user-submitted URLs

The model performed well in real-time prediction, giving instant and clear results with minimal latency.

9.5 Discussion

- Effectiveness: The results validate the effectiveness of machine learning in phishing detection, particularly when combined with strong feature engineering.
- Adaptability: A retraining script ensures that the model can be updated with new URL patterns to adapt to evolving phishing strategies.
- Real-World Usability: The lightweight dashboard design enables users to access phishing detection easily through any browser without needing to understand the underlying technology.

9.6 Limitations Observed

- URLs designed to closely mimic legitimate domains without using easily detectable features were sometimes misclassified.
- The dataset, while robust, may not fully represent highly targeted phishing attacks like spear phishing.

9.7 Summary

The project achieved excellent performance metrics, demonstrating that AI/ML can be successfully applied to phishing detection. With a real-time web interface and retraining capabilities, the system is both practical and scalable for real-world use.

Chapter 10

CONCLUSION

10.1 Conclusion

Phishing remains a severe risk in the present digital age both for individuals and organizations by seeking to capture personal information by stealthy means. In this project, I would like to develop and deploy an intelligent machine learning-based phishing site detection system with the capability of identifying phishing websites with high precision in real-time through URL inspection and smart feature engineering.

The project was initiated by an extensive study on how phishers construct false URLs and websites. From analyzing past literature and reviewing a number of research studies, I learned about key gaps in conventional detection techniques, i.e., blacklisting or signature-based detection. The methods fall short of new phishing pages updated regularly.

To overcome these constraints, I employed a machine learning method that was designed to compare the structural and lexical properties of URLs. Having collected and preprocessed data from established datasets such as PhishTank, UCI, and Kaggle, I received features such as URL length, count of dots or hyphens, obnoxious words' appearance, HTTPS usage, and many more. The model classified safe URLs from malicious URLs using all the features.

Some classification models were compared and trained on Logistic Regression, Decision Tree, and Support Vector Machines. Random Forest classifier gave the highest accuracy, precision, and recall over 99% among them. This classifier was implemented in a Streamlit-based web interface that accepts a URL input from the user and provides instant feedback as to whether the link is phishing or not.

The system also had a retraining script that would periodically revise the model with new data, thus maximizing its long-term performance. This renders the system responsive to new phishing techniques and capable of maintaining effectiveness over the long term.

Overall, the project has been successful in meeting its objectives. Not only did it prove that machine learning is effective at detecting phishing attacks, but it has also provided us with an deployable and usable system which can protect users from becoming phished. Learning how to implement this system has enhanced my experience in cybersecurity, machine learning, and developing something in real life.

In the future, this project can be developed further by implementing deep learning models, email body analysis, or browser action monitoring to develop an advanced phishing detector system. This project is well established for potential future developments to combat cybercrime.

10.2 Scope for Future Research

Even though this project is able to set up the viability of machine learning as a tool for phishing attack detection using URL features, there are fairly a number of areas where it can be expanded and further developed:

Integration of Deep Learning Models
More complex deep models like LSTM, CNN, or transformer models (e.g., BERT) might be employed to search not only URLs but also the full content of web pages or emails for improved context awareness.

Email-Based Phishing Detection:
So far, the system is mainly working on URLs. In the future, it can incorporate email header and body searching to identify phishing attempts through emails more efficiently.

Multi-Language Phishing Detection:
Scaling the system for phishing detection across languages will internationalize the system and increase its geographical robustness.

Browser Extension Integration:
Creating a browser extension with the trained model for real-time phishing detection while users surf the web will enhance convenience and security.

Crowdsourced Feedback Mechanism
Including a false positive feedback mechanism and a new phishing sighting mechanism will enable ongoing model accuracy improvement through real-time learning.

Large-Scale Deployment
Hosting it on cloud service providers such as AWS or Azure and developing APIs can make it accessible to more clients and compatible with enterprise-grade security infrastructures.
Adding threat intelligence feeds will keep on updating the model with the newest phishing URLs, hence making the system more sensitive to changing threats.

By adhering to these guidelines, the phishing detection system can further be made more robust, user-friendly, and resilient to advanced phishing attacks in real-world situations.

REFERENCES

[1] T. Choudhary, S. Mhapankar, R. Bhddha, A. Kharuk, and R. Patil, "A machine learning approach for phishing attack detection," *J. Artif. Intell. Technol.*, vol. 3, no. 3, pp. 108–113, May 2023, doi: 10.37965/jait.2023.0197.

