WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

IDENTIDICATION OF PHISHING WEBSITE

Sabin Ahammed Z[1], Mrs. A. Shajeetha Banu[2]

UG Student, Department of Computer Science, Sri Krishna Adithya College of Arts and Science, Coimbatore.
Head of Department, Department of Computer Science, Sri Krishna Adithya College of Arts and Science, Coimbatore.

A B S T R A C T :

In the modern digital landscape, phishing attacks have become one of the most widespread and dangerous cyber threats, targeting individuals and organizations by tricking them into revealing sensitive information such as usernames, passwords, and financial details. Cybercriminals create fraudulent websites that closely resemble legitimate ones, making it difficult for users to distinguish between real and fake sites. As a result, there is a growing need for automated phishing detection systems that can analyze website characteristics and alert users to potential threats. This project aims to develop a phishing website identification system using HTML, CSS, and JavaScript for the frontend and Python for the backend, providing a simple yet effective solution for detecting malicious websites.

Keywords: Login, Verify, Secure, Confirm, Bank, Pay, Account, Billing, Support, Helpdesk, Service Free, Win, Gift, Offer

Introduction

The *frontend* will serve as an intuitive interface where users can input website URLs for verification. It will be designed with *HTML for structure, CSS for styling, and JavaScript for interactivity*, ensuring a seamless user experience. The *backend*, powered by *Python*, will handle the core functionality of the system, employing *machine learning algorithms, heuristic-based analysis, and API integrations* with cybersecurity databases to determine whether a given URL is legitimate or potentially harmful. Various website attributes, including *domain age, SSL certificate status, URL structure, and website content*, will be analyzed to detect common phishing patterns. The primary objective of this project is to *develop a reliable and user-friendly phishing detection tool* that helps users identify potentially harmful websites before interacting with them. By combining a *straightforward HTML-based interface* with a *powerful Python-driven backend*, this project aims to *reduce the risks associated with phishing attacks and enhance online security*. With the increasing reliance on digital platforms for financial transactions, communication, and personal data storage, having an *effective phishing detection system* is crucial in preventing cyber fraud and ensuring a safer browsing experience for users worldwide.

Literature Survey

Phishing is a significant cybersecurity threat where attackers impersonate legitimate entities to deceive users into revealing sensitive information such as usernames, passwords, and financial details. The identification of phishing websites has been an active area of research, leading to the development of various detection techniques, including blacklist-based, heuristic-based, machine learning-based, and deep learning-based approaches.

Blacklist-Based Approaches

One of the earliest methods used to identify phishing websites is the blacklist-based approach, where URLs known to be malicious are stored in a database and compared against incoming URLs. Major web browsers and cybersecurity organizations maintain such blacklists (e.g., Google Safe Browsing and PhishTank). Zhang et al. (2007) studied the effectiveness of blacklists and found that they are useful for identifying known threats but struggle with newly created phishing sites. The primary limitation of this method is its inability to detect zero-hour phishing attacks, as blacklists require prior knowledge of a phishing website before it can be flagged.

Heuristic-Based Approaches

To overcome the limitations of blacklists, researchers introduced heuristic-based approaches that rely on predefined rules and features to detect phishing websites. These features include URL length, presence of special characters, number of subdomains, and use of HTTPS. Garera et al. (2007) developed a rule-based classification system that considers these features to flag suspicious websites. However, heuristic methods can suffer from high false positive rates as they rely on manually defined rules that may not generalize well to new attack patterns.

Machine Learning-Based Approaches

Machine learning (ML) techniques have significantly improved phishing detection by allowing models to learn patterns from large datasets. ML-based approaches extract various features from URLs, webpage content, and metadata to classify websites as legitimate or phishing. Ma et al. (2009) proposed an ML model using lexical features of URLs and achieved high accuracy in detecting phishing websites. Various algorithms such as Decision Trees, Support Vector Machines (SVM), and Random Forests have been explored for this purpose. A key advantage of ML-based methods is their adaptability to evolving phishing tactics, unlike rule-based systems. However, these models require frequent retraining with updated datasets to maintain accuracy.

