DETECTION OF PHISHING WEBSITE

USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

DHIVYA J	**-**	**IV CSE A**	**(610519104019)**
AMSAVENI S M	**-**	**IV CSE A**	**(610519104003)**
GAYATHRI S	**-**	**IV CSE A**	**(610519104024)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

DHIRAJLAL GANDHI COLLEGE OF TECHNOLOGY

SALEM - 636 309

ANNAUNIVERSITY :: CHENNAI 600 025

MAY 2023

BONAFIDE CERTIFICATE

Certified that this project report **“DETECTION OF PHISHING WEBSITE USING MACHINE LEARNING”**is the bonafide work of **“DHIVYA J (610519104019), AMSAVENI S M (610519104003), GAYATHRI S (610519104024)”** who carried out the work under my supervision.

Dr.N.Thillaikarasi M.E., Ph.D.,MIE.,
HEAD OF THE DEPARTMENT,
Department of Computer Science
and Engineering

Dhirajlal Gandhi College of
Technology
Salem – 636 309

Ms. J. Maheswari M.E.,
SUPERVISOR,
Department of Computer Science
and Engineering

Dhirajlal Gandhi College of
Technology
Salem – 636 309

Submitted for University Viva Voce Examination held on ________________.

------------------------- -------------------------

Internal Examiner **External Examiner**

ABSTRACT

Phishing is a popular tactic among cybercriminals because it is easier to persuade someone to click a malicious link that appears to be authentic than it is to bypass computer security measures. One common approach used in phishing attacks is to create fake websites and emails that closely resemble legitimate ones. This is achieved by using logos, slogans, and other elements that are typically associated with the targeted organization or individual. When users click on the links provided in these emails, they are directed to fake websites where they are asked to provide sensitive information such as login credentials, bank account details, or other personal information. To detect phishing websites, machine learning models such as Random Forest, Decision Tree, and Multilayer Perceptron are utilized and compared for their accuracy and efficiency. The existing machine learning models used to detect phishing websites have some limitations. For example, they have low latency and lack a specific user interface. Furthermore, there is a need for an effective comparison of different algorithms used for detecting phishing websites. Machine learning models are used to detect phishing websites, but there is a need for improvement in their accuracy, efficiency, and user interface. It is essential to stay vigilant and double-check the authenticity of emails, websites, and communication channels to avoid falling victim to phishing attacks.

திட்டப்பணிசுருக்கம்

ஃபிஷிங் என்பது சைபர் கிரைமினல்கள் மத்தியில் பிரபலமான ஒரு தந்திரமாகும், ஏனெனில் கணினி பாதுகாப்பு நடவடிக்கைகளை புறக்கணிப்பதை விட நம்பகத்தன்மை கொண்டதாக தோன்றும் தீங்கிழைக்கும் இணைப்பைக் கிளிக் செய்ய ஒருவரை வற்புறுத்துவது எளிது. ஃபிஷிங் தாக்குதல்களில் பயன்படுத்தப்படும் ஒரு பொதுவான அணுகுமுறை, சட்டப்பூர்வமானவற்றை ஒத்திருக்கும் போலி இணையதளங்கள் மற்றும் மின்னஞ்சல்களை உருவாக்குவதாகும். லோகோக்கள், முழக்கங்கள் மற்றும் இலக்கு அமைப்பு அல்லது தனிநபருடன் பொதுவாக தொடர்புடைய பிற கூறுகளைப் பயன்படுத்துவதன் மூலம் இது அடையப்படுகிறது. இந்த மின்னஞ்சல்களில் கொடுக்கப்பட்டுள்ள இணைப்புகளை பயனர்கள் கிளிக் செய்யும் போது, உள்நுழைவு சான்றுகள், வங்கிக் கணக்கு விவரங்கள் அல்லது பிற தனிப்பட்ட தகவல்கள் போன்ற முக்கியமான தகவல்களை வழங்குமாறு கேட்கப்படும் போலி இணையதளங்களுக்கு அவர்கள் அனுப்பப்படுவார்கள். ஃபிஷிங் இணையதளங்களைக் கண்டறிய, ரேண்டம் ஃபாரஸ்ட், டெசிஷன் ட்ரீ மற்றும் மல்டிலேயர் பெர்செப்ட்ரான் போன்ற இயந்திர கற்றல் மாதிரிகள் பயன்படுத்தப்பட்டு அவற்றின் துல்லியம் மற்றும் செயல்திறனுக்காக ஒப்பிடப்படுகின்றன.

ஃபிஷிங் இணையதளங்களைக் கண்டறிய தற்போதுள்ள இயந்திர கற்றல் மாதிரிகள் சில வரம்புகளைக் கொண்டுள்ளன. எடுத்துக்காட்டாக, அவை குறைந்த தாமதத்தைக் கொண்டுள்ளன மற்றும் குறிப்பிட்ட பயனர் இடைமுகம் இல்லை. மேலும், ஃபிஷிங் இணையதளங்களைக் கண்டறிவதற்குப் பயன்படுத்தப்படும் பல்வேறு அல்காரிதங்களின் திறம்பட ஒப்பிட வேண்டிய அவசியம் உள்ளது. ஃபிஷிங் இணையதளங்களைக் கண்டறிய இயந்திர கற்றல் மாதிரிகள் பயன்படுத்தப்படுகின்றன, ஆனால் அவற்றின் துல்லியம், செயல்திறன் மற்றும் பயனர் இடைமுகம் ஆகியவற்றில் முன்னேற்றம் தேவை. ஃபிஷிங் தாக்குதல்களுக்கு பலியாகாமல் இருக்க விழிப்புடன் இருப்பது மற்றும் மின்னஞ்சல்கள், இணையதளங்கள் மற்றும் தகவல் தொடர்பு சேனல்களின் நம்பகத்தன்மையை இருமுறை சரிபார்ப்பது அவசியம்.

ACKNOWLEDGEMENT

Dedicating this project to the **ALMIGHTY GOD** whose abundant grace and mercy enabled the project's successful completion

We express our deep gratitude to our beloved **Shri. Dhirajlal A. Gandhi,** Founder Chairman, Dhirajlal Gandhi College of Technology, who gave us the golden opportunity to do this wonderful project.

We convey our sincere thanks to **Shri. D. Manoj Kumar,** Chairman, Dhirajlal Gandhi College of Technology, who supports us in all our endeavors.

We owe a genuine gratitude to **Smt. Archana Manojkumar, M.E.,** Secretary, for providing support for this project.

We are greatly indebted to thank **Dr. A. Selvaraj, M.E., Ph.D, MISTE.,**principal, for providing support for this project.

We express our warm thanks to **Dr. N. Thillaikarasi M.E., Ph.D., MIE.,** Professor and Head, Department of Computer Science and Engineering, for her encouragement and inspiration.

We are grateful to **Ms.J .Maheswari M.E.,** Assistant Professor, Department of Computer Science and Engineering, our supervisor, for her guidance and encouragement.

We are greatly thankful to **Ms. P. Vimala, M.E.,** Assistant Professor and **Mr. S. Sankar, M.E.,** Assistant Professor, Department of Computer Science and Engineering, our project coordinators, for their valuable support and guidance along the project.

We are grateful to **Ms. N .Radha, M.E.,** Assistant Professor and to **Mr**. **R. Makendran,M.E.,** Assistant professor, Department of Computer Science and Engineering, our class Advisors, for their guidance and support.

Special thanks to our Faculty members, Supporting Staffs, Friends, our beloved Family and to those who are not mentioned above for the endless support given to us.

DHIVYA J

(610519104019)

AMSAVENI S M

(610519104003)

GAYATHRI S

(610519104024)

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
1	**INTRODUCTION**	1
	1.1 Machine Learning	
	1.2 Purpose of Machine learning	
	1.3 Description of the project	
	1.4 Problem Analysis	
2	**LITERATURE SURVEY**	7
3	**SYSTEM ANALYSIS**	13
	3.1 Existing Solution	
	3.1.1 Disadvantages	
	3.2 Proposed Solution	
	3.2.1 Advantages	
4	**SYSTEM SPECIFICAION**	16
	4.1 Software Specification	
	4.1.1 Python	
	4.1.2 Flask	
	4.1.3 Tensor Flow	

4.1.4 Keras

4.1.5 Ubuntu

5 **FEASIBILITY STUDY** 22

5.1 Economical Feasibility

5.2 Technical Feasibility

5.3 Operational Feasibility

6 **SYSTEM DESIGN** 26

6.1 Data flow diagram

6.1.1 System Architecture

6.1.2 Use case Diagram

6.1.3 Flow Diagram

6.2MODULE DESCRIPTION

6.2.1 Data collection

6.2.1 Data preprocessing

6.2.3 Feature Extraction

6.2.4 Model Training

6.2.5 User Web Interface

7 **SYSTEM TESTING** 35

7.1 Types Of Testing

7.1.1 Unit Testing

7.1.2 Integration Testing

7.1.3 Validation Testing

7.1.4 Output Testing

7.1.5 System Testing

7.1.6 Performance Testing

7.1.7 Procedure Testing

8 CONCLUSION AND FUTURE ENHANCEMENT 39

8.1 Conclusion

8.2 Future Enhancement

APPENDIX I

Source Code

APPENDIX II

Screenshots

REFERENCES

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
6.1.1	SYSTEM ARCHITECTURE	26
6.1.1	FLOW DIAGRAM	27
6.1.2	USE CASE DIAGRAM	28
6.2.1	DATA COLLECTION	30
6.2.2	DATA PREPROCESSING	31
6.2.3	FEATURE EXTRACTION	32
6.2.4	MODEL TRAINING	33
6.2.5	USER WEB INTERFACE	34
A.2.1	WEB PAGE	47
A.2.2	URL PAGE	47
A.2.3	URL ENTER PAGE	48
A.2.4	RESULT PAGE	48

LIST OF ABBREVIATIONS

ML	MACHINE LEARNING
MLP	MULTI LAYER PERCEPTRON
HTML	HYPERTEXTMARKUP LANGUAGE
CSS	CASCADING STYLE SHEETS
URL	UNIFORM RESOURCE LOCATOR
HTTP	HYPER TEXT TRANSFER PROTOCOL
HTTPS	HYPER TEXT TRANSFER PROTOCOL SECURE

CHAPTER-1

INTRODUCTION

In today’s fast-paced world technology has become an essential part ofeveryone’s life. Technology has been greatly escalating and thereby making our experiences comfortable. Nowadays our presence and business havebeen dependent on the internet and various online platforms. People perform various activities in their day-to-day life that includes accessing online shopping websites, banking websites, educational websites, and social media. Nonetheless, all these websites ask for our data and some of them consist of sensitive information that may be bank details or card details. And as a result of all this, hackers have found an easy way of attacking other personal information and tracking their behaviour. There are many types of attacks including Man-in-the-middle Attacks, Dos Attacks, SQL injection, Phishing Attacks, and many more. Out of all these websites, phishing has been considered a great threat to a user’s vital information. The social engineering trick isused to manipulate the users and thereby duping them with the legitimate-looking URL which is a fake URL. It is difficult for a naive user to spot whether the URLis legitimate or fake. Also, we have made use of the Random Forest Algorithm due to its high accuracy, robustness, and good performance. And based on characteristic classification the system will differentiate the provided URL and will conclude whether the given URL is legitimate or a phishing URL. By which the user will be able to figure out that he might endanger his information if he visits that particular URL. And hence this system helps in guarding and thereby providing a possible solution towards the issue.

