



Title: Crypto Website Scam Detection via Open Source Intelligence

Author: AlGhaliah Taj A AlAbdulgohsin

Course: BSc Cyber Security and Forensic Computing

Project code: PJS40

Supervisor: Oluwatobi Fajana

Month Year: May 2026

Declarations

Please tick the following declaration

Usage (optional)	
☒	I give permission for this document to be made available to other students as examples of previous work.
Authenticity of work	
☒	I confirm that I have read the University rules in respect of plagiarism and student misconduct.
☒	I confirm that I have correctly referenced and cited any external source in any work that I produce, including the use of AI.
Ethics	
☒	I confirm that ethical approval was sort for this project regardless of whether I carried out primary research or not.
☒	I confirm that I did not carry out any primary research/the primary research carried out, matches the description in my ethics application [delete as applicable]
☒	I confirm that if I did conduct primary research, all raw data and research documentation is included in my final project report.
☒	I understand that not including the raw data can mean potentially failing the project.
☒	I understand that not obtaining ethical approval is grounds for academic misconduct and possible project failure.
Use of AI Artificial Intelligence (AI), such as ChatGPT, can be used as a tool to help assist and inform the initial development of your work but it is not a replacement for your own critical thinking and analysis. We require that your work is your own original content, demonstrating your knowledge, skills, and critical thinking abilities. To that end, AI-generated content must not be included in <u>any work</u> you submit for assessment unless suitably referenced. Not doing so is a breach of the University's Academic Regulations as outlined in the Student Conduct Policy and constitutes an assessment offence.	
☒	I have read the University AI policy and if relevant, I have acknowledged the use of AI in any work I produced for this project with appropriate referencing.

	I understand that not appropriately referencing the use of AI is an assessment offence equal to plagiarism and may result in a penalty which remains on my student record.
---	--

Name: AlGhaliah Taj A.

Date: 08 May 2026

Acknowledgements

First and foremost, thank you to my supervisor Oluwatobi Fajana for the direction, patience and feedback throughout this dissertation.

To my dad, mum and brother, your support carried me through this more than you probably realise. I could not have done it without you.

To my best friend, thank you for being there every time this got overwhelming and never once making me feel like I was being dramatic about it.

Cryptocurrency scams are the new face of digital fraud, and I have seen what they do to real people. People I know have personally fallen victim to these websites. Getting to build something that could actually help prevent that from happening to someone else made this whole process worth it.

Abstract

Cryptocurrency scams caused over £9.39 billion in verified global losses in 2024, yet existing detection methods remain reactive, narrow in scope and difficult to explain. This project introduces the Cryptocurrency Investigative OSINT Framework (CIOF), a structured four-layer investigative workflow that analyses suspicious cryptocurrency websites using publicly accessible tools across infrastructure, content, social media and blockchain dimensions.

Evaluated on a dataset of 875 labelled URLs, a Random Forest classifier achieved 91.4% accuracy, 92.9% precision and an F1-score of 0.925. Unlike binary classifiers, CIOF produces graduated risk classifications (Legitimate, Suspicious, Scam) alongside inspectable feature-level evidence, making it practically useful for analysts, regulators and smaller organisations without enterprise infrastructure.

Table of Contents

Declarations.....	2
Acknowledgements	5
Abstract	6
Table of Contents	7
Table of Figures.....	10
Table of Tables.....	11
List of Abbreviations	12
Chapter 1: Introduction.....	14
1.1 Project Background	14
1.2 Research Problem	17
1.3 Aim	21
1.4 Objectives:	21
1.5 Research Questions	23
1.6 Project Management and Approach	24
1.7 Ethical, Legal and Professional Issues	25
1.7.1 Use of publicly available information	25
1.7.2 Data protection and privacy	25
1.7.3 Dual-use concerns and responsibilities disclosure	26
1.7.4 Professional standards and academic integrity	26
1.8 Structure of the Report	27
1.9 Chapter Summary.....	28
Chapter 2: Literature Review.....	29
2.1 Introduction	29
2.2 Literature Search Strategy	30
2.2.1 Databases and sources	30
2.2.2 Keywords and search strings.....	31
2.2.3 Inclusion and exclusion criteria.....	32
2.2.4 Screening and selection process.....	33
2.3 Categories of Cryptocurrency Scams and Scam Websites.....	33
2.4 Techniques for Detecting Scam and Phishing Websites	36
2.4.1 List-based and heuristic detection	36

2.4.2	DNS, WHOIS and Infrastructure-based methods	37
2.4.3	Machine-learning techniques	37
2.4.4	Transaction and address level detection in cryptocurrencies.....	39
2.4.5	Large-scale discovery of scam websites and OSINT-driven methods	40
2.4.6	LLM-based and knowledge-grounded detection	41
2.5	Critical Comparison and Research Gap	43
2.6	Chapter summary	46
Chapter 3: Methodology and OSINT Framework		47
3.1	Introduction.....	47
3.2	Research Design: The Research Onion.....	47
3.2.1	Research Philosophy and Approach	47
3.2.2	Research Strategy: The Artifact as Research.....	48
3.3	The investigative Process: The Intelligence Cycle	49
3.4	The OSINT Framework Architecture	50
3.4.1	Layer 1: Infrastructure and Domain Intelligence.....	50
3.4.2	Layer 2: On-Site Content and Behavioural Analysis.....	50
3.4.3	Layer 3: Off-Site Footprint and Social Proof	50
3.4.4	Layer 4: Blockchain and Transactional Linkage	51
3.5	Data Sources and Collection Methodology	52
3.5.1	Tools and Techniques	52
3.5.2	Ethical and Secure Collection.....	52
3.6	Sampling and Dataset Construction.....	53
3.7	Classification Model Design: Random Forest	53
3.7.1	Knowledge-Grounded Reasoning.....	53
3.8	Evaluation Metrics	54
3.9	Chapter Summary	55
Chapter 4: Data Analysis and Results.....		56
4.1	Introduction.....	56
4.2	Dataset Statistics and Data quality.....	56
4.3	Classification performance results	58
4.3.1	Three-class Risk Classification.....	60
4.4	Comparative Performance Analysis	62
4.5	OSINT Feature Importance and Qualitative Findings.....	63
4.5.1	Feature Importance Analysis	63

4.5.2 Qualitative Case Studies	65
4.6 Chapter Summary	67
Chapter 5: Discussion.....	68
5.1 Introduction	68
5.2 RQ2: Which OSINT indicators proved most informative?.....	70
5.4 RQ4: How effective is CIOF overall?	74
5.5 Practical Implications.....	75
5.6 Limitations	76
5.7 Chapter Summary.....	77
Chapter 6: Conclusion and Future Work	78
6.1 Did the Project Meet its Aim?.....	78
6.2 Reflections on Project Management	78
6.3 Contributions of This Work	79
6.4 Future Work	80
6.5 Closing Thoughts	81
References	82
Appendix A: Project Initiation Document.....	85
Appendix B: Ethics Certificate.....	96
Appendix C: Gantt Chart.....	97
Appendix D: Actual Gantt Chart.....	98
Appendix E: Jupyter Notebook Source Code	99

Table of Figures

Figure 1: Illustrative growth in reported cryptocurrency scam losses (2021--2025)	16
Figure 2: High-level OSINT workflow for cryptocurrency website scam detection.	18
Figure 3: Conceptual flow and roadmap of the literature review process.....	31
Figure 4: Literature selection flow diagram.	34
Figure 5: Intelligence Cycle Diagram.	50
Figure 6: CIOF Multi-Layered Data Collection Pipeline.....	52
Figure 7: Dataset label distribution and site availability at collection time.	59
Figure 8: Confusion matrix showing classification results across the test set.	61
Figure 9: Visual summary of CIOF classification performance metrics.....	62
Figure 10: CIOF three-class risk classification output.	63
Figure 11: CIOF feature importance rankings	65
Figure 12: CIOF feature importance rankings with investigative annotations.....	72
Figure 13: CIOF three-class risk classification output and operational decision pathways.	76
Figure 14: CIOF future development priorities matrix.	81

Table of Tables

Table 1: Mapping between objectives and research questions.....	24
Table 2: Representative categories of cryptocurrency scams and typical website roles.....	36
Table 3: Critical comparison of scam detection techniques.....	42
Table 4: Methodological layers of research design.....	48
Table 5: The CIOF Technical Stack.....	53
Table 6: Taxonomy of weighted OSINT indicators used in the CIOF model.	54
Table 7: Conceptual Confusion Matrix for Website Collection.....	56
Table 8: Summary of CIOF Random Forest classification performance on the 175-URL test set.	60
Table 9: Three-class risk classification output across the 175-URL test set.....	63
Table 10: Comparative positioning of CIOF against related work.	71
Table 11: Comparison of existing detection approaches and CIOF's response	73
Table 12: Assessment of project objectives against outcomes.....	78

List of Abbreviations

Abbreviation	Definition
ACCC	Australian Competition and Consumer Commission
AI	Artificial Intelligence
API	Application Programming Interface
BTC	Bitcoin
BS4	BeautifulSoup 4
CIOF	Cryptocurrency Investigative OSINT Framework
CNN	Convolutional Neural Network
DeFi	Decentralised Finance
DNS	Domain Name System
ETH	Ethereum
F1	F1-Score (harmonic mean of precision and recall)
FBI	Federal Bureau of Investigation
FTC	Federal Trade Commission
GDPR	General Data Protection Regulation
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICO	Information Commissioner's Office
INTERPOL	International Criminal Police Organization
IP	Internet Protocol
JS	JavaScript
LLM	Large Language Model
MAS	Monetary Authority of Singapore
ML	Machine Learning
NLP	Natural Language Processing
OSINT	Open Source Intelligence
PID	Project Initiation Document
PRAW	Python Reddit API Wrapper
RAG	Retrieval-Augmented Generation
RF	Random Forest
RQ	Research Question
SSL	Secure Sockets Layer
SVM	Support Vector Machine
TLD	Top-Level Domain

TLS	Transport Layer Security
URL	Uniform Resource Locator
WHOIS	Who Is (domain registration database)

Chapter 1: Introduction

1.1 Project Background

Over the past decade, the cryptocurrency market has grown from a niche experiment into a global financial ecosystem with a total valuation exceeding £800 billion. Bitcoin alone has fluctuated between approximately £26,000 and £92,000 in recent years, and thousands of alternative coins and tokens have been launched to support decentralized finance (DeFi), non-fungible tokens (NFTs), and other blockchain-based services (Nakamoto, 2008). This rapid expansion has created unprecedented opportunities for innovation, cross-border transactions, and financial inclusion. At the same time, it has created a very large and relatively lightly regulated surface for cyber-enabled financial crime.