Deep Learning-Based Approaches

Recent advancements in deep learning have further enhanced phishing detection capabilities. Deep learning models, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), can analyze website screenshots, text content, and URL structures with high precision. Bahnsen et al. (2017) developed a character-level Recurrent Neural Network (RNN) model that analyzes URLs to detect phishing attempts without requiring manual feature extraction. Similarly, CNN-based approaches have been used to detect phishing websites by analyzing website layouts and images. Deep learning models outperform traditional ML approaches in detecting obfuscated and adversarial phishing attacks but require substantial computational resources for training and deployment.

Hybrid Approaches

Given the limitations of individual methods, hybrid approaches combining multiple techniques have been proposed. For instance, a system may use a blacklist for quick filtering, heuristics for additional checks, and machine learning for final classification. Abdelnabi et al. (2020) introduced a hybrid approach integrating natural language processing (NLP) and deep learning to analyze website content and URLs, achieving superior detection accuracy compared to standalone methods.

Challenges and Future Directions

Despite advancements in phishing detection, several challenges remain. Attackers continuously evolve their techniques by using homographic attacks (e.g., replacing ‘o’ with ‘0’), cloaking mechanisms to evade detection, and generating dynamic phishing pages. Future research directions include developing real-time detection systems, improving adversarial robustness, and leveraging blockchain-based security mechanisms to enhance trust verification.

Phishing is a significant cybersecurity threat where attackers impersonate legitimate entities to deceive users into revealing sensitive information such as usernames, passwords, and financial details. The identification of phishing websites has been an active area of research, leading to the development of various detection techniques, including blacklist-based, heuristic-based, machine learning-based, deep learning-based, and hybrid approaches. Blacklist-based approaches rely on databases of known malicious URLs, such as Google Safe Browsing and PhishTank, but are ineffective against newly emerging phishing websites. Heuristic-based methods, on the other hand, use predefined rules such as URL length, the number of special characters, and HTTPS usage to detect phishing attempts, but they often suffer from high false positive rates and can be bypassed by attackers modifying website structures.

Machine learning (ML) techniques have significantly improved phishing detection by allowing models to learn patterns from large datasets, extracting features from URLs, webpage content, and metadata to classify websites as legitimate or phishing. Algorithms such as Decision Trees, Support Vector Machines (SVM), and Random Forests have been widely used, offering adaptability to evolving phishing tactics. However, ML models require frequent retraining with updated datasets to maintain accuracy and can be susceptible to adversarial attacks where attackers manipulate website attributes to evade detection. Recent advancements in deep learning have further enhanced phishing detection capabilities. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can analyze website screenshots, text content, and URL structures with high precision. Researchers have also explored Transformer-based models such as BERT and GPT for phishing detection by analyzing natural language text in phishing emails and website content.

Hybrid approaches that combine multiple detection techniques have been proposed to enhance robustness against phishing attacks. For example, a hybrid system may integrate a blacklist for quick filtering, heuristics for additional checks, and machine learning for final classification. Feature engineering plays a crucial role in phishing detection, with key features including lexical attributes (URL length, number of dots, and special characters), host-based indicators (domain age and WHOIS information), content-based attributes (HTML tag analysis and embedded JavaScript), behavioral patterns (user interaction and redirection chains), and network-based attributes (SSL certificate validity and DNS record analysis). By leveraging multiple features, phishing detection models can improve generalization and resilience against evasion tactics.

Despite these advancements, several challenges remain in phishing detection. Attackers continuously evolve their techniques using homographic attacks (e.g., replacing ‘o’ with ‘0’), cloaking mechanisms, and dynamically generated phishing pages. The rise of AI-generated phishing attacks introduces additional complexity, as attackers leverage machine learning to craft highly convincing phishing content. Future research directions focus on real-time detection, improving adversarial robustness, exploring blockchain-based verification for website authenticity, and enhancing the interpretability of AI-driven phishing detection models.

In conclusion, the identification of phishing websites has evolved from simple blacklisting to sophisticated AI-driven approaches. While each method has strengths and weaknesses, combining heuristic, ML, and deep learning techniques offers the most effective defense against evolving phishing threats. Ongoing research aims to improve detection accuracy, reduce false positives, and develop lightweight solutions for real-time prote

Conclusion

The identification of phishing websites has progressed from simple blacklisting to sophisticated AI-driven approaches. While each method has strengths and weaknesses, a combination of heuristic, ML, and deep learning techniques offers the best defense against evolving phishing threats. Future research should focus on improving detection accuracy, reducing false positives, and developing lightweight solutions for resource-constrained devices. By continuously enhancing detection methodologies, cybersecurity researchers can help mitigate the growing risk of phishing attacks.