1.1 Machine Learning

Machine learning is a branch of artificial intelligence (AI) that focuses on the development of algorithms and models capable of learning from data and making predictions or decisions without being explicitly programmed. It involves the construction and training of mathematical models that can automatically improve their performance over time through exposure to large volumes of data.

There are several key components and concepts in machine learning:

1. Training Data: Machine learning models require a substantial amount of labeled training data to learn patterns and relationships. The training data consists of input features (also known as variables or attributes) and corresponding target labels or outcomes.

2. Feature Extraction: Feature extraction involves selecting or transforming the relevant information from raw data that will be used as input for the machine learning model. This step helps in representing the data in a format suitable for analysis.

3. Model Representation: Machine learning models can take different forms, such as decision trees, neural networks, support vector machines (SVM), or probabilistic graphical models. Each model represents a hypothesis space that captures the potential patterns and relationships in the data.

4. Model Training: During the training phase, the machine learning model is presented with the labeled training data. It learns from the data by adjusting its internal parameters or weights to minimize the difference between predicted outputs and the true labels. This process is typically achieved using optimization algorithms like gradient descent.

Machine learning can be categorized into various types, including:

Supervised Learning: Models are trained using labeled data, with input features and corresponding target labels. The model learns to map input features to the correct output based on the provided examples.

- Unsupervised Learning: Models learn patterns and relationships in unlabeled data without explicit target labels. Common techniques include clustering, dimensionality reduction, and anomaly detection.

- Semi-Supervised Learning: This approach combines labeled and unlabeled data to train the model. It leverages the structure of unlabeled data to improve performance, especially when labeled data is scarce or expensive to obtain.

- Reinforcement Learning: Agents learn through interactions with an environment, receiving rewards or penalties based on their actions. The model learns to optimize its decision-making process by maximizing cumulative rewards.

1.2 Purpose of Machine learning

The purpose of machine learning is to enable computers and systems to learn from data and improve their performance without being explicitly programmed. By developing algorithms and models that can automatically detect patterns, relationships, and trends in data, machine learning empowers systems to make accurate predictions, take informed decisions, and solve complex problems.

Machine learning plays a crucial role in extracting valuable insights and knowledge from large volumes of data, often referred to as "big data." It enables businesses, organizations, and individuals to leverage the potential hidden within their data to gain a competitive advantage, optimize processes, enhance decision-making, and drive innovation.

Machine learning has broad applicability across various domains and industries. It is used in image and speech recognition, natural language processing, recommendation systems, fraud detection, financial forecasting, healthcare diagnostics, autonomous vehicles, robotics, and more. Its purpose is to enable systems to adapt and improve their performance over time, making them more intelligent, efficient, and capable of handling complex tasks.

Overall, the purpose of machine learning is to harness the power of data and enable systems to learn, evolve, and make accurate predictions or decisions, thereby driving advancements, innovation, and efficiency in various fields.

1.3 Description of the project

The detection of phishing websites involves the process of identifying and mitigating websites that aim to deceive users and extract sensitive information. It encompasses various techniques and strategies to identify the telltale signs of phishing attempts and protect users from falling victim to these fraudulent websites.

Detection methods often include analyzing the website's URL for suspicious elements such as misspellings, additional characters, or deviations from legitimate domain names. This scrutiny helps identify websites that mimic trusted brands or organizations to trick users into revealing their confidential data.

SSL certificate verification is another important aspect of detection. Phishing websites often use self-signed or expired certificates, which can be identified during the verification process. Ensuring the presence of a valid and trusted SSL certificate helps users verify the authenticity and security of a website.

By employing a combination of URL analysis, content analysis, SSL certificate verification, reputation services, behavioral analysis, machine learning, and user feedback, detection systems aim to identify and mitigate phishing websites, safeguarding users' sensitive information and enhancing online security. Regular updates and continuous research are essential to stay ahead of evolving phishing techniques and provide robust protection against these threats.

1.4 Problem Analysis

The detection of phishing websites faces several challenges and problems that can impact its effectiveness. Analyzing these issues is crucial for understanding the limitations and areas for improvement in the current detection systems.

One significant problem is the presence of false positives and false negatives. False positives occur when legitimate websites are incorrectly flagged as phishing sites, causing inconvenience and potential mistrust among users. False negatives, on the other hand, happen when sophisticated phishing attacks go undetected, leaving users vulnerable to scams.

Another problem lies in the evolving nature of phishing techniques. Attackers continuously adapt their methods to evade detection. New phishing tactics, such as polymorphic phishing, which dynamically alters the content or URLs of websites, pose a challenge for static analysis or blacklisting mechanisms. Phishing websites can employ obfuscation techniques to hide their malicious intent. Encryption, obfuscated code, or camouflage techniques make it difficult for existing detection systems to analyze the content and accurately identify phishing indicators.

User education and awareness are also critical factors in the fight against phishing, but they present a challenge in the current detection systems. While these systems focus primarily on technical detection mechanisms, addressing the human factor through user education is equally important.

CHAPTER 2

LITERATURE SURVEY

TITLE:A systematic literature review on phishing website detection techniques

AUTHORS: Asadullah Safi [a], Satwinder Singh

PUBLISHED: Feb 2023

Phishing is a fraud attempt in which an attacker acts as a trusted person or entity to obtain sensitive information from an internet user. In this Systematic Literature Survey (SLR), different phishing detection approaches, namely Lists Based, Visual Similarity, Heuristic, Machine Learning, and Deep Learning based techniques, are studied and compared. For this purpose, several algorithms, data sets, and techniques for phishing website detection are revealed with the proposed research questions. A systematic Literature survey was conducted on 80 scientific papers published in the last five years in research journals, conferences, leading workshops, the thesis of researchers, book chapters, and from high-rank websites. The work carried out in this study is an update in the previous systematic literature surveys with more focus on the latest trends in phishing detection techniques. This study enhances readers' understanding of different types of phishing website detection techniques, the data sets used, and the comparative performance of algorithms used. Machine Learning techniques have been applied the most, i.e., 57 as per studies, according to the SLR.

TITLE: Machine learning approach for phishing website detection

AUTHORS: Rutuja R. Patil,Gagandeep Kaur,Himank Jain,Ayush Tiwari,Soham Joshi,Keshav Rao

PUBLISHED: 14 Jun 2022

The past year saw our world afflicted by COVID-19 undergo a digital transformation which led to a majority of people and organizations gravitate towards the internet. A remote working environment complicated the pre-existent crisis of phishing where the vulnerable population incurred huge losses at the hands of internet miscreants. A phishing attack comprises an attacker that creates fake websites to fool users and steal client-sensitive data which may be in form of login, password, or credit card details. Timely detection of phishing attacks has become more crucial than ever. Hence in this paper, we provide a thorough literature survey of the various machine learning methods used for phishing detection. This thesis will discuss in detail, different approaches used by various authors over the past few years. This survey aims to identify and narrow down the best machine learning algorithms that can be adopted to develop a hybrid model which can be implemented to detect whether a website is legitimate or phishing in nature.

TITLE:A Comprehensive Survey on Identification and Analysis of Phishing Website based on Machine Learning Methods

AUTHORS: Mohammed HazimAlkawaz; Stephanie Joanne Steven; Asif Iqbal Hajamydeen;

PUBLISHED: May 2021

Phishing is a cybercrime which is carried out by imitating a legal website to trick users to steal their personal data, including usernames, passwords, account numbers, national insurance numbers, etc. Phishing frauds may be the most widespread cybercrime used today. Machine learning focuses on computer algorithms which improves automatically through experience. Machine learning methods were utilized to detect phishing URLs that typically evaluates an URL based on a feature or set of features extracted from it. This paper presents an approach to identify phishing websites using trained machine learning models. It also delivers a detailed analysis of phishing attacks with a comparison on machine learning approaches used for analysis and classification of phishing and legitimate websites

.

TITLE:A Survey of Machine Learning-Based Solutions for Phishing Website Detection

AUTHORS: Lizhen Tang and Qusay H. Mahmoud

PUBLISHED:20 August 2021

With the development of the Internet, network security has aroused people's attention. It can be said that a secure network environment is a basis for the rapid and sound development of the Internet. Phishing is an essential class of cybercriminals which is a malicious act of tricking users into clicking on phishing links, stealing user information, and ultimately using user data to fake logging in with related accounts to steal funds. Network security is an iterative issue of attack and defense. The methods of phishing and the technology of phishing detection are constantly being updated. Traditional methods for identifying phishing links rely on blacklists and whitelists, but this cannot identify new phishing links. Therefore, we need to solve how to predict whether a newly emerging link is a phishing website and improve the accuracy of the prediction. With the maturity of machine learning technology, prediction has become a vital ability. This paper offers a state-of-the-art survey on methods for phishing website detection. It starts with the life cycle of phishing, introduces common anti-phishing methods, mainly focuses on the method of identifying phishing links, and has an in-depth understanding of machine learning-based solutions, including data collection, feature extraction, modeling, and evaluation performance. This paper provides a detailed comparison of various solutions for phishing website detection.

TITLE:Phishing Detection using Random Forest, SVM and Neural Network with Backpropagation

AUTHORS: Smita Sindhu; Sunil ParameshwarPatil; Arya Sreevalsan;

PUBLISHED: October 2020

Phishing is a common attack used to obtain sensitive information using visually similar websites to that of legitimate websites. With the growing technology, phishing attacks are on the rise. Machine Learning is a very popular approach to detect phishing websites. This paper explains the existing machine learning methods that are used to detect phishing websites. The paper explains the improved Random Forest classification method, SVM classification algorithm and Neural Network with backpropagation classification methods which have been implemented with accuracies of 97.369%, 97.451% and 97.259% respectively.