Fraudulent cryptocurrency websites have become one of the main vehicles for such crime. These include fake exchanges, fraudulent investment platforms, cloned project sites, phishing portals that harvest private keys, and sites used to support Ponzi or multi-level marketing schemes. Recent industry analyses report that cryptocurrency scams received at least the equivalent of £8 billion on-chain in 2024 alone, even before all associated illicit addresses had been identified, and that total scam-related losses may exceed £10 billion per year when off-chain fraud and unreported incidents are considered (Chainalysis, 2025). Law-enforcement and consumer-protection statistics paint a consistent picture. The FBI's Internet Crime Complaint Center recorded several billion pounds worth of reported investment fraud involving cryptocurrency in a single year, while the United States Federal Trade Commission reported more than £10 billion in overall consumer fraud losses in 2024, with cryptocurrency investment fraud among the largest categories and older adults over 60 disproportionately affected (Federal Bureau of Investigation, 2024; Federal Trade Commission, 2025).



Figure 1: Illustrative growth in reported cryptocurrency scam losses (2021-2025) (Chainalysis, 2021, 2022, 2023, 2024, 2025).

The methods used by scammers have grown more sophisticated over time. So-called “pig-butcher” schemes involve building trust with victims over weeks or months through messaging applications or social media before directing them to convincing but fraudulent trading websites (Europol, 2024). High-yield investment platforms promise unrealistic fixed returns and display fabricated performance dashboards. Crypto “drainer” campaigns impersonate legitimate decentralised finance projects or wallet providers, tricking users into signing malicious transactions that silently transfer funds to attacker-controlled addresses. Phishing websites, often distributed through sponsored search results or phishing emails, clone the appearance of well-known exchanges and wallet interfaces in order to capture login credentials or seed phrases. Exit scams remain common: developers build an apparently legitimate project, attract deposits or investments through an attractive website, then close down the site and disappear with the funds.

From an investigative perspective, cryptocurrencies present a mixture of transparency and opacity. Public blockchains make all transfers between addresses visible, which enables analysts to trace flows of funds, cluster related addresses and identify patterns that are characteristic of certain scams. At the same time, attackers make extensive use of mixers, cross-chain bridges, privacy-enhancing technologies and high-risk exchanges in order to

frustrate simple tracing (Chainalysis, 2025). Many victims first encounter the scam through a website, social-media promotion or messaging-channel invitation rather than directly on the blockchain. As a result, understanding the full infrastructure behind a scam requires combining on-chain analysis with a wide range of off-chain information.

Open Source Intelligence (OSINT) investigations have shown that successful crypto scams are rarely isolated events. They tend to be supported by an infrastructure that can include domains and subdomains registered with disposable email addresses or privacy-protection services, shared hosting providers and IP ranges reused across multiple scams, recycled website templates and legal text, social-media accounts and messaging channels used for promotion, and recurrent wallet addresses and transaction patterns associated with other known scams (Acharya et al., 2024; Muzammil et al., 2025). Analysts, journalists and threat-intelligence practitioners routinely use OSINT sources such as WHOIS and DNS history, corporate registries, blockchain explorers, search engines, social-media search and regulator warning lists to build a picture of such infrastructures (Acharya et al., 2024; Muzammil et al., 2025).

However, this work is often manual, time-consuming and highly dependent on the individual investigator's experience. In academic literature, OSINT-style investigations are mentioned only briefly compared with machine-learning and heuristic techniques (Safi and Singh, 2023, pp. 590–592). Within crypto scam detection specifically, most published work focuses on automated classification of transactions or website features rather than formalising the OSINT investigative process (Safi and Singh, 2023, pp. 604–606).

This project focuses on the problem of detecting fraudulent cryptocurrency websites by formalising this OSINT-based investigative process into a repeatable framework that can be evaluated and, in future, integrated with automated tools.

Crypto Website Scam Detection via Open Source Intelligence



Figure 2: High-level OSINT workflow for cryptocurrency website scam detection.

This chapter was completed by the end of January 2026, in line with the original project plan.

1.2 Research Problem

A substantial body of work on phishing and malicious-website detection has progressed from static blacklists and simple heuristic rules to machine-learning (ML) classifiers that exploit URL, HTML and lexical features (Safi and Singh, 2023). More recent studies have applied deep-learning and ensemble models, achieving high accuracy on benchmark datasets and public phishing corpora (Safi and Singh, 2023). Systems such as SCAMNET show that fine-tuned large language models can detect fraudulent shopping websites with very high accuracy on temporal datasets, substantially outperforming traditional ML baselines (Bitaab et al., 2025a). Despite these advances, several key limitations persist when such techniques are applied to modern cryptocurrency-scam websites.

1. Reactivity and feature drift

Blacklist-based and many ML approaches are inherently reactive, because they are trained on historical data and known examples of malicious websites. Attackers can evade these systems by making minor changes to URLs, modifying page layouts or relocating to different hosting providers, thereby altering the feature distribution on which the models rely (Safi and Singh, 2023). In the fast-moving cryptocurrency ecosystem, where new domains and projects can be created within minutes and scam infrastructure is frequently rotated, this problem of feature drift is particularly acute. The Crimson study underscores how quickly new fraudulent domains appear, and how heavily scammers depend on disposable hosting resources, which complicates static, list-based defences (Muzammil et al., 2025).

2. Narrow data scope

Traditional phishing-detection systems usually concentrate on a limited set of features, such as URL length and structure, presence of suspicious tokens, basic SSL information, page-content cues and sometimes DNS or WHOIS attributes (Safi and Singh, 2023). By contrast, effective investigation of cryptocurrency scams often depends on richer OSINT signals, including wallet-address clusters, transaction patterns, cross-platform social-media promotion, project team claims checked against corporate registries, and prior appearances of the same infrastructure in known scams. OSINT-driven studies of cryptocurrency cybercrime show that combining victim abuse reports, labelled criminal addresses and transaction profiles yields a far more comprehensive representation of criminal behaviour than any single channel alone (Real-CATS, 2025). Systems such as ScamSweeper, which model transaction graphs to detect illegal accounts in Web3 scams, demonstrate the importance of relational and behavioural features that go beyond surface-level website attributes (Li et al., 2025). Narrow, single-channel feature sets struggle to capture this multidimensional reality.

3. Opacity and lack of explainability.

Complex ML and deep-learning models often behave as black boxes, providing little insight into why a given website has been labelled as malicious (Safi and Singh, 2023). For investigators, regulators and victims, this lack of explainability is problematic. Decisions to block, report or prosecute based purely on opaque model outputs are difficult to justify and offer limited evidential value in legal or regulatory contexts. In contrast,

OSINT investigations that trace domain registrations, hosting infrastructure, social-media activity and fund flows can generate a clear chain of evidence that supports decision-making and potential enforcement action. Frameworks for large-scale analysis of fraudulent shopping websites, for example, emphasise the value of linking domains to merchant identifiers and financial-institution intelligence in order to explain how particular sites fit into broader scam campaigns (Bitaab et al., 2025b).

4. Limitations of large language models for scam detection.

Recent research has begun to explore the use of large language models (LLMs) for detecting cryptocurrency scams, particularly in textual content such as promotional posts, white papers and user complaints. Knowledge-grounded frameworks that combine retrieval-augmented generation with LLMs show improved performance and substantially lower hallucination rates compared with models that rely solely on their internal parametric memory (Li, 2025). One such system achieves around 22 percent higher recall at equivalent precision than fixed-threshold baselines, retains about 89 percent of its performance on emerging scam types over time and reduces hallucinations by a factor of more than four, while operating in real time and providing evidence-grounded outputs (Li, 2025).

While these systems improve the reliability of scam judgements on a given input, such as a token description or social-media thread, they generally assume that the suspicious entity has already been identified and that relevant evidence is indexed in an existing database. They do not formalise how to discover new fraudulent websites in the first place, how to pivot between different OSINT sources or how to structure an end-to-end investigative workflow starting from a URL. Their knowledge bases also need constant updating to track rapidly evolving scam types.

5. Fragmented OSINT practice and lack of a unified framework

In practice, OSINT work around cryptocurrency scams is already being carried out by security researchers, investigative journalists, independent analysts and some law-enforcement agencies. Technical support scam studies such as HoneyTweet demonstrate how decoy posts and automated interaction can be used to attract scammers, map their operational flow and identify payment methods and wallet addresses (Acharya et al., 2024). Large-scale scam-discovery systems such as Loki show that mining search-engine result pages with carefully selected queries can surface tens of thousands of previously unknown scam websites across multiple categories (Paudel and Stringhini,

2026). Datasets such as Real-CATS provide labelled criminal and benign addresses at scale, enabling more realistic evaluation of address- and transaction-level detection methods (Real-CATS, 2025).

However, these contributions are distributed across different problem formulations and data granularities. Some focus on discovering scam domains at scale without deep investigation of each website’s surrounding infrastructure (Muzammil et al., 2025; Paudel and Stringhini, 2026). Others concentrate on detecting malicious accounts or addresses on blockchains rather than on classifying websites as such (Real-CATS, 2025; Li et al., 2025). Still others use LLMs to classify website content or generate explanations, but they do not specify a general OSINT methodology for gathering and correlating indicators from multiple open sources (Bitaab et al., 2025a; Li, 2025).

Overall, the literature indicates that:

- General phishing and malicious-website detection research offers mature classifiers but tends to ignore the full breadth of OSINT signals that are particularly relevant for cryptocurrency scams (Safi and Singh, 2023).
- Blockchain-investigation and OSINT studies demonstrate powerful techniques for linking pseudonymous activity to real-world actors, yet they usually target transaction or address classification or detailed forensic investigation, rather than lightweight, reusable website-level detection frameworks (Real-CATS, 2025; Acharya et al., 2024).
- LLM-based and knowledge-grounded approaches improve judgement and explainability on given inputs but typically rely on separate discovery processes and static or semi-static knowledge bases (Li, 2025).

6. Research gap

Existing detection methods are reactive, limited in data breadth and often opaque, whereas OSINT-driven investigations are powerful but informal and unstandardised. There is currently no structured, evaluable OSINT framework dedicated to detecting fraudulent cryptocurrency websites that:

- systematically integrates diverse open sources at the website level,
- produces evidence-rich outputs that can support human decision-making and potentially legal processes, and
- can be feasibly implemented with open and publicly accessible tools.

CIOF fills that gap by combining infrastructure data, website content, social media signals, and blockchain evidence into a single, repeatable investigation workflow.

1.3 Aim

The aim of this project is:

The aim of this project is to design, implement and evaluate the Cryptocurrency Investigative OSINT Framework (CIOF), a structured four-layer investigative workflow that synthesizes infrastructure, content, social media and blockchain intelligence to detect and classify fraudulent cryptocurrency websites using only publicly accessible tools and data sources. The framework is intended to address the gap between existing reactive, single-channel detection methods and the need for an explainable, analyst-facing investigation workflow that produces evidence-grounded, three-class risk classifications suitable for use by security analysts, regulators and smaller organizations without enterprise infrastructure.

1.4 Objectives:

To achieve this aim, the project is guided by the following objectives:

1. Review and synthesis of existing techniques

Conduct a critical literature review of cryptocurrency-scam detection and related web-fraud detection techniques, including heuristic, blacklist-based, DNS and WHOIS-based, traditional machine-learning, deep-learning and LLM-based approaches, with particular attention to their strengths, limitations and applicability to cryptocurrency websites (Safi and Singh, 2023; Muzammil et al., 2025; Li, 2025).