Table 1 Summary of related work on Identification of Phishing Website

Literature	Identification of phishing website			
	DNN	SVM	GBI	COPY MOVE
J.Malathi, et al. 2019 [2]	YES	YES		

F. Matern, et al. 2020 [3]			YES	
Anushka Singh and Jyotsna Singh, 2022 [5]	YES			
S. B. G. T. Babu and C. S. Rao, 2020 [8]				YES
H. Chen, et al. 2020 [12]	YES			YES

Collectively, the summary of various techniques used as per the recent literature for website phishing as shown in Table 2

PAPER	***Technique***
Syed Sadaf Ali, et al. 2022 [1]	***Recompression of Images***
J.Malathi, et al. 2019 [2]	***SVM***
F. Matern, et al. 2020 [3]	***Gradient-Based Illumination***
Anushka Singh and Jyotsna Singh, 2022 [5]	***ResNet***
F. Marra, et al. 2020 [6]	***ResNet***
S. B. G. T. Babu and C. S. Rao, 2020 [8]	***Copy-move***
S. alZahir and R. Hammad, et al. 2020 [10]	***Expectation Maximization***

SYSTEM METHODOLOGY

A. System Methodology for Identification of Phishing Websites

The system methodology for identifying phishing websites consists of multiple stages, including data collection, feature extraction, model training, and classification. The proposed approach integrates machine learning and deep learning techniques to enhance accuracy and resilience against evasion tactics.

B. **Data Collection:**

A comprehensive dataset is gathered from sources such as PhishTank, OpenPhish, and Alexa's top-ranked legitimate websites.Data includes phishing and legitimate URLs, website content, and metadata.

C. **Feature Extraction:**

Extracted features include lexical features (URL length, presence of special characters), content-based features (HTML structure, JavaScript analysis), host-based features (domain age, SSL certificate status), and network features (DNS information, WHOIS lookup).Image-based features are analyzed using CNNs to detect phishing attempts based on website appearance.

D. **Model Training:**

Machine learning algorithms (Decision Trees, SVM, Random Forest) and deep learning techniques (CNN, RNN) are trained on labeled datasets.Feature selection techniques such as Principal Component Analysis (PCA) are employed to reduce dimensionality and enhance performance.

E. **Phishing Detection and Classification:**

The trained model classifies websites as phishing or legitimate based on extracted features.A hybrid approach combines heuristic analysis, blacklist filtering, and machine learning to improve detection accuracy.Suspicious websites undergo further analysis using deep learning-based visual and content similarity detection.

F. **Evaluation and Performance Metrics:**

The model is evaluated using accuracy, precision, recall, and F1-score to measure its effectiveness.Robustness against adversarial attacks is tested to ensure resilience against evolving phishing strategies.

G. **Deployment and Real-Time Detection:**

The final model is integrated into web browsers and email security systems for real-time phishing detection.APIs and browser extensions are developed to flag phishing websites and alert users.In conclusion, the proposed system methodology leverages multiple detection techniques to enhance phishing detection accuracy.

