ACADEMIA MILITAR

Detecting Phishing Websites Using Artificial Intelligence and Computer Vision for Brand Protection and Cyber Defence

Author: Carlos Eduardo Carvalho Vaz Pires

Supervisor: Prof. José Alberto de Jesus Borges

Master in Information Warfare

Thesis to obtain the Master of Science Degree in Information Warfare

Lisbon, September 2023

ACADEMIA MILITAR

Detecting Phishing Websites Using Artificial Intelligence and Computer Vision for Brand Protection and Cyber Defence

Author: Carlos Eduardo Carvalho Vaz Pires

Supervisor: Prof. José Alberto de Jesus Borges

Master in Information Warfare

Thesis to obtain the Master of Science Degree in Information Warfare

Lisbon, September 2023

PREFACE

"I am your enemy, the first one you've ever had who was smarter than you. There is no teacher but the enemy. No one but the enemy will ever tell you what the enemy is going to do. No one but the enemy will ever teach you how to destroy and conquer. Only the enemy shows you where you are weak. Only the enemy tells you where he is strong. And the only rules of the game are what you can do to him and what you can stop him from doing to you. I am your enemy from now on. From now on I am your teacher."

Orson Scott Card, Ender's Game

DEDICATION

To my partner, for your patience, understanding, and unwavering support. Your love has been my refuge and motivation.

ACKNOWLEDGMENTS

I sincerely thank the individuals and institutions whose support and assistance have been instrumental in completing this thesis.

First and foremost, I extend my most profound appreciation to my advisor, Prof. José Alberto de Jesus Borges, for his unwavering guidance, expertise, and encouragement throughout this research endeavour. His insightful availability, feedback and commitment have been invaluable.

I want to thank my team, who acted as a development stakeholder and provided insights on identifying the most critical requirements the developed platform should have. Their feedback was crucial in prioritizing the most critical components.

Finally, I would also like to thank my colleagues at the Information Warfare Masters' for their camaraderie, intellectual exchange, and moral support throughout my academic journey: Alexandre Aleluia, Artur Alves, Luís Santos, Luís Velho, Orlando Sousa and Sérgio Cruz.

RESUMO

A sobreposição de eventos ciber associados a adversários com um elevado nível de sofisticação e a grupos apoiados por Estados ou a atuarem no seu interesse evidencia a necessidade de fortificar os controlos de cibersegurança. O cibercrime é cada vez predominante, adaptativo e diversificado, com os adversários a desenvolverem e aplicarem estratégias cada vez mais sofisticadas para atacar pessoas e empresas. Os ataques de engenharia social com recurso a *phishing* continuam a ser uma ameaça dominante, em que os adversários, estando cada vez mais sofisticados e motivados pelo elevado retorno financeiro, refinam as suas táticas e técnicas. As tecnologias de visão computacional são uma ferramenta inestimável para a identificação e categorização de *websites* de *phishing*, contribuindo assim para a ciberdefesa e cibersegurança. A proeminência das Redes Neurais Convolucionais demonstra a sua notável capacidade em identificar com precisão *websites* de *phishing*. Neste trabalho é proposto uma ferramenta inovadora denominada por BlitzPhish, baseada em inteligência artificial e tecnologias de visão computacional, para a identificação, deteção e resposta a ataques de *phishing*. O objetivo principal do BlitzPhish é o de automatizar o processo de classificação, minimizando o tempo entre a deteção e a tomada de uma ação de resposta ao ataque de *phishing*. Quando treinado com conjunto de imagens selecionadas de capturas de ecrã de *websites* reais, o classificador destacou-se pela sua eficácia na classificação de *websites* de *phishing* assim como no número reduzido de falsos-positivos, tendo obtido 95.76% na métrica de F1 e 91.57% na métrica de *Matthews Correlation Coefficient*. Sugere-se como trabalho futuro, para melhorar a precisão e a sensibilidade do classificador, e potenciar o desenvolvimento de novas estratégias de cibersegurança, a integração de mecanismos de *fuzzy logic* com recurso a outras variáveis.

Palavras-chave: Ciberdefesa; Cibersegurança; Inteligência Artificial; *Phishing*; Redes Neuronais Convolucionais

ABSTRACT

The urgency of fortifying cybersecurity measures has never been more pronounced, with cyber threats increasingly intertwined with state-sponsored activities and sophisticated threat groups. Cybercrime is becoming more prevalent, adaptive, and diverse, with adversaries employing increasingly sophisticated strategies to target individuals and corporations. Phishing attacks have remained dominant due to the lucrative gains criminals derive from them, with adversaries refining their methods and elevating their precision and technical sophistication. Computer vision technologies are a powerful tool for identifying and categorising phishing websites, contributing to cyber defence and cybersecurity. The prominence of Convolutional Neural Networks showcases their remarkable capacity to discern diverse phishing websites accurately. A novel tool named BlitzPhish is proposed to identify, detect and mitigate phishing attacks, leveraging the power of computer vision technology and artificial intelligence. BlitzPhish's primary objective is to automate the classification process, reducing the dwell time between detection and executing courses of action to respond to phishing attacks. When applied to a real-world curated dataset, the proposed classifier achieved relevant results with an F1-Score of 95.76% and a Matthews Correlation Coefficient value of 91.57%, highlighting the classifier's effectiveness in identifying phishing domains with minimal false classifications, affirming its suitability for the intended purpose. Future work involves the integration of fuzzy logic to consider additional variables to enhance the classifiers' precision and recall and develop new cybersecurity strategies.

Keywords: Artificial Intelligence; Convolutional Neural Networks; Cyber defence; Cybersecurity; Phishing

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDIX

LIST OF ACRONYMS

Acronym	Definition
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interfaces
APT	Advanced Persistent Threat
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
BRIEF	Binary Robust Independent Elementary Features
CNN	Convolutional Neural Networks
CoA	Course of Action
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
CV	Computer Vision
DBIR	Data Breach Investigations Report
DCT	Discrete Cosine Transformation
DOI	Digital Object Identifier
DOM	Document Object Model
EC	Exclusion Criteria
EMD	Earth Mover's Distance
ENISA	European Union Agency for Cybersecurity
ETL	ENISA Threat Landscape
EUROPOL	European Union Agency for Law Enforcement Cooperation
FAST	Features from Accelerated Segment Test
FBI	Federal Bureau of Investigation
FN	False Negatives
FP	False Positives
GUI	Graphical User Interface
HOG	Histogram of Oriented Gradients
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IC	Inclusion Criteria
IC3	Internet Crime Complaint Center

Acronym	Definition
IOCTA	Internet Organised Crime Threat Assessment
KNN	K-Nearest Neighbor
LR	Linear Regression
LSTM	Long Short-Term Memory
MCC	Matthews Correlation Coefficient
ML	Machine Learning
MISP	Malware Information Sharing Platform
MMOD	Max-Margin Object Detection
NASA	National Aeronautics and Space Administration
NB	Naïve Bayes
NCD	Normalized Compression Distance
NIST	National Institute of Standards and Technology
OCR	Optical Character Recognition
ORB	Oriented FAST and Rotated BRIEF
PCA	Principal Component Analysis
QC	Quality Criteria
QR Codes	Quick Response Codes
RF	Random Forest
RGB	Red Green Blue
ROC	Receiver Operating Characteristic
RQ	Research Question
SE	System Engineering
SHA-256	Secure Hash Algorithm 256
SIFT	Scale Invariant Feature Transform
SLR	Systematic Literature Review
SMS	Short Message Service
SQ	(Research) Subquestion
SSIM	Structural Similarity Index Measure
SURF	Speed Up Robust Features
SVM	Support Vector Machine
TF-IDF	Term Frequency – Inverse Document Frequency
TLS	Transport Layer Security

Acronym	Definition
TN	True Negatives
TP	True Positives
URL	Uniform Resource Locator

INTRODUCTION

Cybercrime poses a significant threat to today's society, making the fight against cyber-attacks a critical concern for businesses and governments (Huang et al., 2019). This threat amplifies when nation-states back cybercriminals or cybercriminals act in nation-states interests, such as the Lazarus Group, also known as Advanced Persistent Threat 38 (APT38), which performed one of history's biggest known cyber heists (Roy et al., 2023), or the Sandworm Group, which attacked the Ukrainian power grid in 2015 (Herr & Armbrust, 2015), arguably aimed at delivering military effects in the physical world, making it both a cybersecurity and a cyber defence problem. Both attacks resulted from the initial exploitation of human factors through phishing (Herr & Armbrust, 2015; Roy et al., 2023). Phishing attacks succeed because people often are unaware of their vulnerabilities and lack knowledge about the risks of their actions (Desolda et al., 2022).

In the latest Internet Organised Crime Threat Assessment (IOCTA) report, which provides an overview of the cybercrime threat landscape in the European Union, EUROPOL[1] noted that cybercrime continues to rise, evolving and diversifying, with adversaries using increasingly sophisticated tactics to target individuals and businesses (EUROPOL, 2021). The same observation holds in the United States, where the FBI's Internet Crime Report, published by the Internet Crime Complaint Center (IC3) division, showcases increased complaints and financial losses over the years (FBI, 2021).

Similarly, the European Union Agency for Cybersecurity (ENISA) annually releases the ENISA Threat Landscape (ETL) report. In the 2022 edition, ENISA highlighted eight prime threat groups due to their predominance and impact when materialised. One of the listed threat group categories pertains to social engineering, which includes phishing attacks (ENISA, 2022).

The American communications company Verizon annually publishes the Data Breach Investigations Report (DBIR) to provide insights into the analysis of thousands of incidents and data breaches worldwide. The latest report stated that phishing remains popular among adversaries as it "is where their targets are reachable" (Widup et al., 2022, p. 34). Phishing and social engineering attacks have prevailed over the last few years as criminals continue to profit significantly from them without significantly reinventing their *modus*

[1] The European Union Agency for Law Enforcement Cooperation

operandi. Instead, they continue to improve the attacks, making them more refined, targeted and technically advanced (EUROPOL, 2021).

The work developed in this research has already resulted in three conference papers (Pires & Borges, 2022a, 2022b, 2023b) and one journal article submitted to the ACM Computing Surveys Journal (Pires & Borges, 2023a). An extended version of (Pires & Borges, 2023b) will be submitted to the Expert Systems with Applications Journal.

This document consists of an introduction (this chapter), four development chapters and conclusions.

Chapter 1 explores using Artificial Intelligence and Computer Vision for brand protection and phishing website detection. The main contribution of this chapter is the definition of the body of concepts used in this research.

Chapter 2 details the methodology used for the systematic literature review (Chapter 3) and the computational implementation and assessment (Chapter 4). This chapter specifies the main research question and the research subquestions.

Chapter 3 presents the results of the systematic literature review on how artificial intelligence, using computer vision techniques, can support detecting and classifying phishing websites from screenshots taken from browser-rendered websites to support an organisation's brand protection efforts. This chapter delves into addressing the first five research subquestions.

Chapter 4 details the development of BlitzPhish, a tool that solves a practical brand protection phishing problem and evaluates it against other tools identified during the systematic literature review (Chapter 3). This chapter addresses the sixth research subquestion.

Finally, in the concluding chapter, an answer to the main research question is formulated supported by the answers to the research subquestions obtained in the previous development chapters. The chapter also provides future research directions.

This research also contains several appendices, referenced from Chapter 1 through Chapter 4, that provide additional relevant information and supporting materials.

CHAPTER 1 – ON THE DETECTION OF PHISHING WEBSITES USING ARTIFICIAL INTELLIGENCE AND COMPUTER VISION FOR BRAND PROTECTION

1.1. Key Concepts

This research considers the following definitions for key concepts:

Artificial Intelligence: The study of creating "intelligent entities–machines that can compute how to act effectively and safely in a wide variety of novel situations" (Russell & Norvig, 2022, p. 19);

Brand: "A name, symbol, logo, design or image, or any combination of these, which is used to identify a product or service and distinguish it from those of competitors" (Kotler et al., 2019, p. 377);

Computer Vision: Technologies to understand and interpret visual data from the world, process images, and recognise objects (Goodfellow et al., 2016).

As Khonji, Iraqi, and Jones (2013, p. 2091) noted, phishing has several definitions since "the phishing problem is broad and incorporates varying scenarios". The National Institute of Standards and Technology (NIST) glossary has seven definitions of phishing depending on the publication context (NIST, 2023). This research considers the definition coined by (Khonji et al., 2013, p. 2092): "Phishing is a type of computer attack that communicates socially engineered messages to humans via electronic communication channels in order to persuade them to perform certain actions for the attacker's benefit".

1.2. For Brand Protection

Chikada (2019) noted that a brand is an organisation's most valuable asset. Chikada also stated that "phishing, social media and unauthorised websites pose the most threat to a brand and its consumers alike" (Chikada, 2019, p. 6) and that "getting a brand-protection strategy wrong leads to a loss of trust, a damaged reputation and reduced revenues" (Chikada, 2019, p. 8). Adversaries often use legitimate organisations' logos in their phishing emails and web pages (van den Hout et al., 2022) to trick the intended victim by impersonating a trusted organisation. As such, it is vital for organisations to "monitor for impersonation and the misuse of brands and trademarks and prevent those with ill intent from fooling consumers into thinking they're engaging with a brand" (Chikada & Gupta, 2017, p. 354).

1.3. For Phishing

Phishing is usually perceived as a social engineering attack to acquire sensitive information (Salahdine & Kaabouch, 2019). Phishing attacks contain subcategories, such as spearphishing, a more refined and sophisticated version of the typical phishing attack aimed at a specific user or group, or whaling, a spearphishing attack aimed at users in corporate or otherwise high positions (ENISA, 2022; Salahdine & Kaabouch, 2019). These attacks can also be classified based on the payload delivery method used to target the intended victim. On the ELT report, ENISA defined phishing delivered via email as a phishing email attack, delivered via SMS as a smishing attack, and delivered via a phone call as a vishing attack (ENISA, 2022). Other authors provided additional definitions, such as qrishing for phishing attacks delivered via Quick Response Codes (QR Codes) (Vidas et al., 2013).

MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK), a curated knowledge base for cyber adversary behaviour (Strom et al., 2020), currently defines the phishing technique on three different tactics: phishing for information (MITRE, 2023c) on the reconnaissance tactic (TA0043), phishing (MITRE, 2023d) on the initial access tactic (TA0001) and internal spearphishing (MITRE, 2023a) on the lateral movement tactic (TA0008). Each tactic pertains to a specific attack phase the adversary exploits at a given moment in the attack chain. As defined by MITRE, the reconnaissance phase occurs when the adversary is gathering information to plan future operations; the initial access phase occurs when the adversary is trying to get into the network; and the lateral movement phase occurs when the adversary is trying to move through on the compromised environment (MITRE, 2023e).

To perform an attack using the method described in ATTA&CK's Phishing for Information technique, the adversary usually has to set up some infrastructure to impersonate a legitimate entity and trick the victim into visiting the website. Although this is not always the case, as an adversary can perform an attack just by using direct email or message exchange (MITRE, 2023c) or by sending the victim an attachment file (MITRE, 2023b), the objective of this technique is to get the victim into disclosing sensitive information such as credentials or other actionable information (MITRE, 2023c).

Attack methods described under ATT&CK's Phishing technique (T1566 under TA0001), which consists of executing code on the victim systems, and Internal Spearphishing technique (T1534 under TA0008), which involves moving laterally on the organisation's network, fall outside the scope of this work.

A study based on a year-long network monitoring dataset concluded that the average phishing attack using websites to impersonate a legitimate entity, from the first to the last victim, persists for approximately 21 hours (Oest et al., 2020). In contrast, the detection "by anti-phishing entities occurs on average nine hours after the first victim" access (Oest et al., 2020, p. 361).

Deploying a phishing kit by an adversary is also straightforward, as it usually simply requires uploading and extracting a phishing kit to a remote server. When free hosting providers are used, the adversaries are relieved from having to deal with custom domains or certificate generation, streamlining the process (Oest et al., 2018). For these reasons, swift detection by the defenders is critical to minimise the impact on the victims and the brands. Artificial Intelligence can help with this problem by detecting cyber threats and making real-time decisions (Chan et al., 2019).

Specialised detection applications extract classification features from phishing websites. Yang, Zhao and Zeng (2019) suggested a multidimensional feature phishing detection framework based on URL, web page code, and text features. Xiang, Hong, Rose and Cranor (2011) proposed a comprehensive feature-based approach named CANTINA+ that uses web page source code, search engines and third-party services to detect phishing websites. Since these detection techniques are well-known to the adversaries, they often implement countermeasures on their phishing kits to subvert them. P. Zhang et al. (2021) detailed several methods adversaries use to cloak phishing websites using client-side techniques. Ndichu, Kim, Ozawa, Misu and Makishima (2019) noted that adversaries could use javascript to create dynamic content to hide their maliciousness and evade detection.

Jain and Gupta (2017) broadly classified visual similarity approaches into six categories: document object model (DOM), visual features, CSS features, pixel-based features, visual perceptions and hybrid features. Mahdavifar and Ghorbani (2019) surveyed the application of deep learning techniques to cybersecurity, including detecting phishing attacks. Zhao, Masood and Seneviratne (2021) reviewed computer vision methods in network security, grouping visual similarity-based phishing detection methods into three categories: image feature-based, image hashing-based and neural network-based.

CHAPTER 2 –METHODOLOGY

2.1. Research Questions

This research aims to collect insights into how artificial intelligence, using computer vision techniques, can support detecting and classifying phishing websites from screenshots taken from browser-rendered websites to support an organisation's brand protection efforts. This research also explores developing and implementing a system that detects if a screenshot acquired from a suspicious website showcases a phishing page targeting a specific protected brand. For these purposes, the main research question (RQ) is defined as:

RQ1: How can computer vision methodologies be used for phishing website detection and classification based on browser-rendered screenshots to support an organisation's brand protection efforts?

Initially, five subquestions (SQ) were defined to focus the research, help derive the answer to the main research question during the systematic literature review, and investigate the current state-of-the-art:

SQ1: What are the current state-of-the-art techniques for detecting and classifying phishing websites using computer vision methodologies?

SQ2: What are the challenges and limitations of using computer vision methods for detecting and classifying phishing websites?

SQ3: What are the methodologies to perform image-based feature extraction?

SQ4: What are the methodologies and their effectiveness in classifying phishing websites?

SQ5: How can these methodologies be used to help brand owners detect fraudulent websites?

An additional SQ proposes to explore the applicability of this research to solve practical brand protection phishing problems:

SQ6: How to recognise if a browser-rendered screenshot is displaying a phishing website for a specific protected brand?

It is worth noting that although the Systematic Literature Review (SLR) performed during this research to answer SQ1 to SQ5 focuses mainly on brand protection, the conclusions concerning artificial intelligence and computer vision are broad and will likely apply to other malicious threats perpetrated and delivered through phishing such as malware distribution and identity theft (Thomas, 2018).

2.2. Design of the Systematic Literature Review

An SLR aims to identify, evaluate, and synthesise all available evidence on a specific research question. It is a methodologic, rigorous and transparent approach that follows a set of predetermined steps to ensure the quality and reliability of the results.