TITLE: A comprehensive survey of AI-enabled phishing attacks detection techniques

AUTHORS: Abdul Basit, Maham Zafar, Xuan Liu, Abdul RehmanJaved, ZuneraJalil & KashifKifayat

PUBLISHED: 23 October 2020

In recent times, a phishing attack has become one of the most prominent attacks faced by internet users, governments, and service-providing organizations. In a phishing attack, the attacker(s) collects the client's sensitive data (i.e., user account login details, credit/debit card numbers, etc.) by using spoofed emails or fake websites. Phishing websites are common entry points of online social engineering attacks, including numerous frauds on the websites. In such types of attacks, the attacker(s) create website pages by copying the behavior of legitimate websites and sends URL(s) to the targeted victims through spam messages, texts, or social networking. To provide a thorough understanding of phishing attack(s), this paper provides a literature review of Artificial Intelligence (AI) techniques: Machine Learning, Deep Learning, Hybrid Learning, and Scenario-based techniques for phishing attack detection. This paper also presents the comparison of different studies detecting the phishing attack for each AI technique and examines the qualities and shortcomings of these methodologies. Furthermore, this paper provides a comprehensive set of current challenges of phishing attacks and future research direction in this domain.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SOLUTION

In recent times, machine learning has played a pivotal role in the detection of phishing websites. Advanced machine learning algorithms have been developed and applied to analyze various features and patterns inherent in website content and behavior. These algorithms can effectively learn from vast amounts of data, enabling them to accurately identify and classify potential phishing sites. By analyzing factors such as URL structure, content analysis, website reputation, SSL certificate validity, and user behavior, machine learning models can identify subtle indicators and similarities with known phishing patterns. This approach allows for real-time detection and proactive mitigation of phishing attempts. The continuous evolution and refinement of machine learning models contribute to their increasing effectiveness in identifying sophisticated and previously unseen phishing techniques. By leveraging the power of machine learning, organizations and individuals can enhance their defenses against phishing attacks and safeguard sensitive information from falling into the wrong hands.

3.1.1 Disadvantage

While existing systems for detecting phishing websites have made significant progress, they still face certain limitations and disadvantages. Here are some common disadvantages of these systems:

1. False Positives and False Negatives: Existing systems may generate false positives, flagging legitimate websites as phishing sites, or false negatives, failing to identify sophisticated phishing attacks. False positives can lead to inconvenience for users, while false negatives pose a risk to their security.

2. Evolving Phishing Techniques: Phishing techniques constantly evolve, with attackers finding new ways to deceive users. Existing systems may struggle to keep

up with these evolving tactics and may not accurately detect newly developed phishing websites.

3. Polymorphic Phishing: Phishing websites can employ polymorphic techniques, where the website's content or URL is dynamically changed to evade detection. Existing systems relying on static analysis or blacklisting may struggle to identify these dynamically altered phishing websites.

It's important to continually enhance and update existing systems to address these disadvantages and keep pace with the evolving nature of phishing attacks. Combining multiple detection techniques, integrating threat intelligence, and incorporating user feedback can help improve the overall effectiveness of phishing detection systems.

3.2 PROPOSED SOLUTION

A proposed system for the detection of phishing websites combines several techniques to enhance online security. This system includes URL analysis, where suspicious patterns and misspellings are identified. The content of web pages is analyzed for signs of forgery or inconsistencies.The main goal of this project is to create a domain authentication system that would detect if a given domain url is legit or fake website created to perform fraud. Multiple ML models will be tested for this problem. For this dataset MLP gave the highest accuracy (99%) with suitably balanced precision and recall. The behavior of websites is monitored for suspicious activities, such as early data requests or unexpected redirects. Blacklists and whitelists of known phishing and legitimate websites are utilized. Additionally, browser extensions are developed to provide real-time warnings to users. While no system can offer absolute accuracy, employing a combination of these techniques and regular updates can significantly improve phishing website detection.

3.2.1 Advantages of Proposed system

The proposed system for the detection of phishing websites offers several advantages over existing methods, enhancing the overall effectiveness of detecting and mitigating phishing threats. These advantages include:

1. Improved Accuracy: The proposed system incorporates advanced techniques such as machine learning, which can learn from vast amounts of data. This improves the accuracy of detection, reducing false positives and false negatives compared to traditional rule-based systems.

2. Adaptability to Emerging Threats: By leveraging machine learning and real-time threat intelligence, the proposed system can adapt to evolving phishing techniques.

3. Integration of User Education: While technical detection mechanisms are crucial, the proposed system recognizes the importance of user education. It integrates user awareness initiatives, providing guidance and information to users about phishing techniques, common scams, and safe online practices.

Overall, the proposed system offers improved accuracy, adaptability to emerging threats, enhanced detection of targeted phishing, the ability to overcome obfuscation techniques, timely updates, integration of user education, and a focus on continuous improvement.

CHAPTER 4

SYSTEM SPECIFICATION

Software Requirements:

Operating System	: Linux-Ubuntu
Server Side	: Python 3.11,Flask
Client Side	: HTML, CSS, Javascript
Libraries	: TensorFlow,keras,regex
Platform	: Sublime Text

Hardware requirements:

CPU type	: Intel Pentium 4
Clock speed	: 3.0 GHz
Ram size	: 512 MB
Hard disk capacity	: 40 GB
Monitor type	: 15 Inch color monitor
Keyboard type	: internet keyboard

4.1 Software Specification

4.1.1 Python

Python is a versatile and widely used programming language known for its simplicity and readability. Created by Guido van Rossum and released in 1991, Python has gained popularity for its clean and expressive syntax. The language emphasizes code readability by using indentation and whitespace to define code blocks, making it easier to write and understand code.

One of the notable features of Python is its support for multiple programming paradigms, including procedural, object-oriented, and functional programming. This flexibility allows developers to choose the approach that best suits their needs. Python's extensive standard library provides modules for a wide range of purposes, such as file I/O, networking, web development, and more. This eliminates the need for external dependencies and enables developers to accomplish complex tasks efficiently. The language's popularity is further amplified by its rich ecosystem of third-party libraries and frameworks. Python offers numerous powerful libraries for tasks like data analysis (e.g., NumPy and Pandas), web development (e.g., Django and Flask), machine learning (e.g., TensorFlow and scikit-learn), and much more

Python has a large and active community of developers who contribute to its growth and development.With its wide range of applications, including web development, data analysis, scientific computing, machine learning, automation, and scripting, Python has become a popular choice for both beginners and experienced developers. Its simplicity, versatility, and extensive ecosystem make it a powerful language for various programming tasks.

4.1.2 Flask

Flask is a lightweight and popular web framework for Python that enables developers to build web applications quickly and efficiently. It was developed by Armin

Ronacher and released in 2010. Flask is known for its simplicity, minimalistic design, and flexibility, making it an excellent choice for small to medium-sized projects.

One of the key features of Flask is its simplicity. The framework provides a straightforward and intuitive interface that allows developers to get started quickly. Flask follows a minimalist approach, providing only the essential tools needed for web development, while allowing developers to add additional functionality through extensions.

Flask is built on the Werkzeug WSGI toolkit and the Jinja templating engine. Werkzeug provides a comprehensive set of tools for handling HTTP requests, routing, and other web-related tasks

Flask is commonly used in the development of small to medium-sized web applications, RESTful APIs, and prototypes. Its simplicity and minimalistic design make it a preferred choice for projects that do not require the complexity of larger frameworks. Flask's lightweight nature also makes it well-suited for deploying applications to cloud platforms or containers.

Overall, Flask is a powerful and flexible web framework that simplifies the process of building web applications in Python. Its simplicity, minimalistic design, and extensive ecosystem of extensions make it a popular choice among developers seeking a lightweight framework with the flexibility to customize and scale their applications.

4.1.3 Tensorflow

TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem for building and deploying machine learning models. TensorFlow is designed to be efficient, scalable, and flexible, making it suitable for a wide range of applications, from research and prototyping to production-level deployments.

At its core, TensorFlow represents computations as data flow graphs. In this graph-based model, nodes represent mathematical operations, and edges represent the flow of data between operations.

One of the key features of TensorFlow is its ability to handle large-scale datasets and complex models. It provides powerful abstractions and APIs for creating and training deep neural networks, which are popular for tasks such as image and speech recognition, natural language processing, and more. TensorFlow's extensive collection of pre-built neural network layers and models, called TensorFlow Hub, makes it easier for developers to leverage existing architectures and accelerate their development process.

Another noteworthy feature of TensorFlow is its support for distributed computing. It allows users to distribute computations across multiple devices, machines, or even clusters, which is crucial for training large-scale models on massive datasets. Overall, TensorFlow is a versatile and powerful framework for machine learning and deep learning. Its efficient computational model, scalability, distributed computing support, and extensive ecosystem make it a preferred choice for developing and deploying state-of-the-art machine learning models and systems.

4.1.4 Keras

Keras is a high-level neural networks API written in Python. It is designed to be user-friendly, modular, and extensible, making it a popular choice for building and experimenting with deep learning models. Keras provides a simple and intuitive interface to create, train, and evaluate various types of artificial neural networks, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and more.

One of the key features of Keras is its focus on enabling fast experimentation. It offers a wide range of pre-built layers, activation functions, optimizers, and loss functions that can be easily combined to construct complex neural network architectures. Additionally, Keras seamlessly integrates with popular deep learning libraries such as TensorFlow, allowing users to take advantage of the efficient computation capabilities provided by TensorFlow while enjoying the simplicity and ease-of-use of the Keras API.

Keras supports both sequential and functional model architectures. Sequential models are created by stacking layers on top of each other in a linear manner, while functional models allow for more complex network designs with multiple inputs and outputs, shared layers, and branching architectures.

Overall, Keras provides a high-level abstraction for building deep learning models, enabling researchers and practitioners to quickly prototype and experiment with different architectures, while also offering the necessary tools for fine-tuning and optimizing models for specific tasks.

4.1.5 Ubuntu

Ubuntu is an open-source operating system based on the Linux kernel. It is one of the most popular and widely used distributions of Linux, known for its user-friendly interface, stability, and security. Ubuntu is developed by Canonical Ltd. and is available for free.

Ubuntu offers a versatile computing environment suitable for desktops, laptops, servers, and cloud-based systems. It provides a robust and customizable user interface called the Unity desktop environment (until version 17.04) and GNOME (starting from version 17.10). The interface is designed to be intuitive and user-friendly, making it accessible to both beginners and experienced users.

One of the notable features of Ubuntu is its package management system, which allows users to easily install, update, and remove software applications from a vast repository of packages. The APT (Advanced Package Tool) package management system ensures efficient and reliable package installation, dependency resolution, and system updates.

Moreover, Ubuntu emphasizes open-source principles, promoting the use of free and open software, and encouraging the sharing of knowledge and collaboration within the Linux community.

In summary, Ubuntu is a versatile and user-friendly operating system based on Linux, offering a stable and secure environment with a vast selection of software applications. It is known for its ease of use, extensive community support, and commitment to open-source principles.

CHAPTER 5

FEASIBILITY STUDY

A feasibility study on the detection of phishing websites involves evaluating the practicality and viability of creating a system or solution to identify and prevent phishing attacks. The study considers several key factors. First, it assesses the technical feasibility by examining available technologies and techniques for phishing website detection, such as algorithms, machine learning approaches, and heuristics. It determines if reliable methods exist to distinguish legitimate websites from phishing ones. Next, the availability of relevant datasets for training and evaluating the detection system is considered. Comprehensive datasets of known phishing websites, legitimate websites, and potentially malicious URLs are crucial for effective development and validation.

Additionally, the implementation effort is assessed, including the necessary technical expertise, time requirements for development and deployment, and any potential financial implications. Legal and ethical considerations are also taken into account, ensuring compliance with privacy and data protection regulations when handling user data or accessing URLs.

By conducting a comprehensive feasibility study encompassing these factors, valuable insights can be gained regarding the practicality and viability of developing a phishing website detection system. This study enables informed decision-making regarding the potential benefits, challenges, and necessary steps for implementing an effective solution.