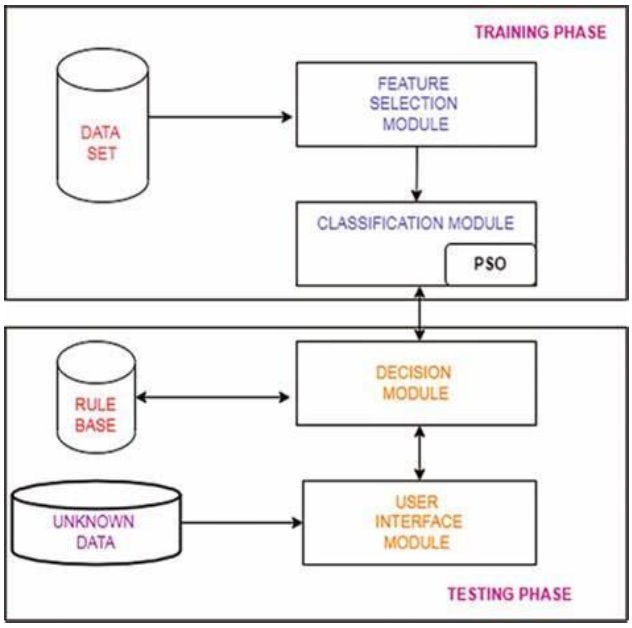


Fig.1 System architecture for Identification of phishing website

Convolutional Neural Network

Convolutional Neural Networks (CNNs) have indeed become a popular tool for detecting forgery images. CNNs are a type of deep learning algorithm that can be trained to extract features from images and classify them into different categories. They are inspired by the human visual system and consist of multiple layers of inter connected neurons that perform convolution operations on the input image to extract features. One of the advantages of using CNNs for image forensics is their ability to detect subtle artifacts that may not be visible to the naked eye. For example, when an image is manipulated, such as by copy-pasting a fragment from one image to another, there may be slight variations in the pixel values or texture that are indicative of the manipulation. CNNs can learn to detect these differences and classify the image as either genuine or fake. Overall, CNNs have shown great promise in a variety of computer vision and image processing applications, including image forensics. With the increasing prevalence of digital manipulation in today's world, the ability to detect forged images has become more important than ever, and CNNs provide a powerful tool for this purpose. The input layer in a convolutional neural network (CNN) is where the images from the dataset are fed. The images are usually in the form of 3-dimensional arrays, with the first two dimensions representing the height and width of the image (the number of pixels), and the third dimension representing the red, green, and blue (RGB) colors present in each pixel. In the feature-extraction part of the CNN architecture, the input image is passed through a series of convolutional layers, which apply a set of learnable filters to extract features from the image. Each filter produces a feature map that highlights a particular pattern or feature in the input image. These feature maps are then passed through activation functions like ReLU to introduce non-linearity and avoid the vanishing gradient problem. After the feature extraction process, the output of the last convolutional layer is flattened into a 1-dimensional vector and fed into a series of fully connected layers for classification. The fully connected layers use the extracted features to make predictions about the class of the input image. The final output layer usually employs the softmax function to generate a probability distribution over the classes, indicating the most likely class for the input image.

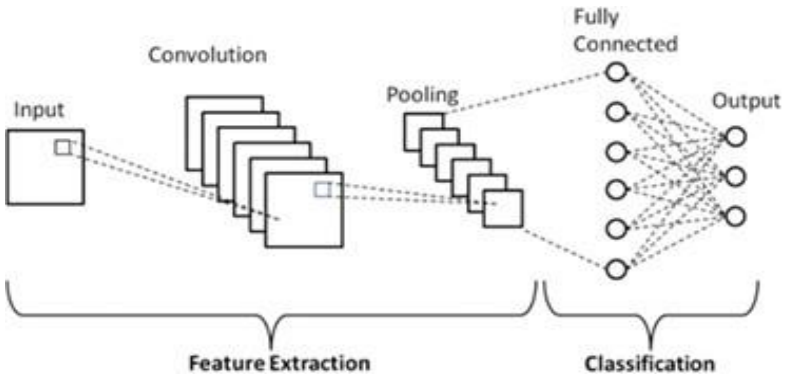


Fig. 2 CNN Architecture

That's a good summary of the main functions of each layer in a convolutional neural network. Here's a bit more detail on each layer: Convolutional layer: This layer applies a set of filters to the input image or feature map, generating a set of output feature maps. Each filter looks for a specific pattern or

feature in the input, and the output feature maps highlight where those features are present in the input. By stacking multiple convolutional layers, the network can learn increasingly complex and abstract features. Pooling layer: This layer downsamples the output feature maps from the convolutional layer, reducing their spatial size and number of parameters. The most common type of pooling is max pooling, which takes the maximum value within a small region of the feature map. This helps to capture the most salient features while discarding redundant information, and also makes the network more robust to variations in input position and scale.

Fully-connected layer: This layer takes the flattened output of the previous layer and applies a set of weights to produce a vector of class probabilities. The weights are learned during training using backpropagation and gradient descent. The softmax activation function ensures that the output probabilities sum to 1, allowing the network to make a single prediction for the input image. The number of neurons in the fully-connected layer corresponds to the number of output classes.