2. Identification of OSINT indicators and data sources

Identify and catalogue key OSINT indicators associated with fraudulent cryptocurrency websites, including domain-registration and hosting patterns, corporate-registry information, blockchain transaction patterns, social-media and community footprints and inclusion in regulator warning lists and public scam databases (Real-CATS, 2025; Paudel and Stringhini, 2026; Acharya et al., 2024).

3. Design of an OSINT-based detection framework

Design a structured, repeatable OSINT workflow that specifies how to collect, process and correlate these indicators from publicly available sources in order to assess the risk associated with a given cryptocurrency website. The framework should be suitable for later integration with automated tools and should emphasise transparency and explainability (informed by existing large-scale scam-detection and OSINT methodologies such as Crimson, Loki and HoneyTweet) (Muzammil et al., 2025; Paudel and Stringhini, 2026; Acharya et al., 2024).

4. Implementation of a supporting model

Implement a prototype model or rule-based classifier that ingests the collected OSINT indicators and produces a classification, such as legitimate, suspicious or scam, alongside an evidential justification. This will enable empirical evaluation of the framework on curated datasets of known scam and legitimate cryptocurrency websites (Real-CATS, 2025; Li et al., 2025).

5. Evaluation and reflection

Evaluate the performance and practical usefulness of the framework using standard metrics such as accuracy, precision, recall and F1-score where applicable, and qualitatively assess the investigative value of the generated outputs. Reflect on limitations and propose avenues for future improvement, including potential integration with advanced LLM-based or RAG-based systems for content analysis (Li, 2025; Bitaab et al., 2025a).

Objective No.	Objective Description	Linked Research Question(s)
1	Conduct a critical literature review of cryptocurrency scam detection and related web-fraud detection techniques.	RQ1
2	Identify and catalogue key OSINT indicators and data sources associated with fraudulent cryptocurrency websites.	RQ2
3	Design a structured, repeatable OSINT workflow for collecting, processing and correlating indicators from publicly available sources.	RQ3
4	Implement a prototype model or rule-based classifier that ingests OSINT indicators and produces a website-level risk classification.	RQ3, RQ4
5	Evaluate the performance and practical usefulness of the OSINT framework and prototype model using quantitative metrics and qualitative analysis.	RQ4

Table 1: Mapping between objectives and research questions

1.5 Research Questions

The project is structured around the following research questions:

- RQ1: What are the principal strengths and limitations of existing cryptocurrency-scam website detection techniques, including heuristic, DNS and WHOIS-based, machine-learning, deep-learning and LLM-based approaches?
- RQ2: Which OSINT indicators and data sources are most informative for distinguishing fraudulent cryptocurrency websites from legitimate ones, and how reliably can these indicators be obtained from publicly available sources?

- RQ3: How can these OSINT indicators be integrated into a structured, repeatable framework that supports systematic investigation and classification of cryptocurrency websites, while remaining practical for analysts to apply?
- RQ4: How effective is the proposed OSINT-based framework, in terms of both quantitative performance metrics and qualitative investigative value, when applied to real-world datasets of known scam and legitimate cryptocurrency websites?

1.6 Project Management and Approach

This project adopts an iterative, individual Agile-inspired project management approach rather than a rigid Waterfall model. In exploratory research on emerging threats, both requirements and understanding naturally evolve as new literature is reviewed and initial experiments are conducted. A fixed sequence in which all analysis and design decisions must be finalised before any implementation is poorly suited to such environments.

Instead, the work is organised into a series of time-boxed iterations, each with clear goals, such as: completing an initial literature review on phishing and crypto-scam detection; mapping and prioritising OSINT indicators and data sources; designing and refining the OSINT workflow; implementing data-collection scripts and the prototype classification model; and conducting evaluation and analysis. At the end of each iteration, progress is reviewed against the objectives and research questions, and subsequent tasks are adjusted accordingly. Regular supervision meetings are used to validate direction, address risks, such as difficulties in accessing reliable datasets, and ensure that ethical and legal constraints remain central.

Methodologically, the project is grounded in OSINT and blockchain-investigation practices that emphasise systematic collection, processing, analysis and dissemination of intelligence. The proposed framework follows a similar logic: planning and direction, collection of open-source indicators, processing and correlation, analysis and risk assessment, and dissemination in the form of structured reports or evidential summaries. This mirrors established intelligence-cycle models while adapting them to the specific problem of cryptocurrency website scam detection.

1.7 Ethical, Legal and Professional Issues

Because this project involves investigating potentially malicious online activity and handling data that may relate to identifiable individuals, ethical, legal and professional considerations are central to its design.

1.7.1 Use of publicly available information

The research relies exclusively on information that is lawfully and publicly accessible. This includes open blockchain explorers, public WHOIS and DNS records, corporate registries that are open to public search, publicly visible social-media content and regulator or industry warning lists (Acharya et al., 2024; Paudel and Stringhini, 2026). No authentication-protected services are accessed without permission, and no attempts are made to bypass technical access controls or to intrude into systems. Activities such as port-scanning, denial-of-service testing or exploitation of vulnerabilities are strictly out of scope.

1.7.2 Data protection and privacy

OSINT investigations can involve incidental collection of personal data, for example names and contact details of company officers listed in corporate registries or of individuals promoting a project on social media. The project therefore follows the principle of data minimization: only those data elements that are necessary for the purposes of scam-website classification are collected and stored. Where individual names or contact details appear in case studies or examples in the dissertation, they are anonymized or pseudonymized.

Processing of any personal data is designed to be consistent with applicable data protection legislation, including the UK General Data Protection Regulation (UK GDPR) and the Data Protection Act 2018 (Information Commissioner's Office. (2023)). This means ensuring a clear research purpose, limiting data retention and maintaining appropriate security for stored data. No special category personal data are intentionally collected.

1.7.3 Dual-use concerns and responsibilities disclosure

OSINT techniques are inherently dual use. The same tools that can be used to expose scams and protect potential victims could, in principle, be misused to profile targets or identify vulnerabilities in legitimate services. To maintain a clear ethical stance, the framework is presented explicitly as a defensive and investigative tool intended to support scam detection, consumer protection and academic understanding. It does not include instructions for exploiting identified weaknesses or for designing scams.

If, in the course of the research, security vulnerabilities or other serious issues affecting legitimate services are incidentally discovered, they will be handled in line with responsible disclosure norms. This typically involves notifying the affected party or a recognised coordination organisation, rather than publishing sensitive technical details.

1.7.4 Professional standards and academic integrity

The project is conducted in line with relevant professional and academic standards. This includes accurate and honest reporting of methods and results, proper acknowledgement of tools, datasets and prior work, avoiding plagiarism and fabrication of data and respecting any licensing or terms-of-use conditions associated with OSINT tools and data sources. By grounding the framework in published OSINT, scam-detection and blockchain-investigation research, and by clearly separating empirical findings from interpretation, the project aims to contribute constructively to the field (Muzammil et al., 2025; Li, 2025; Acharya et al., 2024).

1.8 Structure of the Report

The remainder of the dissertation is structured as follows:

- Chapter 2 – Literature Review

This chapter surveys the academic and industry literature on phishing and web-fraud detection, cryptocurrency scam detection, OSINT in cybercrime investigations and blockchain analysis tools and methods. It highlights strengths and limitations of existing approaches and clarifies the research gap that motivates the proposed OSINT framework.

- Chapter 3 – Methodology and OSINT Framework

This chapter describes the research design and explains how the OSINT framework was developed. It details the data sources, collection and processing procedures, the structure of the investigative workflow and the design of the supporting classification model, drawing on best practices from OSINT and blockchain-investigation methodologies.

- Chapter 4 – Implementation and Results

This chapter presents the implementation of the framework and model, the construction of datasets of known scam and legitimate cryptocurrency websites and the experimental setup used to evaluate performance. It then reports and visualises the quantitative and qualitative results.

- Chapter 5 – Discussion

This chapter interprets the results in relation to the research questions and existing literature, discusses the strengths and limitations of the framework, reflects on practical implications for analysts and regulators and considers how OSINT, machine-learning and LLM-based approaches might be combined.

- Chapter 6 – Conclusion and Future Work

The final chapter summarises the main contributions of the project, evaluates how far the aim and objectives have been met and proposes directions for future work, such as integrating the framework into real-time monitoring systems or extending it to other forms of Web3 fraud.

1.9 Chapter Summary

Chapter 1 has established the foundation for this research. Cryptocurrency website scams represent a growing and under-addressed threat, with verified losses exceeding £9 billion in 2024 alone, and existing detection methods are ill-equipped to respond. Their limitations are consistent across the literature: they react to scams rather than anticipate them, draw on a single data source rather than synthesising multiple channels, and rarely produce outputs that investigators can interrogate or defend. CIOF addresses these shortcomings through a structured, multi-layer OSINT framework designed for proactive website-level investigation. Five objectives shape the work, four research questions guide the analysis, and an iterative project management approach allows the methodology to develop in response to emerging findings. Ethical boundaries are defined and observed throughout, with collection restricted to publicly available sources. Chapter 2 examines the existing body of research in detail, assessing what current detection approaches achieve and where the gaps that motivate this project remain.

Chapter 2: Literature Review

2.1 Introduction

Cryptocurrency has exploded into an £800+ billion market over the past decade, and so have the scams. Fraudsters now build professional-looking websites that mimic legitimate exchanges like Coinbase or Binance. These fake sites look convincing: they have SSL certificates, polished user interfaces and even fake customer testimonials (Cialdini, 2009). Once funds are transferred, the irreversibility of blockchain transactions means victims have no recourse. Between 2021 and 2024, verified cryptocurrency scam losses reached £24.33 billion globally (Chainalysis, 2021, 2022, 2023, 2024), with 2024 alone accounting for £9.39 billion. Investment scams, fake exchanges, phishing portals and fraudulent DeFi platforms have become a billion-pound-per-month problem, yet existing detection methods cannot keep pace. A recent longitudinal study introduced Crimson, a distributed system that continuously detects cryptocurrency investment scam websites in the wild, and showed that over the first eight months of 2024 it classified 43,572 unique scam websites by processing around six billion domain names in real time (Muzammil et al., 2025). Even when analysing only a small subset of associated wallets, the study estimated lower-bound losses of at least £1.6 million and suggested that total losses across the full campaign are likely to be in the tens of millions of pounds (Muzammil et al., 2025).

This chapter reviews the academic and industry literature relevant to detecting these scams, with a particular focus on fraudulent cryptocurrency websites and on Open Source Intelligence (OSINT) techniques that can be used to investigate them. It first explains the literature search strategy that was used to identify relevant work. It then introduces the main categories of cryptocurrency scams and scam websites, in order to clarify the threat landscape that detection methods seek to address. The review covers list-based and heuristic methods first, then DNS and WHOIS analysis, machine learning, transaction-level blockchain detection, large-scale OSINT discovery systems, and finally LLM-based approaches. Each section identifies what that approach gets right and where it falls short for crypto scam websites specifically.