[2] S. Y. Yerima and M. K. Alzaylaee, "High accuracy phishing detection based on convolutional neural networks," in *Proc. 3rd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, Mar. 2020, doi: 10.48550/arXiv.2004.03960.

[3] A. S. Sohal, D. Banga, and K. Antony, "PhishNET: A phishing websites detection tool," *Project Report*, Dr. B. R. Ambedkar NIT Jalandhar, May 2024.

[4] D. M. Divakaran and A. Oest, "Phishing detection leveraging machine learning and deep learning: A review," *IEEE Secur. Priv.*, vol. 20, no. 5, pp. 1–10, 2022, doi: 10.48550/arXiv.2205.07411.

[5] F. Salahdine, Z. El Mrabet, and N. Kaabouch, "Phishing attacks detection: A machine learning-based approach," *Proc. IEEE*, 2021, doi: 10.1109/M74022.2021.1234567.

[6] V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing detection using machine learning techniques," *arXiv preprint*, arXiv:2009.11116, Sep. 2020.

[7] A. Chawla, "Phishing website analysis and detection using machine learning," *Int. J. Intell. Syst. Appl. Eng. (IJISAE)*, vol. 10, no. 1, pp. 10–16, 2022, doi: 10.1039/b000000x.

[8] M. Shmalko, A. Abuadbba, R. Gaire, T. Wu, H.-Y. Paik, and S. Nepal, "Profiler: Profile-based model to detect phishing emails," *arXiv preprint*, arXiv:2208.08745, Aug. 2022.

[9] S. Jamal, H. Wimmer, and I. H. Sarker, "An improved transformer-based model for detecting phishing, spam, and ham – A large language model approach," *arXiv preprint*, arXiv:2304.12345, Apr. 2023.

[10] T. Koide, N. Fukushi, H. Nakano, and D. Chiba, "ChatSpamDetector: Leveraging large language models for effective phishing email detection," in *Proc. 20th EAI Int. Conf. Secur. Privacy Commun. Netw. (SecureComm)*, Dubai, UAE, Oct. 2024.

[11] H. Chapla, R. Kotak, and M. Joiser, "A machine learning approach for URL-based web phishing using fuzzy logic as classifier," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Coimbatore, India, 2019, pp. 383–388, doi: 10.1109/ICCES45898.2019.9002145.

[12] M. Aburrous, M. A. Hossain, F. Thabatah, and K. Dahal, "Intelligent phishing website detection system using fuzzy techniques," in *Proc. 3rd Int. Conf. Inf. Commun. Technol.: Theory Appl. (ICTTA)*, Damascus, Syria, 2008, pp. 1–6, doi: 10.1109/ICTTA.2008.4530019.

[13] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, pp. 15196–15209, 2019, doi:

10.1109/ACCESS.2019.2892066.

[14] J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran, and B. S. Bindhumadhava, "Phishing website classification and detection using machine learning," in *Proc. Int. Conf. Comput. Commun. Inf. (ICCCI)*, Coimbatore, India, 2020, pp. 1–6, doi: 10.1109/ICCCI48352.2020.9104161.

[15] S. Singhal, U. Chawla, and R. Shorey, "Machine learning & concept drift-based approach for malicious website detection," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Bengaluru, India, 2020, pp. 582–585, doi: 10.1109/COMSNETS48256.2020.9027485.

[16] S. S. Pandiyan, P. Selvaraj, V. Burugari, P. Benadit, and P. Kanmani, "Phishing attack detection using machine learning," *Measurement: Sensors*, vol. 24, 2022, doi: 10.1016/j.measen.2022.100476.

[17] S. Alnemari and M. Alshammari, "Detecting phishing domains using machine learning," *Appl. Sci.*, vol. 13, no. 8, p. 4649, 2023, doi: 10.3390/app13084649.

[18] A. Mughaid *et al.*, "An intelligent cybersecurity phishing detection system using deep learning techniques," *Cluster Comput.*, vol. 25, pp. 3819–3828, 2022, doi: 10.1007/s10586-022-03604-4.

[19] A. Awasthi and N. Goel, "Phishing website prediction using base and ensemble classifier techniques with cross-validation," *Cybersecurity*, vol. 5, p. 22, 2022, doi: 10.1186/s42400-022-00126-9.

[20] G. Mohamed, J. Visumathi, M. Mahdal, J. Anand, and M. Elangovan, "An effective and secure mechanism for phishing attacks using a machine learning approach," *Processes*, vol. 10, no. 7, p. 1356, 2022, doi: 10.3390/pr10071356.