Kitchenham and Charters (2007) stated that systematic review protocols should adhere to a predetermined set of steps to ensure the quality and reproducibility of the results. These procedures involve formulating the research question, identifying relevant studies, selecting studies according to established inclusion and exclusion criteria, extracting and synthesising data from selected studies, and evaluating the quality of the evidence collected.

This section describes the steps to conduct an SLR based on the protocol proposed by Kitchenham and Charters (2007) and explains how the studies were selected, analysed, and synthesised.

It is worthy of note that this research seeks to identify how computer vision methodologies can assist in determining if a website is hosting a phishing web page by disguising it as a company's specific brand, thereby enabling businesses to develop or improve their brand protection and customer protection programmes. This systematic literature review does not intend to survey how to detect suspicious domains or URLs but to determine if a screenshot acquired from a specific rendered website showcases a phishing kit featuring a protected brand.

2.2.1. Procedure for the Study Selection

This research used a six-step methodology to ensure that documents matching the highest quality passed through a phased selection based on strict criteria. This research procedure also specifies how disagreements between the author and the advisor were managed to avoid bias during document selection. The study selection procedure synthesised in Figure 1 provides a visual perspective of the overall process.

The definitions of the Inclusion Criteria (IC) and the Exclusion Criteria (EC) are detailed in Subsection 2.2.3.

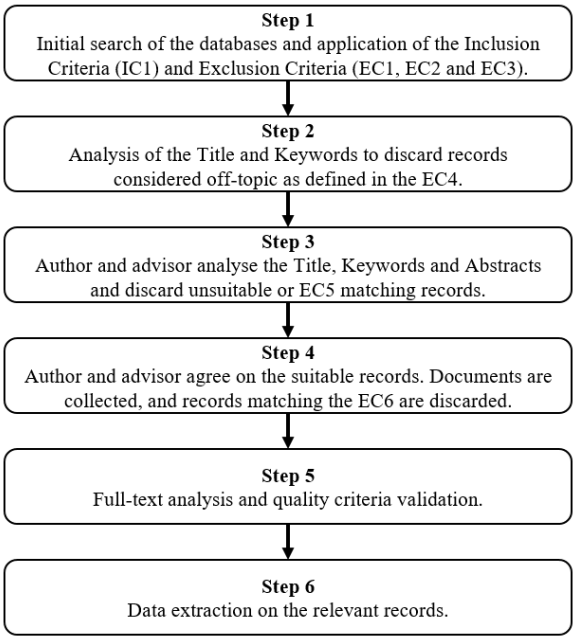


Figure 1: Methodology overview for the systematic literature review

Step one addressed the initial search using the criteria detailed in the inclusion criteria IC1. It also applied EC1, EC2 and EC3 exclusion criteria to the query output.

Step two covered the title and keywords screening to determine whether the document was considered off-topic, possibly off-topic or on-topic as defined in EC4. This step excluded all documents deemed off-topic.

In step three, both the author and the advisor read the title, keywords and abstracts and classified the documents as “relevant”, “maybe relevant”, and “not relevant”. The document analysis was done independently to avoid bias. Papers not complying with the EC5 resulted in a “not relevant” classification for this research.

In step four, the author and the advisor agreed on and collected the relevant documents for this research. EC6 was applied to the academic works unavailable to either the author or the advisor due to restricted access to some academic databases.

Step five consisted of a full-text analysis resulting in a set of documents complying with the quality criteria described in Subsection 2.2.4 for the final step.

Finally, the relevant information for answering the five subquestions and supporting the main research question was extracted in step six, as detailed in Subsection 2.2.5.

2.2.2. Databases for Literature Searching & Research Strategy

A manual search was performed on the ACM, IEEE Xplore, SCOPUS and Web of Science databases for this SLR.

As this research aims to determine how artificial intelligence, more specifically using computer vision for feature extraction, can detect and classify phishing screenshots from browser-rendered websites, the keywords “computer vision”, “image classification”, and “image processing” were used on for the queries on all searches. The OR boolean operator ensured the research was as thorough as possible. Combining keywords using the AND boolean operator with the “phishing” keyword resulted in the following queries on the selected databases:

ACM: [[All: “computer vision”] OR [All: “image classification”] OR [All: “image processing”]] AND [All: “phishing”]

IEEE Xplore: ((“All Metadata”: “computer vision”) OR (“All Metadata”: “image classification”) OR (“All Metadata”: “image processing”)) AND (“All Metadata”:phishing)

SCOPUS: (TITLE-ABS-KEY(“computer vision”) OR TITLE-ABS-KEY(“image classification”) OR TITLE-ABS-KEY(“image processing”)) AND TITLE-ABS-KEY(phishing)

Web Of Science: (ALL=(“computer vision”) OR ALL=(“image classification”) OR ALL=(“image processing”)) AND ALL=(phishing)

The “ALL” data field was used to query the ACM and Web of Science databases, whilst the “All Metadata” data field was used to query the IEEE Xplore database.

Upon reviewing the SCOPUS database query output, it was observed that the “ALL” field generated diverse material outside the SLR’s scope. This behaviour was determined to be caused by query matches on documents’ references rather than on documents themselves, as the “ALL” field also queries documents’ references. Therefore, the “TITLE-ABS-KEY” field, which includes abstracts, keywords, and document titles, was used, restricting the database queries on the SCOPUS database.

2.2.3. Definition of the Inclusion & Exclusion Criteria

The Inclusion Criteria detail the characteristics an academic work must have to be included in the SLR, whilst the Exclusion Criteria describe the elements considered irrelevant to the research. Defining the IC and EC helped reduce bias and increase the review’s validity by ensuring that only studies relevant to the research question were

included and that the review was manageable regarding the number of studies to be included and analysed.

This research considers the following IC and EC:

IC1: Abide into the structure for the queries identified in Subsection 2.2.2 on the ACM, IEEE Xplore, SCOPUS and Web of Science databases.

EC1: Language - Documents not written in English.

EC2: Date - Documents published before 2004.

EC3: Redundancy/Duplication - Duplicated documents listed in several database sources (based on the DOI).

EC4: Scope - Documents considered "off-topic".

EC5: Originality - Secondary studies such as literature reviews or surveys.

EC6: Availability - Documents without the full text available.

Some examples of "off-topic" documents excluded by EC4 were papers about website defacement detection techniques, documents on the creation of phishing websites for adversarial emulation purposes, and documents related to phishing email detection techniques and non-directly related to phishing website detection based on its screenshot.

2.2.4. Definition of the Study Quality Criteria

The study Quality Criteria (QC) help ensure the relevant studies for this review. Relevant studies are more likely to provide accurate and reliable results, increasing the review's validity and credibility. Additionally, it allowed for identifying any limitations or weaknesses in the studies included, which helped to interpret the review's findings in context. Furthermore, this assessment was an essential step in the data synthesis process, as it helped to weigh the evidence and make judgments about the overall quality and strength of the evidence base. Table 1 provides the weights of the QC used in this research.

Table 1: Definition of the weights for the quality criteria

ID	Definition	Weight
QC1	Does the document use computer vision or image processing techniques to extract features or classify rendered web page screenshots?	2
QC2	Does the document present the applied methodologies and their effectiveness?	2
QC3	Is the research methodology adequately documented?	2
QC4	Is the purpose of the document clear?	1
QC5	Is the dataset used during the research explicitly identified and available?	1
QC6	Does the dataset used during the research have at least 1,000 screenshot samples?	1
QC7	Was the document cited in the last year (2022)?	1

The weights assigned to each QC reflect the relative importance of each criterion for this research. During the SLR's step 5, illustrated in Figure 1, all the inputted studies were evaluated and classified according to these criteria.

2.2.5. Data Extraction

The systematic process aimed to collect and organise relevant data from the studies included in the review in a consistent and structured manner is named data extraction. It allows for an accurate and comprehensive comparison while seeking to answer the research question and subquestions.

The most relevant metadata required to answer the central research question RQ1 and research subquestions SQ1 to SQ5 was identified and consolidated into a data extraction form template whose primary objective was to structure the extracted data and allow for a seamless consultation and comparison between studies. Appendix A details the developed data extraction form template in Table A1.

In this research, the data extraction process consisted of the qualitative reading of the relevant documents, identified as the output of SLR's step 5 by the author, illustrated in

Figure 1. The advisor mentored and audited the entire process to prevent any bias and ensure the quality of the process.

All studies that reached the SLR's step 6, illustrated in Figure 1, were analysed, and the information considered relevant to this research was extracted and consolidated in a study data extraction sheet.

2.3. Computational System Engineering Design

The field of computation design plays a pivotal role in developing complex systems. NASA defines system engineering as "a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system" (Hirshorn et al., 2017, p. 3).

By adhering to the systematic and iterative approach outlined in NASA's System Engineering (SE) engine, significant insights are gained into the ideas, techniques, and best practices associated with computation design while emphasising its crucial role in system design and development. Figure 2 showcases the nine core processes that represent the execution of a project as defined by NASA's SE.

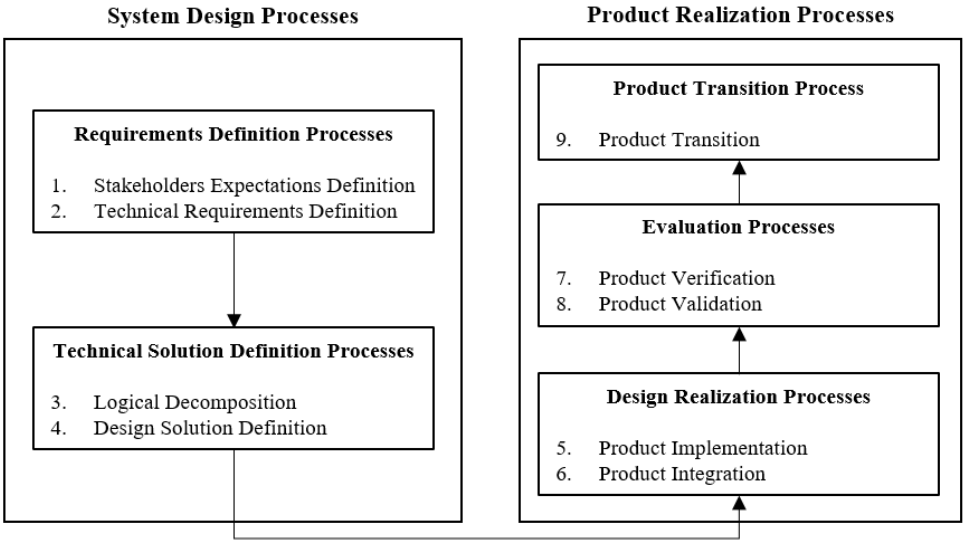


Figure 2: Systems engineering tasks

Source: Adapted from "NASA Systems Engineering Handbook" by Hirshorn et al., 2017, p. 6

This section provides insights into the System Design, specifically the Requirements Definition Process for the BlitzPhish tool. The Technical Solution Definition Processes and Product Realization Processes are explored in Chapter 4 as they are derived from both this section's output and the Systematic Literature Review in Chapter 3. The Evaluation Processes are also detailed in Chapter 4, showcasing the research results. Finally, this research does not explore the Product Transition Process, as the final product was not integrated into any existing system or transitioned to the end user for this research.

2.3.1. Requirements Definition

The Requirements Definition Processes include the Stakeholder Expectations Definition process, which aims to identify the stakeholders, particularly the end users, and how they intend to use the product and the Technical Requirements Definition process, where the end users expectations are converted into technical requirements, articulated as "shall" statements (Hirshorn et al., 2017). These requirements were then prioritised according to the MoSCoW technique (Hudaib et al., 2018) in "must have" (Mo) requirements, "should have" (S) requirements, and "could have" (Co) requirements. The "won't have" (W) requirements were not included as Technical Requirements.

The stakeholders' expectations for this research directly link with the technical requirements definitions as the author derives both. The following list summarizes the end users' expectations:

Expectation 1: To develop a tool that solves a practical brand protection phishing problem.

Expectation 2: The developed tool must derive an answer to research subquestion SQ6.

Expectation 3: The developed tool must support the realization of an answer to the main research question RQ1.

The developed tool shall meet the following Technical Requirements, prioritised according to the MoSCoW criteria:

Requirement 1 (Mo): A Machine Learning (ML) classifier must be used to evaluate whether a browser-rendered screenshot showcases a phishing web page targeting a specific brand or website requiring protection.

Requirement 2 (Mo): It must perform ML model training and evaluation using a custom brand dataset that requires protection.

Requirement 3 (Mo): It must be modular, allowing components to be executed in distributed environments if needed.

Requirement 4 (S): It should be developed in Python as it supports extensive libraries and frameworks designed for machine learning, allows for rapid prototyping and iterative development, is cost-effective and has a broad industry adoption.

Requirement 5 (S): It should provide the end user with a PHP-developed web interface, allowing them to explore the acquired data and execute operations.

Requirement 6 (S): It should support the complete workflow, artefact collection, and management from identification and detection to response.

Requirement 7 (S): It shall provide an Application Programming Interface (API) to support the integration with external systems or between modules.

In summary, a well-defined roadmap has been established for designing and deploying the BlitzPhish tool by identifying the stakeholders' expectations and converting them into prioritised technical needs.

CHAPTER 3 – SYSTEMATIC LITERATURE REVIEW

3.1. Results of the Systematic Literature Review

Figure 3 summarises the number of inputs and outputs in each step of the SLR research process. In early January 2023, the procedure documented in Subsection 2.2.1 guided the development of the first step of the research methodology.

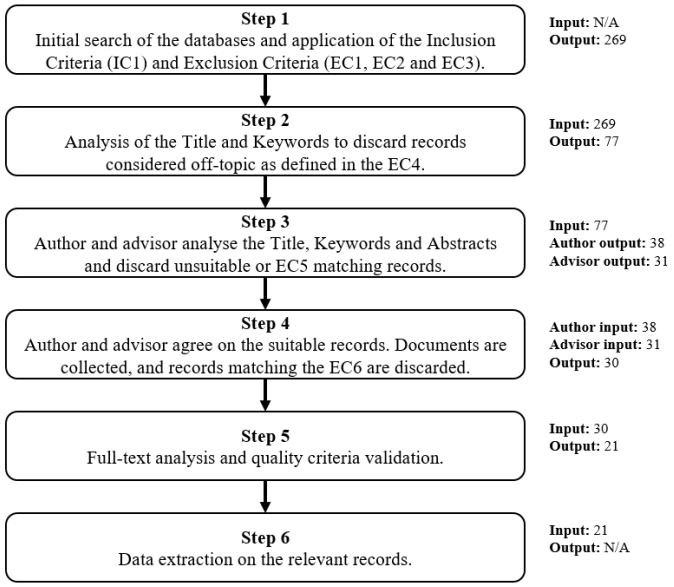


Figure 3: Results of the systematic literature review methodology applied to this research

The initial search on the ACM, IEEE Xplore, SCOPUS and Web of Science databases identified 314 studies matching the Subsection 2.2.2 queries. The EC1, EC2 and EC3 were applied to this list, resulting in 269 documents.

In step two, the 269 documents' titles and keywords were screened, and off-topic papers were excluded as per EC4, resulting in 77 articles. As 192 articles were deemed off-topic in step two, a TOP20 frequency word check was performed on the excluded articles' titles to validate the consistency and effectiveness of EC4. Table 2 presents the result of this frequency word check. The keywords related to phishing or computer vision mentioned in Table 2, such as "authentication", "phishing", and "visual", were used to confirm the correct application of EC4 to the related documents and that applying EC4 had not mistakenly suppressed relevant documents.

Table 2: TOP 20 words used on document titles excluded by exclusion criterion 4

Title Words	Number of Occurrences	Title Words	Number of Occurrences
security	19	based	9
detection	15	automated	9
adversarial	13	data	9
learning	13	detecting	8
authentication	13	privacy	8
attacks	11	survey	8
System	11	machine	7
mobile	10	secure	7
Visual	10	phishing	6
malware	10	against	6

The author and the advisor independently analysed titles, keywords and abstracts for the 77 remaining articles in step three. The EC5 was also evaluated against the dataset, and matching documents were labelled “not relevant”. At the end of this phase, both the author and the advisor had classified the documents as being “relevant”, “not relevant”, or “maybe relevant” to this research. The author classified 39 documents as “not relevant”, 10 as “maybe relevant”, and 28 as “relevant”. The advisor classified 46 documents as “not relevant” and 31 as “relevant”.

In step four, the author and the advisor met and agreed on the studies for this research. The EC6 was applied to restrict the SLR analysis to academic works with full text available to the author and the advisor, resulting in 30 documents.

Step five consisted of the in-depth analysis of the 30 documents and their assessment given the study quality criteria described in Subsection 2.2.4. Only records with a weighted average value of 5 or greater were considered for step 6, resulting in the 21 documents. The

complete list of studies that cleared the selection procedure is detailed in Table 3, ordered by ascending publication year.

Table 3: List of studies analysed on step 6 ordered by publishing year ascending

Study ID	Document Title	Study Reference
S-01	Cantina: a content-based approach to detecting phishing web sites	(Y. Zhang et al., 2007)
S-02	Counteracting Phishing Page Polymorphism: An Image Layout Analysis Approach	(Lam et al., 2009)
S-03	Detecting visually similar Web pages: Application to phishing detection	(Chen et al., 2010)
S-04	PhishZoo: Detecting Phishing Websites by Looking at Them	(Afroz & Greenstadt, 2011)
S-05	Judging a site by its content: learning the textual, structural, and visual features of malicious web pages	(Bannur et al., 2011)
S-06	Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach	(H. Zhang et al., 2011)
S-07	Utilisation of website logo for phishing detection	(Chiew et al., 2015)
S-08	A Computer Vision Technique to Detect Phishing Attacks	(Rao & Ali, 2015)
S-09	Use of HOG descriptors in phishing detection	(Bozkir & Sezer, 2016)
S-10	Phish-IRIS: A New Approach for Vision Based Brand Prediction of Phishing Web Pages via Compact Visual Descriptors	(Dalgic et al., 2018)
S-11	Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild	(Tian et al., 2018)

Study ID	Document Title	Study Reference
S-12	Image Based Phishing Detection Using Transfer Learning	(Phoka & Suthaphan, 2019)
S-13	VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity	(Abdelnabi et al., 2020)
S-14	Intelligent phishing detection scheme using deep learning algorithms	(Adebowale et al., 2020)
S-15	LogoSENSE: A companion HOG based logo detection scheme for phishing web page and E-mail brand recognition	(Bozkir & Aydos, 2020)
S-16	Privacy-Preserving Phishing Web Page Classification Via Fully Homomorphic Encryption	(Chou et al., 2020)
S-17	Combining Text and Visual Features to Improve the Identification of Cloned Webpages for Early Phishing Detection	(van Dooremaal et al., 2021)
S-18	Phishing Detection Using Computer Vision	(Khandelwal & Das, 2022)
S-19	SenseInput: An Image-Based Sensitive Input Detection Scheme for Phishing Website Detection	(Lin et al., 2022)
S-20	Phishing Website Detection Using Deep Learning	(Siddiq et al., 2022)
S-21	Leveraging Deep Learning Image Classifiers for Visual Similarity-based Phishing Website Detection	(Trinh et al., 2022)

Step six analysed the remaining 21 documents and extracted the information pertinent to this research using the form template outlined in Table A1. The consolidated document's sheets enhanced comparison and quick reference, facilitating answering subquestions SQ1 to SQ5 and supporting answering the main research question RQ1.