Three key considerations involved in the feasibility analysis are

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

5.1 Economical Feasibility

An economical feasibility study for the detection of phishing websites involves assessing the financial viability and cost-effectiveness of implementing a system or solution for identifying and preventing phishing attacks. Several key considerations are involved in this study. First, a cost analysis is conducted to evaluate the expenses associated with developing and implementing the detection system.

Operational costs are also assessed, encompassing the ongoing expenses of maintaining and running the detection system. These may include server hosting, data storage, software updates, and regular maintenance and monitoring activities. The study then examines the potential return on investment (ROI) by determining the financial benefits and returns. This involves considering factors such as the reduction in losses due to successful phishing attacks, potential savings in incident response and recovery costs, and the ability to attract and retain security-conscious customers or users.

A cost-savings comparison is made to evaluate the costs associated with implementing the detection system versus the potential costs of not having such a system.

By conducting a thorough economical feasibility study encompassing these factors, organizations can assess the financial viability and cost-effectiveness of implementing a phishing website detection system. This study provides insights into the potential costs, benefits, and financial implications associated with the system, aiding in informed decision-making regarding its feasibility and economic viability.

5.2 Technical Feasibility

The technical feasibility of developing a system for the detection of phishing websites involves evaluating various technical aspects to determine its practicality and viability. Several key considerations are involved in this assessment. First, an evaluation of available technologies, tools, and techniques is conducted to identify suitable options for detecting and identifying phishing websites. This includes analyzing algorithms, machine learning approaches, data analysis methods, and heuristic-based systems that are commonly used in the field.

The availability and quality of relevant datasets necessary for training and evaluating the detection system are also assessed. Obtaining comprehensive datasets of known phishing websites, legitimate websites, and potentially malicious URLs is crucial for effective model training and evaluation. Furthermore, the feasibility of selecting and processing features that can distinguish phishing websites from legitimate ones is determined. This involves analyzing various indicators such as URL structures, website content, SSL certificates, domain age, and IP reputation.

By conducting a comprehensive technical feasibility assessment considering these factors, organizations can determine the practicality and viability of developing a phishing website detection system. This evaluation helps in making informed decisions regarding the technical approach, resource requirements, and potential challenges associated with the system's development and implementation.

5.3 Operational Feasibility

Operational feasibility refers to assessing the practicality and viability of implementing a system for detecting phishing websites from an operational perspective. Several key considerations are involved in evaluating operational feasibility.

Firstly, resource availability is assessed to determine if the necessary resources for implementing and maintaining the detection system are accessible. This includes evaluating the availability of human resources with the required expertise, such as cybersecurity professionals or data analysts, as well as the availability of hardware, software, and infrastructure needed to support the system.

Risk management is an essential aspect, involving the identification and evaluation of risks associated with the system's operation. This includes considering potential false positives or false negatives, the impact of system downtime, and the effectiveness of backup and recovery procedures. Strategies to mitigate risks and ensure the system operates within acceptable risk thresholds are developed.

By conducting a comprehensive assessment of operational feasibility, organizations can determine whether the detection system can be implemented and operated effectively within their operational environment.

CHAPTER 6

SYSTEM DESIGN

6.1 DATA FLOW DIAGRAM

6.1.1 System Architecture

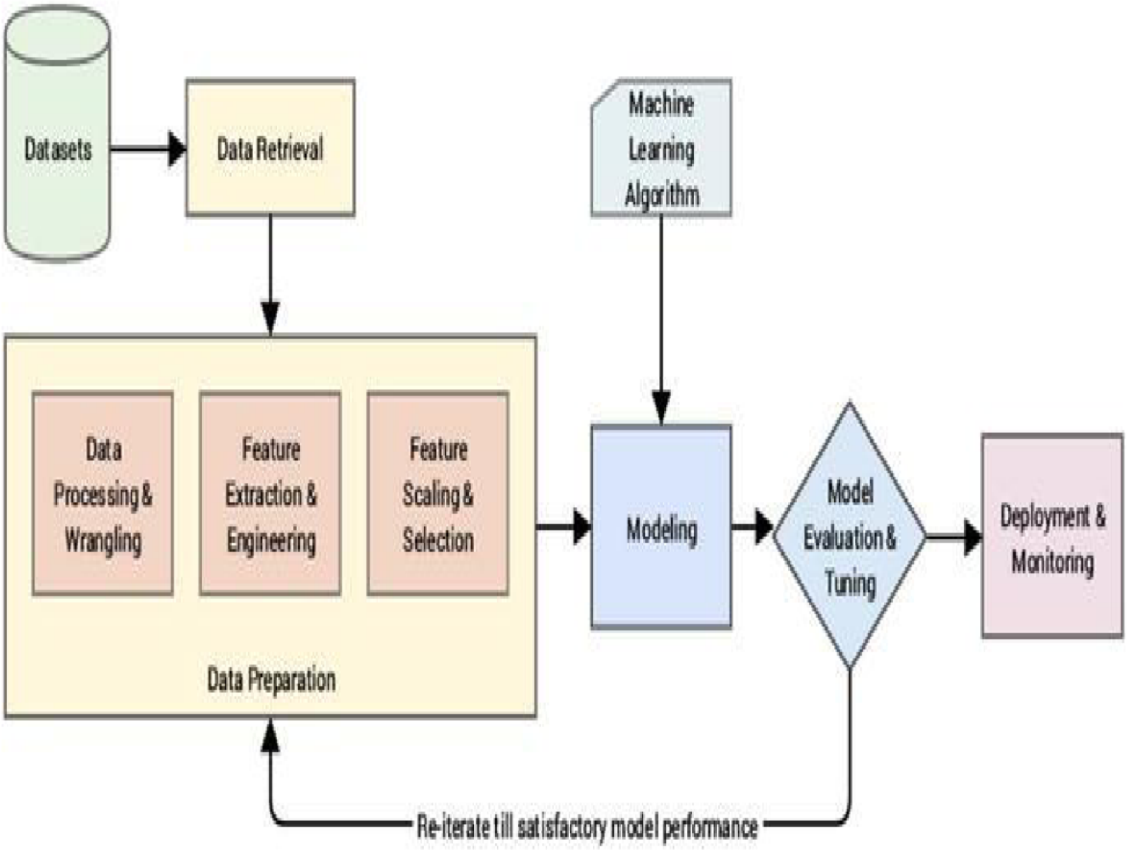

Datasets
Data Retrieval
Machine Learning Algorithm
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate till satisfactory model performance

Fig 6.1.1System architecture

6.1.1 Flow Diagram

6.1.2 Use Case Diagram

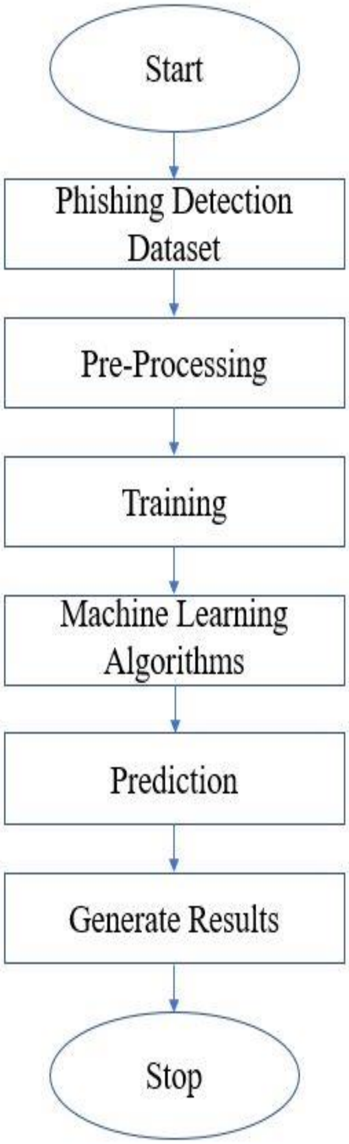


Fig 6.1.1

6.1.2 Use Case Diagram

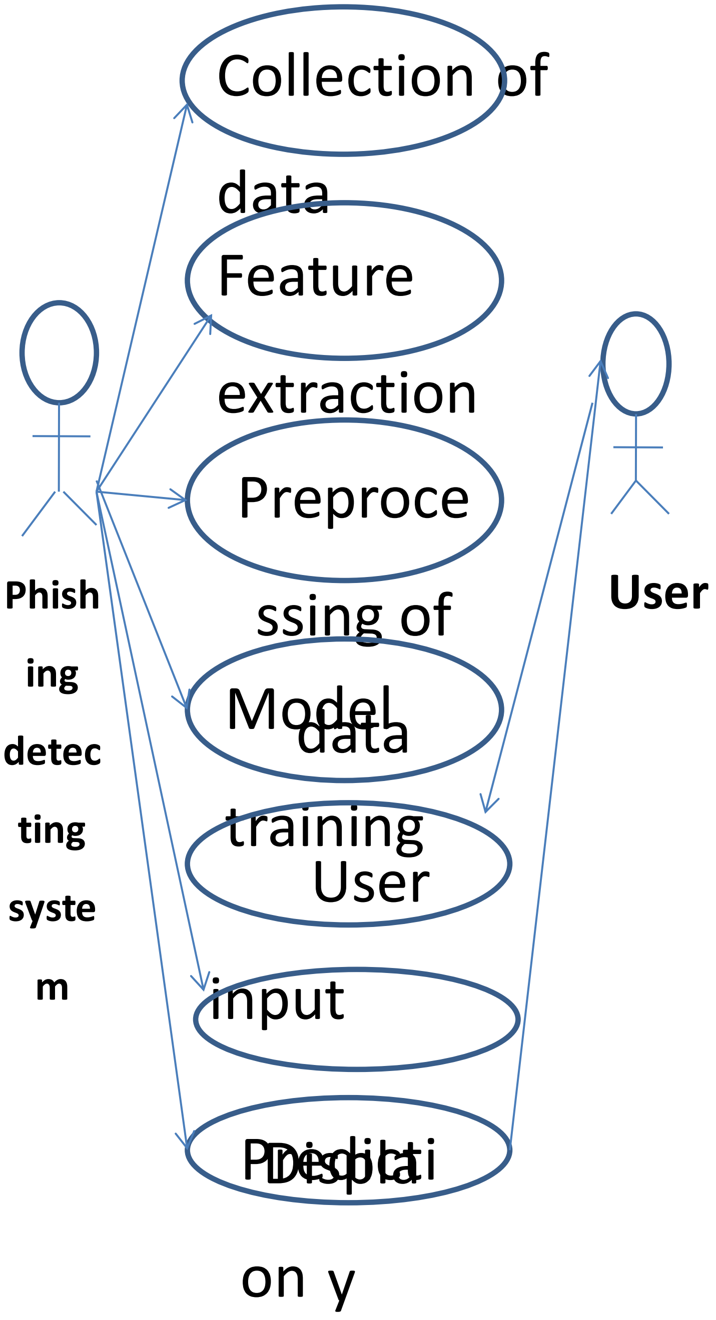

Collection of data
Feature extraction
Preprocessing of data
Model training
User input
Prediction
Display
Phishing detecting system
User

Fig 6.1.2

6.2 MODULE DESCRIPTION

- Data collection
- Feature Extraction
- Data preprocessing
- Model Training
- User Web Interface

6.2.1 Data Collection

Data collection is one of the most important steps in any machine learning workflow as the efficiency of our production model is directly proportional to the quality of our training dataset. In order to train a production ready classification model which can optimally predict malicious phishing URLs, we need to train the model on similar datasets.