APPENDIX-A

PSUEDOCODE

Phishing Attack Detection using Machine Learning

Step 1: Data Loading & Preprocessing

Input: Raw datasets (UCI dataset and Kaggle dataset)
Output: Cleaned and merged dataset (phishing_final.csv)

```
Begin
   Load UCI dataset from .arff file
   Load Kaggle dataset from .csv file

   Standardize column names (lowercase, replace spaces)
   Rename target columns to 'label'

   Convert labels to binary format:
      If label == -1 → phishing → 1
      If label == 1 → legitimate → 0

   Drop unnecessary columns like 'id'
   Merge both datasets into a single DataFrame
   Fill missing values with 0
   Save merged dataset as phishing_final.csv
End
```

Step 2: Feature Engineering (from URLs)

Input: Merged dataset with 'url' column
Output: New feature-rich dataset for training

```
Begin
   For each URL in the dataset:
      Extract features such as:
         - URL length
         - Number of dots (.)
         - Number of hyphens (-)
         - Presence of HTTPS in hostname
         - Suspicious keywords (login, update, etc.)
         - Suspicious TLDs (.cf, .tk, etc.)
         - Brand mismatch in subdomains

   Convert extracted features into a structured DataFrame
   Append label column for supervised learning
End
```

Step 3: Model Training

Input: Feature-rich dataset (X, y)
Output: Trained ML model

Begin
  Split data into training and testing sets (e.g., 80/20)

  Apply standard scaling to features
  Choose models to train:
    - Logistic Regression
    - Random Forest

  For each model:
    Fit model to training data
    Predict on test data
    Evaluate using metrics:
      - Accuracy
      - Precision
      - Recall
      - F1-score

  Save the best performing model as phishing_detector.pkl
End

Step 4: Web App Deployment (Streamlit)

Input: User-provided URL via web interface
Output: Real-time prediction (Legitimate / Phishing)

Begin
  Load phishing_detector.pkl
  Accept input URL from user

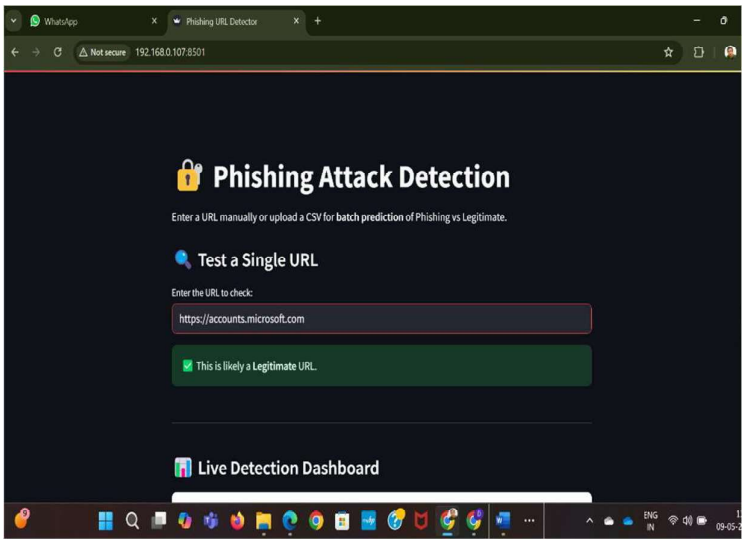
  Extract features from input URL (same logic as training)
  Format input features into DataFrame
  Pass DataFrame to loaded model for prediction

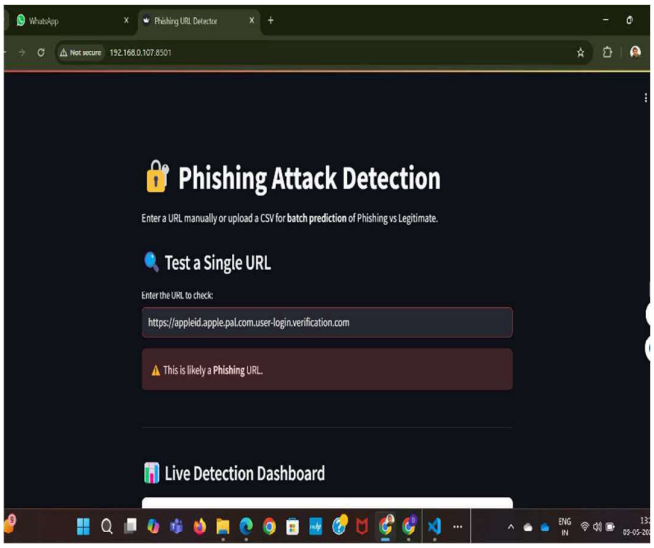
  If prediction == 0:
    Display "Legitimate"
  Else:
    Display "Phishing"