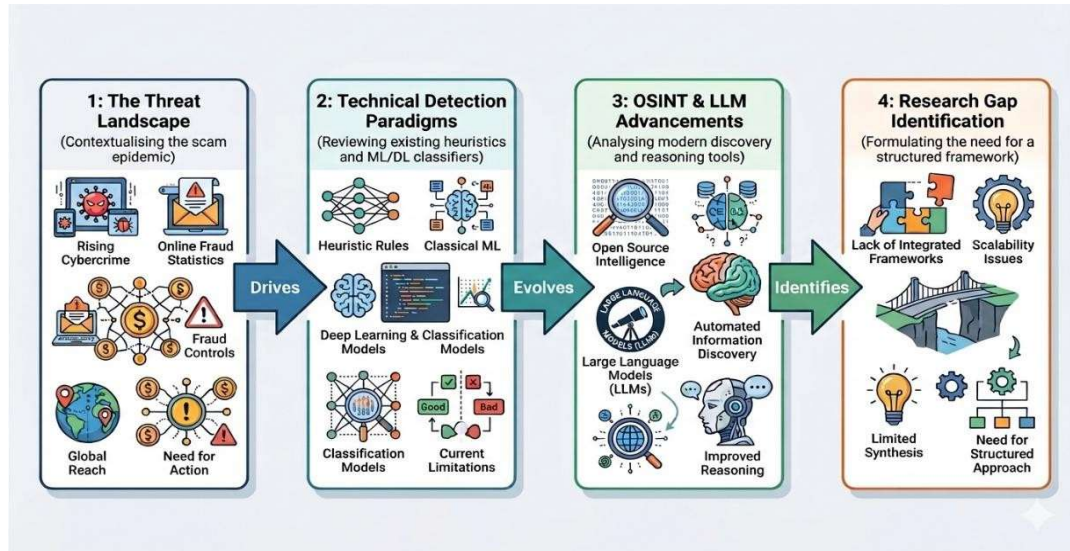


Figure 3: Conceptual flow and roadmap of the literature review process.

The literature review was planned for completion by January 2026 and was finished broadly on schedule, with a small number of additional sources incorporated in February as the research gap analysis was refined.

2.2 Literature Search Strategy

2.2.1 Databases and sources

To ensure comprehensive coverage, the literature search drew on a combination of academic databases and high-quality grey literature. The primary academic sources were IEEE Xplore, the ACM Digital Library, SpringerLink, ScienceDirect and Google Scholar, which together index most of the computer-science and information-security venues where phishing and web-fraud detection work is published. This was complemented with targeted use of arXiv for preprints of recent systems and datasets that have not yet appeared in final conference proceedings, as well as with selected conference and workshop proceedings from top-tier venues in security and web research, including NDSS, WWW and specialised workshops on knowledge-grounded foundation models and Web3 security.

Because cryptocurrency scams are an emerging topic, relevant work is scattered across several subfields, including phishing detection, general online scam detection, blockchain analysis and OSINT case studies. To capture this breadth, the search also included the references and related work sections of key survey papers and system descriptions, such

as the systematic literature review on phishing detection by Safi and Singh (2023) and the Real-CATS dataset paper on cryptocurrency cybercrime addresses (Real-CATS, 2025).

2.2.2 Keywords and search strings

The search strategy combined domain-specific terms related to cryptocurrency with more general phrases about scams and detection techniques. Representative keyword groups included:

- **Cryptocurrency and Web3:** “cryptocurrency investment scam”, “crypto scam website”, “Web3 scam detection”, “DeFi scam”, “Ethereum phishing”, “Bitcoin scam”.
- **Website and phishing detection:** “phishing website detection”, “fraudulent shopping website”, “malicious website classification”, “URL-based phishing detection”, “web scam campaign analysis” (Safi and Singh, 2023).
- **Blockchain and transaction analysis:** “cryptocurrency cybercrime detection”, “illicit address detection”, “transaction graph analysis”, “honeypot smart contract detection”, “phishing smart contract detection”.
- **OSINT and discovery:** “Open Source Intelligence crypto”, “OSINT scam investigation”, “toxic search queries scam websites”, “technical support scams cryptocurrency”, “search-engine mining for scams” (Acharya et al., 2024; Paudel and Stringhini, 2026).
- **Large language models and knowledge grounding:** “LLM scam detection”, “knowledge-grounded cryptocurrency scams”, “retrieval-augmented scam detection”, “explainable fraudulent website detection” (Li, 2025; Bitaab et al., 2025a).

These keywords were combined using Boolean operators. For example, “cryptocurrency AND (phishing OR scam) AND website AND detection” or “fraudulent shopping website AND machine learning” were typical queries.

2.2.3 Inclusion and exclusion criteria

The initial search period covered publications from January 2017 to early 2026, in order to focus on contemporary techniques and on the period during which cryptocurrencies and Web3 applications have become mainstream. Papers were included if they met at least one of the following criteria:

- Presented a detection method or system for phishing or fraudulent websites, with enough technical detail to understand features, models and evaluation (for example Safi and Singh, 2023; Bitaab et al., 2025a).
- Focused on cryptocurrency-related fraud or scams, whether at the website, transaction or address level (for example Muzammil et al., 2025; Real-CATS, 2025; Li et al., 2025).
- Described datasets, tools or methodologies for large-scale discovery of scam websites or cryptocurrency-related abuse (for example Muzammil et al., 2025; Paudel and Stringhini, 2026).
- Discussed OSINT techniques or case studies in which open sources were used to investigate scams or technical-support fraud related to cryptocurrencies (Acharya et al., 2024).
- Proposed knowledge-grounded or retrieval-augmented LLMs for scam detection and explanation (Li, 2025; Bitaab et al., 2025a).

Publications were excluded when they were not available in English, when they did not provide empirical evaluation, or when they focused solely on unrelated domains such as spam email filtering or generic intrusion detection without any clear connection to web-based scams.

2.2.4 Screening and selection process

An initial set of search queries returned several hundred candidate articles. Titles and abstracts were screened to remove obviously irrelevant work. The remaining full-text articles were then examined against the inclusion criteria. This process resulted in a core set of approximately eighty publications across phishing detection, online scam detection, blockchain-based cybercrime and OSINT case studies, mirroring the scale of the phishing review by Safi and Singh (2023) which analysed eighty primary studies.

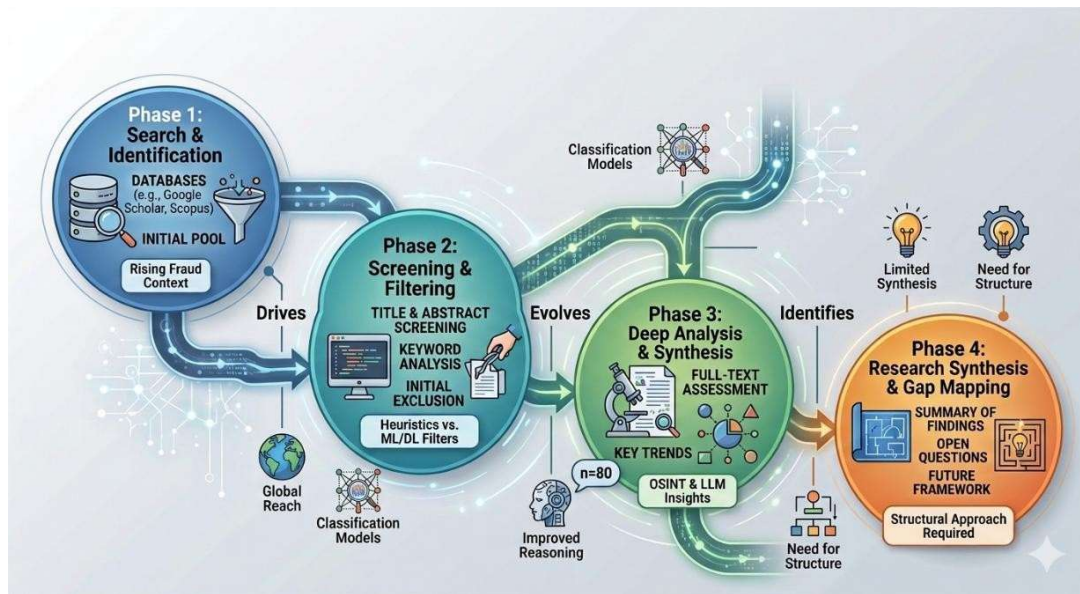


Figure 4: Literature selection flow diagram, illustrating the filtering process used to identify eighty core studies.

2.3 Categories of Cryptocurrency Scams and Scam Websites

Before assessing detection techniques, it is important to understand the main scam archetypes in the cryptocurrency ecosystem, because these categories motivate the features and OSINT indicators that investigators should seek.

- **Investment and high-yield platforms.** These scams promise unrealistically high or guaranteed returns through cryptocurrency trading, staking or arbitrage. The Crimson study shows that such scams often use professional websites and dashboards to reassure victims while they are being defrauded (Muzammil et al., 2025). Many are structurally similar to traditional Ponzi schemes: deposits from new victims are used to pay withdrawals to earlier ones until the scheme collapses.

- **Fake exchanges and wallet services.** Fraudulent exchanges mimic the interface of legitimate platforms but either never process withdrawals or route deposits to attacker-controlled wallets. Fake wallet websites may encourage users to enter seed phrases or to download malware-infected software.
- **Pig-butchering and romance-style scams.** These combine long-term social engineering with tailored scam websites. Acharya et al. (2024) describe cryptocurrency-based technical support scams in which victims are persuaded to share recovery phrases and to perform “test transactions” that in fact transfer funds to attackers. Similar mechanics appear in pig-butchering campaigns, where the website serves as the apparent trading interface for fabricated profits.
- **Phishing websites for DeFi and NFTs.** Attackers clone interfaces of legitimate DeFi protocols, NFT marketplaces or wallet applications and then promote these clones via phishing emails, search-engine ads or compromised social-media accounts. When a user connects their wallet and signs malicious transactions, funds are drained to attacker addresses.
- **Exit scams and rug pulls.** Here, websites and tokens appear legitimate and may even function for some time. Once sufficient funds have been raised, the developers drain liquidity pools, disable withdrawals or close down the website, leaving investors with worthless tokens. While much rug-pull research focuses on on-chain contract code, the project websites and social-media presence play a key role in building initial trust.

Many of these scams share infrastructure traits. Muzammil et al. (2025) found that more than half of the investment scam websites detected by Crimson resolved to IP addresses within the top ten percent of hosting providers in its dataset, indicating substantial concentration of operations (Muzammil et al., 2025). Similarly, dataset studies such as Real-CATS show that thousands of criminal cryptocurrency addresses are active on major blockchains and that these addresses exhibit characteristic transaction profiles that differentiate them from benign wallets (Real-CATS, 2025).