Data sources: Public datasets, Kaggle datasets etc

Data collection in machine learning is a critical process that involves gathering and acquiring the necessary data to train and evaluate machine learning models. It begins with defining the data requirements based on the problem statement and objectives of the task at hand.

Overall, effective data collection forms the foundation for training accurate and robust machine learning models. It requires meticulous planning, attention to data quality, and adherence to ethical considerations. Properly collected and curated data is vital for successful machine learning applications.

Fig 6.2.1 Data Collection

6.2.2 Feature Extraction

Feature extraction in machine learning is a fundamental process that involves transforming raw data into meaningful features suitable for training machine learning models. The process begins by considering the raw data in its original form, which could be numerical values, text documents, images, audio signals, or sensor readings. From there, relevant features are selected based on domain knowledge and data exploration, ensuring they have a significant impact on the model's performance.

Once the features are chosen, they undergo transformation to prepare them for input into machine learning algorithms. This may involve scaling or normalizing numerical features to establish a consistent range, encoding categorical variables into numerical representations, or converting non-numeric data, such as text or images, into numerical feature vectors using methods like bag-of-words or deep learning embeddings.Feature extraction is an iterative process, involving experimentation, evaluation, and refinement. Different techniques are explored, and their impact on model performance is assessed, allowing for the fine-tuning of the feature set.By performing effective feature extraction, noise is reduced, model efficiency is improved, and the crucial information necessary for accurate predictions is captured. It requires a combination of domain knowledge, data exploration, and the application of appropriate techniques to select, transform, and engineer features that effectively represent the underlying patterns in the data, ultimately enhancing the performance of machine learning models.

Some of the URL features we captured from the datasets we collected are listed below :

- Length of Url
- Count of Digits
- Count of Number Of Directories
- Use of Shortening URL or not

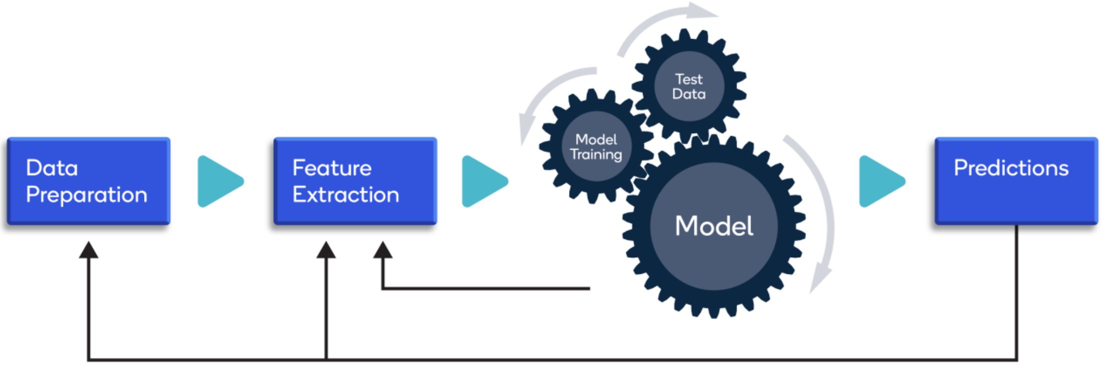


Fig 6.2.3 Feature Extraction

6.2.3 Data Preprocessing

Data preprocessing is a crucial step in machine learning (ML) that involves transforming raw data to make it suitable for training ML models and improving their performance. Several techniques are used for data preprocessing. Firstly, data cleaning addresses missing data and outliers. Missing data can be handled by removing samples or imputing them with estimated values. Outliers, extreme values that deviate significantly, can be treated by truncation or winsorization. Finally, the dataset is split into training, validation, and test sets. The training set is used for model training, the validation set aids in hyperparameter tuning and model selection, and the test set evaluates the final model's performance on unseen data. Data preprocessing is an iterative process, where different techniques are combined and applied based on the data characteristics and ML requirements. Its purpose is to ensure data is in a suitable form for training ML models, enhancing their accuracy, robustness, and ability to generalize to new data.

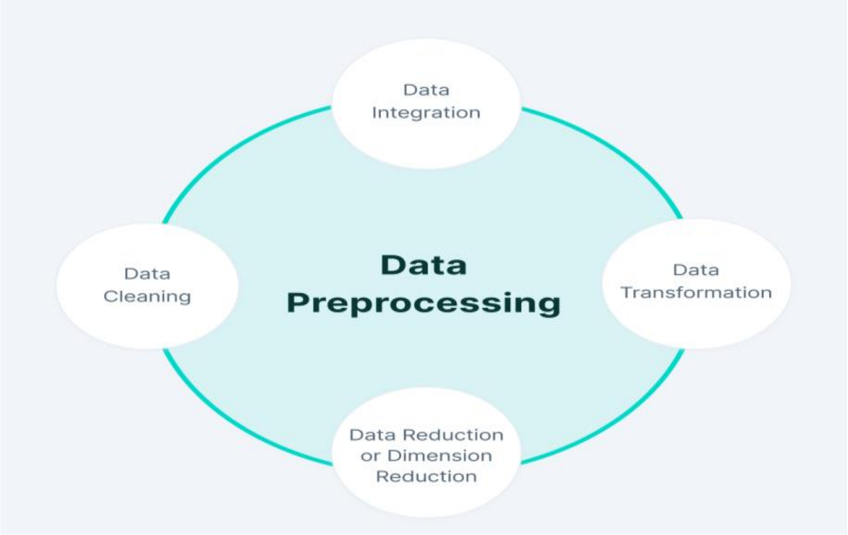


Fig 6.2.2 Data Preprocessing

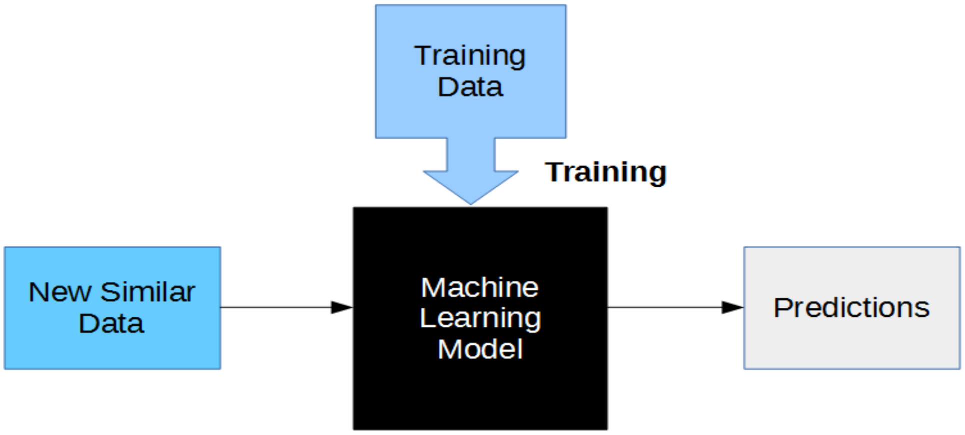
6.2.4 Model Training & Evaluation

After the data has been brought to a suitable format after processing and feature engineering, we can start with training machine learning models. Depending on the type of data given, the accuracy and performance of various machine learning algorithms can also vary.

The machine learning models considered to train the dataset in this project are the following :

- Decision Trees
- Random Forest
- Neural Network (Multilayer Perceptron)

Training a multilayer perceptron (MLP) model in machine learning involves several key steps. Initially, the training dataset needs to be prepared, which typically includes data collection, preprocessing, and splitting into training and validation sets



.

Fig 6.2.4 Model Training And Evalution

6.2.5 User Web Interface

A user web interface (UI) is the visual and interactive part of a website or web application that allows users to engage with the system. It serves as the gateway for users to access the functionalities and content provided by the web application. The primary objective of a user interface is to offer an intuitive and user-friendly experience, enabling users to navigate, perform tasks, and access information effortlessly.

A typical user web interface consists of various components. The navigation menu provides a structured layout for accessing different sections or pages of the website. The content display area presents the main content of the web application, which can include text, images, forms, and interactive elements. Forms and input fields are used to collect user inputs or trigger specific actions, while buttons and icons offer interactive elements to perform actions or access specific features.

Fig 6.2.5 User Web Interface

CHAPTER 7

SYSTEM TESTING

The System testing is stage in which is the system tested to check whether the system works accurately and efficiently before it was implemented. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

7.1 TYPES OF TESTING

- Unit Testing
- Integration Testing
- Validation Testing
- Output Testing
- System Testing
- Performance Testing
- Procedure Testing

7.1.1 Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design, the module. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by-step instructional document. However, automation is efficient for achieving this, and enables the many benefits listed in this article. Conversely, if not planned carefully, a careless manual unit test case may execute as an integration test case that involves many software components.

Thus preclude the achievement of most if not all of the goals established for unit testing. This is also known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself.

During unit testing, modules are tested in isolation:

- If all modules were to be tested together it may not be easy to determine which module has the error.

- Unit testing reduces debugging effort several folds.

- Programmers carry out unit testing immediately after they complete the coding of a module.

7.1.2 Integration Testing

Data can be lost across the interface; one module can have an adverse effect on others. Integration testing is a systematic testing for constructing program structure. While at the same time conducting tests to uncover errors associated within the interface. After the software has been integrated a set of high order sets are conducted. The objective is to take unit tested modules and combine them test it as a whole. Thus, in the integration-testing step, all the errors uncovered are corrected for the next testing steps.

After different modules of a system have been coded and unit tested

- Big Bang Approach
- Top-Down Approach
- Bottom Approach
- Mixed Approach

7.1.3 System Testing

System testing involves:

•Validating a fully developed system against its requirements.

•System testing is done against the requirements in the SRS

•This is the last phase of testing before the product is delivered

•System testing consists of three kinds

•Alpha testing

•Beta testing

•Acceptance testing

•Alpha - System testing is carried out by the test team within the developing organization.

•Beta - System testing performed by a select group of friendly customers.

•Acceptance - System testing performed by the customer himself to determine whether the system should be accepted or rejected.

7.1.4 Validation Testing

The outputs that come out of the system are as a result of the inputs that go in to the system. So, for the correct and the expected outputs the inputs that go in to the system should be correct and proper.

So, this testing is done to check if the inputs are correct and they are validated before it goes in to the system for processing.

7.1.5 Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2ways-one is on screen and another is printed format.

7.1.6 Performance Testing

Performance testing is designed to test the run-time performance of software within the context of an integrated system. It requires both hardware and software instrumentation. It is often necessary to measure resource utilization in an exacting fashion.