3.2. Discussion

This section delves into examining the state-of-the-art phishing website detection using computer vision. As the digital landscape undergoes constant transformation, the strategies adopted by adversaries aiming to deceive and exploit unwary users adapt accordingly. The present section conducts a comprehensive analysis of the existing body of literature from January 2004 to January 2023 using a structured approach.

3.2.1. Answer to the Research Subquestion 1

Detecting phishing websites by analysing the web page appearance is not a novel idea, though it has evolved in methods and methodologies over the years.

In 2007, Y. Zhang et al. (2007) published a paper where they rolled out a content-based approach to detecting phishing websites named CANTINA. Although this approach does not use computer vision to detect phishing, it uses several features extracted from a website to classify it as phishing or not phishing, including Term Frequency – Inverse Document Frequency (TF-IDF), which measures the importance of a word in a document's body. CANTINA was evaluated against some standard toolbars, implemented through browser add-ons to detect phishing and performed well. The authors envisioned that an adversary could attack the TF-IDF algorithm by modifying the website. They suggested simple computer vision methodologies could mitigate these adversarial countermeasures and CANTINA and TF-IDF limitations (Y. Zhang et al., 2007).

In 2009, Lam et al. (2009) proposed an image layout analysis approach to counteract phishing page polymorphism. As Lam et al. (2009, p. 271) stated, "Phishing pages have to be quite similar to the authentic pages in order to deceive users". Thus, their work focused on developing a technique to detect polymorphic phishing pages. They used web page layout similarity using images rather than pixel information. They took partial similarity into account, thus maintaining high accuracy even when adding, replacing or purging colours or image blocks. Lam et al. (2009) compared their work with the image comparison approach proposed by Fu, Wenyin and Deng (2006), which used Earth Mover's Distance (EMD) algorithm to compare the similarity between a phishing web page and its legitimate counterpart, having concluded that their method was more efficient and accurate.

Chen et al. (2010) proposed the usage of a Normalized Compression Distance (NCD) technique to determine the distance between a legitimate site and its potential phishing counterpart. Their work considered that a website "must be treated as indivisible entities (i.e., a whole) to be congruent to human perceptions" and applied this technique to protect

specific websites or brands rather than trying to detect phishing websites at large (Chen et al., 2010, p. 35).

The PhishZoo tool, developed by Afroz and Greenstadt (2011), uses a mixed approach to detect phishing websites. It profiles a legitimate website using several features such as the URL, the HTML code and images such as logos. When a user visits a website, features are extracted and matched to determine the likelihood of being phishing or not. The image feature extraction and comparison use the Scale Invariant Feature Transform (SIFT) algorithm, which calculates the Euclidean distance between image key points. The authors stated that their approach would be helpful to large organisations seeking to find phishing sites to improve website access blocking or to issue takedown notices.

Bannur et al. (2011) also used SIFT as part of their approach to detecting phishing websites. They experimented using Linear Regression (LR) and Support Vector Machine (SVM) as classifiers for phishing websites, using several content-based features and several visual features extracted with SIFT (sift-stats). They concluded that although the visual features significantly impacted computation requirements, using the sift-stats features permitted this information to be included in the model while minimising the computation impact. In their work, combining structural with the sift-stats visual features resulted in significant drops in error rates.

In 2011, H. Zhang et al. (2011) proposed a framework that employs a Bayesian approach to detect phishing websites based on textual and visual content. Their work extends the EMD method Fu et al. (2006) presented into a hybrid framework that uses textual features extracted from HTML code and image features, rendered web page screenshots. Images are classified using a visual similarity assessment based on EMD, and text is classified using Naïve Bayes (NB) rules. The results of both classifiers are combined using a data fusion algorithm implemented using a Bayesian approach. The authors also considered integrating these results into browser plugins or APIs to protect users from phishing web pages targeting sites that need high-security attention.

Chiew et al. (2015) used ML algorithms to identify and extract logos from the content downloaded while accessing the website. Contrary to other methods, the website logo is recognised from the cached resources downloaded while accessing the website and not pulled from a web page screenshot. The logo is then matched to a domain by using Google Image search. This work assumed an exclusive relationship between a logo and a domain name, which is seldom true, especially when dealing with multinational organisations or organisations that provide services to different geographies.

In 2015, Rao and Ali (2015) experimented with the Speed Up Robust Features (SURF) algorithm to detect phishing websites. Their work considered the pre-existence of a list with "known good" website screenshots and their matching URLs. A suspicious website is compared for similarities with the ones on this "known good" dataset, and if the similarity is higher than the threshold and the URL does not match the known one, the website is deemed phishing.

Histogram of Oriented Gradients (HOG) descriptors is a type of feature descriptor used in computer vision and image processing, introduced by Dalal and Triggs in 2005 as a method for pedestrian detection (Dalal & Triggs, 2005). In 2016, Bozkir and Sezer (2016) explored using the HOG descriptors in phishing detection. Like the work from Rao and Ali (2015) using SURF, a suspicious website screenshot image is compared to a "known good" dataset. The website is labelled as suspicious if the similarity exceeds the predefined threshold.

In 2018, Dalgic et al. (2018) employed several MPEG7 and MPEG7-like compact visual descriptors to classify browser-rendered website screenshots. Their dataset included phishing screenshots for some heavily phished brands and an "other" category where they tagged legitimate websites. They used a Random Forest (RF) and an SVM as a classifier methodology, having evaluated their method against the HOG implementation by Bozkir and Sezer (2016).

Domain squatting, also known as cybersquatting, is the practice of registering, trafficking or using a domain name to profit from someone else's trademark. Domain squatters often register domain names similar to popular or trademarked names or brands, hoping to sell the domain name, generate revenue through advertising, redirect traffic to another website, or use it for phishing or other malicious purposes (Buber et al., 2017). In 2018, Tian et al. (2018) proposed an ML classifier to detect phishing web pages from squatting domains. This approach had novel features, such as retrieving the suspicious website screenshot by rendering the website on a browser and manipulating its user-agent header to bypass adversarial countermeasures. Their work used HTML elements as input features for the classifier. Instead of using the screenshot as input for the classifier, they extracted textual elements from the screenshot using Optical Character Recognition (OCR) and fed those features to the classifier. They used NB, RF and K-Nearest Neighbor (KNN) algorithms to assess performance.

Transfer Learning refers to transferring knowledge, e.g., models, derived in one setting and exploiting them to improve generalization in another domain (Goodfellow et al.,

2016). Instead of starting the training process from scratch, the pre-trained model provides a starting point, transferring its learnt capabilities to a new model trained on the target task or domain. This technique can result in faster training times, improved performance, and reduced need for large amounts of labelled data for the new job.

Phoka and Suthaphan (2019) proposed a method to detect phishing pages based on the whole image of a website screenshot. They used Convolutional Neural Networks (CNN) to perform image classification in a binary (phishing / not-phishing) and multiclass (5 different brands) classification problem. They concluded that training a CNN model takes considerable time and requires a large dataset. As such, the authors used data augmentation to expand the training dataset and Transfer Learning, using a pre-trained model, to minimize the training time.

Abdelnabi et al. (2020) developed a similarity-based phishing detection framework called VisualPhishNet. This framework used a triple CNN and a similarity metric to classify phishing web pages with new appearances, mitigating some evasion and perturbation techniques the adversary may implement on the phishing kit. They employed visual similarity generalization to detect unseen phishing pages that target trusted websites despite having different contents.

Adebowale et al. (2020) detailed a new phishing detection system that combined CNN and Long Short-Term Memory (LSTM) algorithms to build a hybrid classification model named Intelligent Phishing Detection System. This system used hybrid features extracted from images, text and frames. The authors concluded that CNN performed better regarding time lapsed but was, on average, less accurate than LSTM. Combining the two methods led to better results while addressing the problem of a large dataset and better classification performance. As a preprocessing for the CNN, the authors cropped the images on the springing box and removed the wrong ones, whereas, for the LSTM, different features were collected and saved as a Comma-Separated Values (CSV) file. They used the AlexNet as the CNN model and the holdout cross-validation for the LSTM offline model.

Bozkir and Aydos (2020) proposed a HOG-based brand logo detection scheme named LogoSENSE. Their work focuses on recognizing a target brand on a phishing website rather than the entire website by detecting logos using Max-Margin Object Detection (MMOD). King (2015) introduced MMOD as a machine-learning algorithm for image object detection. It relies on maximizing the margin between positive and negative samples. Positive samples refer to the objects of interest that need to be detected, and negative refer to the background or other elements that are not of interest. The MMOD algorithm uses an SVM classifier on

a set of labelled images, which is trained to learn the visual features of the positive and negative samples. It then uses these features to detect objects in new images. The research by Bozkir and Aydos (2020) intended to recognize zero-hour phishing websites targeting predetermined brands.

Not directly linked to detecting and classifying phishing websites using computer vision but relevant to this field, Chou et al. (2020) proposed a method that allowed the classification of phishing web pages while preserving privacy using homomorphic encryption. Homomorphic encryption is a cryptographic technique that enables computation on encrypted data without decrypting it first. It allows for the processing and analysis of sensitive data without compromising privacy, confidentiality, or security. The authors considered the existence of a third-party cloud service that evaluates whether a given website screenshot is phishing. They stated that uploading an image to this third-party provider represents a severe privacy risk and potentially has a considerable bandwidth footprint. As such, they proposed a method to locally extract features from a screenshot, seal them using homomorphic encryption and only then query the cloud service. The cloud service computes an encrypted phishing score and returns the result. The phishing results are then decrypted and evaluated by the client to take action.

In 2021, van Dooremaal et al. (2021) proposed a method to improve the detection of cloned web pages by combining textual and visual features. The method extracts visual features from screenshots and textual features from a web page's DOM. The authors used several classifiers, such as EMD, Discrete Cosine Transformation (DCT), Euclidean distance for pixel similarity (PSIM), Structural Similarity Index Measure (SSIM), and Oriented FAST and Rotated BRIEF (ORB), to calculate the performance against the individual and combined features. Extracted image regions with identifiable information are reverse-searched on a search engine to determine whether the page is legitimate.

In 2022, Khandelwal and Das (2022) discussed employing Transfer Learning and Representation Learning techniques using various off-the-shelf CNN architectures such as VGGNet, GoogLeNet, ResNet and DenseNet to extract image features and apply RF and SVM to build the machine learning classifier. Since their work considered the detection of phishing for multiple brands, they used a multiclass classifier instead of a binary classifier.

Lin et al. (2022) proposed an image-based sensitive input detection scheme that detects HTML forms asking for user information such as usernames and passwords. Their work uses Faster-RCNN architecture by Ren, He, Girshick and Sun (2015) to address the input detection problem, OCR to extract sensitive information from the input image and the

tree-based LightGBM classifier. It implements the CANTINA+ rule-based approach proposed by Xiang et al. (2011) and extends it with new features, resulting in more accurate results.

Siddiq et al. (2022) proposed using the Conv2D CNN model for phishing detection. Using a publicly available textual dataset, they converted each sample into a numerical array using the OneHotEncoding technique, ending up with a multidimensional vector per sample. OneHotEncoding is a technique used to represent categorical data as numerical data. Each value transforms into a binary vector of 1s and 0s, representing a possible category value. Since Conv2D addresses image classification problems, the authors reshaped the samples into an 8x8 matrix.

Trinh et al. (2022) explored the Transfer Learning technique using pre-trained image classification models to extract features for an ML model to classify a suspected phishing website binarily. The author's method consisted of two stages: first, training the image classification models with Deng et al. (2009) ImageNet dataset; second, through Transfer Learning, the pre-trained models support extracting features from screenshots, while ML algorithms classify binarily the websites as phishing or no phishing.

In sum, the current state-of-the-art techniques for detecting and classifying fraud websites using computer vision methodologies are characterised primarily by the extensive use of CNNs for feature extraction and classification. Recent research in the field, from 2020 to 2022, has consistently demonstrated the superiority of CNNs, mainly when applied to whole-page screenshots as input data. While some studies employ non-visual techniques such as text and website frames to complement visual analysis, CNNs remain the predominant method for obtaining reliable results. Although challenges such as dataset imbalances persist and processing costs are more pronounced in real-time analysis, the demonstrated progress highlights the critical role that computer vision techniques play in enhancing cybersecurity strategies.

3.2.2. Answer to the Research Subquestion 2

Like the state-of-the-art analysis, the challenges and limitations of using computer vision methods in phishing detection have evolved. New approaches, techniques, and higher computer processing capabilities have mitigated some challenges and constraints identified in older studies.

Y. Zhang et al. (2007) found that the TF-IDF algorithm struggled with East Asian languages and that an adversary could attack this algorithm using several techniques. Some

of these techniques included changing the phishing page by converting text into images or having invisible text on the page that is imperceptible to the user but negatively impacts the algorithm. Their method, CANTINA, also depends on Google Page Rank, which can be attacked. For instance, an adversary can compromise a legitimate website and use it to host the phishing website used on the campaign or allow the phishing domain to be indexed by the search engine before launching the phishing campaign. Using Google also represents a challenge regarding the latency introduced by performing the search query and waiting for the results.

Lam et al. (2009), while comparing their work with the EMD algorithm, concluded that EMD is not resilient against changes in the image aspect ratio as it requires all the compared screenshots to have the same width and height. As a result, if an adversary modifies the phishing page's aspect ratio, it will decrease the efficiency of the EMD method.

The PhishZoo tool by Afroz and Greenstadt (2011), which uses SIFT to extract the logo from a website, fails to find a match if the logo is rotated more than 30 degrees with other website elements changes. This limitation can only be circumvented by decreasing the threshold used in their work, though the authors state that it would render the false positive ratio unacceptable. According to the authors, the adversary can combine the manipulation of the logo and HTML files to avoid detection. The only way to protect against this attack is to use screenshots of the rendered websites. However, the authors state that this approach would reduce accuracy and increase the number of false positives. Additionally, even though matching every website image with all the protected website artefacts increases accuracy, it slows the website rendering to unacceptable levels if used in real-time.

Bannur et al. (2011) stated that visual features were more expensive to compute than web page structural features. Their work also identified the limitation of grouping all malicious web pages in just one class. Detection of malicious web pages might require different approaches and the selection of various features depending on the website footprints. Additionally, to keep the system updated, adapted and tuned to new attacks and pages, the system would have to be held online, which could expose it to exploitation attempts. In such cases, the utility of a malicious web page filter is highly connected to the user experience.

The framework by Chiew et al. (2015) extracted logo image files downloaded during the website loading process to perform a reverse Google image search and detect the legitimate website domain. The authors found that a web page can have multiple logos, such as a login form that allows users to authenticate with federated identities, such as social

media identity providers. However, they claim this is not a critical limitation as the proposed method aggregates the results of all comparison queries and returns successfully if at least one match exists. Another restriction of such a method is that several web page images can have similar characteristics to a logo. As such, the authors claimed that an effective logo extraction process ought to improve the accuracy of the phishing detection process. They suggested using a rendered web page screenshot and applying object segmentation techniques to extract the web page logo for enhanced logo extraction.

The method by Rao and Ali (2015), which relies on using the SURF algorithm to detect phishing websites, fails if the phishing website style is changed significantly or replaced with advertisements. The authors stated that even an unsophisticated user could identify the differences and be awry if this happens. While recognizing the potential for their research to detect zero-day phishing attacks, alternative approaches to improve the computation cost and accuracy results went unmentioned.

Bozkir and Sezer (2016) used HOG descriptors to capture visual cues from website layouts and detect the similarity of unknown websites to a known database of known good websites. This method is an efficient and easy way to compare layout signatures. However, the authors suggested that replacing all images with a placeholder before performing feature extraction could enhance the approach. This way, the process would not be mangled when websites present different backgrounds on each access, in what they called “image content invariance”. Similar to Lam et al. (2009) work, another identified limitation is that the approach requires all images to have the same width and height.

Dalgic et al. (2018) proposed a new approach named Phish-IRIS and compared it against the research by Bozkir and Sezer (2016). The authors claimed to have produced a lightweight solution that could be used as a browser add-on or on a mobile device to detect phishing and allowed for images of different heights. However, the authors had to create a new class in their multiclass classifier to support all legitimate websites not protected by their solution. Including “other” web page screenshots lacking standard features posed an open-set classification challenge. Compounding this, the authors found no available datasets for this research and thus generated their own, releasing it publicly to facilitate further work without wasted efforts recreating proprietary data.

Tian et al. (2018) found that phishing kits could cloak their activity as a detection countermeasure depending on the browser and the operating system used to render the website. The crawler they used to acquire the screenshots only had two profiles defined, so the authors stated they may have failed to detect sites that targeted users on platforms other

than the ones mimicked by the crawler. Bypassing adversarial countermeasures is a challenge and a novelty of this research. The authors identified their research focusing only on the "popular brands" defined on the Alexa ranking[2] as a limitation. In future work, they also proposed adding other specific domains as classes to the multiclass classifier. Finally, the authors had difficulty comparing their classifier's performance with other studies as most works do not open-source their tools. They listed the difficulty and the cost of obtaining certain features for large-scale datasets, which were also identified as issues.

Phoka and Suthaphan (2019) actively identified the labour-intensive task of manually labelling sub-images, rendering their Transfer Learning approach impractical for real applications. Their work failed to account for pages featuring dynamic or event-driven content updates. Added images or content on login pages could negatively impact the proposed method. However, these limitations did not universally affect all phishing websites, so the effectiveness of the approach depended on the specific phishing problem addressed. Some sites fell outside the limitations, but others faced challenges that could undermine the approach.

Abdelnabi et al. (2020) work, similar to the study by Dalgic et al. (2018), found that the existing datasets had some limitations, including the one introduced by Dalgic et al. (2018), which, according to the authors, has a limited number of trusted websites and the screenshots of the phishing websites collected from phishing reports rather than from the live websites. To overcome this limitation, the authors released a new dataset named Visual-Phish. During their research, the authors analysed the false negatives generated by their model. They inferred that extracting text from a screenshot via OCR or incorporating logo feature detection could be used as model optimization. Another issue pertained to the screenshot width and height, which the authors solved by fixating the browser window size at the time of the screenshot acquisition. Finally, maintaining a trusted domain list was identified as a potential issue affecting the model if a website changes its domain. However, the authors suggest a rolling trusted-list update mechanism, removing the model retrain requirement.

LogoSENSE work by Bozkir and Aydos (2020) performed well, though several factors, such as image size and the upsampling zoom factor, may be impactful. The authors observed that web pages with high heights have their detection speed drastically affected. Additionally, the MMOD schema frequently accesses unnecessary areas of the visual

[2] https://www.alexa.com/

content, impacting the precision score and the running time. This schema is affected by the semi-rigid representation of the HOG features, which the authors planned to overcome by exploring the usage of a CNN detection module in future work.