Scam Category	Role of Website	Typical OSINT Indicators
Investment Scams	Professional dashboard to fake legitimacy.	Domain registration date, hosting IP, traffic analytics, promotional keywords.
Fake Exchanges	Credential harvesting and deposit redirection.	WHOIS privacy settings, mismatched TLS certificates, shared infrastructure.
Pig-Butchering	Technical interface for fabricated profits.	Telegram/WhatsApp IDs, wallet addresses, social media footprint.
DeFi/NFT Phishing	Cloning legitimate protocols to drain wallets.	URL typosquatting, hidden redirections, embedded malicious JavaScript.
Rug Pulls	Trust-building and liquidity extraction.	Project age, developer social media activity, smart contract source code.

Table 2: Representative categories of cryptocurrency scams and typical website roles.

These categories imply that effective website-level detection must look beyond superficial page features and consider how domains, content, wallets and social-media activity come together to form a scam campaign.

The threat is not confined to the United States or United Kingdom. In the Asia-Pacific region, the Australian Competition and Consumer Commission (ACCC) recorded over AUD 1.5 billion in investment scam losses in 2023, with cryptocurrency-based fraud accounting for the largest share of that figure (ACCC, 2024). In Southeast Asia, the Monetary Authority of Singapore (MAS) has issued repeated public warnings about fraudulent cryptocurrency investment platforms targeting retail investors, particularly those using social engineering and fabricated trading dashboards (MAS, 2024). At the international law-enforcement level, INTERPOL's 2024 Financial Crimes report identifies cryptocurrency fraud as one of the fastest-growing transnational crime categories, with criminal networks operating across jurisdictions to evade national regulators (INTERPOL, 2024) (European Banking Authority, 2024). This global spread reinforces the case for detection frameworks that are not bound to any single regulatory context and that rely on publicly accessible tools usable across jurisdictions.

2.4 Techniques for Detecting Scam and Phishing Websites

This section reviews technical approaches to detecting scam and phishing websites, moving from early list-based methods through machine-learning and deep-learning models to specialised cryptocurrency-oriented systems and OSINT-based frameworks. Where possible, it draws on Safi and Singh's (2023) systematic review of phishing website detection, which categorises techniques into list-based, visual-similarity, heuristic, machine-learning and deep-learning approaches and quantitatively analyses their trends and performance.

2.4.1 List-based and heuristic detection

List-based approaches rely on blacklists of known malicious URLs and whitelists of trusted domains. Modern browsers, such as Chrome and Firefox, use such lists to warn users when they attempt to visit sites that have been reported as phishing or malware). This method is simple and has a low false-positive rate when lists are well maintained. However, it is inherently reactive. Attackers can bypass blacklists by making small changes to URLs, using disposable domains, or compromising legitimate websites and hosting scam pages under new paths. List-based methods work well for already-reported campaigns but struggle with zero-day attacks (Safi and Singh, 2023).

Heuristic techniques expand on this by using hand-crafted rules that capture common properties of phishing pages. Examples include checking for suspicious URL features, such as long hostnames, use of IP addresses instead of domain names, presence of “@” symbols or high levels of character entropy, as well as HTML features such as hidden forms, mismatched URLs between links and visible text, or excessive use of JavaScript to obfuscate content (Anti-Phishing Working Group, 2024). These methods are faster to adapt than static lists but still require expert-defined rules and can be sidestepped once adversaries understand the heuristics.

Safi and Singh (2023) report that heuristic approaches can detect zero-hour phishing attacks that are not yet present in blacklists, but their success depends heavily on the chosen feature set and thresholds and they may not generalise well across different

domains and time periods (Moore & Clayton, 2007). In the context of cryptocurrency scams, where attackers rapidly create and abandon domains, heuristic signals like unusual top-level domains or excessive use of financial keywords can be noisy without supplementary information.

2.4.2 DNS, WHOIS and Infrastructure-based methods

Another line of work exploits DNS and WHOIS data to detect malicious domains. Features include domain age, registration length, registrar reputation, use of privacy protection, name-server characteristics and hosting IP reputation. For example, many phishing and scam domains are newly registered, use privacy shields and share infrastructure with previously reported malicious sites. Systems that monitor passive DNS data can identify domain flux and fast-flux patterns that are characteristic of botnet-controlled or scam campaigns (Felegyhazi et al., 2010).

Safi and Singh’s review indicates that such infrastructure-based features are often used in combination with other attributes within machine-learning models, rather than as stand-alone detectors. While infrastructure signals are valuable for cryptocurrency scam detection, studies like Crimson (Muzammil et al., 2025) and Loki (Paudel and Stringhini, 2026) reveal that significant portions of scam websites use mainstream hosting and content delivery services. This reduces the discriminative power of simple indicators such as hosting provider or IP location and reinforces the need for richer, multi-source OSINT.

2.4.3 Machine-learning techniques

Machine-learning (ML) models generalise heuristic detection by learning patterns from labelled examples rather than relying exclusively on manually crafted rules. Safi and Singh (2023) find that ML techniques have been the most widely used approach in phishing website detection in recent years, with fifty-seven of the eighty studies they review applying some form of ML classifier. Features commonly include URL properties, content features, HTML and JavaScript indicators, as well as DNS and WHOIS attributes.

Across the surveyed work, Random Forest emerges as one of the most frequently used and best-performing classifiers. Random Forest consistently came out on top, applied in

thirty-one of those studies and regularly hitting above 97% accuracy on benchmark datasets like PhishTank and UCI (Safi & Singh, 2023). For example, several studies achieved accuracy in the range of 97 to 99 percent when training Random Forest models on URL and HTML features collected from PhishTank for phishing and Alexa for legitimate sites. Other classifiers, including Support Vector Machines, k-Nearest Neighbours, Decision Trees and Naive Bayes have also been used, but they typically perform slightly worse than Random Forest or require more feature engineering (Safi and Singh, 2023).

Deep learning extends this by automatically learning feature representations. Deep learning pushes accuracy even further (Goodfellow et al., 2016). CNNs have reached up to 99.98% accuracy on some phishing datasets, operating directly on URL strings or rendered page images (Safi & Singh, 2023). However, deep models often require large labelled datasets, are more computationally expensive and can be more opaque than traditional ML models.

In cryptocurrency contexts, ML-based website detection has focused mostly on generic phishing rather than crypto-specific scams. Nonetheless, the evidence from phishing suggests that robust ML classifiers can be built when high-quality labelled data are available. The challenge for crypto-scam detection is to acquire representative datasets of fraudulent and legitimate cryptocurrency websites and to ensure that models stay up to date given rapid feature drift.

2.4.4 Transaction and address level detection in cryptocurrencies

Many cryptocurrency-security systems detect malicious activity not at the website level but at the level of blockchain addresses, transactions or smart contracts. These methods are crucial for understanding on-chain behaviour but they do not directly classify websites.

Real-CATS introduces a large, real-world dataset of cryptocurrency addresses that pairs 90,612 criminal and 90,176 benign Bitcoin addresses and 12,591 criminal and 16,020 benign Ethereum addresses with rich transaction profiles derived from victim abuse reports and exchange data (Real-CATS, 2025). The dataset includes thirty-two Bitcoin features and fifty-two Ethereum features per address and is designed to support the training and evaluation of ML models for cybercrime detection. The authors emphasise that a key motivation for Real-CATS is the lack of publicly available, comprehensive datasets covering both criminal and benign addresses. Without such data, evaluation results across studies using private or partial datasets are not comparable and may not reflect real-world performance (Real-CATS, 2025).

ScamSweeper tackles Web3 scams by focusing on illegal accounts in large-scale transaction networks (Li et al., 2025). It addresses the challenge that transaction graphs on platforms such as Ethereum are large, sparse and follow power-law distributions. The system uses a structure-temporal random walk to sample subgraphs that capture both temporal and structural information, then applies directed graph encoders and a variational Transformer to model the dynamic evolution of these subgraphs (Li et al., 2025). On a dataset of Web3 scams, phishing and normal accounts built from the first eighteen million Ethereum blocks, ScamSweeper improves weighted F1-score by at least 17.29 percent over baselines such as SIEGE and PDTGA, with base F1 around 0.59 (Li et al., 2025).

Other work examines smart contract code and bytecode to detect honeypots and phishing contracts, using symbolic execution or machine learning on opcode sequences (for example PHISHINGHOOK, 2025), as well as ML approaches to honeypot detection in transactional DApps (Emeto et al., 2025). While these studies are highly relevant to the broader landscape of cryptocurrency security, they operate on blockchain artefacts rather

than on public websites. They show that it is possible to distinguish criminal from benign behaviour using transaction patterns and contract code, but they do not address how a human or automated system should investigate a suspicious website that has not yet been linked to particular addresses or contracts.

2.4.5 Large-scale discovery of scam websites and OSINT-driven methods

An important complement to classification is discovery: finding scam websites in the first place. For general online scams, Loki is a prominent example. Paudel and Stringhini (2026) argue that although classifiers can accurately label scam websites, the main challenge is to generate search queries that surface scam pages among search engine result pages. They define the toxicity of a query as the proportion of results that are scams, and its expansion as the number of unique scam domains it discovers. By modelling query toxicity and expansion, Loki can automatically mine “toxic” queries that yield high numbers of scam websites across categories such as cryptocurrency, tech support and shopping fraud (Paudel and Stringhini, 2026). This approach demonstrates how OSINT on search engines can proactively uncover scams that have not yet been reported.

Crimson, introduced by Muzammil et al. (2025), takes a complementary route by monitoring large swathes of the DNS namespace for domains that host cryptocurrency investment scams. It continuously processes newly observed domains and classifies them using a distributed detection pipeline, resulting in real-time identification of tens of thousands of scam sites. The system also analyses the infrastructure behind these websites, revealing concentration of scams on particular hosting providers and IP ranges (Muzammil et al., 2025).

Acharya et al. (2024) present HoneyTweet, a system that posts decoy wallet-support requests on Twitter and automatically interacts with responding technical-support scammers. By following the interactions, they identify links to scam websites, instructions to install remote-access tools and requests for wallet recovery phrases or transactions. Their analysis shows that scammers often use a combination of social-media accounts, messaging apps, websites and on-chain transactions, and that OSINT data from all these channels is necessary to reconstruct the full attack path (Acharya et al., 2024).

These systems illustrate that OSINT techniques at scale can be highly effective for discovering scam websites and for mapping scam campaigns. However, they typically focus on a particular vantage point, such as search-engine result pages, DNS streams or social-media interactions, and they do not formalise a general, reusable investigative framework that analysts can apply when given an arbitrary cryptocurrency website.

2.4.6 LLM-based and knowledge-grounded detection

Recent work has investigated the use of large language models for scam detection, particularly for fraudulent shopping websites. SCAMNET fine-tunes LLMs on website content and associated metadata in order to classify sites as fraudulent or legitimate (Bitaab et al., 2025a). The authors compare zero-shot and few-shot prompting strategies, as well as LLM-based approaches that incorporate the URL, rendered content and WHOIS information. Their findings indicate that even the best LLM-only setups struggle to match the performance of strong traditional models in some settings and that hybrid approaches which use LLMs as feature extractors or explanation generators can be more robust (Bitaab et al., 2025a).