7.1.7 Procedure Testing

Determine the clarity of the documentation on operation and the user of the system by having users do exactly what manual request. In case of this project work system testing and unit testing are mainly used.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

In conclusion, the detection of phishing websites is of utmost importance in ensuring online security and protecting users from fraudulent activities. Phishing websites attempt to deceive users by mimicking legitimate platforms and tricking them into disclosing sensitive information. Various techniques and approaches have been developed to identify and combat these threats.

Machine learning algorithms have proven to be effective in phishing website detection by analyzing patterns, features, and characteristics of malicious websites. Features such as URL properties, domain reputation, content analysis, and user behavior can be utilized to train ML models for accurate classification.

However, it is important to acknowledge the limitations and challenges associated with phishing website detection. Phishers continually adapt and employ sophisticated techniques to evade detection systems, necessitating the constant evolution and improvement of detection methods.

Overall, the ongoing research and development in phishing website detection aim to provide users with a safer online environment. By leveraging machine learning and adopting proactive measures, we can effectively identify and mitigate the risks posed by phishing websites, safeguarding users' sensitive information and maintaining trust in online interactions.

8.2 FUTURE ENHANCEMENT

In the future, there are several potential enhancements that can be explored to improve phishing website detection:

1. Deep Learning Approaches: Deep learning techniques, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), can be further explored to extract more complex features and patterns from phishing websites. These models have the potential to capture intricate relationships in data and improve detection accuracy.

2. Adversarial Detection: Phishers constantly adapt their techniques to evade detection systems. Developing robust adversarial detection methods that can identify and counter sophisticated adversarial attacks on phishing detection models is a promising direction for future research.

3. Behavioral Analysis: By analyzing patterns in user interactions, mouse movements, or browsing behavior, it is possible to identify suspicious activities and detect potential phishing attempts.

4. Continuous Learning and Adaptability: Phishing techniques evolve over time, so detection systems should continuously learn and adapt. Employing online learning techniques that can update detection models in real-time based on new data and emerging patterns can significantly improve detection performance.

By exploring these future enhancements, the field of phishing website detection can evolve and develop more robust and effective systems to combat the ever-evolving threat of phishing attacks, ultimately safeguarding users and their sensitive information in the digital landscape.

APPENDIX –I

SOURCE CODE

```
from flask import Flask, render_template,request
fromtensorflow import keras
import re
fromurllib.parse import urlparse
app = Flask(__name__)
@app.route("/")
def main():
returnrender_template("index.html")
@app.route("/result",methods = ["POST","GET"])
def result():
ifrequest.method == "POST":
globalurl
url = request.form.get("url")
status = "Please enter valid Url"
model_path = r"Malicious_URL_Prediction.h5"
if url:
prediction = get_prediction(url,model_path)
value = f"This Website is {prediction}% legit"
returnrender_template("result.html",value=value)
returnrender_template("index.html",status=status)

defget_prediction(url, model_path):

model = keras.models.load_model(model_path)

url_features = extract_features(url)

prediction = model.predict([url_features])

i = prediction[0][0] * 100

i = round(i,3)

    v = 100 - i

    v = round(v,3)

return v

defextract_features(url):

url_features = []

i = hostname_length(url)

url_features.append(i)

i = url_length(url)

url_features.append(i)

i = fd_length(url)

url_features.append(i)

i = get_counts(url)

url_features = url_features + i

i = digit_count(url)
```

```
url_features.append(i)

i = letter_count(url)

url_features.append(i)

i = no_of_dir(url)

url_features.append(i)

i = having_ip_address(url)

url_features.append(i)


returnurl_features

deffd_length(url):

urlpath = urlparse(url).path

try:

returnlen(urlpath.split('/')[1])

except:

return 0

defdigit_count(url):

digits = 0

fori in url:

ifi.isnumeric():

digits = digits + 1

return digits
```

```
defletter_count(url):

letters = 0

fori in url:

ifi.isalpha():

letters = letters + 1

return letters

defno_of_dir(url):

urldir = urlparse(url).path

returnurldir.count('/')

defhaving_ip_address(url):

match = re.search(

        '(([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.'

        '([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\/)|'  # IPv4

        '((0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\/)'

        '(?:[a-fA-F0-9]{1,4}:){7}[a-fA-F0-9]{1,4}', url)  # Ipv6

if match:

return -1

else:

return 1
```

```
defhostname_length(url):

returnlen(urlparse(url).netloc)

defurl_length(url):

returnlen(urlparse(url).path)

defget_counts(url):

count_features = []

i = url.count('-')

count_features.append(i)

i = url.count('@')

count_features.append(i)

i = url.count('?')

count_features.append(i)

i = url.count('%')

count_features.append(i)

i = url.count('.')

count_features.append(i)


i = url.count('=')

count_features.append(i)

i = url.count('http')

count_features.append(i)
```

```
i = url.count('https')

count_features.append(i)

i = url.count('www')

count_features.append(i)

returncount_features

if __name__ == "__main__":

app.run(debug=True)
```

APPENDIX II

SCREENSHOTS

Web page:

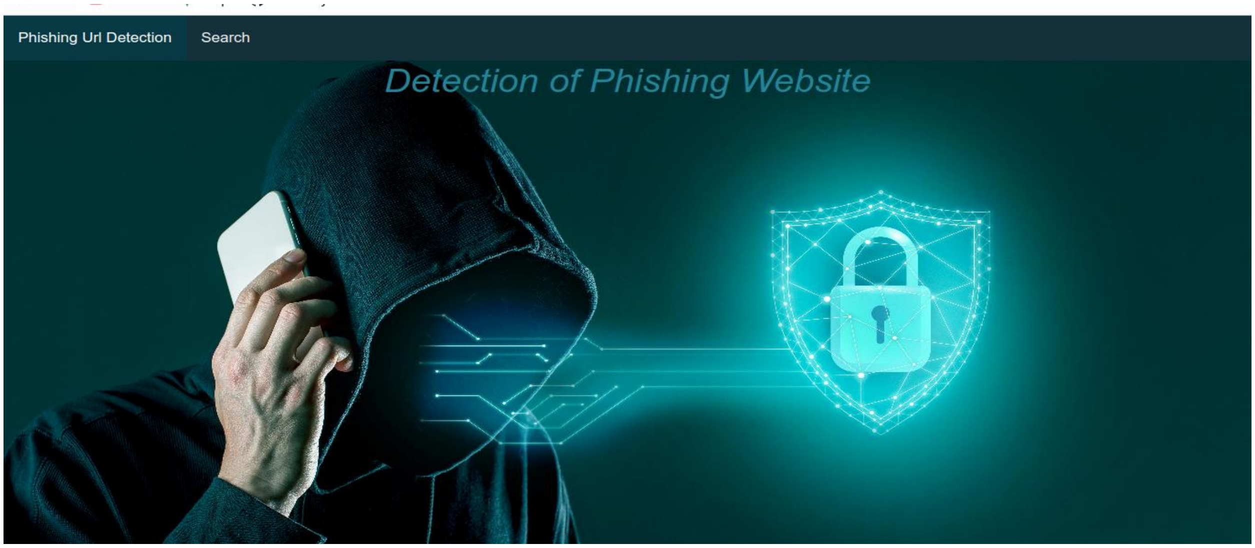


Fig A.2.1 Web Page

URL Page:

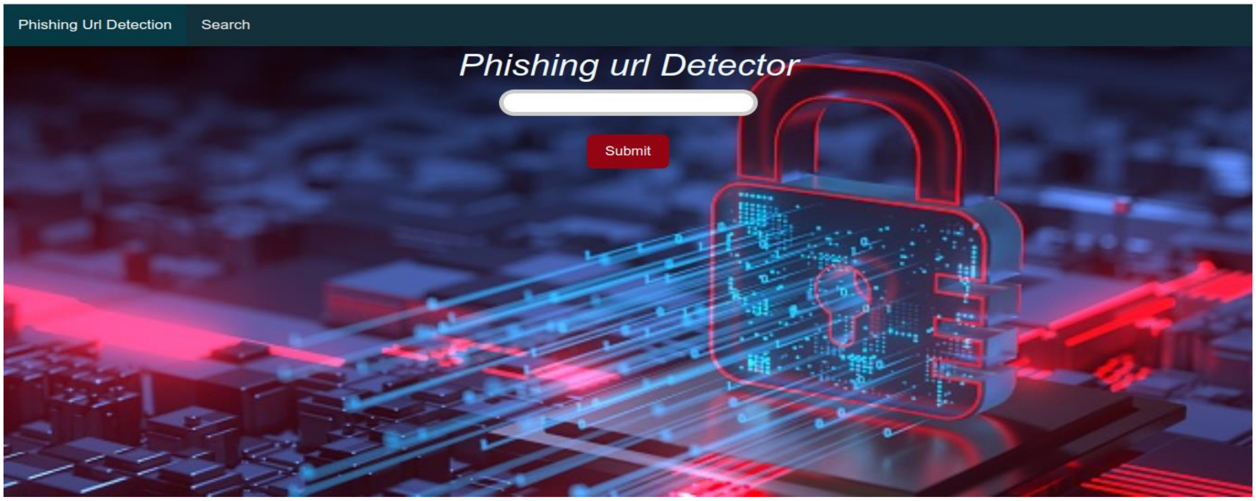
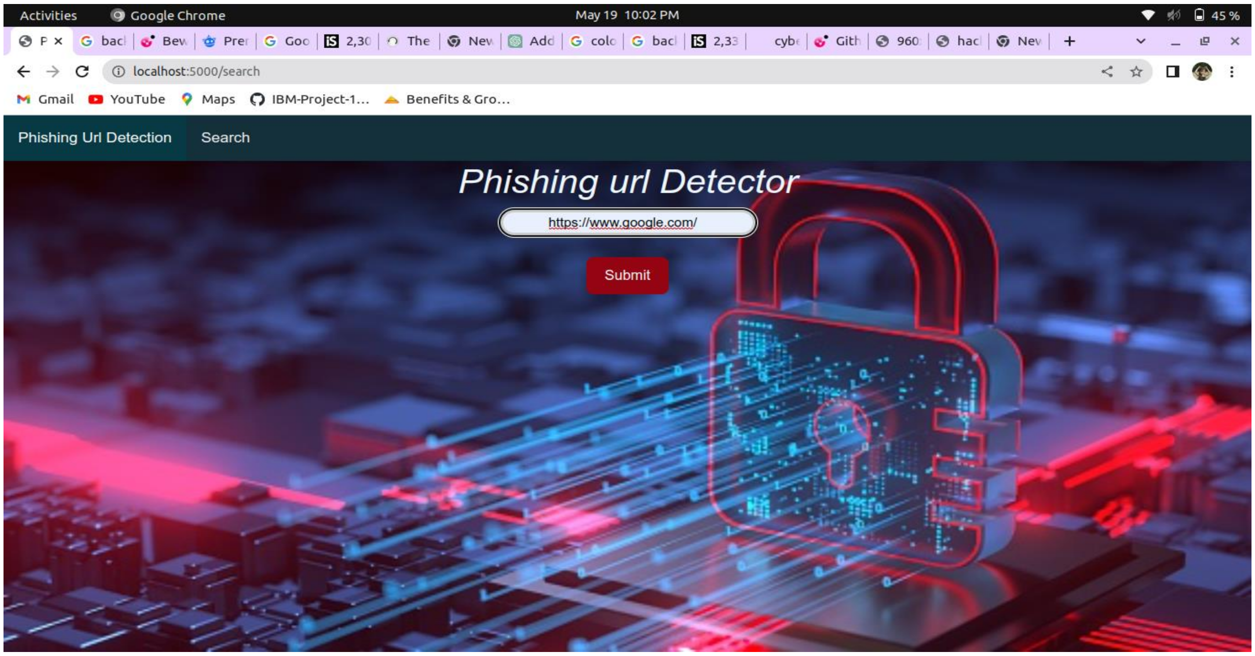