Chou et al. (2020) focused on a framework to ensure that privacy was maintained when using a cloud service to determine if a given screenshot is phishing while being performant and with a low bandwidth footprint. The problem the authors proposed to address is that a user may insert sensitive data on phishing websites before a cloud phishing detection model checks the submitted website features. Seeking to ensure data security as information transfers to cloud-based platforms, the authors centred their work around preserving confidentiality through the model training and classification processes via homomorphic encryption. This technique allowed training the model on encrypted data in the cloud and returning an encrypted classification to the user-provided features without decryption. By leveraging homomorphic encryption, the authors developed an approach to gain the computational benefits of cloud resources while keeping sensitive information safe from exposure during all stages of remote processing.

In 2021, van Dooremaal et al. (2021) proposed using textual and visual features to detect phishing websites. The authors have shown that whilst the visual features positively impact classification, as their approach relied on reverse image searching on an external search engine, it could be affected by several constraints, such as low internet speed or data cap. Another challenge was identifying the number of regions to search inside the image as a trade-off for identification accuracy. The lower the number of regions to compare, the faster the algorithm runs, though less accurately.

Lin et al. (2022) used OCR to extract text from images, specifically text used on forms requesting user-sensitive data. Such an approach allows for overcoming rule-based detection countermeasures implemented by adversaries. The authors concluded that phishing dataset bias was causing gaps in the validation and testing datasets regarding the ability to detect phishing websites, as showcased in the results recall metric.

Siddiq et al. (2022) concluded that although their Conv2D CNN model took very little time to train, feature extraction may be time-consuming in real-time systems. Additionally, the authors identified the required hyperparameters fine-tuning to improve accuracy as needing additional research.

Finally, Trinh et al. (2022) identified that the complexity of the visual feature extraction process is a drawback for adopting visual-similarity phishing detection methods. They proposed to mitigate this issue by using Transfer Learning techniques, which employ

pre-trained deep-learning models to extract features from screenshots to optimize the feature extraction process, using its output as input for the classifier. The authors also identified the training performance issues caused by the dataset imbalances as a problem requiring additional research.

To summarise, the prevalence of imbalanced datasets across analysed studies surfaces as a critical issue, negatively affecting classifier performance. Additionally, while computer vision techniques have advanced, processing costs remain a concern, particularly in real-time analysis scenarios requiring immediate action, such as browser add-ons.

3.2.3. Answer to the Research Subquestion 3

The feature extraction methodologies from the analysed studies were extracted and summarised to answer this research subquestion. Table B1 in Appendix B describes the feature extraction methodologies used in the reviewed studies pertinent to this study.

It is worthy of note that several studies (S-01; S-03; S-12; S-20) were found to use pre-existing datasets where feature extraction was not performed, relied solely on textual or URL-based features, and did not consider visual features, or the feature extraction process documentation is incomplete. Although relevant to this research, these studies were not considered particularly useful in answering this research subquestion.

Other studies (S-04; S-05; S-06; S-11; S-14; S-19) were analysed as mixed visual features with non-visual ones, extracting both. Only the extraction methodologies related to images or computer vision were considered for this research subquestion.

In conclusion, CNNs are utilised extensively in image-based feature extraction techniques for phishing website detection. Recent research in this field emphasises the prevalence of CNNs, which are helpful for autonomously extracting relevant features from screenshots of entire web pages. These CNN-based methods systematically analyse visual elements, layout, and content to distinguish between phishing and legitimate websites. Additionally, some studies supplement visual analysis with non-visual techniques, such as text or website frame analysis, enhancing the feature extraction process.

3.2.4. Answer to the Research Subquestion 4

Similarly to SQ3, which provided a summary of the visual feature extraction methodologies, the classification methodology and their effectiveness in classifying phishing websites were extracted and summarised from the analysed studies. The extraction is described in Table C1 in Appendix C, and the used metrics are described in Appendix D.

To answer this research subquestion, classifiers using visual and non-visual features were extracted to provide the most detailed answer. The "Notes" column of Table C1 identifies the studies that used hybrid features or relied solely on non-visual features.

One specific study (S-16) was found to focus solely on feature extraction and did not adequately document the used classifier. As such, it was not considered for this subquestion.

In summary, the methodologies employed in classifying phishing websites primarily centre on using CNNs for classification, with SVMs also featuring as a notable choice in some cases. Recent research spanning 2020-2022 demonstrates CNNs' dominance, particularly when analysing whole-page screenshots as input data. These CNN-based algorithms efficiently distinguish between phishing and legitimate websites by exploiting visual cues, layout irregularities, and content anomalies. While SVMs, a more classic machine learning model, are employed occasionally, they are frequently recommended in conjunction with CNNs to overcome specific limitations. Problem-specific properties and the amount of available training data determine the selection of these models.

3.2.5. Answer to the Research Subquestion 5

The analysed studies were grouped into the two groups detailed in Table 4 to derive an answer to this research subquestion. The groups are:

Studies that are brand agnostic and focus on detecting all sorts of phishing websites, no matter what has been previously classified. These studies focused on protecting users from zero-day phishing attacks independently of any brand displayed on the phishing website. They resulted in an open-set classification problem and used methods to determine whether an unknown website should be classified as suspicious. Examples of such studies are studies that used search engines to distinguish legitimate domains from suspicious ones or whose model was trained using generic phishing datasets, not brand-oriented ones.

Studies focused on protecting specific brands or websites, using either a binary classifier to determine if a given website was phishing or a multiclass classifier to determine which brand was targeted by the phishing web page.

One study (S-16) was identified focusing solely on feature extraction and did not adequately document the used classifier. As such, it was not considered for this subquestion.

Table 4: List of studies grouped by brand protection classification categories

Brand Protection Classification Problem	Study ID
Generic (brand agnostic)	S-01; S-02; S-05; S-07; S-11; S-13; S-14; S-17; S-19; S-20; S-21
Targeted (brand / protected websites oriented)	S-03; S-04; S-06; S-08; S-09; S-10; S-12; S-15; S-18

Some studies (S-01; S-07; S-17) relied on search engines to determine the legitimate website pair using specific attributes. Other studies (S-02; S-05; S-13; S-14; S-19; S-20; S-21) used generic datasets based on websites reported to widely used phishing-reporting websites such as PhishThank or were collected indiscriminately from the targeted brands. These studies were classified as brand-agnostic. Additionally, although (S-11) considered specific brands to orient the research, they used more than 700 brands as a pivot, and the authors stated that the brands were selected due to being "famous". For these reasons, this particular study was also assigned as brand-agnostic.

The studies that were deemed brand-targeted either used lists of sensitive websites that required protection (S-03; S-04; S-06; S-08; S-09) or focused on a small amount of highly phished web pages (S-15) or specific login web pages exploited by adversaries (S-12). Additionally, (S-10; S-18) research used a dataset with well-defined phished brands.

In conclusion, these methodologies have the potential to significantly aid brand owners in detecting fraudulent websites that illegally exploit their brands for fraudulent activities. By focusing on the characteristics of phishing websites, these computer vision techniques can specifically target websites attempting to impersonate or defraud a specific brand or high-value website. Even though many studies have a broader phishing focus, some are oriented towards individual brand detection, presenting an opportunity for organisations interested in protecting their online identities and assets.

3.3. Chapter Conclusion

The study of SQs in Section 3.2 concluded that computer vision techniques evolved to extract features and classify phishing websites, solving ongoing challenges. Over the years, researchers actively used computer vision to address emerging issues, enhancing

extraction and classification abilities. The SQ analysis confirmed that techniques progressed, overcoming technical challenges with solutions for new problems arising.

Phishing websites often have characteristics that distinguish them from legitimate websites. Such characteristics may include fake URLs, inconsistencies in the design and layout, and questionable content, such as requests for personal information or urgent calls to action. By analysing these characteristics, developing new detection models and processes may support identifying whether a website will likely be phishing.

The findings presented in response to research subquestions SQ1 to SQ5 demonstrate the progression of computer vision techniques, mainly focusing on the prevalence of Convolutional Neural Networks (CNNs) for feature extraction and classification. Notably, the dominance of CNNs is evident in studies leading up to the present research period.

As showcased in the answers to SQ1, SQ2, SQ3, and SQ4, computer vision techniques to detect and classify phishing websites have evolved and can be successfully used for phishing detection and classification. Although optimization exists through adding structural and semantic features, current research provides relevant phishing classification results harnessing computer vision techniques. These methods performed well, leaving room to reinforce outcomes further by incorporating non-visual features, yet they actively demonstrated promising outcome capabilities for visual detection alone.

The feature extraction and classification techniques extracted during the answer to SQ3 and SQ4 demonstrate the predominance of CNNs for both tasks, mainly when focusing on the three years (2020-2022) before this research.

The analysed studies published during this period (S-13 to S-21) often used whole-page screenshots as the input for the feature extraction process. Focusing on this period, some non-visual techniques, such as text and website frames, were still used to complement the visual ones. In one case (S-16), OCR from graphical elements extracted textual features. In this example, the classification relied on textual features mined from web page screenshots, although extraction came indirectly through images rather than direct text analysis. The approach drew on screenshot content but classified using textual elements, providing an alternate processing path that bypassed direct text mining.

For classification, the usage of SVM and CNN was prevalent when focusing on the same period. One study (S-15) relied on more traditional machine learning models, such as SVM, and suggested using CNNs to surmount some identified limitations in future research. The model's choice depends on the problem's specific characteristics and the available data for training and testing the models.

It is worth noting that dataset imbalances are critical across some of the analysed studies as they negatively impact classifier performance. Additionally, although processing costs are still a concern for feature extraction, classification, and model training, they are more prevalent in endpoint software, such as browser add-ons, that require real-time analysis than in server solutions that deal with websites in batches.

Though many studies did not directly help organizations defend against malicious websites unlawfully using brands to defraud or target customers, focusing instead on broad phishing in general terms, as shown in SQ5's answer, some studies actively oriented around specific brands. Most works branched more widely, lacking brand-specific goals, but a few studies focused on individual company marks for enhanced validation. As organisations implementing an anti-fraud and brand monitoring system care about protecting their websites and brands, this is an opportunity. While phishing protection systems that protect users from generic phishing websites have to deal with a vast dataset of potential phishing artefacts and features, narrowing this down to a couple of specific brands or sensitive websites that require protection eases the extraction and classification problem. As put by H. Zhang (2011, p. 1534), a company "probably only cares about their own site, so it makes sense for them to detect fake versions of their own brand".

As such, studies grouped as "targeted" in Table 4 may support brand protection. Notably, restricting datasets to specific sites and brands requiring protection would likely allow the use of most "generic" studies for this targeted purpose.

This SLR highlights the potential for computer vision techniques in brand protection against phishing, with CNNs being the most prevalent in recent studies. As computer vision techniques continue to mature, they hold the potential to mitigate the impact of phishing attacks on brands and individuals. While challenges such as dataset imbalances and processing costs persist, the demonstrated progress reinforces the role of these techniques in enhancing cybersecurity strategies. The field is poised to contribute significantly to protecting online identities and assets by staying attuned to ongoing advancements and adapting to evolving threats.

CHAPTER 4 – COMPUTATIONAL IMPLEMENTATION AND ASSESSMENT

This section details the development of a cybersecurity tool meticulously designed to combat the growing threat of phishing attacks. The tool, BlitzPhish, employs a multifaceted approach, encompassing website detection, artefact acquisition, computer vision analysis, and responsive measures, all orchestrated to safeguard users and organizations from the insidious dangers of phishing websites.

4.1. Selection of Computer Methodologies

The primary objective of this work is the implementation of a CNN algorithm to classify whether a website screenshot is phishing. Although this is the focus of this research, additional modules and functionalities were developed for the completeness of the overall solution. Hence, this research implements a website detection and response platform with a web GUI (Graphical User Interface) and a web API (Application Programming Interface).

The detection, classification and response components were developed in Python version 3.10 with TensorFlow version 2.13 (Abadi et al., 2015) with the Keras Neural Network library version 2.6 (Chollet, 2015) for the classification tasks. Selenium WebDriver[3] was used with Google Chrome in headless mode for screenshot acquisition. The system operator's web GUI and the API to ingest the detected websites or domains were developed using PHP version 8. The framework and the web components use a MariaDB database version 10.6 to store data.

Using Python for the framework development, with TensorFlow[4] and Keras[5] for the CNN development, was a compelling choice, empowering the development of sophisticated CNNs efficiently for several reasons. Python, as an intuitive and versatile programming language, makes it accessible. TensorFlow is an open-source deep-learning library that provides a robust and efficient framework for building complex neural networks. Keras, a high-level API built on TensorFlow, simplifies designing and training CNN models, thus allowing the focus to be on architecture design and hyperparameter tuning.

On the other hand, PHP stood out as an optimal choice for crafting the web GUI and API when considering the seamless integration with prevalent web servers such as Apache

[3] https://www.selenium.dev
[4] https://www.tensorflow.org/
[5] https://keras.io/

and Nginx and databases such as MariaDB, its comprehensive documentation and the enablement of an agile development cycle effortlessly.

4.2. Technical Solution Definition

This section details the processes associated with NASA's SE Technical Solution Definition processes within the System Design, as showcased in Figure 2 in Subsection 2.3.

4.2.1. Logical Decomposition

This process aims to create detailed functional requirements that meet the stakeholders' expectations by identifying "what" should be achieved. A system architecture is created, and the requirements are decomposed (Hirshorn et al., 2017).

This research proposes a platform, BlitzPhish, developed using the technologies described in Subsection 4.1, comprised of three modules to detect, acquire and evaluate whether a given website or domain showcases a phishing web page targeting a protected website or brand. The BlitzPhish platform also considers a fourth module to execute a Course of Action (CoA) upon websites and domains (entities) classified as phishing by the Evaluator Module. A web GUI enables a system operator to observe and query the database efficiently, and a web API enables the integration and extensibility of the tool with external components.

The Evaluator Module comprises a CNN classifier and a CNN trainer. The CNN trainer creates a classification model using pre-classified screenshots, whereas the CNN classifier uses the model to evaluate and predict the classification of new screenshots. The CNN trainer must consider the system operator's False Positives (FP) and False Negatives (FN) manual classification override to make the model more resilient.

While the Detector Module runs persistently and asynchronously from the rest of the system, even allowing it to be run remotely by integrating with the rest of the components using the web API, the Acquirer, Evaluator (CNN classifier component), and Responder modules must run sequentially and routinely at a pre-designated schedule.

4.2.2. Design Solution Definition

NASA's SE defines the Design Solution Definition process as translating "the high-level requirements derived from the stakeholder expectations and the outputs of the Logical Decomposition Process into a design solution" (Hirshorn et al., 2017, p. 77).

BlitzPhish High-Level Design

Figure 4 provides the High-Level Design for the BlitzPhish system.

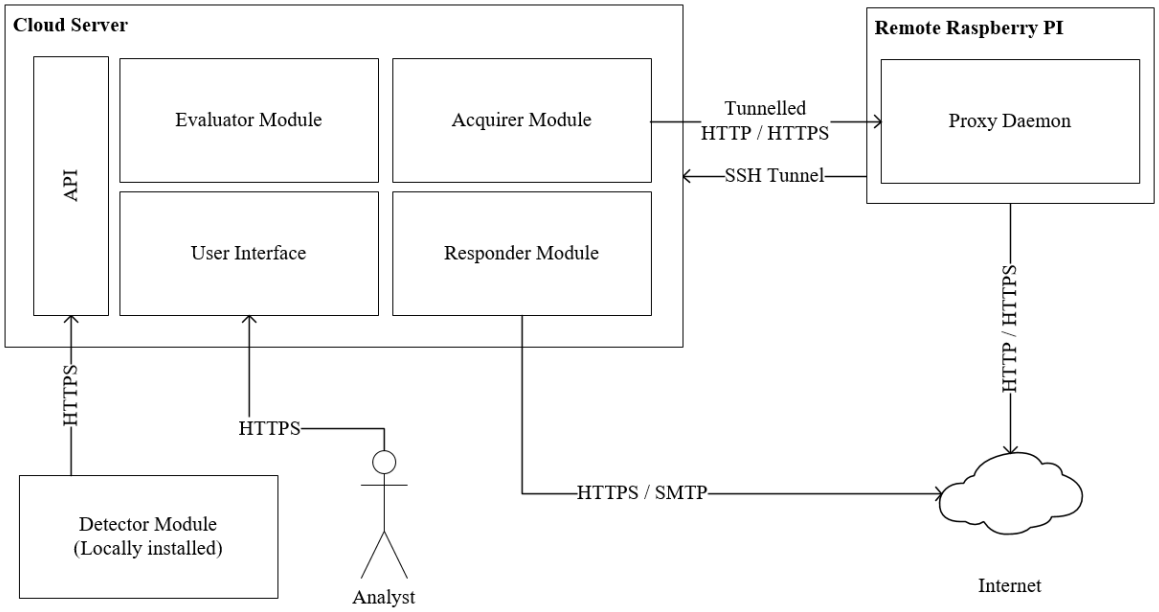


Figure 4: BlitzPhish high-level design

As the Detector Module can be installed locally or on a remote system, it is represented as a system external to the Cloud Server.

The remote Raspberry PI server allows the platform's web traffic to be masqueraded as potential victim traffic by originating on a consumer-grade IP address instead of a Cloud Service Provider, to which the adversary may have deception countermeasures in place. This way, the product is more resilient against simplistic adversarial deception techniques and can acquire a screenshot as seen by the targeted user on a regular end-user device.

Detector Module

BlitzPhish's first module pertains to the detector functionality of suspicious websites. This module extracts websites and domains from internal and external sources, running asynchronously from the other modules and reporting the detected domains and websites, referred to as "entities", to the acquirer module via the web API. As this module interacts with the rest of the system via a web API due to the modularity requirement listed in Subsection 2.3.1, it can be set up on an external system, apart from the rest of the modules,

or have multiple versions of this module set up to report suspicious domains and websites from different sources.

Some examples of potential internal data sources are sandboxes or internally reported phishing emails. Potential external data sources include monitoring domain registration databases, monitoring the registration and usage of digital certificates by exploiting the Certificate Transparency logs[6], and publicly reported phishing emails for suspicious keywords.

Acquirer Module

BlitzPhish's second module pertains to an application that evaluates the reported domain or website stored on the database via the web API and processes it. This module connects to the suspicious websites or domains in the processing queue and acquires a screenshot image. The screenshot image is normalised and stored with a consistent predefined height and weight to maximise the Evaluator's Module training and classification potential.

The Acquirer Module also acquires additional artefacts that may be useful for additional research by the system operator from a threat intelligence point of view or for future correlation, storing them in the database. Examples of potentially relevant information acquired and stored on the database are domain and website IP addresses, digital certificate data, website HTML source code and HTTP response headers, and domain registration information.

Evaluator Module

BlitzPhish's third module contains a "CNN trainer" and a "CNN classifier". The CNN trainer component runs asynchronously from the rest of the system. It is manually triggered by the system operator when deemed necessary and creates a CNN model using a pre-classified phishing dataset. This component must also consider the system operator's manual screenshot classification override output to generate a model that detects new phishing web pages incorrectly classified as false negatives (not being phishing) and correctly classifies legitimate images as false positives (incorrectly classified as phishing).