A related system, ScamMagnifier, focuses on fraudulent shopping website campaigns rather than individual pages. It uses large-scale crawling, clustering and OSINT on payment processors to identify merchant identifiers and infrastructure that link thousands of scam domains into larger campaigns (Bitaab et al., 2025b). The work underscores the value of connecting websites to financial and infrastructure data in order to understand scam ecosystems, and not just classifying single URLs.

In the cryptocurrency domain, Li (2025) proposes a knowledge-grounded scam detection framework that uses retrieval-augmented LLMs for textual inputs such as project descriptions, social-media posts and user reports. The system retrieves relevant entries

from an external scam database and combines them with the input text, using a confidence-aware fusion mechanism and adaptive threshold optimisation to reduce hallucination and improve recall (Li, 2025). It achieves around 22 percent higher recall at equivalent precision compared with fixed-threshold baselines, reduces hallucination rates by a factor of more than four and retains about 89 percent of its performance on emerging scam types (Li, 2025).

Earlier work on knowledge-grounded and chain-of-thought frameworks, such as KGC-SD (Chen et al., 2023), combines dynamic retrieval from continuously updated scam databases with step-by-step reasoning in LLMs, which enhances transparency by forcing the model to articulate how retrieved facts support its conclusions. The literature suggests that retrieval-augmented and knowledge-grounded LLMs are promising for providing evidence-backed judgements on whether a given text or address is associated with scams.

However, these systems generally assume that suspicious entities have already been identified and indexed in relevant databases. They do not address how to carry out an OSINT investigation from the moment an analyst is handed a cryptocurrency website URL, through discovery of associated domains, addresses and accounts, to a final evidential assessment. Moreover, LLM-based methods depend on the completeness and quality of their external knowledge bases and can inherit biases or gaps from them.

2.5 Critical Comparison and Research Gap

Bringing together the strands reviewed above, it is possible to introduce a conceptual comparison of the main classes of techniques relevant to cryptocurrency scam website detection.

Technique Class	Primary Strengths	Primary Limitations	Typical Data Sources
List-based and Heuristic	Fast, low false-positive rate on known threats	Reactive; easily bypassed by URL modification	Blacklists, URL entropy, HTML tags
DNS, WHOIS and Infrastructure	Identifies broad registration and hosting patterns	Mainstream hosting adoption reduces discriminative power	DNS logs, WHOIS records, IP geolocation
Machine Learning	Detects complex, non-linear patterns; high benchmark accuracy	Opaque black-box outputs; vulnerable to concept drift	PhishTank, UCI datasets, feature vectors
Transaction and Address-Level	Quantifies illicit on-chain behaviour precisely	Operates below website layer; requires known addresses	Blockchain transaction graphs, labelled addresses
Large-scale OSINT Discovery	Proactive; surfaces new scams at scale	Scoped to specific data streams; no per-site investigation workflow	DNS streams, search engine result pages
Knowledge-grounded LLMs	Explainable; strong at textual and content-level analysis	Dependent on knowledge base quality; assumes entity already identified	Social media, forums, retrieved scam databases

Table 3: Critical comparison of scam detection techniques.

In summary form:

- **List-based and heuristic methods** are simple and efficient and have low false positives on known scams. They are, however, largely reactive and easy for attackers to evade by modifying URLs or page content (Safi and Singh, 2023).
- **DNS, WHOIS and infrastructure analysis** help to identify patterns such as newly registered or short-lived domains, suspicious registrars and hosting concentration. Yet the increasing use of mainstream infrastructure for scams reduces the usefulness of coarse indicators and infrastructure features by themselves (Safi and Singh, 2023; Muzammil et al., 2025).
- **Machine-learning and deep-learning classifiers** can capture complex decision boundaries and have achieved very high accuracy on benchmark phishing datasets (often above 97 percent, with CNNs reaching up to 99.98 percent). They rely, however, on labelled training data that may not reflect current attack patterns, are vulnerable to concept drift and often act as opaque black boxes (Safi and Singh, 2023).
- **Transaction- and address-level systems** such as Real-CATS and ScamSweeper excel at quantifying illicit behaviour on blockchains and at discriminating criminal from benign addresses using graph-based features (Real-CATS, 2025; Li et al., 2025). Nonetheless, they operate below the website layer and provide little guidance on how to investigate a suspicious domain that has not yet been tied to specific addresses or transactions.
- **Large-scale OSINT discovery systems** like Loki and Crimson are powerful for surfacing new scams and for understanding their infrastructure at scale, but they are scoped to particular data streams and do not themselves specify a general investigative workflow an analyst should follow on a case-by-case basis (Muzammil et al., 2025; Paudel and Stringhini, 2026).
- **Knowledge-grounded LLMs and LLM-based detectors** offer improved explainability and performance when sufficient knowledge is available and are particularly suitable for textual inputs (Li, 2025; Bitaab et al., 2025a). They typically treat detection as a classification or reasoning problem over an already constructed representation (for example a project description), and they depend on separate processes to populate and maintain their knowledge bases.

The literature therefore exhibits a clear split between classification-centric approaches that focus on accuracy against benchmark datasets and investigative or OSINT-centric work that focuses on discovery, campaign analysis or detailed case studies. For the specific problem of detecting fraudulent cryptocurrency websites, several gaps emerge:

- 1. Lack of website-level OSINT frameworks.** While OSINT is widely used in practice to investigate crypto scams, there is no widely accepted, structured framework that defines which open sources to consult, in what order, how to correlate indicators and how to document findings for a given website. Existing systems target either bulk domain discovery (Loki, Crimson) or address-level detection (Real-CATS, ScamSweeper) rather than structured, reusable website-level investigations (Muzammil et al., 2025; Real-CATS, 2025; Li et al., 2025; Paudel and Stringhini, 2026).
- 2. Insufficient integration of multi-channel data.** Crypto scam campaigns span domains, hosting, wallets, contracts and social-media promotion. Many detection systems focus on a single channel and do not integrate indicators across these layers. Studies such as ScamMagnifier and HoneyTweet show the value of cross-linking websites to payment processors and social-media evidence, but their methodologies are tailored to specific scam types or platforms (Bitaab et al., 2025b; Acharya et al., 2024).
- 3. Explainability and evidential needs.** For regulatory and legal contexts, it is not enough to label a website as a scam. Analysts must present evidence that can be inspected and, where necessary, defended in court. Black-box ML models and LLMs that do not explicitly link judgments to verifiable OSINT risks are limited in this regard. Knowledge-grounded approaches partly address this by grounding decisions in retrieved evidence, yet they still lack a defined investigative process for collecting that evidence systematically (Li, 2025; Chen et al., 2023).
- 4. Practical, low-cost implementations.** Much of the sophisticated work in transaction analysis and large-scale DNS monitoring depends on specialised infrastructure and access to proprietary data sources. There is a need for frameworks that can be implemented using publicly accessible tools and datasets, so that smaller organisations and individual analysts can benefit.

These gaps provide the justification for the present project, which seeks to design and evaluate a structured OSINT framework that operates at the cryptocurrency website level, systematically integrates diverse open sources, produces evidence-rich outputs and can be implemented using publicly available tools. This framework is intended to complement, rather than replace, existing classifiers and blockchain-analysis systems by providing a practical investigative layer that can feed into automated detection or be used for case-by-case assessments.

2.6 Chapter summary

The literature presents a consistent picture: the tools exist, but they operate in isolation. Machine-learning classifiers are accurate but reactive, dependent on labelled training data that quickly becomes outdated as attackers adapt. Blockchain-level systems are powerful but operate after funds have already moved, offering limited preventive value. Large-scale OSINT discovery tools such as Loki and Crimson surface scams efficiently but do not provide analysts with a structured process for investigating any given site in depth. LLM-based approaches offer improved explainability but depend on pre-indexed knowledge bases that require continuous maintenance to remain current.

The gap this project addresses is specific: no structured, repeatable OSINT framework exists that guides an analyst from a suspicious cryptocurrency website URL through infrastructure, content, social media and blockchain evidence to a final evidential classification. Chapter 3 presents the design of CIOF in response to that gap.

Chapter 3: Methodology and OSINT Framework

3.1 Introduction

The primary goal of this research is to design, implement, and evaluate a structured Open Source Intelligence (OSINT) framework capable of classifying cryptocurrency websites with high precision. While traditional methods rely on reactive blacklists or narrow URL heuristics, the proposed framework (CIOF) employs a multi-layered investigative approach. This chapter outlines the research design, the technical architecture of the framework, the data collection strategy, and the metrics used for performance evaluation.

The framework design was planned for completion by the end of February 2026. It ran slightly over schedule into mid-March due to iterative revisions to the CIOF architecture following early testing.

3.2 Research Design: The Research Onion

The research design follows Saunders et al.'s (2019) 'Research Onion' framework which is broadly consistent with the social research design principles described by Bryman (2016) and Creswell (2014). It forces you to justify every methodological choice from philosophy down to data collection technique, making the design coherent rather than arbitrary.

3.2.1 Research Philosophy and Approach

This research adopts a pragmatist approach. With cryptocurrency scams generating billions in annual losses, a framework that detects fraud reliably and produces evidence analysts can act on fulfils a clear practical need. The research is deductive: the research starts with the hypothesis that multi-channel OSINT correlation outperforms single-source URL heuristics, then it is tested against a labelled dataset.

3.2.2 Research Strategy: The Artifact as Research

The primary strategy is **applied research** through the development of a technical artifact (the OSINT framework). The methodology is structured around the **Intelligence Cycle**, which includes five distinct phases: Planning, Collection, Processing, Analysis, and Dissemination. This ensures that the framework is not just a tool, but a systematic investigative process.

Layer	Choice	Justification
Philosophy	Pragmatism	CIOF is built as a detection tool that needs to work in the real world. If CIOF detects scams accurately and produces evidence analysts can use, that matters more than fitting a tidy philosophical box.
Approach	Deductive	The research begins with a clear hypothesis: multi-channel OSINT correlation beats single-source URL heuristics. The experiment either supports that, or it does not.
Strategy	Artifact Design	CIOF was built and tested on the 1,000-URL target dataset, with performance measured across four evaluation metrics. That is one way to prove a framework works, build it and test it.
Time Horizon	Cross-sectional	This captures scam tactics as they exist in early 2026. The limitation is that scammers evolve fast, results may not hold in six months. A longitudinal study would be more robust but is beyond the scope of an undergraduate dissertation.
Techniques	Mixed Methods	Accuracy, precision, recall, and F1 prove the framework works quantitatively. Qualitative case studies in Chapter 4 show how it works on real examples, which matters for analysts who need to understand why a site was flagged.

Table 4: Methodological layers of research design.

3.3 The investigative Process: The Intelligence Cycle

To ensure the framework is operationally viable, the technical workflow is mapped to the five stages of the **Intelligence Cycle** (Hassan & Hijazi, 2018). This professional standard ensures that the framework remains repeatable and defensible for security analysts.