Fig A.2.2 URL Page

URL Enter Page:

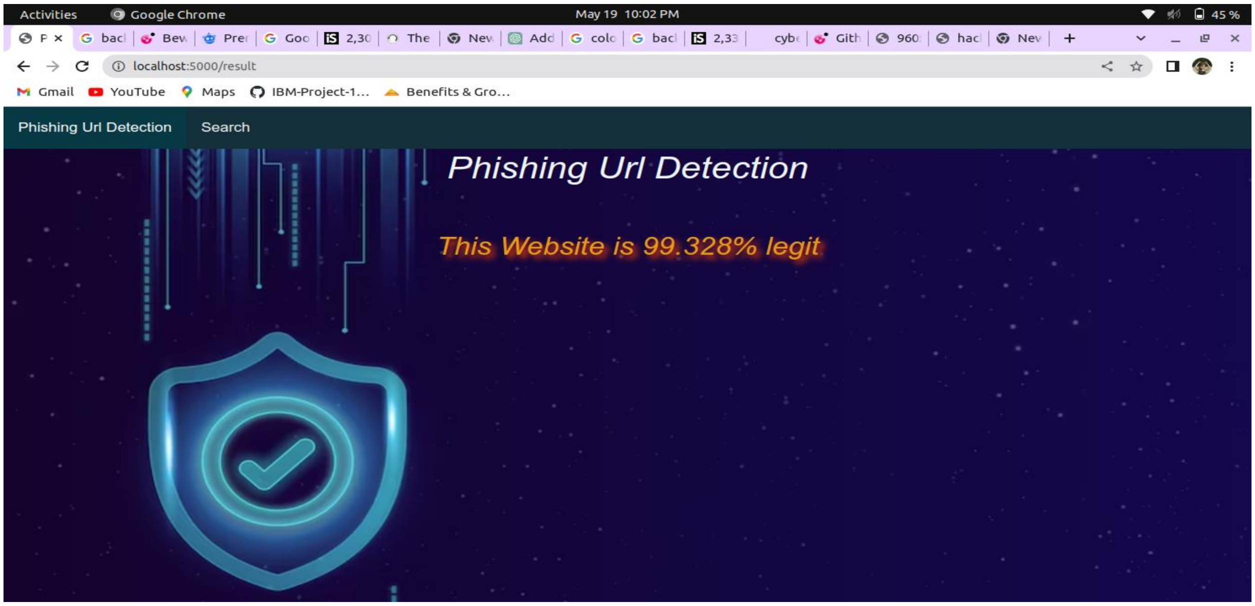


Phishing url Detector

https://www.google.com/

Fig A.2.3 URL Enter Page

ResultPage:



Phishing Url Detection

This Website is 99.328% legit

Fig A.2.4 Result page

REFERENCES

1. R. C. Salih and N. M. Yahya, "Phishing website detection using machine learning: a systematic review," Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 189-204, 2021.

2. S. H. Choi, S. H. Lee, and Y. S. Moon, "Phishing website detection using machine learning and natural language processing," Future Generation Computer Systems, vol. 116, pp. 448-458, 2021.

3. A. R. M. Luthfi, S. A. Fauzi, and N. Hidayati, "Phishing website detection using machine learning," in Proceedings of the 6th International Conference on Computer Science and Computational Intelligence (ICCSCI), pp. 1-6, 2020.

4. K. Srinivasan and S. Madhavi, "Detection of Phishing Websites Using Machine Learning Techniques," in Proceedings of the International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 862-867, 2020.

5. "Detecting Phishing Websites Using Machine Learning" by S. N. Alharthi and H. M. Al-Hassan. Journal of King Saud University - Computer and Information Sciences, Vol. 32, No. 3, May 2020.

6. T. Hasan, F. Faruque, M. Islam, M. F. Hossain, and R. Hasan, "A Deep Learning Based Detection and Prevention System for Phishing Websites," in Proceedings of the 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 185-192, 2019.

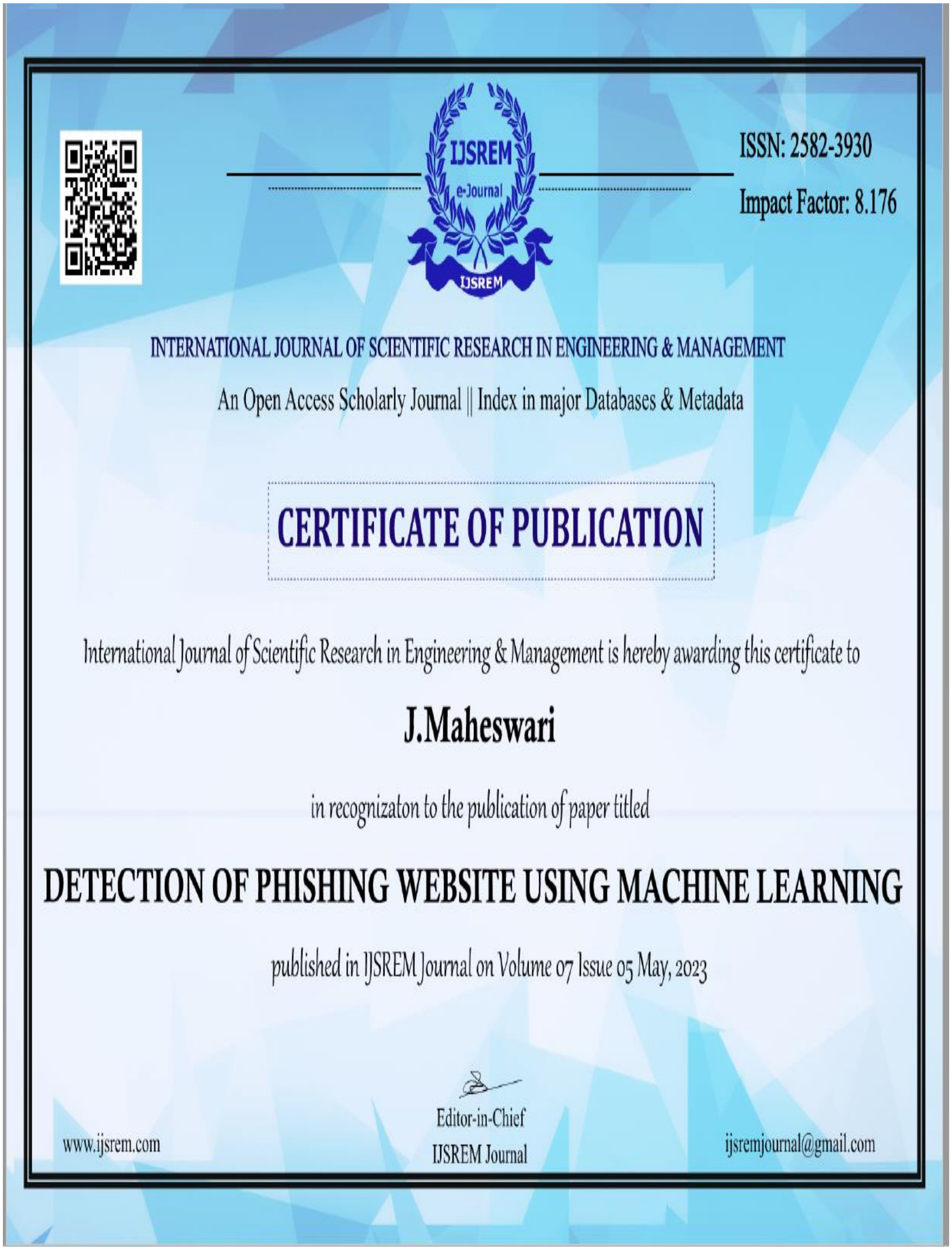
7. "Phishing website detection using machine learning classifiers" by M. Saranya and K. Saravanan. International Journal of Computer Science and Mobile Computing, Vol. 7, No. 2, February 2018.

8. "Phishing website detection using machine learning techniques based on URL features" by A. H. Al-Waisy and H. S. Al-Khafaji. International Journal of Advanced Computer Science and Applications, Vol. 9, No. 7, July 2018.

9. "Detection of Phishing Websites using Machine Learning Techniques" by R. H. Magar and S. S. Pawar. International Journal of Computer Applications, Vol. 171, No. 5, July 2017.

10. "Phishing website detection using machine learning techniques" by C. Vijayakumar and S. Arumugam. International Journal of Computer Applications, Vol. 140, No. 1, April 2016.