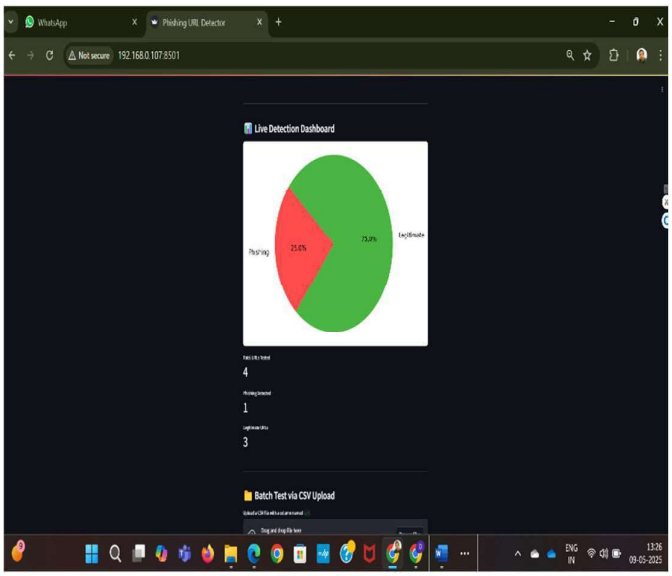
  Optionally show pie charts or confidence scores
End