Identification Of Phishing Website

Phishing is a significant cybersecurity threat where attackers impersonate legitimate entities to deceive users into revealing sensitive information such as usernames, passwords, and financial details. The identification of phishing websites has been an active area of research, leading to the development of various detection techniques, including blacklist-based, heuristic-based, machine learning-based, deep learning-based, and hybrid approaches. Blacklist-based approaches rely on databases of known malicious URLs, such as Google Safe Browsing and PhishTank, but are ineffective against newly emerging phishing websites. Heuristic-based methods, on the other hand, use predefined rules such as URL length, the number of special characters, and HTTPS usage to detect phishing attempts, but they often suffer from high false positive rates and can be bypassed by attackers modifying website structures.Machine learning (ML) techniques have significantly improved phishing detection by allowing models to learn patterns from large datasets, extracting features from URLs, webpage content, and metadata to classify websites as legitimate or phishing. Algorithms such as Decision Trees, Support Vector Machines (SVM), and Random Forests have been widely used, offering adaptability to evolving phishing tactics. However, ML models require frequent retraining with updated datasets to maintain accuracy and can be susceptible to adversarial attacks where attackers manipulate website attributes to evade detection. Recent advancements in deep learning have further enhanced phishing detection capabilities. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can analyze website screenshots, text content, and URL structures with high precision. Researchers have also explored Transformer-based models such as BERT and GPT for phishing detection by analyzing natural language text in phishing emails and website content.Hybrid approaches that combine multiple detection techniques have been proposed to enhance robustness against phishing attacks. For example, a hybrid system may integrate a blacklist for quick filtering, heuristics for additional checks, and machine learning for final classification. Feature engineering plays a crucial role in phishing detection, with key features including lexical attributes (URL length, number of dots, and special characters), host-based indicators (domain age and WHOIS information), content-based attributes (HTML tag analysis and embedded JavaScript), behavioral patterns (user interaction and redirection chains), and network-based attributes (SSL certificate validity and DNS record analysis). By leveraging multiple features, phishing detection models can improve generalization and resilience against evasion tactics.Despite these advancements, several challenges remain in phishing detection. Attackers continuously evolve their techniques using homographic attacks (e.g., replacing ‘o’ with ‘0’), cloaking mechanisms, and dynamically generated phishing pages. The rise of AI-generated phishing attacks introduces additional complexity, as attackers leverage machine learning to craft highly convincing phishing content. Future research directions focus on real-time detection, improving adversarial robustness, exploring blockchain-based verification for website authenticity, and enhancing the interpretability of AI-driven phishing detection models.

Author(s)	Year	Approach	Key Findings
Zhang et al.	2007	Blacklist-Based	Effective against known phishing sites but fails against zero-hour attacks.
Garera et al.	2007	Heuristic-Based	Uses predefined rules but has high false positives.
Ma et al.	2009	Machine Learning	Lexical feature-based classification improves detection accuracy.
Bahnsen et al.	2017	Deep Learning	RNN model detects phishing URLs without manual feature extraction.
Abdelnabi et al.	2020	Hybrid Approach	Combines NLP and deep learning for superior accuracy.

Fig.3 A CNN which acts a backbone for the model

SYSTEM IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively. Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively. The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system. The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation. The system may require some software to be developed. The web

Hardware System Specification:

System : Intel Dual Core

Hard Disk : 500GB

Ram : 4GB

Software Specification Operating System:

Operating System : Windows Family

Front End : HTML,CSS

Back End : Python

HTML:

HTML (HyperText Markup Language) is the standard language used to create and design webpages on the internet. It is a markup language that structures the content of a webpage by using various tags and elements. HTML allows developers to define headings, paragraphs, links, images, tables, and other types of content.

CSS:

CSS (Cascading Style Sheets) is a stylesheet language used to control the presentation and layout of HTML elements on a webpage. While HTML provides the structure of a webpage, CSS is used to define how that content should look, including aspects like colors, fonts, spacing, and positioning.

JAVASCRIPT:

JavaScript (JS) is a powerful scripting language used to create dynamic and interactive content on webpages. While HTML provides the structure and CSS defines the design, JavaScript enables functionalities like animations, form validation, and real-time updates. It allows developers to manipulate HTML elements, handle user events, and interact with APIs to enhance the user experience. JavaScript can be applied in three main ways: inline (directly within an HTML element), internally.

PYTHON

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It is widely used for web development, data analysis, artificial intelligence, machine learning, automation, and more. Python's syntax is clear and concise, which makes it easy for beginners to learn and for experienced developers to write efficient code. Python supports multiple programming paradigms, including procedural, object- oriented, and functional programming.