PUBLICATION



ISSN: 2582-3930

Impact Factor: 8.176

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

J.Maheswari

in recognizaton to the publication of paper titled

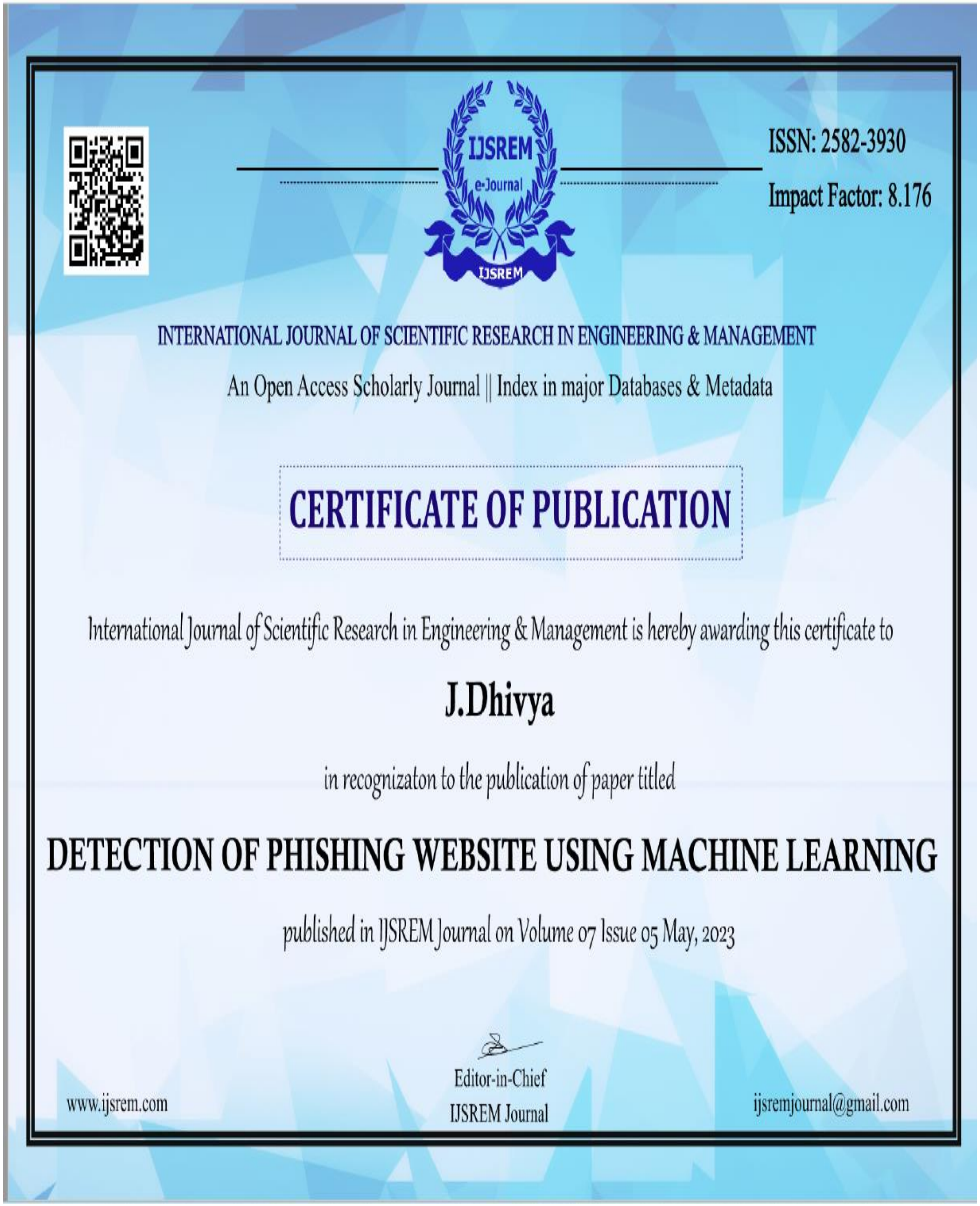
DETECTION OF PHISHING WEBSITE USING MACHINE LEARNING

published in IJSREM Journal on Volume 07 Issue 05 May, 2023

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

ijsremjournal@gmail.com



ISSN: 2582-3930

Impact Factor: 8.176

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

J.Dhivya

in recognizaton to the publication of paper titled

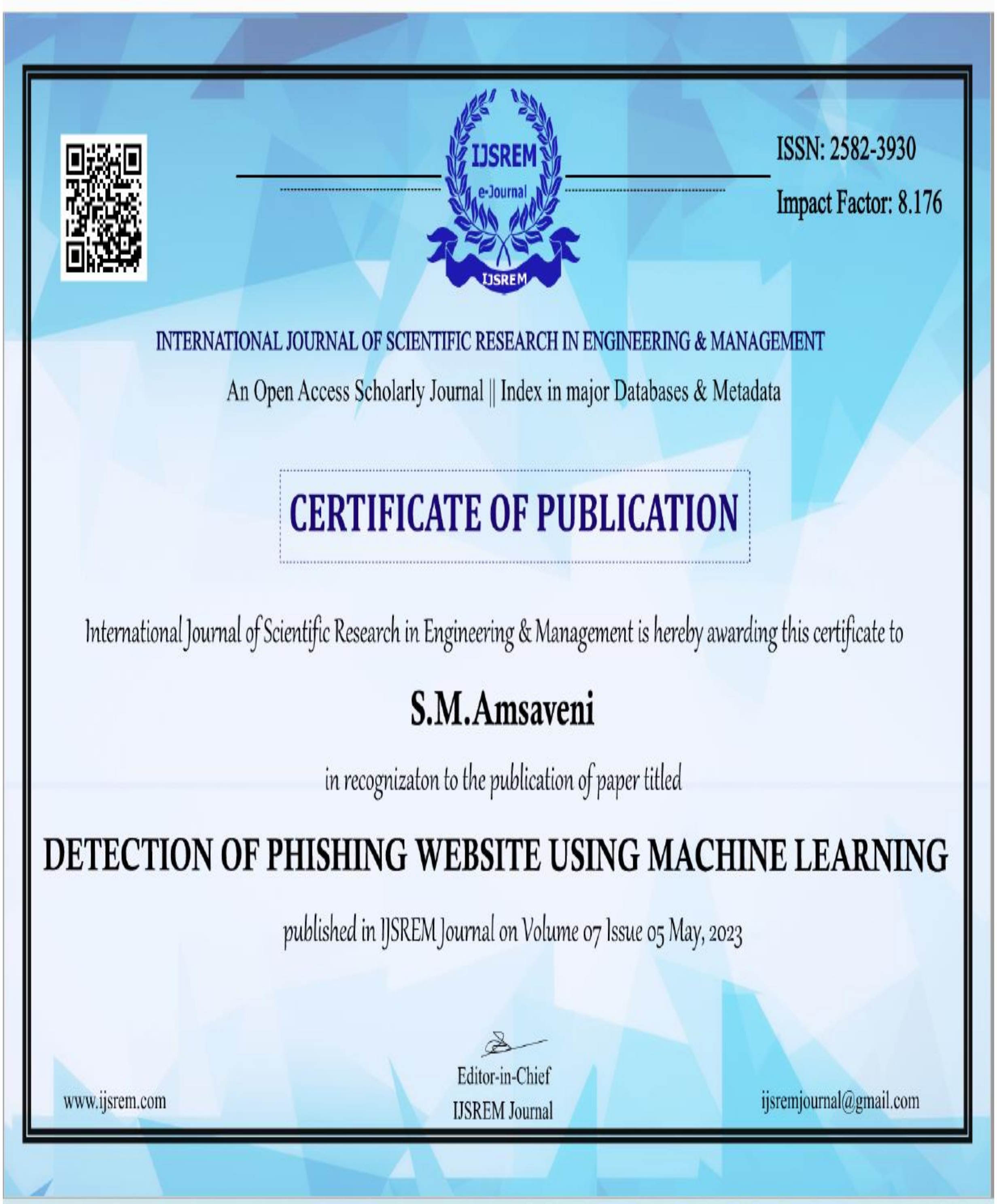
DETECTION OF PHISHING WEBSITE USING MACHINE LEARNING

published in IJSREM Journal on Volume 07 Issue 05 May, 2023

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

ijsremjournal@gmail.com



ISSN: 2582-3930

Impact Factor: 8.176

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

S.M.Amsaveni

in recognizaton to the publication of paper titled

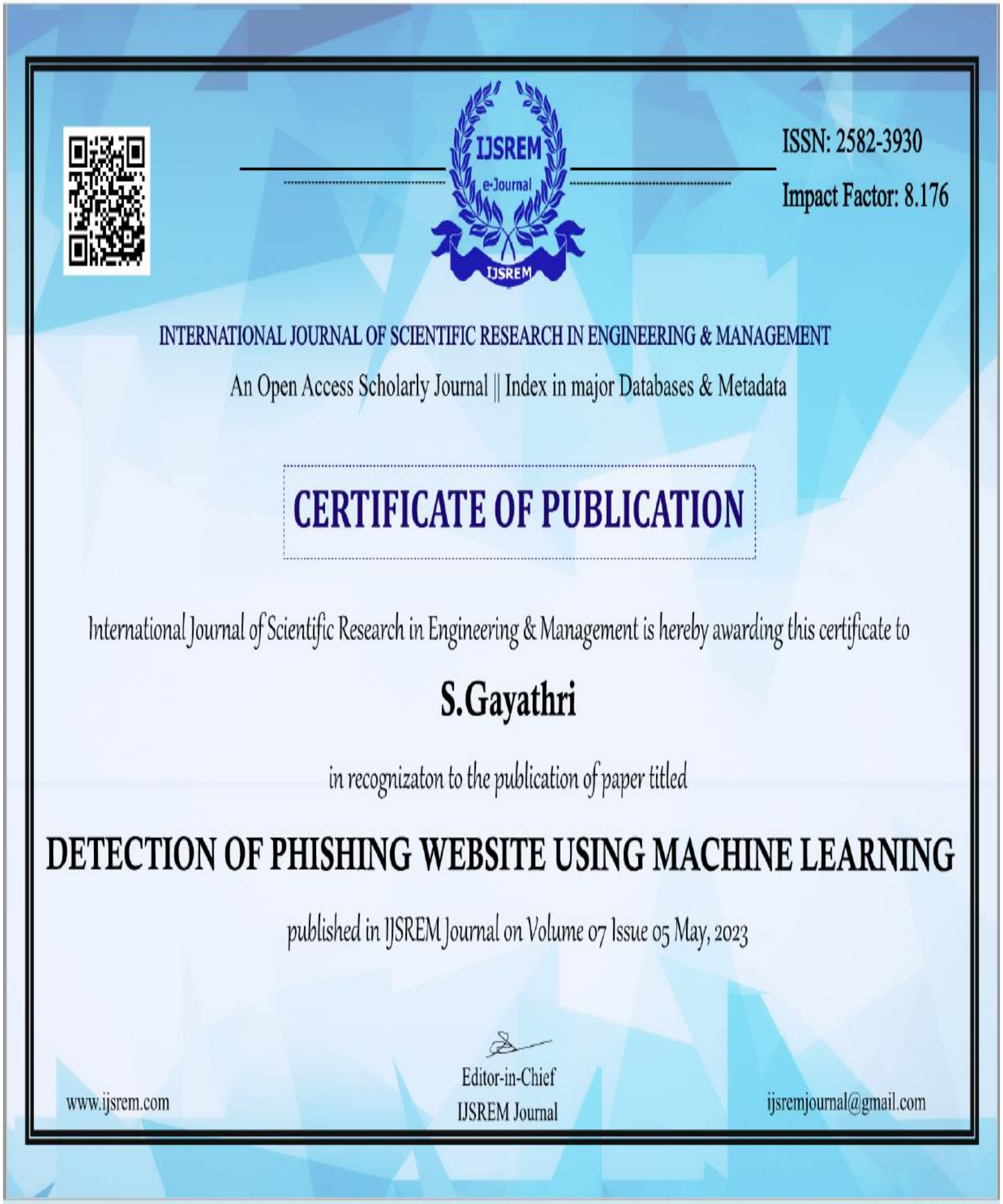
DETECTION OF PHISHING WEBSITE USING MACHINE LEARNING

published in IJSREM Journal on Volume 07 Issue 05 May, 2023

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

ijsremjournal@gmail.com



ISSN: 2582-3930

Impact Factor: 8.176

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

S.Gayathri

in recognizaton to the publication of paper titled

DETECTION OF PHISHING WEBSITE USING MACHINE LEARNING

published in IJSREM Journal on Volume 07 Issue 05 May, 2023

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

ijsremjournal@gmail.com