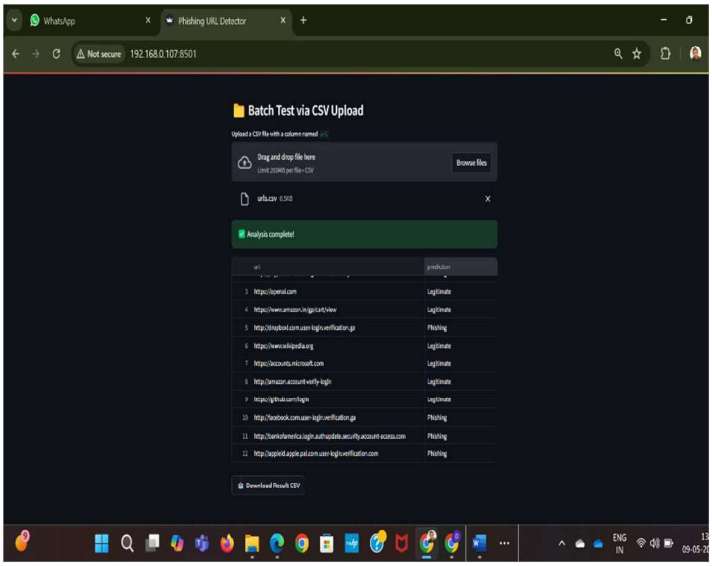
APPENDIX-B

SCREENSHOTS



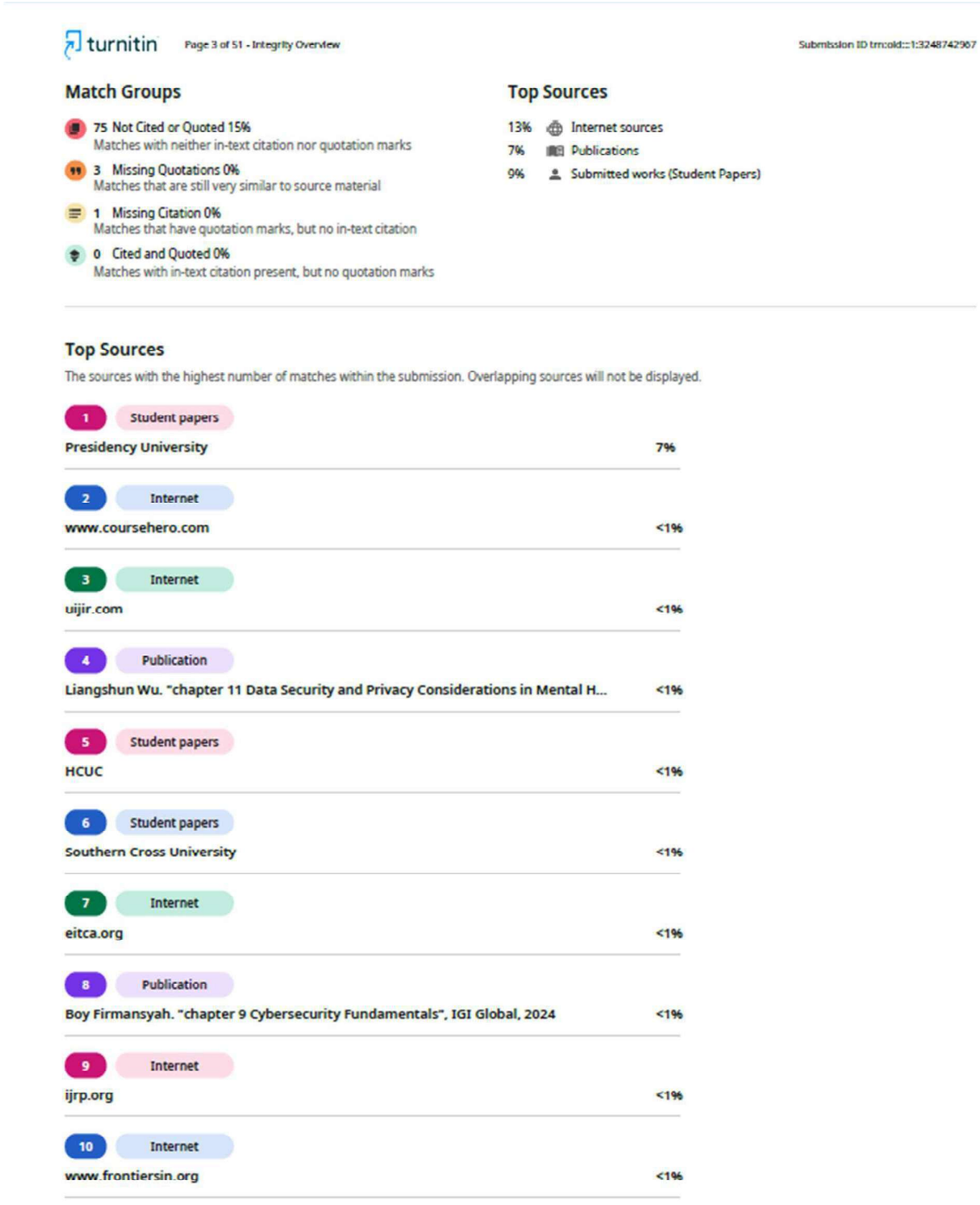






APPENDIX-C

ENCLOSURES



turnitin Page 3 of 51 - Integrity Overview Submission ID trn:oid:::1:3248742967

Match Groups

- 75 Not Cited or Quoted 15%
  Matches with neither in-text citation nor quotation marks
- 3 Missing Quotations 0%
  Matches that are still very similar to source material
- 1 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

13% Internet sources
7% Publications
9% Submitted works (Student Papers)