Datase

The CASIA v2.0 database contains a total of 10,000 images, divided into two subsets: a training set of 5,000 images and a testing set of 5,000 images. Each subset includes eight categories of images: animal, architecture, article, character, nature, plant, scene, and texture. The images are in JPEG format and have a size of either 256 x 384 or 384 x 256 pixels.CASIA V2.0 dataset is used for image forgery detection. Two classes make up this dataset: actual photos and tampering detection. There are 7354 images, which are classified into real images and altered images in JPG format.

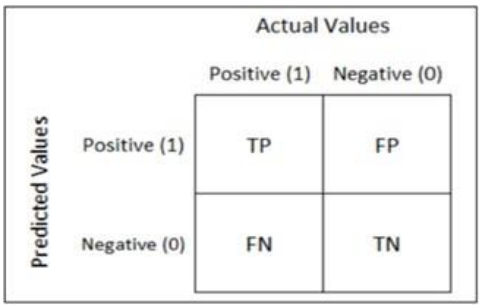
Dataset	**Size**	**Categories**	**Format**
CASIA V2.0	**5 GB**	**8 categories of images**	**JPEG**

Table 3 Details of CASIA Dataset

Fig.5. Images from the CASIA V2.0 Dataset

Confusion Metrics

A confusion matrix is a table that is commonly used to evaluate the performance of a classification algorithm by comparing the predicted labels to the true labels of a set of test data. The matrix displays the number of true positive, false positive, true negative, and false negative predictions made by the algorithm

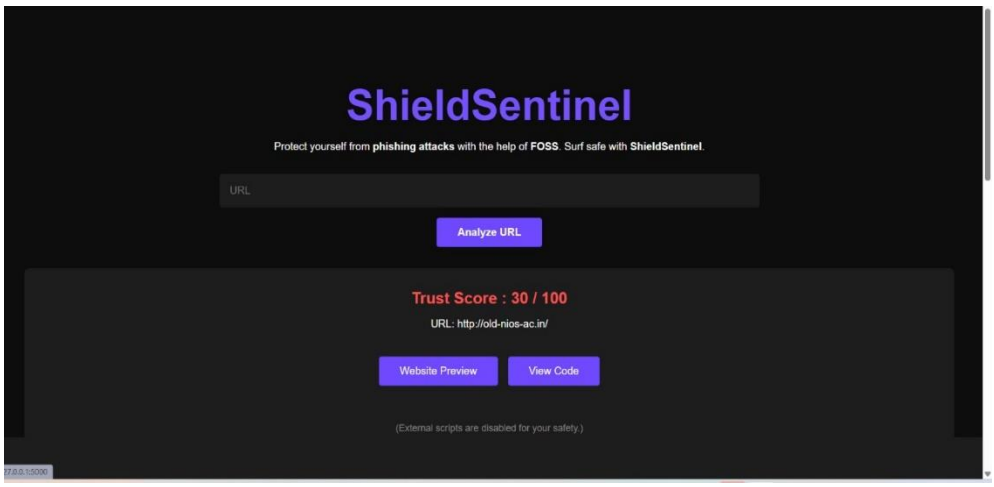


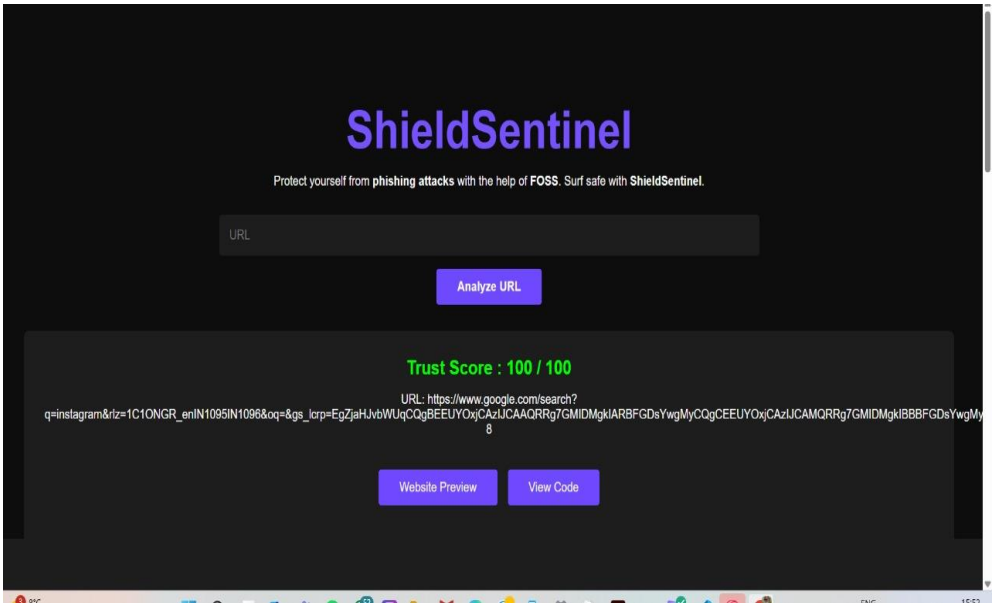
The above table is a confusion matrix that summarizes the performance of a binary classification model, andit includes four possible outcomes: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). True positives occur when the model correctly predicts a positive outcome, and true