The CNN classifier component is a binary classification program that uses the trained model to predict the class ("phishing" or "other") of an image. It runs sequentially following

[6] Certstream - https://certstream.calidog.io/

the Acquirer Module execution by querying the database for newly acquired screenshots, classifying them according to a probability threshold, and storing the results in the database.

Responder Module

BlitzPhish's fourth module pertains to an application that runs sequentially following the Evaluator Module and applies the pre-defined CoA upon websites and domains whose latest screenshots received a "phishing" classification.

Some examples of possible CoA are reporting the website or domain to the Google Safe Browsing project[7], automatically filling an abuse complaint with the domain registrar's abuse email address, reporting the domain to the national CERT, or requesting a lawful takedown to a third-party provider specialised in taking down abusive websites and domains.

Web GUI

BlitzPhish's web GUI is a front-end application where the system operator monitors the output of the underlying modules. It allows the system operator to add websites and domains to the database for processing, displays the acquired screenshots and artefacts, the screenshots' prediction results, allows for manual classification override of FP and FN, and the manual execution of the predefined CoA.

Web API

BlitzPhish's web API implements interfaces that allow external platforms to interact with the system by ingesting data on the platform or querying the platform for the stored data. Examples of external platforms that may ingest or query the system are the Detector Module (that, due to modularity requirements, is considered an external application) or a Threat Intelligence Platform (TIP) that queries the database for newly discovered phishing websites and supports an organisation's Priority Intelligence Requirements (PIR).

4.3. Design Realization

This section details the processes associated with NASA's SE Design Realization processes within the Product Realization, as showcased in Figure 2 in Subsection 2.3.

[7] https://safebrowsing.google.com/

4.3.1. Product Implementation

NASA's SE defines the Product Implementation process as the generation of "a specified product of a project or activity through buying, making/coding, or reusing previously developed hardware, software, models, or studies to generate a product appropriate for the phase of the life cycle" and that "the product should satisfy the design solution and its specified requirements" (Hirshorn et al., 2017, p. 91).

During this research, the BlitzPhish application modules were coded in Python 3.10 and the web GUI and API in PHP 8, as detailed in Subsection 4.1. The interaction between these applications is summarised in Table 5 by system process. These system processes detail the core functions the product implements (Process Name) and the product module interaction on how it is implemented (Module Flow).

Table 5: BlitzPhish module interaction view organised by the system process

Process Name	Process Schedule	Module Flow
Entity Monitor	Real-Time.	The Detector Module reports detected entities via the web API.
Entity Batch Processor	System scheduled (regular intervals).	The Acquirer Module processes the entity queue, launching the Evaluator's Module classifier component and finishing with the Responder Module execution.
CNN Model Trainer	System Operator Triggered.	The system operator triggers the Evaluator's Module trainer component to create a trained CNN Model used by the classifier component.
The system operator's platform access	Real-Time	The web GUI is always available to the system operator via a web browser.
External platform interaction	Real-Time	The web API is always available to support external platform integration. The Detector Module uses it to add entities to the processing queue.

The Detector Module relies upon the Certificate Transparency logs. These logs, consisting of newly registered digital certificates reported by the browsers, were monitored for keywords often used to target a specific multinational financial institution brand using the Certstream Python library[8]. The chosen keywords focused on phishing domains used by adversaries to launch phishing attack campaigns on the organisation's customers with the intent of acquiring online banking platform credentials. The Detector Module proactively adds domains detected in real-time by Certificate Transparency logs matching chosen sensitive keywords to a database using the web API for acquisition and monitoring.

The Acquirer Module was developed using Python and performs several functions. First, it queries the database for reported suspicious domains and websites referenced as "entities". It then runs in batch mode at a specific prescheduled time controlled by the server's operating system to verify if the reported entity is on an allowed list. If the entity is on an allowed list, as it is a legitimate website, this module archives it by updating its status on the database, as it does not require further processing. If not, this module collects the entity's IP address information and the website TLS Certificate data. It also performs a domain lookup, storing all this information on the database for historical purposes and further correlation by a system operator. Finally, the module component uses Selenium with the Google Chrome WebDriver in headless mode to acquire a rendered website screenshot, HTTP response headers and HTML source code. The WebDriver is configured to send a mobile device user agent on the HTTP request header, and the window size is configured to 414x896 to mimic a mobile device browser. The HTTP request is tunnelled to a local proxy where traffic routing rules can be configured independently of the rest of the system. The acquired screenshot SHA-256 hash is calculated, and the screenshot file is stored using the hash concatenated with the .png file extension as the filename. The HTTP response headers and the HTML source code are stored in textual files with their SHA-256 hash as the filename, concatenated with the .txt file extension. The database is updated to include the acquired screenshot, HTTP response headers, HTML source code file references, and the acquisition timestamp, linking them to the entity record.

The Evaluator Module comprises a CNN trainer and a CNN classifier, developed in Python using the Machine Learning framework TensorFlow (Abadi et al., 2015) and the Keras Neural Network library (Chollet, 2015).

[8] https://github.com/CaliDog/certstream-python

The CNN trainer component implements a CNN to perform screenshot image classification. It parses the dataset directory where the screenshots are stored and uses the subfolders organising the images to binarily classify them between “phishing” and “other” classes. Images belonging to the “phishing” class are stored in the “dataset/phishing/” subfolder, whereas images belonging to the “other” class are stored in the “dataset/other/” subfolder. The component then randomly selects 70% of the screenshots for the training dataset, 15% for the validation dataset and 15% for the testing dataset.

The designed CNN model used by the CNN training component implements the Keras Sequential API Neural Network model, with nine layers stacked sequentially on each other. The first layer was a Conv2D convolutional layer with 32 filters of 3x3. The Rectified Linear Unit (ReLU) activation function was used, with the input shape set for a 414x896 image with three colour channels (RGB) to match the dataset screenshot images. The second layer was a MaxPooling2D to downsample the input into a 2x2 window to reduce computational complexity. The third layer added a second Conv2D convolutional layer with 64 filters of 3x3 and ReLU activation. The fourth layer is similar to the second layer, and MaxPooling2D was applied to downsample the input into a 2x2 window. The fifth layer added a third Conv2D convolutional layer with 128 filters of 3x3 and ReLU activation.

Similarly to the second and fourth layers, the sixth layer applied MaxPooling2D to downsample the input into a 2x2 window. The seventh layer flattened the output from the previous layer into a 1-dimensional array, preparing the data for the fully connected layers that followed. The eight-layer added a fully connected Dense layer with 128 neurons and applied the ReLU activation function. Finally, the ninth layer added a Dense layer with one single neuron and the sigmoid activation function, used for binary classification where the output is a probability score between 0 and 1, indicating the likelihood of the input image belonging to a specific class.

The Neural Network was compiled using the Adam optimiser, with binary cross-entropy loss and accuracy as the evaluation metric.

Data augmentation techniques were used on the training dataset to enhance the size and quality of the dataset (Shorten & Khoshgoftaar, 2019). These techniques allowed for the dynamic creation of multiple image versions, making the model more robust (Russell & Norvig, 2022). The training images were augmented by randomly flipping the images horizontally and zooming in or out by a maximum of 20%. All images had their pixels rescaled by a factor of 1/255 to normalize each RGB channel value in the range of [0,1]. This transformation resulted from using the ImageDataGenerator function.

The CNN model was trained using ten epochs, storing the training output in a Keras model file. The screenshot images for the training and validation datasets were parsed from the dataset directory, with the class mode set to binary, making it a two-class dataset, where 0 indicates the “other” class, and 1 indicates the “phishing” class, in a 1-dimensional array.

The CNN classifier component loads the saved module and makes predictions on the newly collected screenshots. Due to the binary classifier, screenshots received a “phishing” label if the predicted probability exceeded 0.5 or an “other” label if below 0.5. The classifier component then adds each screenshot classification to the database for further processing by the Responder Module.

The Evaluator Module also supports model retraining so it can handle “other” entities wrongly classified as “phishing” (false positives) and “phishing” entities wrongly classified as “other” (false negatives). This capability also supports model enhancement by using newly detected phishing images for training, making it more accurate and kept updated. For this purpose, a dataset maintenance script was developed in Python that adds the newly acquired and correctly classified screenshot images to the proper dataset subfolders, false negatives to the “phishing” training subfolder, and false positives to the “other” training subfolder. Following the dataset update, the already described CNN training component triggers a new training process.

The Responder Module was developed in Python and executed the pre-defined CoA on the newly discovered and positively predicted phishing websites or domains. Three actions were defined: an alert email informing of the newly discovered phishing website or domain with the attached screenshot requesting the domain takedown, a Telegram chat notification, and an automatic report to the Google Safe Browsing project.

The PHP web GUI was developed in PHP to facilitate the platform operation by the end users and to present them with the acquired artefacts in a user-friendly way. It also allows the system operator to override image classification (FP and FN), trigger a CoA on malicious websites or manually add websites or domains for further processing. It implements an access control system to prevent unauthorized system usage.

The web API was developed in PHP and implements interfaces that allow other platforms to interact with the system. These interfaces can be used for data ingestion by the Detector Module or any other data ingestion system and data consumption by an external platform. API keys and secrets validate the API access to restrict access to the platform to authorised users only.

4.3.2. Product Integration

NASA's SE defines the Product Integration process as the "engineering of the subsystem interactions and their interactions with the system environments (both natural and induced)" and that "in this process, lower-level products are assembled into higher-level products and checked to make sure that the integrated product functions properly and that there are no adverse emergent behaviors" (Hirshorn et al., 2017, p. 97).

BlitzPhish was installed on a cloud server with 32 GB of RAM, 8 CPU, 600 GB disk space storage and Ubuntu 22.04 Server as Operating System. These system specifications are not the minimal requirements to run BlitzPhish, though they were selected to handle the system at scale and to support the CNN model training process. Additionally, two remote Raspberry Pi 2 microcomputers with Ubuntu Core 22 were configured as proxy systems to handle all HTTP traffic via consumer-grade access points. The following software was installed on the cloud server via the Ubuntu package management system (apt):

- Python 3.10 with the modules required for the application to run:
  - Certstream 1.12;
  - Keras Neural Network 2.6;
  - Selenium-wire 4.6;
  - TensorFlow 2.13;
- Snap 2.60 with the latest Chromium package for the Google Chrome Browser;
- Apache 2.4 Web Server to support the system operator's web interface and publish the application API;
- PHP 8 to support the system operator's GUI and the web API;
- MariaDB 10.6 as a relational database;
- Postfix 3.6 as the Mail Transfer Agent used to handle the alert emails;
- Privoxy 3.0 HTTP Proxy to handle and route the Google Chrome HTTP requests. Traffic is routed to the locally configured HA Proxy;
- HA Proxy 2.4 to load balance the remote SOCKS5 proxy access points where the HTTP traffic is tunnelled.

Remotely, a set of Raspberry Pi 2 microcomputers, running the Dante 1.4 SOCKS5 Proxy daemon connected via two independent consumer-grade Internet access points, connects to the cloud server via SSH and port-forwards the locally configured Dante Proxy daemon to the server's HA Proxy.

4.4. Dataset

The dataset used to train the CNN model pertained to websites and domains detected using prototypes that later led to the development of the Detector and the Acquirer modules. Similarly to the Detector and Acquirer modules, this code acquired screenshots from domains detected using Certificate Transparency logs that matched specific phished keywords from March 2023 to July 2023. A system operator manually classified the acquired screenshots as “phishing” and “other” by moving the files into two separate filesystem folders.

The class “other” was named that way, as the acquired screenshot may still display a phishing page for brands that do not require protection or for websites that are not considered high-value assets and, as such, are out of the detection scope. Since “phishing pages have to be quite similar to the authentic pages in order to deceive users” (Lam et al., 2009), screenshots were also acquired from high-value websites requiring protection and manually labelled as “phishing”.

A curated dataset of 31,273 unique screenshot files was acquired during the five-month acquisition period. The system operator acquired and manually labelled the screenshot files into two classes. The classes consist of 1,111 samples labelled “phishing” and 30,162 samples labelled “other”. Therefore, it is evident that an imbalanced dataset exists for the class “phishing” (3.55%) in comparison to the class “other” (96.45%). The screenshot files were stored using their SHA-256 hash as the filename. It is worth noting that web pages displaying similar content with minimal changes generated different files.

Since the complete dataset was imbalanced and favoured the “other” class, the undersampling technique (Estabrooks & Japkowicz, 2001) was applied. A randomiser Python script implementing a random undersampling algorithm to balance the dataset was developed. This algorithm balanced the class distribution by randomly eliminating instances from the largest class (Mohammed et al., 2020). The script outputs a balanced dataset for training purposes comprising 2,222 images, with 1,111 in the “phishing” class and 1,111 in the “other” class.

4.5. Evaluation and Results

This section details the processes associated with NASA’s SE Evaluation processes within the Product Realization, as showcased in Figure 2 in Subsection 2.3.

4.5.1. Results

The presented results aimed to quantify the CNN model's effectiveness in capturing and recognizing complex patterns within the data. For this purpose, five evaluation metrics were selected: Accuracy, Precision, Recall, F1 Score, and the Matthews Correlation Coefficient (MCC).

These metrics, detailed in Appendix D, were chosen because they are standard in classification problems and can provide unique insights into various model performance elements. A fundamental measure, the Accuracy metric, gives a general overview of the model's correctness in the classification task. The F1 Score, which considers Precision and Recall, provides a balanced view by accounting for FP and FN. Finally, the MCC considers True Positives (TP), FP, True Negatives (TN), and FN, encompassing the model's performance across all categorization outcomes.

It is worth noting that although the training dataset has 2,222 screenshots, only 1,554 were used for training, 332 for validation and 332 for testing, summing up to 2,218. The four-image gap is the consequence of dividing the dataset into training (70%), validation (15%) and testing (15%), as described in Subsection 4.3.1 and rounding down the image classes to an even number to ensure a balanced class distribution.

Ten epochs for the training were allowed during Product Implementation to obtain a classification model. Figure 5 presents the Receiver Operating Characteristic (ROC) Curve and the Confusion Matrix tested for the resulting model with the training, testing, and validation subsets and the entire dataset.

The ROC Curve, further described in Appendix D, sheds light on the discriminative prowess of the model and is an invaluable tool for assessing the model's capacity to distinguish between different classes by varying the classification threshold. This analysis is pivotal in quantifying the trade-off between the True Positive and False Positive rates, providing a nuanced perspective on the model's classification performance.

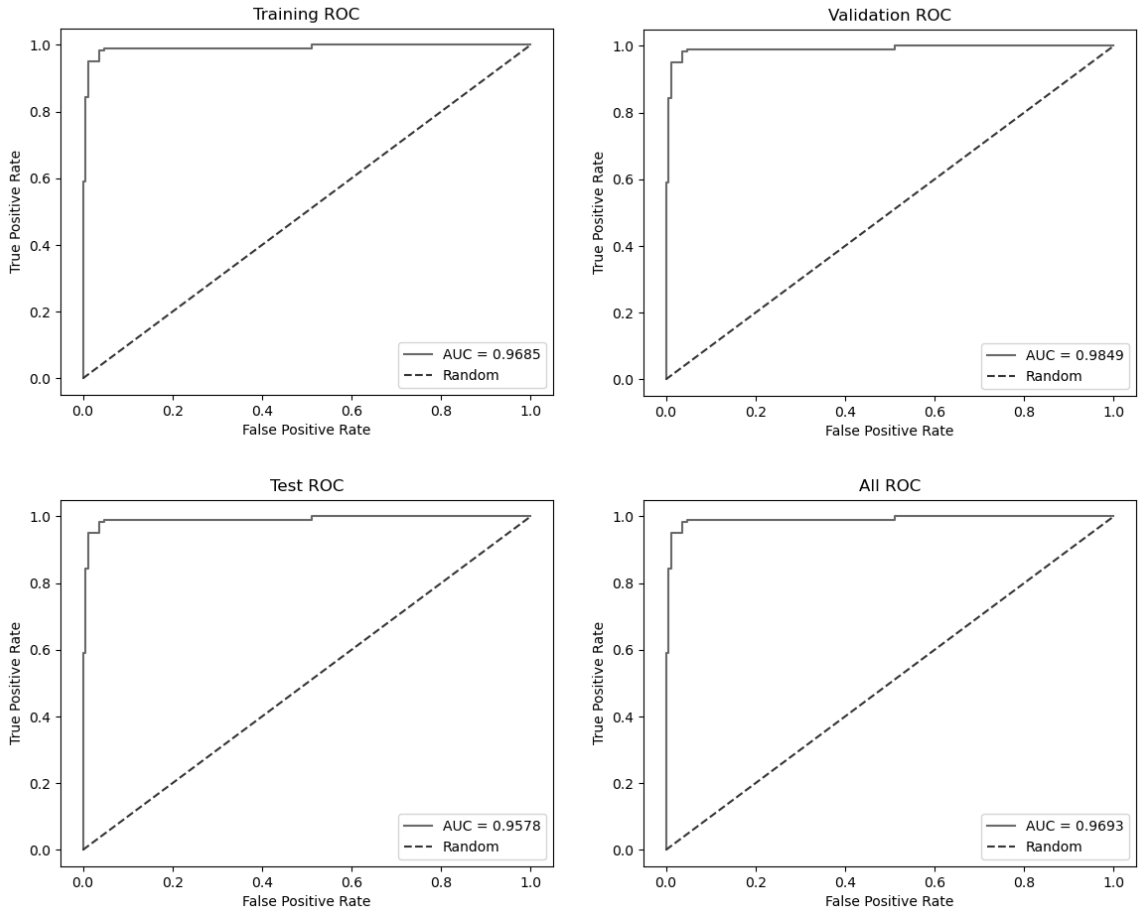


Figure 5: Receiver operating characteristic curves for the trained model

The Confusion Matrix, as described in Appendix D, is pivotal in evaluating the classification performance of the trained model, offering an insightful and granular depiction of the model's predictive accuracy. The depicted Confusion Matrix in Figure 6 showcases the model's TN in the top left corner, TP in the bottom right corner, FN in the bottom left corner and FP in the top right corner.