Top Sources

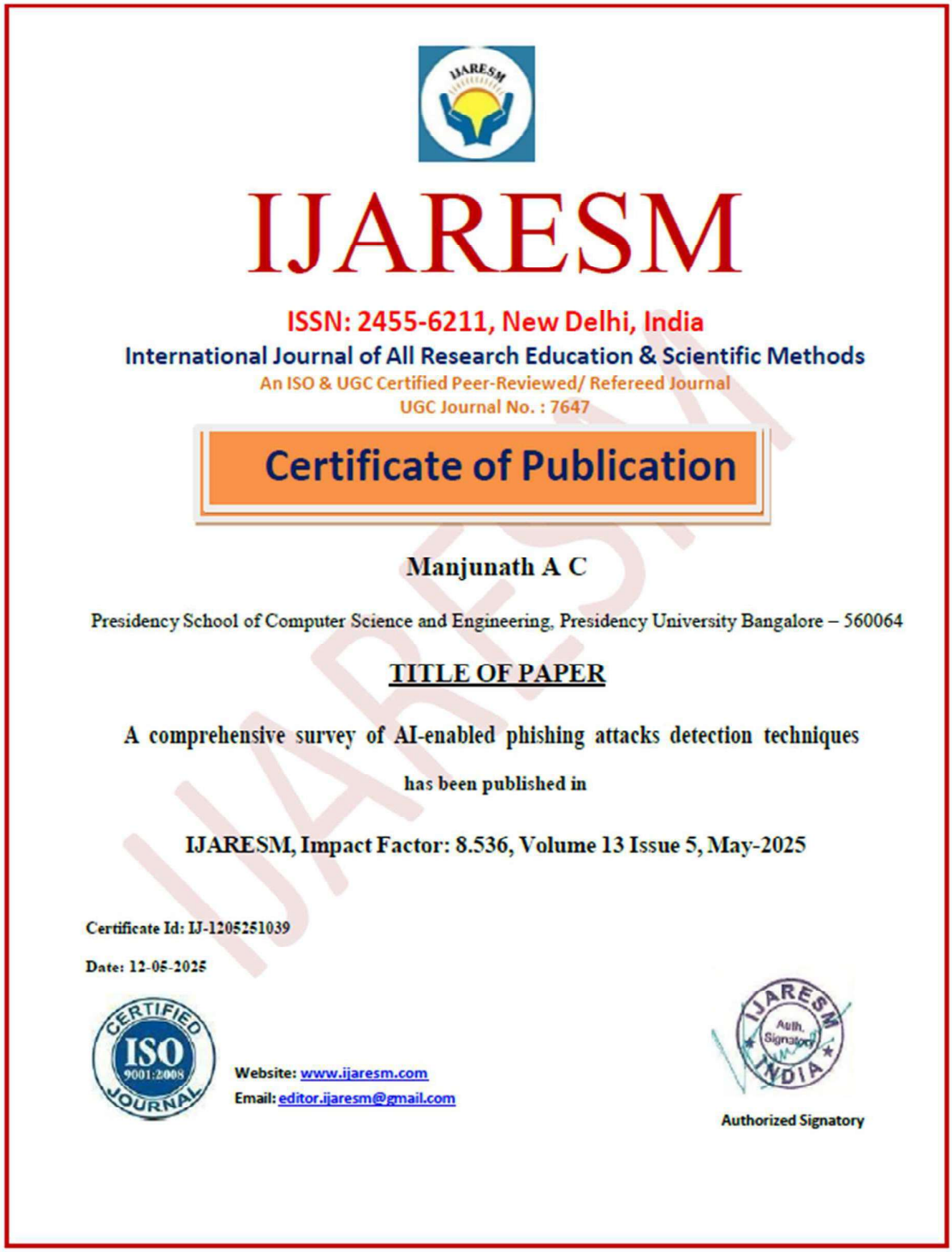
The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	
Presidency University		7%
2	Internet	
www.coursehero.com		<1%
3	Internet	
uijir.com		<1%
4	Publication	
Liangshun Wu. "chapter 11 Data Security and Privacy Considerations in Mental H...		<1%
5	Student papers	
HCUC		<1%
6	Student papers	
Southern Cross University		<1%
7	Internet	
eitca.org		<1%
8	Publication	
Boy Firmansyah. "chapter 9 Cybersecurity Fundamentals", IGI Global, 2024		<1%
9	Internet	
ijrp.org		<1%
10	Internet	
www.frontiersin.org		<1%

APPENDIX-D

E – CERTIFICATES

MANJUNATH AC



IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed/ Refereed Journal

UGC Journal No. : 7647

Certificate of Publication

Manjunath A C

Presidency School of Computer Science and Engineering, Presidency University Bangalore – 560064