negatives occur when the model correctly predicts a negative outcome. False positives occur when the model predicts a positive outcome, but the actual outcome is negative, and false negatives occur when the model predicts a negative outcome, but the actual outcome is positive.

RESULTS ANDANALYSIS

The original image and its ELA-converted counterpart are shown in Fig. 7 and Fig. 8 of the dataset, respectively. And the fake website and its corresponding ELA-converted image are shown in Figs. 9 and 10, respectively. In Fig. 11, the red line represents the model's training loss and training accuracy, while the blue line represents the model's validation loss and validation accuracy. The model is iteratively trained and has an accuracy of 78.08%





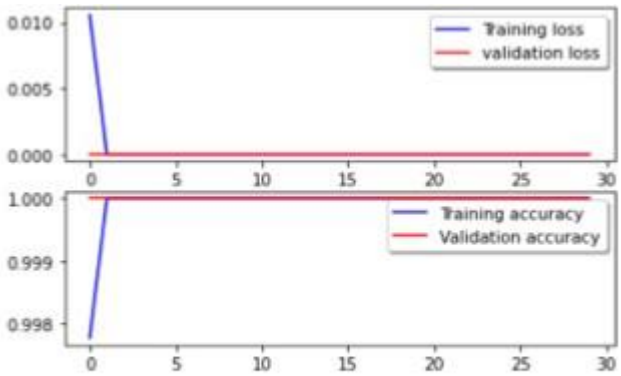


Fig. 11: Evaluation between training loss w.r.t. validation loss and training accuracy w.r.t. validation accuracy

5. CONCLUSION

The identification of phishing websites using Python and HTML provides a robust and scalable approach to combating cyber threats. By leveraging web scraping, feature extraction, and machine learning-based classification, this method enhances the accuracy of phishing detection. The integration of deep

learning models such as CNNs further improves the identification of phishing websites based on visual and structural similarities. The deployment of real-time detection systems through web applications and browser extensions significantly strengthens online security. Future improvements can focus on enhancing model generalization, reducing false positives, and integrating blockchain-based verification to improve trustworthiness. Overall, this approach offers an effective solution to mitigate phishing attacks and protect users from online fraud.

REFERENCES

1. **Learning Python",** *Mark Lutz*, O'Reilly Media, Fifth Edition, July 2013.
2. **"Python Crash Course",** *Eric Matthes*, No Starch Press, Second Edition, May 2019.
3. **"Fluent Python",** *Luciano Ramalho*, O'Reilly Media, First Edition, August 2015.
4. **"Python Cookbook",** *David Beazley and Brian K. Jones*, O'Reilly Media, Third Edition, May 2013.
5. **"Automate the Boring Stuff with Python",** *Al Sweigart*, No Starch Press, Second Edition, November 2019.
6. **"Python Programming: An Introduction to Computer Science**", *John M. Zelle*, Franklin, Beedle & Associates Inc., Third Edition, 2016.
7. **"Effective Python: 90 Specific Ways to Write Better Python**", *Brett Slatkin*, Addison-Wesley, Second Edition, November 2019.
8. **"Think Python: How to Think Like a Computer Scientist",** *Allen B. Downey*, Green Tea Press, Second Edition, August 2015.
9. **"Python for Data Analysis**", *Wes McKinney*, O'Reilly Media, Second Edition, October 2017.
10. **"Introduction to Machine Learning with Python",** *Andreas C. Müller and Sarah Guido*, O'Reilly Media, First Edition, September 2016