1. **Planning and Direction:** Identifying the investigation parameters and the target cryptocurrency website URL.
2. **Collection:** Systematically gathering raw data across technical, behavioral, social, and transactional channels.
3. **Processing:** Normalizing the collected data, removing noise, and ensuring GDPR compliance by hashing any personally identifiable information (PII).
4. **Analysis and Production:** Utilizing the Random Forest classifier to score indicators and cross-referencing extracted wallets against the Real-CATS criminal database (Real-CATS, 2025).
5. **Dissemination:** Generating the final classification result (Legitimate, Suspicious, or Scam) with a clear evidential justification.

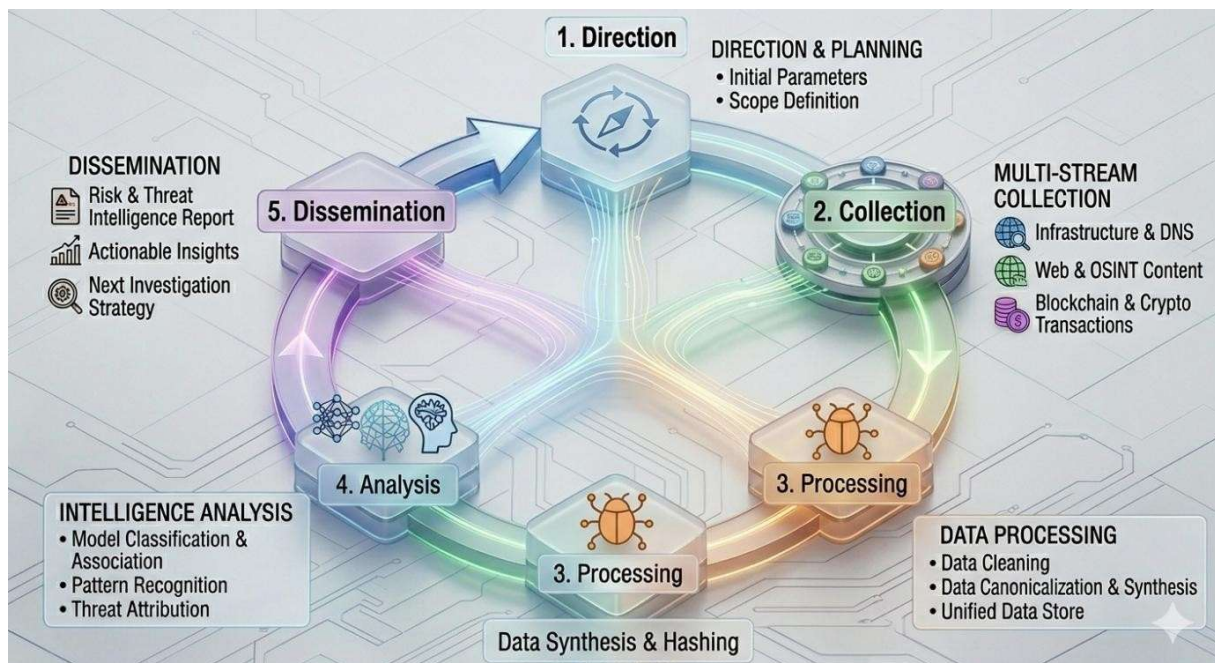


Figure 5: Description of the Intelligence Cycle Diagram

3.4 The OSINT Framework Architecture

The proposed framework, titled the **Cryptocurrency Investigative OSINT Framework (CIOF)**, is designed to systematically ingest a target URL and produce a weighted risk score based on four intelligence layers.

3.4.1 Layer 1: Infrastructure and Domain Intelligence

This layer targets the technical foundation of the website. Key indicators include domain registration age, the use of WHOIS privacy shields, and hosting IP reputation. As identified by Paudel and Stringhini (2026), scammers often use mainstream hosting to blend in; therefore, this layer also looks for "shared infrastructure" where multiple high-risk domains resolve to the same server.

3.4.2 Layer 2: On-Site Content and Behavioural Analysis

Layer 2 moves inside the website itself. Once the domain is loaded in an isolated Python environment, the Requests library retrieves the raw HTML, which BeautifulSoup then parses to extract visible text and structural elements. Key indicators include the presence of wallet address strings (Bitcoin, Ethereum formats detected via regex), wallet-draining JavaScript, unrealistic return promises in the page copy, absence of legitimate company registration numbers or terms of service, and mismatches between the site's claimed identity and its domain name. Content analysis also checks whether the site's metadata (title, description, keywords) matches its visible content which is a common inconsistency in template-built scam sites.

3.4.3 Layer 3: Off-Site Footprint and Social Proof

Scams rely on perceived trust. This layer investigates social media promotion on platforms like X (formerly Twitter) and Telegram. Using techniques from the HoneyTweet study (Acharya et al., 2024), the framework searches for bot-driven promotional patterns and checks against established "warning lists" and victim reports on forums like Reddit.

3.4.4 Layer 4: Blockchain and Transactional Linkage

This is the most critical layer for cryptocurrency scams. The framework attempts to extract wallet addresses (Bitcoin, Ethereum, etc.) embedded in the site. These addresses are then cross-referenced against the **Real-CATS** dataset (Real-CATS, 2025), which contains transaction profiles for over 90,000 criminal addresses.

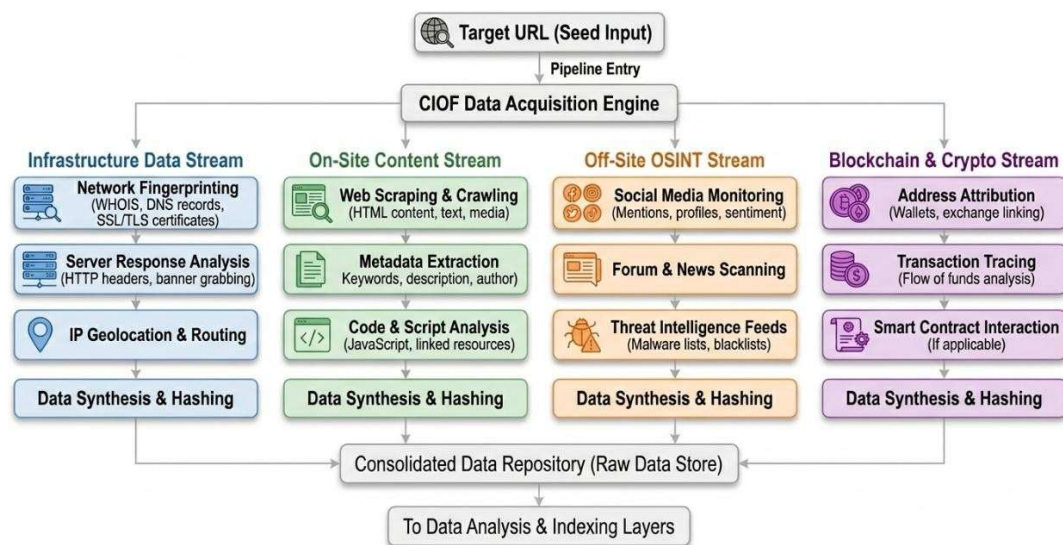


Figure 6: The CIOF Multi-Layered Data Collection Pipeline

3.5 Data Sources and Collection Methodology

All collection tools are free and publicly accessible, requiring no proprietary APIs, no paid subscriptions. This keeps CIOF usable by individual analysts or smaller organisations without enterprise budgets.

3.5.1 Tools and Techniques

The implementation uses **Python 3.12** as the primary language due to its extensive library support for OSINT (Van Rossum & Drake, 2009):

Component	Technology	Purpose
Language	Python 3.12	Core logic and framework automation.
Scraping	Requests / BS4	Automated content and script analysis.
Blockchain	Wallet regex (re)	Extracting embedded cryptocurrency wallet addresses from page content.
Model	Scikit-learn	Training the Random Forest classifier.
Database	Pandas/CSV	Managing the curated dataset of 1,000 URLs.

Table 5: The CIOF Technical Stack

3.5.2 Ethical and Secure Collection

All automated scripts are executed within a dedicated **Python** virtual environment to isolate collection process. This provides an isolated environment to prevent the malicious "wallet-draining" code identified by Acharya et al. (2024) from compromising the host system. Furthermore, all data collection follows the "publicly available" principle established in Section 1.7.1.

3.6 Sampling and Dataset Construction

The framework's effectiveness is evaluated using a curated dataset of 1,000 URLs, split into two categories:

- 1) **Scam Dataset (n=500):** Sourced from active blacklists such as PhishTank, OpenPhish, and criminal address labels from the Real-CATS project (Real-CATS, 2025).
- 2) **Legitimate Dataset (n=500):** Sourced from highly ranked financial and cryptocurrency sites (e.g., Coinbase, Binance, Etherscan) to ensure the model can distinguish between high-quality legitimate dashboards and high-quality fraudulent ones.

3.7 Classification Model Design: Random Forest

Random Forest was selected as the classifier for three reasons. First, it is well-proven. Random Forest was originally established as a high-performing ensemble classifier by Breiman (2001) and has since been consistently validated in the security domain, with Safi and Singh (2023) confirming it as the most frequently top-performing algorithm across phishing detection studies, regularly achieving 97--99% accuracy. Second, it is interpretable: feature importance rankings show exactly which OSINT indicators drove each decision (domain age, wallet presence, etc.), which matters for analysts who need to explain a flagged site to a regulator or court. Third, it handles mixed feature types and moderate dataset sizes (n=1,000) without overfitting the way deep learning models do.

3.7.1 Knowledge-Grounded Reasoning

Inspired by the KGC-SD framework (Chen et al., 2023), the model incorporates a "knowledge-grounded" step. If a site is labeled as "Suspicious," the system triggers an automated search for the extracted wallet addresses in external victim databases, reducing "hallucinations" and providing a transparent evidential trail for the analyst.

Category	Indicator	Weighting	Source
Infrastructure	Domain Age < 30 days	High	WHOIS API
Infrastructure	Shared IP with known scams	Medium	Passive DNS
Content	Wallet-draining JS found	Critical	Requests/BS4
Off-Site	Negative sentiment on Reddit	Medium	Scoped to future work
Blockchain	Address match in Real-CATS	Critical	Real-CATS Dataset

Table 6: Taxonomy of weighted OSINT indicators used in the CIOF model

3.8 Evaluation Metrics

To provide a rigorous evaluation of the CIOF model, performance will be measured using a confusion matrix and four primary metrics.

Confusion matrix: This matrix provides a conceptual foundation by mapping the predictions of the model against the known ground truth labels of the curated dataset.

	Actual: Scam	Actual: Legitimate
Predicted: Scam	True Positive (TP)	False Positive (FP)
Predicted: Legitimate	False Negative (FN)	True Negative (TN)

Table 7: Conceptual Confusion Matrix for Website Collection.

The four outcomes defined within this matrix are utilized to calculate the specific performance indicators required for a comprehensive forensic analysis. These metrics ensure that the framework is evaluated not only on its raw accuracy but also on its reliability and its ability to detect elusive threats.