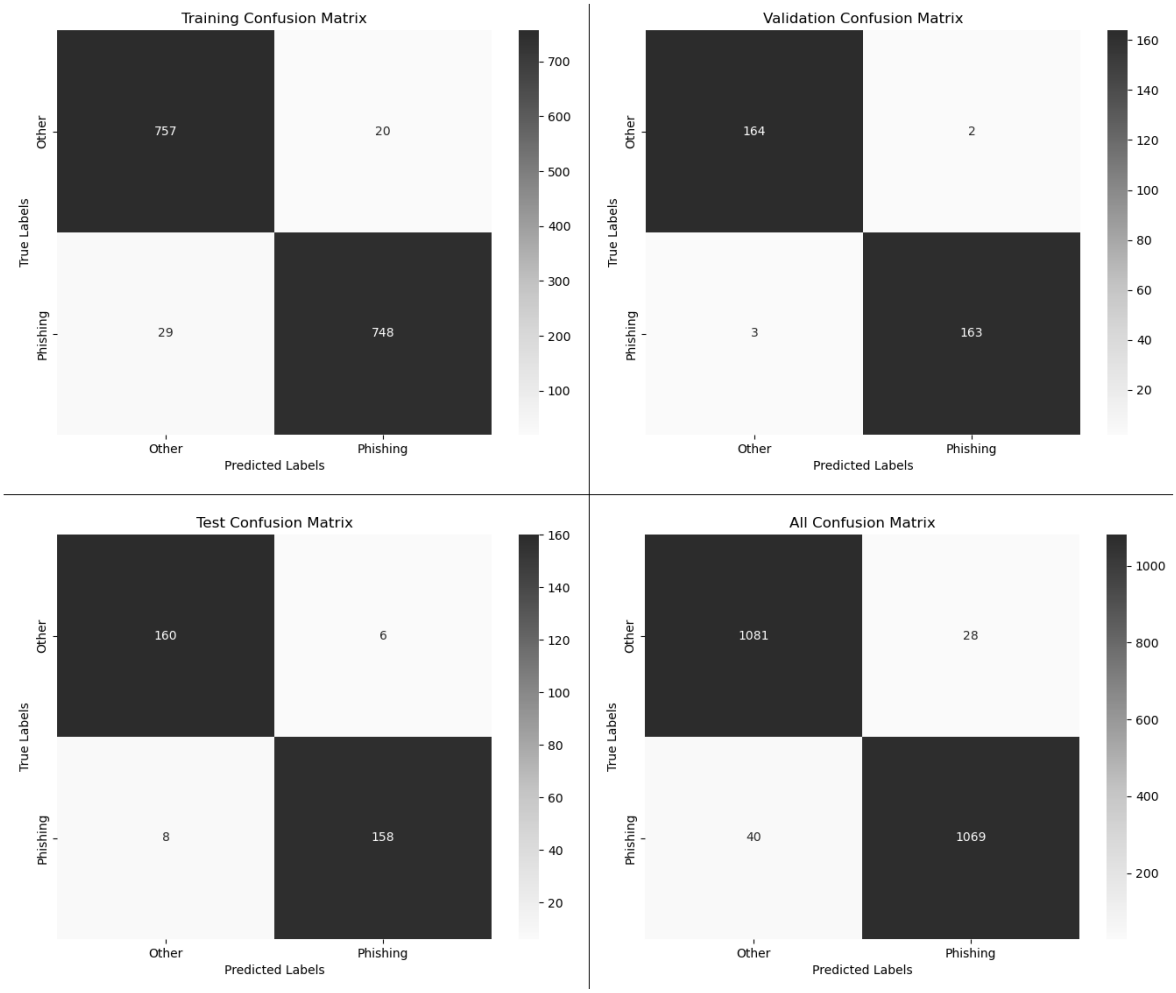


Figure 6: Confusion matrices for the trained model

The results presented on the Confusion Matrix depicted in Figure 6 supported the computation of Accuracy, Precision, Recall, F1 Score and the MCC, shown in Table 6. These metrics are further detailed in Appendix D.

Although Table 6 has the data for all datasets, the test dataset evaluated the model as it pertains to the dataset not used to train the model (as the train and validation datasets were), providing an independent model validation and reducing overfitting risk.

Table 6: Evaluation metrics for the trained model

	Train subset	Validation subset	Test subset	Dataset
Accuracy	96.85%	98.49%	95.78%	96.93%
Precision	97.40%	98.79%	96.34%	97.45%
Recall	96.27%	98.19%	95.18%	96.39%
F1 Score	96.83%	98.49%	95.76%	96.92%
MCC	93.70%	96.99%	91.57%	93.87%

For the test dataset, in terms of accuracy, the ratio between a model's correct predictions and the total number of cases, the model attained 95.78%. For the F1-Score, the harmonic mean of Precision (ratio between the number of TP and the total number of predicted positives) and Recall (ratio between the number of TP and the total number of positive cases), the model attained 95.76%. Finally, for the MCC, which considers the TP, TN, FP, and FN to provide a balanced performance measure, the model got 91.57%.

4.5.2. Product Verification

NASA's SE defines the Product Verification process as the process that "proves that an end product (whether built, coded, bought, or reused) for any element within the system structure conforms to its requirements or specifications", answering the question "was the end product realized right?" (Hirshorn et al., 2017, p. 103).

Verification was performed on each module individually and on the assembled platform to verify if all modules worked correctly aimed at a production version. For the individual module verification, a set of unit tests was designed and run against the functions and classes to confirm they produced accurate and expected outputs for a given input. As the platform relies on externally acquired data, such as DNS records, web server TLS certificate data, HTML source code, HTTP response headers and domain record data, a set of fault injections was also tested to check how reliable the platform was in handling unexpected errors, invalid data, network errors, or adversarial injected malicious payloads.

The CNN-trained model was assessed and verified using 15% of the known "test dataset" dataset. Subsection 4.5.1. presents the results.

The BlitzPhish platform was configured to detect phishing keywords on newly registered domains, acquire and classify the screenshot data, and report findings.

As reporting legitimate websites as phishing due to a classification issue could trigger reputational risks to the reporting party as well as unforeseen costs in case a third party is involved in dealing with the phishing reports, during the product verification phase, the Responder Module was configured to notify the system operators and not to trigger the execution of any pre-defined CoA in case of misclassified entities as phishing (FP).

4.5.3. Product Validation

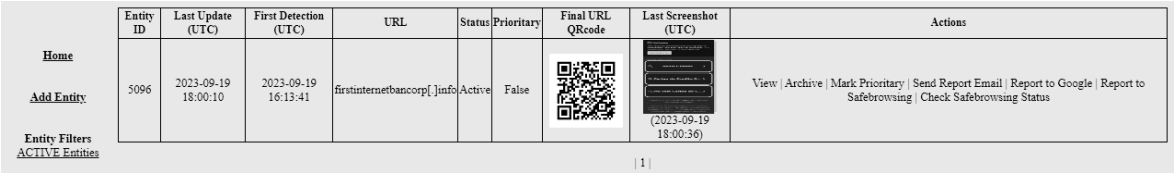
NASA's SE defines the Product Validation process as "performed for the benefit of the customers and users to ensure that the system functions in the expected manner when placed in the intended environment" (Hirshorn et al., 2017, p. 116).

The BlitzPhish product was deployed in a production execution environment and presented to the stakeholders, primarily potential end users, for feedback on its usability, ease of use and completeness against the requirements in Subsection 2.3.1.

A set of use cases was derived from end-user expectations for the validation process. Several publicly available phishing kits that target one of the protected brands were deployed on an isolated server, with the web server being manually added to BlitzPhish via the web GUI. As expected, the classification engine correctly classified all phishing kit screenshots and notified the system operator, validating the Acquirer, Evaluator and Responder modules. Additionally, a domain with one of the monitored keywords was registered. A mock-up website was set up to determine whether the tool detected new domains accordingly. The results were also satisfactory, as the domain was detected minutes after the web server was configured with a digital certificate, validating the Detector module.

The main view of the BlitzPhish tool is depicted in Figure 7. This view displays the domains that have been reported by the Detector module or have been manually added to the tool but have not yet been archived by the system operator. Figure 8 showcases the data acquired from one domain. Both views support the manual triggering of response actions, such as sending reporting alerts via email or adding the domain to phishing databases.

The navigation menu depicted in Figure 8 allows the system operator to navigate the entire tool easily, add new domains manually, and search and filter the product database for indicators. It also presents the system operator with screenshots that the Evaluator module could not predict as "phishing" or "other" so that the operator can manually classify them and retrain the model.

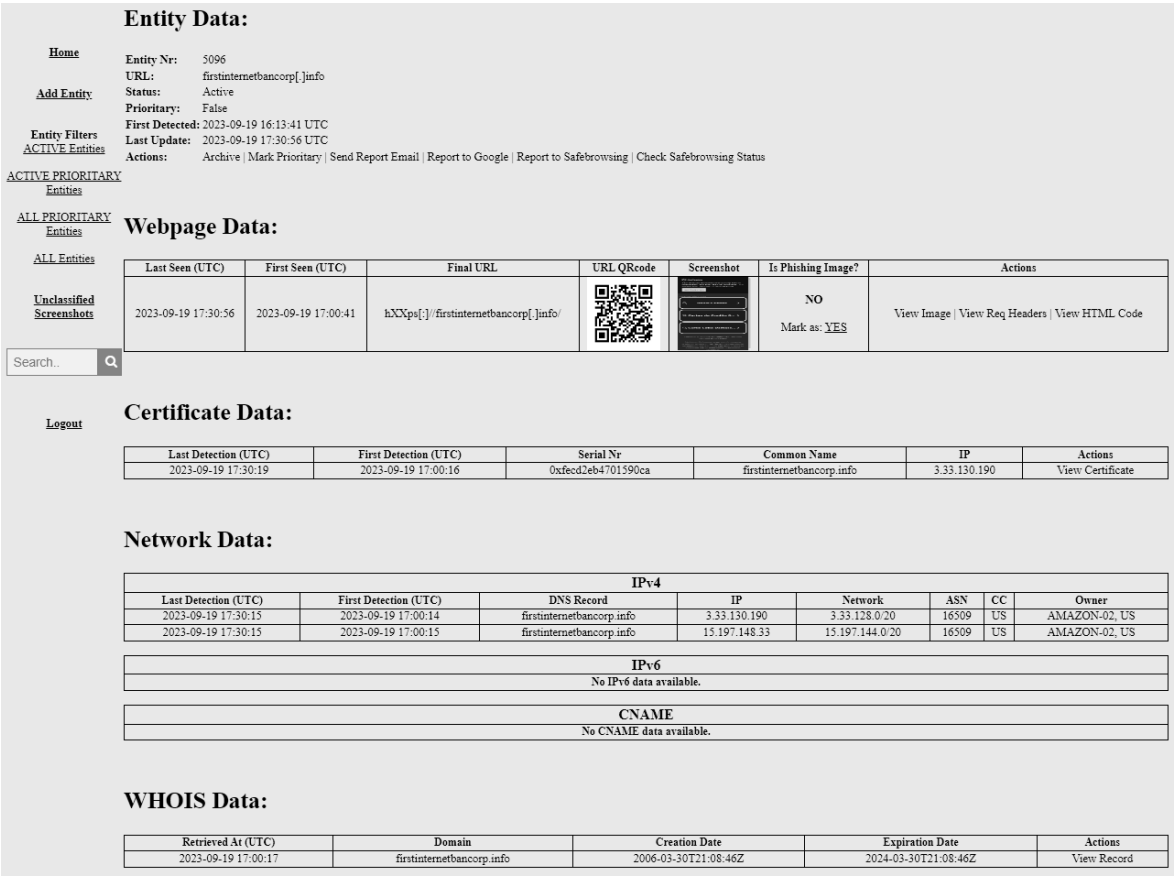


Entity ID	Last Update (UTC)	First Detection (UTC)	URL	Status	Prioritary	Final URL QRcode	Last Screenshot (UTC)	Actions
5096	2023-09-19 18:00:10	2023-09-19 16:13:41	firstinternetbancorp[.]info	Active	False		(2023-09-19 18:00:36)	View | Archive | Mark Prioritary | Send Report Email | Report to Google | Report to Safebrowsing | Check Safebrowsing Status

Home
Add Entity
Entity Filters
ACTIVE Entities

| 1 |

Figure 7: BlitzPhish detected domains view



Entity Data:

Home
Add Entity
Entity Filters
ACTIVE Entities
ACTIVE PRIORITARY Entities
ALL PRIORITARY Entities
ALL Entities
Unclassified Screenshots
Search..
Logout

Entity Nr: 5096
URL: firstinternetbancorp[.]info
Status: Active
Prioritary: False
First Detected: 2023-09-19 16:13:41 UTC
Last Update: 2023-09-19 17:30:56 UTC
Actions: Archive | Mark Prioritary | Send Report Email | Report to Google | Report to Safebrowsing | Check Safebrowsing Status

Webpage Data:

Last Seen (UTC)	First Seen (UTC)	Final URL	URL QRcode	Screenshot	Is Phishing Image?	Actions
2023-09-19 17:30:56	2023-09-19 17:00:41	hXXps[:]//firstinternetbancorp[.]info/			NO Mark as: YES	View Image | View Req Headers | View HTML Code

Certificate Data:

Last Detection (UTC)	First Detection (UTC)	Serial Nr	Common Name	IP	Actions
2023-09-19 17:30:19	2023-09-19 17:00:16	0xfecd2eb4701590ca	firstinternetbancorp.info	3.33.130.190	View Certificate

Network Data:

IPv4							
Last Detection (UTC)	First Detection (UTC)	DNS Record	IP	Network	ASN	CC	Owner
2023-09-19 17:30:15	2023-09-19 17:00:14	firstinternetbancorp.info	3.33.130.190	3.33.128.0/20	16509	US	AMAZON-02, US
2023-09-19 17:30:15	2023-09-19 17:00:15	firstinternetbancorp.info	15.197.148.33	15.197.144.0/20	16509	US	AMAZON-02, US

IPv6
No IPv6 data available.

CNAME
No CNAME data available.

WHOIS Data:

Retrieved At (UTC)	Domain	Creation Date	Expiration Date	Actions
2023-09-19 17:00:17	firstinternetbancorp.info	2006-03-30T21:08:46Z	2024-03-30T21:08:46Z	View Record

Figure 8: BlitzPhish acquired domain data view

With the successful validation of all modules and the feedback from the end users, BlitzPhish has been validated as accurately producing the intended results and automatically detecting new phishing websites, thus minimizing the time from detection to CoA execution.

4.6. Discussion

4.6.1. Answer to the Research Subquestion 6

As demonstrated in the results section, the trained CNN model showcases its prowess in classifying phishing websites using browser-rendered screenshot images. The model

performed exceptionally well, attaining a 95.76% accuracy on the testing dataset, supporting an answer for research subquestion SQ6.

Analysis of the incorrectly classified FP revealed some displayed similarities to protected websites, such as positioned login forms or background images/colours. Others rendered image-matching probabilities between 0.5 and 0.8, higher than the 0.5 threshold but still uncertain. Similar features and marginal matching scores potentially explained misclassification as legitimate when pages posed risks. As described in Subsection 4.3.1, images whose predicted probability was lower than 0.5 were considered from the "other" class and higher than 0.5 from the "phishing" class. Since most false positives were predicted with probabilities between 0.5 and 0.8, setting a probability threshold of 0.2 may improve the false-positive detection ratio by keeping images predicted in the range [0.5, 0.8] as unclassified and requiring human interaction.

The examined FN, on the other hand, was shown to be associated with screenshots that revealed web pages that were not fully loaded at the time of the screenshot or provided a wholly unformatted web page.

The Recall metric is frequently more critical than Precision in a phishing detection challenge, as it is better to have false positives (incorrectly detected websites) than false negatives (misdetections). However, for a system that automatically executes CoA on detected websites, both Precision and Recall are of extreme relevance, as the system cannot execute a CoA on a legitimate website, and detecting phishing is paramount. The test results for this model assessed Precision at 96.34% and Recall at 95.18% for 100% of the images binarily classified as "phishing" or "other".

It is worth mentioning that in a live environment, a system operator can manually classify or override the classification of a website or a domain. The manual classification or reclassification makes the proposed approach more resilient, as the new training dataset will consider the system operator's classification during training, generating a more tailored model. Maintaining an adaptive, up-to-date capability is paramount for new high-value domains requiring safeguarding and protected sites undergoing redesigns, visual overhauls, or rebranding efforts altering appearance factors, which could hinder classification effectiveness if not addressed. Flexibility to incorporate continuous updates represents a key strength for any phishing detection solution aiming to consistently provide robust protection regardless of alterations to brand representation or the emergence of new high-profile targets.

4.7. Chapter Conclusion

In conclusion, the investigation into the classification of phishing websites using browser-rendered screenshot images, as carried out and analysed in Subsection 4.5.1, has yielded significant insights and practical implications. The trained CNN model has effectively addressed the research subquestion SQ6 by demonstrating its proficiency in accurately categorizing phishing web pages targeting protected brands or high-value websites requiring protection, illustrated by the model's outstanding performance, achieving a remarkable accuracy rate of 95.76% on the testing dataset.

Even though the trained model performed remarkably well, assessing the FP revealed intriguing findings. These screenshot images bore significant resemblances to the websites requiring protection, particularly in the arrangement of the login form, background graphics, and colour schemes. Further analysis revealed that while the image-prediction probability for these cases was higher than 0.5, most fell within the range [0.5, 0.8].

The investigation into FN presented a different side of the model's performance. These misclassifications were associated with screenshot images featuring partially loaded web pages or pages lacking any recognizable format.

The inherent trade-off between minimizing FP and FN in phishing detection was addressed by the model's Recall-focused and Precision-focused approach, aligning with the goal of accurate threat identification. In this research, mitigating FP and FN was paramount, as missing out on threats is as critical as not executing a CoA on FP. The model's precision of 96.34% and recall of 95.18% in the binary classification problem attested to its balanced performance in both aspects.

In sum, this research provided a robust solution for detecting phishing websites using computer vision by analysing browser-rendered screenshots. It also highlighted the importance of incorporating human insights into the model's supervised learning process on unclassified images when implementing a probability threshold of 20% to either one of the classes (i.e., requiring manual classification on images predicted with a probability [0.2, 0.8]) or manually associating false positives and false negatives to the correct class, to enhance the training dataset, making the model more robust during the following model retrain process. The insights gained from this research hold significance for cybersecurity strategies, especially within the ever-evolving landscape of online threats and attacks. The success of the CNN model, alongside the highlighted methodological considerations, offers valuable contributions to advancing security measures to respond to phishing threats.

CONCLUSIONS

Cybercrime, with its multifaceted threats, poses a grave challenge to contemporary society, prompting businesses and governments to engage in relentless battles against cyber-attacks (Huang et al., 2019). The domain of cyber warfare intensifies further when orchestrated by nation-states or advanced threat groups, as demonstrated by the Lazarus Group's historic cyber heist and the Sandworm Group's assault on critical infrastructure (Herr & Armbrust, 2015; Roy et al., 2023). A common thread through these incidents is the exploitation of human vulnerability, with phishing attacks emerging as a preferred modus operandi (Pires & Borges, 2023a).

The Systematic Literature Review (SLR) in Chapter 3 delved into the evolution of computer vision techniques for detecting and classifying phishing websites, marking a key milestone in combating these cyber threats. Although some phishing websites exhibit distinct characteristics, including fake URLs, inconsistent design, and dubious content, enabling the development of detection models, others are exact replicas of the original website to trick the intended victims into executing tasks aligned with the adversary's goals whilst minimizing suspicion by the victim. In the latter, only the URL and browsing flow often differ from the original website.

The SLR findings revealed the growth of computer vision methodologies, with Convolutional Neural Networks (CNNs) emerging as a prevailing technique for phishing website detection. The convergence of CNNs as feature extraction and classifiers showcased promising results in enhancing cybersecurity strategies.