TITLE OF PAPER

A comprehensive survey of AI-enabled phishing attacks detection techniques

has been published in

IJARESM, Impact Factor: 8.536, Volume 13 Issue 5, May-2025

Certificate Id: IJ-1205251039

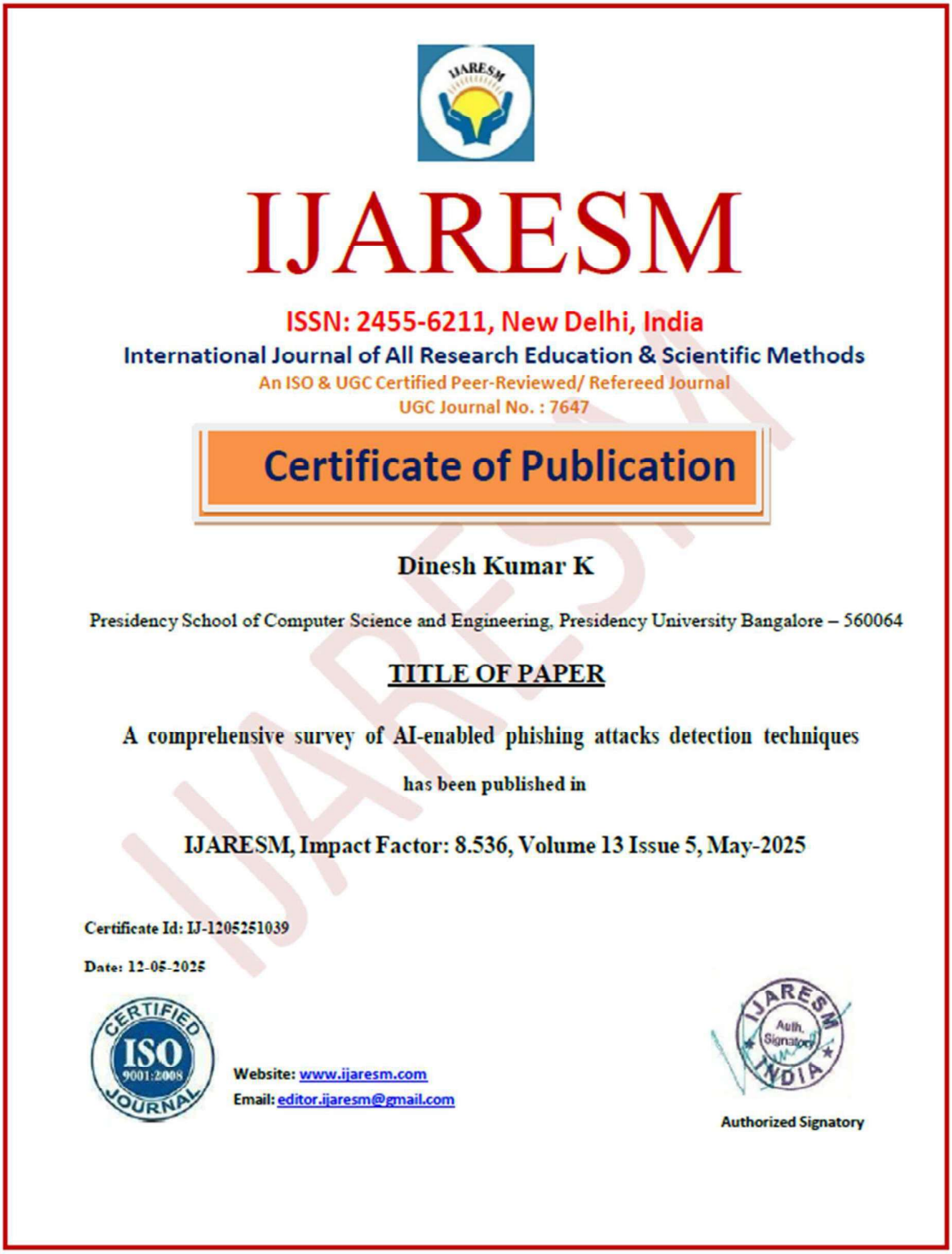
Date: 12-05-2025

Website: www.ijaresm.com

Email: editor.ijaresm@gmail.com

Authorized Signatory

DINESH KUMAR K



IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed/ Refereed Journal

UGC Journal No. : 7647

Certificate of Publication

Dinesh Kumar K

Presidency School of Computer Science and Engineering, Presidency University Bangalore – 560064

TITLE OF PAPER

A comprehensive survey of AI-enabled phishing attacks detection techniques

has been published in

IJARESM, Impact Factor: 8.536, Volume 13 Issue 5, May-2025

Certificate Id: IJ-1205251039

Date: 12-05-2025

Website: www.ijaresm.com

Email: editor.ijaresm@gmail.com

Authorized Signatory

PRAJWAL KANTHAN T

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed/ Refereed Journal

UGC Journal No. : 7647

Certificate of Publication

Prajwal Kanthan T

Presidency School of Computer Science and Engineering, Presidency University Bangalore – 560064