On the other, Chapter 4 provided a Computational Implementation and Assessment, addressing the practical application of computer vision in classifying phishing websites. In this chapter, a CNN model was designed and trained against a curated phishing image dataset. The model was employed for classifying newly discovered suspicious websites that could target protected brands and high-value websites. The trained model balanced approach, focusing on recall and precision for accurate threat identification, mitigated the inherent trade-off between minimizing false positives and false negatives and achieved remarkable accuracy in predicting an image class.

Although the model's high accuracy is easily justified by narrowing down the phishing detection to specific brands and high-value websites, allowing for surgical precision in terms of prediction, exploring the false positives and false negatives predicted by the model in a live environment highlighted the challenge of distinguishing phishing sites

from legitimate ones, emphasizing the nuanced distinctions that pose difficulties in classification. The false negatives were associated with websites whose screenshots showed partially loaded web pages or pages with no recognisable format. In contrast, the false positives bore striking similarities to the websites requiring protection, particularly in the login form layout, background graphics, and colour schemes.

Allowing a system operator to manually classify an unclassified website and override the predicted classification on false positives and negatives is critical to ensure the model is updated and accurate. This supervised learning capability adds to the model's robustness in real-world scenarios where websites change over time and similar websites that may be mislabeled exist.

This research underscored the critical role of computer vision techniques in addressing the escalating threat of cyber-attacks, particularly phishing. The synthesis of insights from the SLR chapter and the development of the platform to identify, detect and respond to phishing websites detailed in the Computational Implementation and Assessment chapter echoes the progress made in theory and practice, answering the main research question: "How can computer vision methodologies be used for phishing website detection and classification based on browser-rendered screenshots to support an organisation's brand protection efforts".

CNNs, at the vanguard of computer vision, demonstrate the capacity to enhance phishing attack response despite the nuanced challenges posed by false positives and negatives. Integrating human-supervised learning in the model's training and retraining processes heightens the model's resilience and adaptability, pointing to breakthroughs in cybersecurity.

This research, extending its scope beyond the boundaries of academia, establishes a connection with the cybersecurity community by directly addressing a relevant cybersecurity concern. It serves as more than a mere academic endeavour, representing a steadfast commitment to fortify our digital environment, preserve the authenticity of our interconnected global community, and cultivate societal resilience in the face of rising cyber risks.

With the evolution of the digital landscape, the insights gained from this research can equip organizations, businesses, and governments with potent tools to thwart emerging cyber threats. As evidenced here, the convergence of human insights and technological innovation marks a formidable defence against the persistence of cyber criminals and their ever-evolving strategies.

The impact of the BlitzPhish tool surpasses immediate threat reduction. It can serve as a valuable intelligence-gathering asset, providing insights into cyber adversaries' tactics, techniques, and procedures in cybersecurity and cyber defence. By analysing the data collected from detected phishing attempts, civilian and military organisations can better understand emerging threats and the evolving cyber threat landscape, leading to the development of more effective countermeasures and enhancing the ability to preempt future attacks, bolstering cybersecurity and cyber defence efforts.

In future work, additional methodologies can be considered to improve the model's recall and precision, such as applying a fuzzy logic model that considers the CNN prediction probability and evaluates it against the domain creation date and the uniqueness of the downloaded resources while accessing the website or domain. Additionally, additional research into a continuous model adaptation to evolving phishing techniques and emerging threats ensures that the detection system remains effective over time. Finally, integration modules for threat intelligence sharing platforms like the MISP Project[9] can be developed to ensure interoperability with cyber threat intelligence sharing communities.

[9] Malware Information Sharing Platform - https://www.misp-project.org/

BIBLIOGRAPHY

Abadi, M., Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, … Xiaoqiang Zheng. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/

Abdelnabi, S., Krombholz, K., & Fritz, M. (2020). VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 1681–1698. https://doi.org/10.1145/3372297.3417233

Adebowale, M. A., Lwin, K. T., & Hossain, M. A. (2020). Intelligent phishing detection scheme using deep learning algorithms. *Journal of Enterprise Information Management*. https://doi.org/10.1108/JEIM-01-2020-0036

Afroz, S., & Greenstadt, R. (2011). PhishZoo: Detecting Phishing Websites by Looking at Them. *2011 IEEE Fifth International Conference on Semantic Computing*, 368–375. https://doi.org/10.1109/ICSC.2011.52

Banerjee, M., Capozzoli, M., McSweeney, L., & Sinha, D. (1999). Beyond kappa: A review of interrater agreement measures. *Canadian Journal of Statistics*, *27*(1), 3–23. https://doi.org/10.2307/3315487

Bannur, S. N., Saul, L. K., & Savage, S. (2011). Judging a site by its content: Learning the textual, structural, and visual features of malicious web pages. *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, 1–10. https://doi.org/10.1145/2046684.2046686

Baratis, E., Petrakis, E. G. M., & Milios, E. E. (2008). Automatic Website Summarization by Image Content: A Case Study with Logo and Trademark Images. *IEEE Transactions on Knowledge and Data Engineering*, *20*(9), 1195–1204. https://doi.org/10.1109/TKDE.2008.34

Bozkir, A. S., & Aydos, M. (2020). LogoSENSE: A companion HOG based logo detection scheme for phishing web page and E-mail brand recognition. *Computers & Security*, *95*, 101855. https://doi.org/10.1016/j.cose.2020.101855

Bozkir, A. S., & Sezer, E. A. (2016). Use of HOG descriptors in phishing detection. *2016 4th International Symposium on Digital Forensic and Security (ISDFS)*, 148–153. https://doi.org/10.1109/ISDFS.2016.7473534

Buber, E., Demir, O., & Sahingoz, O. K. (2017). Feature selections for the machine learning based detection of phishing websites. *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 1–5. https://doi.org/10.1109/IDAP.2017.8090317

Chan, L., Morgan, I., Simon, H., Alshabanat, F., Ober, D., Gentry, J., Min, D., & Cao, R. (2019). Survey of AI in Cybersecurity for Information Technology Management. *2019 IEEE Technology & Engineering Management Conference (TEMSCON)*, 1–8. https://doi.org/10.1109/TEMSCON.2019.8813605

Chen, T.-C., Dick, S., & Miller, J. (2010). Detecting visually similar Web pages: Application to phishing detection. *ACM Transactions on Internet Technology*, *10*(2), 1–38. https://doi.org/10.1145/1754393.1754394

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 6. https://doi.org/10.1186/s12864-019-6413-7

Chiew, K. L., Chang, E. H., Sze, S. N., & Tiong, W. K. (2015). Utilisation of website logo for phishing detection. *Computers & Security*, *54*, 16–26. https://doi.org/10.1016/j.cose.2015.07.006

Chikada, A. (2019). Cyber security and the brand. *Computer Fraud & Security*, *2019*(9), 6–9. https://doi.org/10.1016/S1361-3723(19)30094-6

Chikada, A., & Gupta, A. (2017). Online brand protection. In P. Chaudhry, *Handbook of Research on Counterfeiting and Illicit Trade* (pp. 340–365). Edward Elgar Publishing. https://doi.org/10.4337/9781785366451.00024

Chollet, F. (2015). *Keras*. https://keras.io

Chou, E. J., Gururajan, A., Laine, K., Goel, N. K., Bertiger, A., & Stokes, J. W. (2020). Privacy-Preserving Phishing Web Page Classification Via Fully Homomorphic Encryption. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2792–2796. https://doi.org/10.1109/ICASSP40776.2020.9053729

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *1*, 886–893. https://doi.org/10.1109/CVPR.2005.177

Dalgic, F. C., Bozkir, A. S., & Aydos, M. (2018). Phish-IRIS: A New Approach for Vision Based Brand Prediction of Phishing Web Pages via Compact Visual Descriptors.

2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 1–8. https://doi.org/10.1109/ISMSIT.2018.8567299

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Desolda, G., Ferro, L. S., Marrella, A., Catarci, T., & Costabile, M. F. (2022). Human Factors in Phishing Attacks: A Systematic Literature Review. *ACM Computing Surveys*, *54*(8), 1–35. https://doi.org/10.1145/3469886

ENISA. (2022). *ENISA threat landscape 2022: July 2021 to July 2022*. European Union Agency for Cybersecurity Publications Office. https://data.europa.eu/doi/10.2824/764318

Estabrooks, A., & Japkowicz, N. (2001). A Mixture-of-Experts Framework for Learning from Imbalanced Data Sets. In F. Hoffmann, D. J. Hand, N. Adams, D. Fisher, & G. Guimaraes (Eds.), *Advances in Intelligent Data Analysis* (Vol. 2189, pp. 34–43). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44816-0_4

EUROPOL. (2021). *IOCTA 2021: Internet organised crime threat assessment 2021*. European Union Agency for Law Enforcement Cooperation Publications Office. https://data.europa.eu/doi/10.2813/113799

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

FBI. (2021). *Internet Crime Report 2021*. Federal Bureau of Investigation. https://www.ic3.gov/Media/PDF/AnnualReport/2021_IC3Report.pdf

Fu, A. Y., Wenyin, L., & Deng, X. (2006). Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD). *IEEE Transactions on Dependable and Secure Computing*, *3*(4), 301–311. https://doi.org/10.1109/TDSC.2006.50

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. The MIT Press.

Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning*, *77*(1), 103–123. https://doi.org/10.1007/s10994-009-5119-5

Herr, T., & Armbrust, E. (2015). Milware: Identification and Implications of State Authored Malicious Software. *Proceedings of the 2015 New Security Paradigms Workshop*, 29–43. https://doi.org/10.1145/2841113.2841116

Hirshorn, S. R., Voss, L. D., & Bromley, L. K. (2017). *NASA Systems Engineering Handbook* (Special Publication (SP) NASA SP-2016-6105 Rev2). NASA. http://hdl.handle.net/2060/20170001761

Huang, K., Siegel, M., & Madnick, S. (2019). Systematically Understanding the Cyber Attack Business: A Survey. *ACM Computing Surveys*, *51*(4), 1–36. https://doi.org/10.1145/3199674

Hudaib, A., Masadeh, R., Qasem, M. H., & Alzaqebah, A. (2018). Requirements Prioritization Techniques Comparison. *Modern Applied Science*, *12*(2), 62. https://doi.org/10.5539/mas.v12n2p62

Jain, A. K., & Gupta, B. B. (2017). Phishing Detection: Analysis of Visual Similarity Based Approaches. *Security and Communication Networks*, *2017*, 1–20. https://doi.org/10.1155/2017/5421046

Khandelwal, S., & Das, R. (2022). Phishing Detection Using Computer Vision. In S. Smys, R. Bestak, R. Palanisamy, & I. Kotuliak (Eds.), *Computer Networks and Inventive Communication Technologies* (Vol. 75, pp. 113–130). Springer Singapore. https://doi.org/10.1007/978-981-16-3728-5_9

Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing Detection: A Literature Survey. *IEEE Communications Surveys & Tutorials*, *15*(4), 2091–2121. https://doi.org/10.1109/SURV.2013.032213.00009

King, D. E. (2015). *Max-Margin Object Detection*. https://doi.org/10.48550/ARXIV.1502.00046

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE-2007-01*. School of Computer Science and Mathematics.

Kotler, P., Keller, K. L., Brady, M., Goodman, M., & Hansen, T. (2019). *Marketing management* (4th European edition). Pearson.

Lam, I.-F., Xiao, W.-C., Wang, S.-C., & Chen, K.-T. (2009). Counteracting Phishing Page Polymorphism: An Image Layout Analysis Approach. In J. H. Park, H.-H. Chen, M. Atiquzzaman, C. Lee, T. Kim, & S.-S. Yeo (Eds.), *Advances in Information Security and Assurance* (Vol. 5576, pp. 270–279). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02617-1_28

Lin, S.-C., Wl, P.-C., Chen, H.-Y., Morikawa, T., Takahashi, T., & Lin, T.-N. (2022). SenseInput: An Image-Based Sensitive Input Detection Scheme for Phishing Website

Detection. *ICC 2022 - IEEE International Conference on Communications*, 4180–4186. https://doi.org/10.1109/ICC45855.2022.9838653

Mahdavifar, S., & Ghorbani, A. A. (2019). Application of deep learning to cybersecurity: A survey. *Neurocomputing*, *347*, 149–176. https://doi.org/10.1016/j.neucom.2019.02.056

MITRE. (2023a, January 4). *Internal Spearphishing, Technique T1534—Enterprise | MITRE ATT&CK*. https://attack.mitre.org/techniques/T1534/

MITRE. (2023b, January 4). *Phishing for Information: Spearphishing Attachment, Sub-technique T1598.002—Enterprise | MITRE ATT&CK*. https://attack.mitre.org/techniques/T1598/002/

MITRE. (2023c, January 4). *Phishing for Information, Technique T1598—Enterprise | MITRE ATT&CK*. https://attack.mitre.org/techniques/T1598/

MITRE. (2023d, January 4). *Phishing, Technique T1566—Enterprise | MITRE ATT&CK*. https://attack.mitre.org/techniques/T1566/

MITRE. (2023e, January 4). *Tactics—Enterprise | MITRE ATT&CK*. https://attack.mitre.org/tactics/enterprise/

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. *2020 11th International Conference on Information and Communication Systems (ICICS)*, 243–248. https://doi.org/10.1109/ICICS49469.2020.239556

Ndichu, S., Kim, S., Ozawa, S., Misu, T., & Makishima, K. (2019). A machine learning approach to detection of JavaScript-based attacks using AST features and paragraph vectors. *Applied Soft Computing*, *84*, 105721. https://doi.org/10.1016/j.asoc.2019.105721

NIST. (2023, January 4). *phishing—Glossary | CSRC*. https://csrc.nist.gov/glossary/term/phishing

Oest, A., Safei, Y., Doupe, A., Ahn, G.-J., Wardman, B., & Warner, G. (2018). Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. *2018 APWG Symposium on Electronic Crime Research (eCrime)*, 1–12. https://doi.org/10.1109/ECRIME.2018.8376206

Oest, A., Zhang, P., Wardman, B., Nunes, E., Burgis, J., Zand, A., Thomas, K., Doupé, A., & Ahn, G.-J. (2020). Sunrise to Sunset: Analyzing the End-to-End Life Cycle and Effectiveness of Phishing Attacks at Scale. *Proceedings of the 29th USENIX Conference on Security Symposium*, 361–377.

Oliva, A., & Torralba, A. (2001). Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, *42*(3), 145–175. https://doi.org/10.1023/A:1011139631724

Phoka, T., & Suthaphan, P. (2019). Image Based Phishing Detection Using Transfer Learning. *2019 11th International Conference on Knowledge and Smart Technology (KST)*, 232–237. https://doi.org/10.1109/KST.2019.8687615

Pires, C., & Borges, J. (2023a). Detecting Phishing Websites Using Artificial Intelligence and Computer Vision for Brand Protection. *ACM Computing Surveys (under Review)*.

Pires, C., & Borges, J. (2022a, October). Advanced Phishing Detection Platform for Cyber Threat Intelligence, Cybersecurity & Cyber Defence Purposes. *Proceedings of the ISMS 2022 Conference of the International Society of Military Sciences*.

Pires, C., & Borges, J. (2022b, November). Plataforma avançada de deteção de websites de phishing para efeitos de Cibersegurança e de Ciberdefesa. *Proceedings of the 4.º Encontro de Investigação e Desenvolvimento Em Ciências Militares (ECM 2022)*.

Pires, C., & Borges, J. (2023b, November). Detecting Targeted Phishing Websites for Brand Protection and Cyber Defence Using Computer Vision. *To Appear in the Proceedings of the 2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense 2023)*.

Powers, D. M. W. (2020). *Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation*. https://doi.org/10.48550/ARXIV.2010.16061

Rao, R. S., & Ali, S. T. (2015). A Computer Vision Technique to Detect Phishing Attacks. *2015 Fifth International Conference on Communication Systems and Network Technologies*, 596–601. https://doi.org/10.1109/CSNT.2015.68

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, *28*.

Rourke, L., Anderson, T., Garrison, D. R., & Walter, A. (2001). Methodological issues in the content analysis of computer conference transcripts. *International Journal of Artificial Intelligence in Education*, *12*, 8–22.

Roy, S., Sharmin, N., Acosta, J. C., Kiekintveld, C., & Laszka, A. (2023). Survey and Taxonomy of Adversarial Reconnaissance Techniques. *ACM Computing Surveys*, *55*(6), 1–38. https://doi.org/10.1145/3538704

Russell, S. J., & Norvig, P. (2022). *Artificial intelligence: A modern approach* (Fourth edition, global edition). Pearson.

Salahdine, F., & Kaabouch, N. (2019). Social Engineering Attacks: A Survey. *Future Internet*, *11*(4), 89. https://doi.org/10.3390/fi11040089

Sammut, C., & Webb, G. I. (Eds.). (2017). *Encyclopedia of Machine Learning and Data Mining*. Springer US. https://doi.org/10.1007/978-1-4899-7687-1

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Siddiq, Md. A. A., Arifuzzaman, M., & Islam, M. S. (2022). Phishing Website Detection using Deep Learning. *Proceedings of the 2nd International Conference on Computing Advancements*, 83–88. https://doi.org/10.1145/3542954.3542967

Strom, B., Applebaum, A., Miller, D., Nickels, K., Pennington, A., & Thomas, C. (2020). *MITRE ATT&CK: Design and Philosophy*. MITRE Corporation.

Thomas, J. E. (2018). Individual Cyber Security: Empowering Employees to Resist Spear Phishing to Prevent Identity Theft and Ransomware Attacks. *International Journal of Business and Management*, *13*(6), 1. https://doi.org/10.5539/ijbm.v13n6p1

Tian, K., Jan, S. T. K., Hu, H., Yao, D., & Wang, G. (2018). Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild. *Proceedings of the Internet Measurement Conference 2018*, 429–442. https://doi.org/10.1145/3278532.3278569

Trinh, N. B., Phan, T. D., & Pham, V.-H. (2022). Leveraging Deep Learning Image Classifiers for Visual Similarity-based Phishing Website Detection. *The 11th International Symposium on Information and Communication Technology*, 134–141. https://doi.org/10.1145/3568562.3568629

van den Hout, T., Wabeke, T., Moura, G. C. M., & Hesselman, C. (2022). LogoMotive: Detecting Logos on Websites to Identify Online Scams - A TLD Case Study. In O. Hohlfeld, G. Moura, & C. Pelsser (Eds.), *Passive and Active Measurement* (Vol. 13210, pp. 3–29). Springer International Publishing. https://doi.org/10.1007/978-3-030-98785-5_1

van Dooremaal, B., Burda, P., Allodi, L., & Zannone, N. (2021). Combining Text and Visual Features to Improve the Identification of Cloned Webpages for Early Phishing Detection. *The 16th International Conference on Availability, Reliability and Security*, 1–10. https://doi.org/10.1145/3465481.3470112

Vidas, T., Owusu, E., Wang, S., Zeng, C., Cranor, L. F., & Christin, N. (2013). QRishing: The Susceptibility of Smartphone Users to QR Code Phishing Attacks. In A. A.