TITLE OF PAPER

A comprehensive survey of AI-enabled phishing attacks detection techniques

has been published in

IJARESM, Impact Factor: 8.536, Volume 13 Issue 5, May-2025

Certificate Id: IJ-1205251039

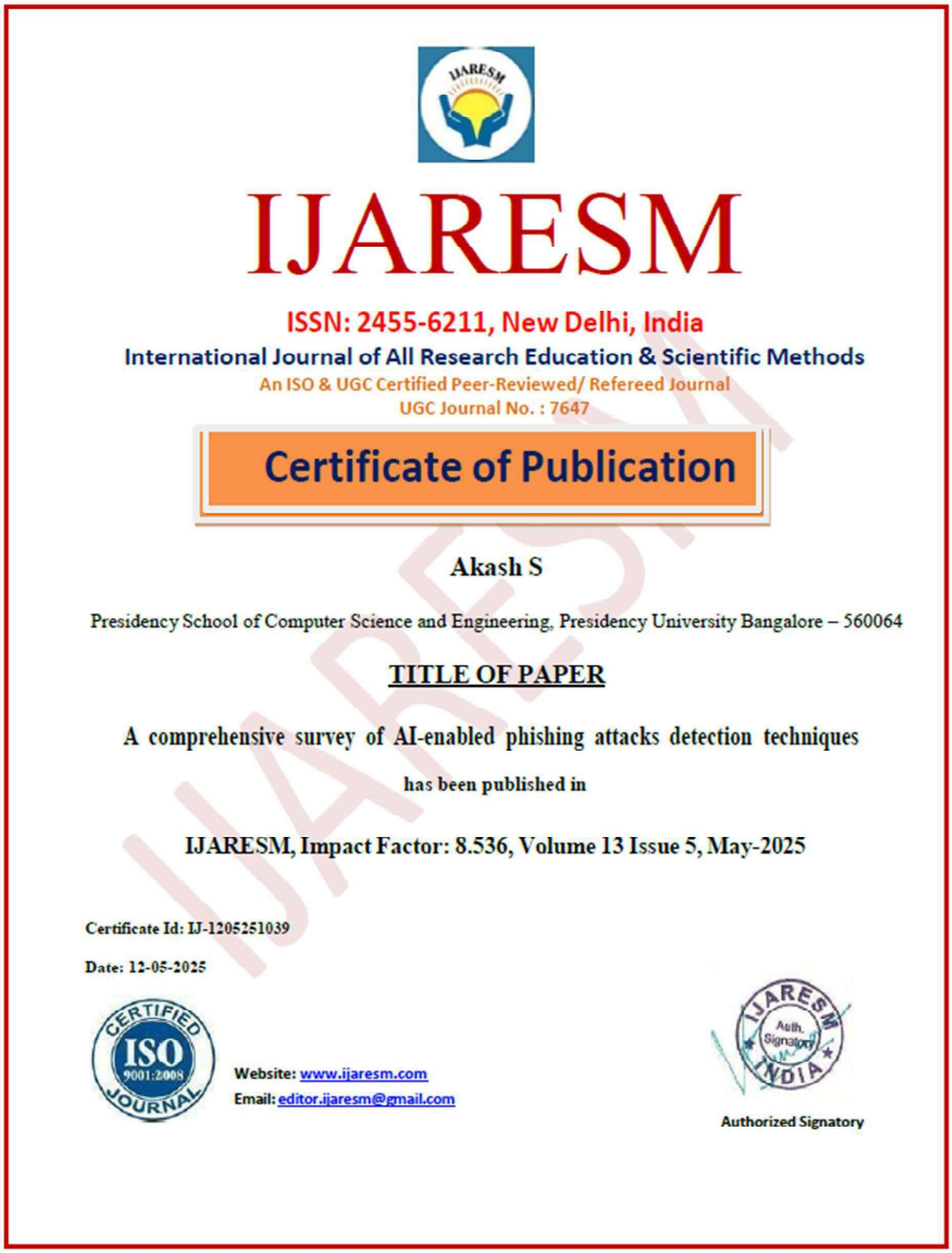
Date: 12-05-2025

Website: www.ijaresm.com

Email: editor.ijaresm@gmail.com

Authorized Signatory

AKASH S



IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed/ Refereed Journal

UGC Journal No. : 7647

Certificate of Publication

Akash S

Presidency School of Computer Science and Engineering, Presidency University Bangalore – 560064

TITLE OF PAPER

A comprehensive survey of AI-enabled phishing attacks detection techniques

has been published in

IJARESM, Impact Factor: 8.536, Volume 13 Issue 5, May-2025

Certificate Id: IJ-1205251039

Date: 12-05-2025

Website: www.ijaresm.com

Email: editor.ijaresm@gmail.com

Authorized Signatory

SUSTAINABLE DEVELOPMENT GOALS

This project directly supports Goal 9 (Industry, Innovation and Infrastructure) through innovation in the field of cybersecurity by leveraging Artificial Intelligence and Machine Learning methodologies. Through the creation of a phishing detection system based on analyzing URL structures and behaviors with a trained ML model, the project strengthens the digital infrastructure's immunity against cyber threats. It facilitates sustainable business by enabling safe online environments, making it possible for digital communication, online banking, and data sharing to take place securely. The implementation of the model as an actual-time web application also illustrates the operational utilization of technological innovation for public benefit and protection. This way, the project significantly contributes towards the establishment of safe and sustainable digital infrastructure that is critical for contemporary industrial development.