Adams, M. Brenner, & M. Smith (Eds.), *Financial Cryptography and Data Security* (Vol. 7862, pp. 52–69). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-41320-9_4

Widup, S., Pinto, A., Hylender, D., Bassett, G., & Langlois, P. (2022). *2022 Data Breach Investigations Report*. https://doi.org/10.13140/RG.2.2.28833.89447

Xiang, G., Hong, J., Rose, C. P., & Cranor, L. (2011). CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites. *ACM Transactions on Information and System Security*, *14*(2), 1–28. https://doi.org/10.1145/2019599.2019606

Yang, P., Zhao, G., & Zeng, P. (2019). Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning. *IEEE Access*, *7*, 15196–15209. https://doi.org/10.1109/ACCESS.2019.2892066

You, Y., Zhang, Z., Hsieh, C.-J., Demmel, J., & Keutzer, K. (2018). ImageNet Training in Minutes. *Proceedings of the 47th International Conference on Parallel Processing*, 1–10. https://doi.org/10.1145/3225058.3225069

Zhang, H., Liu, G., Chow, T. W. S., & Liu, W. (2011). Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach. *IEEE Transactions on Neural Networks*, *22*(10), 1532–1546. https://doi.org/10.1109/TNN.2011.2161999

Zhang, P., Oest, A., Cho, H., Sun, Z., Johnson, R., Wardman, B., Sarker, S., Kapravelos, A., Bao, T., Wang, R., Shoshitaishvili, Y., Doupe, A., & Ahn, G.-J. (2021). CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing. *2021 IEEE Symposium on Security and Privacy (SP)*, 1109–1124. https://doi.org/10.1109/SP40001.2021.00021

Zhang, Y., Hong, J. I., & Cranor, L. F. (2007). Cantina: A content-based approach to detecting phishing web sites. *Proceedings of the 16th International Conference on World Wide Web*, 639–648. https://doi.org/10.1145/1242572.1242659

Zhao, J., Masood, R., & Seneviratne, S. (2021). A Review of Computer Vision Methods in Network Security. *IEEE Communications Surveys & Tutorials*, *23*(3), 1838–1878. https://doi.org/10.1109/COMST.2021.3086475

APPENDIX A – TEMPLATE FOR DATA EXTRACTION

Table A1: Data extraction form template

Field name	Definition
Internal Reference Number	The document tracking number as it pertains to this research.
DOI	Document DOI number.
Publication Year	Document publication year.
Document Title	Document title.
Publication Title	Publication where the research was published.
Authors	Document authors.
Abstract	Document abstract.
ACM CCS Concepts	Document ACM Computing Classification System (CCS) concepts.
Uses CV / IC / IP techniques to address a phishing problem?	Does the document use Computer Vision (CV), Image Classification (IC) or Image Processing (IP) techniques to address a phishing feature extraction or classification problem?
Type of Problem	Which problem does the document address? Feature Extraction, Classification or Both?
Feature Extraction	The feature extraction process, algorithm, methodology, or framework.
Features	Extracted features.
Training Methodology	What is the classifier training methodology? Are they supervised, semi-supervised or unsupervised?

Field name	Definition
ANN Classification Architecture	The Artificial Neural Network (ANN) classification architecture.
Classifier Methodology	Which classifier methodology was used?
Evaluation Dataset	Which dataset was used?
Evaluation Metric	Which evaluation metrics were used to determine the best-performing methodologies?
Implementation Platform	Technology in the research.
Best Results	Which methodologies performed best?
Is Code Available?	Is the implementation code available?
Challenges and Proposed Solutions	The list of challenges and proposed solutions identified by the author.
Notes	Remarks taken during document analysis.

APPENDIX B – METHODOLOGIES FOR IMAGE-BASED FEATURE EXTRACTION

Table B1: Feature extraction methodologies used in the analysed studies

Study ID	Extraction Methodology	Extracted Features	Notes
S-02	Web page layout image (screenshot).	Layout blocks (split image into non-overlapping areas).	None.
S-04	SIFT was applied to the website logo image file.	Website logo SIFT features.	This study relied on "website profiles" generated using additional non-visual features, such as certificate data, URL and HTML content; Logos were matched against the web page logo the user selected, not the complete web page screenshot.
S-05	GIST (Oliva & Torralba, 2001) was applied to generate website screenshot colour histograms;	Colour histograms (gist features); SIFT keypoint descriptors (sift-stats);	In addition to the three visual features described here, this study used URL, structural, page-link, and semantic features.

Study ID	Extraction Methodology	Extracted Features	Notes
	SIFT was applied to identify visual objects on the website screenshot.	SIFT matching images to a repository of common logos (sift-matching).	
S-06	Web page screenshot generated by making use of the graphic device interface API provided by the Microsoft Internet Explorer browser; The generated images were processed into normalized 100x100 images.	Each square image was processed into a feature vector comprised of two components and their corresponding weights: a degraded colour and the centroid of its position distribution in the image.	This study combined both textual and visual features; Although the authors stated that using only visual elements generates false positives, only the visual features extraction methodologies were considered for this SLR; EMD calculates the visual similarity between the suspect phishing and the legitimate web page.
S-07	Use of Machine Learning techniques to identify a logo file from all the downloaded image files; Based on Baratis, Petrakis and Milios's (2008) research.	Website logo image file.	The authors proposed to identify the logo image file from all the downloaded files during the website loading process; The logo file was the only extracted feature, and it was used on a reverse Google Image search.

Study ID	Extraction Methodology	Extracted Features	Notes
S-08	SURF was applied to the suspicious website screenshot.	Screenshot SURF features.	This study applied SURF to the suspicious website screenshot as well as the database of "known good" website screenshots to determine similarities; The algorithm was used for both feature extraction and similarity comparison.
S-09	HOG algorithm was applied to the suspicious website screenshot cropped into a 1024x1024 image area.	HOG descriptors feature vector.	This study performed experiments using two HOG configurations: HOG128, which divides the image into 8x8 cells of 128 pixels, and HOG64, which divides the image into 16x16 cells of 64 pixels each.
S-10	A Spatial Multi-Level Patch Pyramid (SMLPP) framework was applied to the suspicious website screenshot.	Scalable Colour Descriptor (SCD); Colour Layout Descriptor (CLD); Fuzzy Colour and Texture Histogram (FCTH); Colour and Edge Directivity Descriptor (CEDD); Joint Composite Descriptor (JCD).	In addition to the single website screenshot, this study also considered a fine-grained screenshot analysis by diving the image into 2x2, 3x3 and 4x4 grid cells for the SMLPP scheme, allowing for the acquisition of the visual features regardless of the screenshot size and preventing information loss due to cropping.

Study ID	Extraction Methodology	Extracted Features	Notes
S-11	OCR was applied to the suspicious website screenshot.	Text extracted by the OCR process.	This study extracted text from the rendered website screenshot instead of relying on the textual features to bypass countermeasures implemented by the adversary; The study also considers other non-visual features.
S-13	CNN applied to the suspicious website 224x224 resized screenshot.	Feature vector with 512 dimensions.	A 512-dimension feature vector represented each screenshot.
S-14	AlexNet CNN applied to the suspicious website 227x227 resized screenshot.	Feature vector extracted using the AlexNet CNN.	The study also considered other non-visual features, such as text and frames; For the non-image features, LSTM was used; This study's hybrid model combined the image feature using CNN with the non-image feature using LSTM.
S-15	HOG-based MMOD algorithm was applied to the suspicious website screenshot.	HOG descriptors feature vector.	MMOD is not a feature extraction method but rather an object detection algorithm; MMOD combines feature extraction and object detection into a single framework.

Study ID	Extraction Methodology	Extracted Features	Notes
S-16	CNN ResNet-52, where the final output has been removed; OCR is applied to the suspicious website screenshot.	Feature vector from the ResNet-52 output, where the final layer has been removed; TF-IDF encoded textual features.	This study extracted textual elements from the website screenshot using OCR.
S-17	Extracted regions from a suspicious website screenshot containing identifiable information.	Image regions that contain identifiable information.	The study extracted screenshot regions with identifiable information to perform a reverse search on a search engine; The study also considered other non-visual features.
S-18	Several CNNs, including VGGNet, GoogLeNet, Residual Network and DenseNet.	Descriptors generated by the CNNs.	The study used Transfer Learning techniques; This study experimented with images sized 128x128, 256x256 and 512x512; Dimensionality Reduction (Principal Component Analysis – PCA) technique was applied to the extracted features.

Study ID	Extraction Methodology	Extracted Features	Notes
S-19	Faster-RCNN was applied to identify areas with "sensitive input" fields from the suspicious website screenshot; OCR was used to extract text from the "sensitive input" regions.	Text from the "sensitive input" regions of the screenshot image.	The study also considered other non-visual features, such as statistical and textual extracted from the URL and the HTML code.
S-21	CNN models such as VGG16, VGG19, ResNet50, InceptionV3 and Xception trained with ImageNet; Features were extracted from the suspicious website screenshot using the transferred learnt parameters from the saved model.	Feature maps of the convolutional layers.	The suspicious website screenshot and training image screenshots were resized to 224x224; This study used Transfer Learning.

APPENDIX C – METHODOLOGIES AND THEIR EFFECTIVENESS IN CLASSIFYING PHISHING WEBSITES

Table C1: Analysed studies' classification methodologies and effectiveness

Study ID	Best Classification Methodology	Effectiveness	Notes
S-01	Final-TF-IDF.	FPR: 6% TPR: 97%	This classifier was used with non-visual features only; Final-TF-IDF combined the TOP 5 lexical terms extracted from the suspicious website using TF-IDF with the website domain and submitted them to Google to determine if the domain was on the TOP 30 search results.
S-02	Naïve Bayes.	Accuracy: 99.6% FNR: 0.003% FPR: 0.028%	A Naïve Bayes classifier was used to categorize pages based on the similarity score between web page pairs (phishing/original); A 10-fold cross-validation was performed in the classification performance evaluation.
S-03	NCD-based decision threshold.	F1 Score: 97.61% Kappa: 91.69% MCC: 91.93%	This study performed tests with both LZMA and Blocksort compressors; LZMA performed slightly better and is the one whose effectiveness is showcased in this table.

Study ID	Best Classification Methodology	Effectiveness	Notes
		Precision: 99.67% Recall: 95.63%	
S-04	Protected website profile comparison.	Accuracy: 96.1% FPR: 1.4%	This study used hybrid features; A website profile contains several non-visual features, independently compared when determining if a suspicious website is phishing; The effectiveness pertains to using "image" and "visible texts" when matching a profile.
S-05	SVM.	Precision: 97.6% Recall: 96.6%	This study used hybrid features; The effectiveness pertains to an imbalanced dataset with a 1:2 malicious-to-benign ratio.
S-06	Data fusion algorithm based on Bayesian theory.	N/A (see notes)	This study used hybrid features; The overall data fusion classification algorithm evaluated inputs from a textual classifier based on Naïve Bayes and an image classifier based on EMD;

Study ID	Best Classification Methodology	Effectiveness	Notes
			As this study evaluated the algorithm against eight distinct brands, with varying results for each brand, it is impossible to determine its overall efficacy.
S-07	SVM.	FNR: 0.2% FPR: 13% TNR: 87% TPR: 99.8%	The authors stated that their focus was not on the classification. As such, they have selected SVM, which they acknowledge may be suboptimal, though they claimed that "changing to an optimal classifier later is effortless".
S-08	Visual similarity matched against an image database.	FNR: 15.23% FPR: 20.11%	This study compared the extracted features with a known image database using the SURF algorithm; The effectiveness results were measured for a similarity threshold of 70%.
S-09	Visual similarity matched against an image database.	Similarity threshold: 75%	This study compared the extracted features with a known image database using the HOG descriptors; Tests were performed using HOG64 and HOG128, with the latter having the most robust and discriminative results; This study evaluated the similarity scores between several website pairs (phishing/original), where the 75% threshold was the most effective.

Study ID	Best Classification Methodology	Effectiveness	Notes
S-10	SVM.	F1 Score: 90.5% FPR: 8.5% TPR: 90.6%	The MPEG7 descriptor showcasing the best results was SCD with 1+4+9 patches and 3584 features.
S-11	RF.	Accuracy: 90% AUC: 0.97 FNR: 6% FPR: 3%	This study used hybrid features.
S-12	CNN – ResNet V1.	Accuracy: 97.1%	The study introduced phishing detection using Transfer Learning; The 97.1% accuracy pertains to the ResNet V1 CNN model when considering five-class classifications.
S-13	Triplet CNN (VisualPhishNet).	ROC Area: 0.9879 TOP-1: 81.03%	A visual website profile was created by calculating the similarity metric between two same-website pages, even with different content; This study introduced a new dataset featuring 155 websites and 9363 screenshots;

Study ID	Best Classification Methodology	Effectiveness	Notes
			The authors stated that their approach was robust against various evasion attacks.
S-14	Intelligent Phishing Detection Scheme, which combines LSTM and the AlexNet CNN	Accuracy: 93.28% F1 Score: 93.29% Precision: 93.30% Recall: 93.27%	This study used hybrid features; The hybrid model combined the image feature using CNN with the non-image feature using LSTM.
S-15	SVM.	F1 Score: 85.02% Precision: 93.5% Recall: 77.94%	The authors stated that although their work showed promising results, the features used with this classifier caused their approach to have some limitations; The authors proposed to overcome this problem by creating a CNN detection model for future work.
S-17	SSIM (see notes).	Accuracy: 99.66% AUC: 0.9938 F1 Score: 99.77% Precision: 99.55%	This study used hybrid features; The textual features were extracted from the web page DOM, and the visual from a screenshot; Both extracted features (textual and visual) were queried on a search engine, and only the top results were considered as being legitimate web pages;

Study ID	Best Classification Methodology	Effectiveness	Notes
			Although SSIM is not a classifier, it was used to compare the similarity of the suspicious website to the screenshots taken from the search engine's top query results; Logistic Regression was used to compute the threshold for the classification.
S-18	SVM.	F1 Score: 90.9% Recall: 90.8%	In this study, CNN-based models were used to extract features from screenshots; The best result was obtained using an image size of 512*512 with the DenseNet201 CNN architecture, having ingested the presentation of the images into the SVM classifier.
S-19	LightGBM.	F1 Score: 95.87% Precision: 96.78% Recall: 94.98%	This study used hybrid features, using statistical features, textual features extracted from the web page DOM as well as OCR, and visual features; Visual features were used as a source for the OCR extraction process, as well as the input for an object detection problem; The OCR extracts text from the image to determine whether it has sensitive-input text fields, and the object detection problem calculates if the image has a sensitive-input object;

Study ID	Best Classification Methodology	Effectiveness	Notes
			The effectiveness pertains to the overall classification problem, not the object detection problem used as input.
S-20	CNN.	Accuracy: 93.6% Recall: 98.4%	The authors reshaped the samples into an 8x8 matrix because the dataset lacks any images to use the Conv2D image classification; As the authors did not identify the best classification methodology, the extracted classification methodology was determined by prioritizing the recall results over accuracy.
S-21	VGG16 with LR.	AUC: 0.601 Accuracy: 88.68% F1 Score: 83.4% Precision: 86.96% Recall: 88.68%	The study used Transfer Learning with deep-learning image classifiers; A training data imbalance problem existed and negatively impacted training performance.

APPENDIX D – DEFINITIONS FOR THE METRICS

Accuracy: Accuracy is the ratio between a model's correct predictions and the total number of cases (Sammut & Webb, 2017).

$$\text{Accuracy} = \frac{TN + TP}{TN+TP+FN+FP}$$

Area Under Curve (AUC): The AUC is the area under the *Receiver Operating Characteristic (ROC)* curve and evaluates the performance of a binary classification model on a test set (Sammut & Webb, 2017). The AUC ranges from 0 to 1, with 0.5 indicating a random classifier and 1 indicating a perfect classifier (Hand, 2009).

Confusion Matrix: A Confusion Matrix is a summary of the classification performance of a classifier when classifying some test data (Sammut & Webb, 2017). Figure D1 showcases a Confusion Matrix for a Positive / Negative binary classification problem.

		Predicted Class	
		Positive (PP)	**Negative (PN)**
Actual Class	**Positive (AP)**	True Positive (TP)	False Negative (FN)
	Negative (AN)	False Positive (FP)	True Negative (TN)

Figure D1: Positive/negative binary classification Confusion Matrix

Legend: Actual Positive (AP); Actual Negative (AN); Predicted Positive (PP); Predicted Negative (PN)

F1 Score: The F1 Score, also known as *F-measure*, is the harmonic mean of *Precision* and *Recall*. A classifier with a high F1 score possesses high precision and recall (Sammut & Webb, 2017).

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

False Negative Rate (FNR): The FNR, also known as *Miss Rate*, is the ratio between the number of false negatives and the total number of positive cases (Powers, 2020).

$$FNR = \frac{FN}{FN + TP}$$

False Positive Rate (FPR): The FPR, also known as *Fallout*, is the ratio between the number of false positives and the total number of negative cases (Powers, 2020).

$$FPR = \frac{FP}{FP + TN}$$

Kappa: Kappa refers to Cohen's Kappa statistic (Rourke et al., 2001), a chance-corrected measurement of the agreement between two raters (Banerjee et al., 1999).

Matthews Correlation Coefficient: The MCC is a "contingency matrix method of calculating the Pearson product-moment correlation coefficient" (Powers, 2020, p. 42). It measures the quality of a binary classification model and produces a score ranging from -1 to 1, with 1 representing a completely perfect classification, 0 a random classification, and -1 a completely incorrect classification (Chicco & Jurman, 2020).

$$MCC = \frac{TN * TP - FN * FP}{\sqrt{(FP + TP) * (FN + TP) * (FP + TN) * (FN + TN)}}$$

Precision: Precision, also known as *Positive Predictive Value* and *Confidence*, is the ratio between the number of true positives and the total number of predicted positives by the model (Powers, 2020; Sammut & Webb, 2017).

$$\text{Precision} = \frac{TP}{FP + TP}$$

Recall: Recall, also known as *Sensitivity* and *True Positive Rate (TPR)*, is the ratio between the number of true positives and the total number of positive cases (Powers, 2020; Sammut & Webb, 2017).

$$\text{Recall} = \frac{TP}{FN + TP}$$

Receiver Operating Characteristic: An ROC graph is an instrument for visualising, organising, and selecting classifiers according to performance (Fawcett, 2006). The ROC analysis investigates and employs the relationship between a binary classifier's *Sensitivity (TPR)* and *Specificity (TNR)*. It plots the TRP against the FPR (Sammut & Webb, 2017), allowing the comparison of classifiers and choosing the one closest to (0,1) and farther from TPR=FPR (Powers, 2020). Variation of the decision threshold from its maximum to its minimum value produces a linear curve (ROC Curve) from (0,0) to (1,1) (Sammut & Webb, 2017).

TOP-1: Top-1 accuracy means the conventional *Accuracy*. In a multiclass prediction model, the classifier's output must be the expected answer (You et al., 2018).

True Negative Rate (TNR): TNR, also known as *Specificity*, is the ratio between the number of true negatives and the total number of negative cases (Sammut & Webb, 2017).

$$\text{TNR} = \frac{TN}{FP + TN}$$

True Positive Rate (TPR): See *Recall*.