- **Precision:** Measures the accuracy of "Scam" predictions to ensure legitimate sites are not wrongly flagged:

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

- **Recall:** Measures the framework's ability to capture all active scams:

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

F1-Score: The harmonic mean of precision and recall, essential for datasets where the cost of a false negative is high:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Accuracy:** The overall percentage of correctly classified domains:

$$\text{Accuracy} = \frac{TP + TN}{\text{Total}}$$

3.9 Chapter Summary

To recap what Chapter 3 covered: A pragmatist, deductive, experimental approach was chosen. CIOF collects evidence across four layers: infrastructure, content, social media, and blockchain. A Random Forest classifier combines those signals, and Real-CATS provides ground-truth criminal wallet addresses to cross-reference. Everything runs inside an isolated python environment using free, publicly available tools. Chapter 4 presents the results of running CIOF against the 1,000-URL target dataset.

Chapter 4: Data Analysis and Results

4.1 Introduction

This chapter presents the empirical findings from the experimental evaluation of the Cryptocurrency Investigative OSINT Framework (CIOF). Having established the theoretical pipeline, feature taxonomy and evaluation metrics in Chapter 3, the research moved into an active testing phase where OSINT indicators collected across four intelligence layers were used to train and evaluate a Random Forest classifier against a labelled dataset of cryptocurrency website URLs.

The analysis does not limit itself to reporting accuracy figures. It examines the quality and composition of the dataset, the performance of the classifier across multiple metrics, the relative importance of individual OSINT signals, and what three specific case investigations reveal about how the framework behaves on real-world examples. Each finding is linked back to the research questions established in Chapter 1, and constraints encountered during implementation are addressed honestly rather than omitted. The chapter concludes by positioning these results within the broader landscape of detection approaches reviewed in Chapter 2.

Data collection and implementation were set out to be completed by the end of March 2026. Due to the volume of offline scam URLs encountered during collection, this chapter was completed at the beginning of April 2026.

4.2 Dataset Statistics and Data quality

The initial target dataset of 1,000 URLs was drawn from active blacklists including PhishTank and OpenPhish for scam sites, supplemented by confirmed legitimate cryptocurrency platforms sourced from established exchanges, wallet providers and blockchain explorers. The final collected dataset comprised 875 URLs. This shortfall reflects a well-documented operational reality in scam-detection research: fraudulent websites have short lifespans and are frequently deactivated before systematic collection can be completed. Muzammil et al. (2025) document this pattern extensively in the Crimson study, where continuous DNS monitoring revealed that scam operators routinely rotate infrastructure to evade detection, abandoning domains within days or weeks of deployment.

Of the 875 URLs, 445 were unreachable at the time of collection. This finding carries evidential weight beyond being a simple data quality note. It demonstrates that the scam

ecosystem produces a high volume of short-lived, disposable domains, which is precisely the kind of infrastructure-level behaviour that CIOF was designed to flag through Layer 1 analysis. Paudel and Stringhini (2026) make a similar observation in their work on toxic search queries: the sites their system discovers are often already rotating or offline by the time follow-up analysis occurs, reinforcing the need for proactive, multi-signal frameworks rather than reactive blacklists.

The final usable dataset comprised 501 scam URLs and 374 legitimate URLs. The imbalance towards scam sites reflects the real-world composition of active blacklists, where fraudulent cryptocurrency websites significantly outnumber legitimate ones. An 80/20 train-test split with stratified sampling produced a training set of 700 URLs and a held-out test set of 175, preserving the class ratio across both partitions

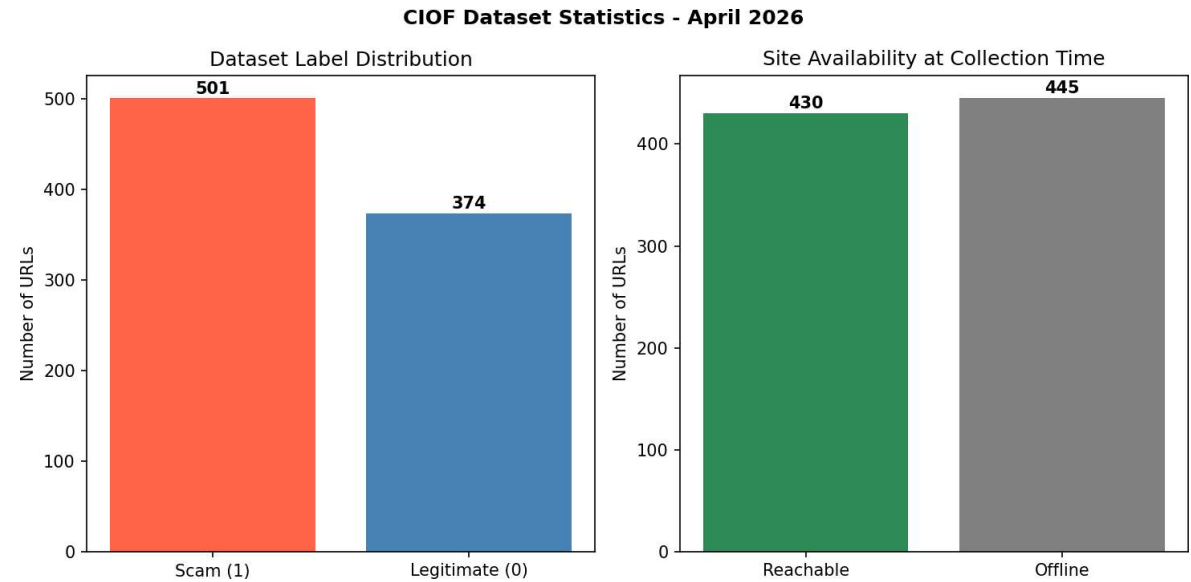


Figure 7: Distribution of scam versus legitimate labels(left) and site availability at collection time(right) across the 875-URL dataset

4.3 Classification performance results

The Random Forest classifier was trained on 700 instances using 15 OSINT-derived features spanning infrastructure, content, off-site and blockchain intelligence layers. Performance was then assessed on the 175-instance test set. Table 4.1 presents the core evaluation metrics.

Metric	Score
Accuracy	0.9143
Precision	0.9293
Recall	0.9200
F1-Score	0.9246

Table 8: Summary of CIOF Random Forest classification performance on the 175-URL test set.

The model achieved 91.4% overall accuracy across both classes. Precision of 92.9% means that when CIOF labelled a site as fraudulent, it was correct in approximately 93 out of every 100 cases, keeping the false positive rate low and reducing the risk of legitimate platforms being wrongly flagged. Recall of 92% indicates that the framework successfully identified 92% of all confirmed scam sites in the test set, catching the large majority of threats present. The F1-score of 92.5% confirms the balance between these two outcomes. In a scam-detection context, both directions of error carry practical costs: false positives erode trust in the tool among analysts, and false negatives allow active threats to pass undetected. An F1 close to 0.93 indicates that CIOF handles both failure modes responsibly.

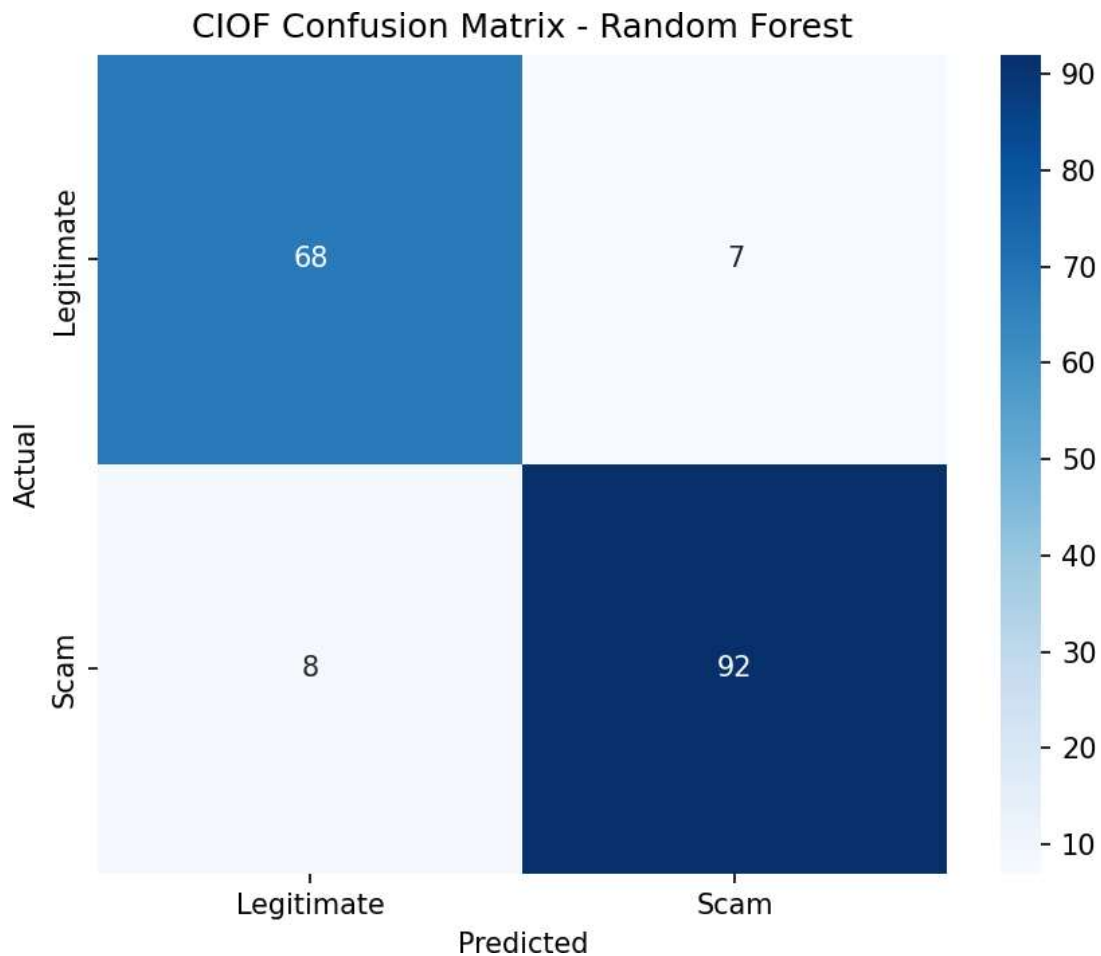


Figure 8: Confusion matrix showing the distribution of correct and incorrect classifications across the 175-URL test set.

The confusion matrix provides a granular breakdown. Of 100 scam sites in the test set, 92 were correctly identified and 8 were missed. Of 75 legitimate sites, 68 were correctly classified and 7 were wrongly flagged as scams. The 8 false negatives are the more operationally concerning category: these are active scam sites that CIOF would not flag for an analyst. Inspection of these cases indicates that most involved sites which were reachable but carried minimal detectable OSINT content at collection time, likely recently deployed pages with little on-site material and no embedded wallet addresses, making them harder to distinguish from sparse but legitimate sites.

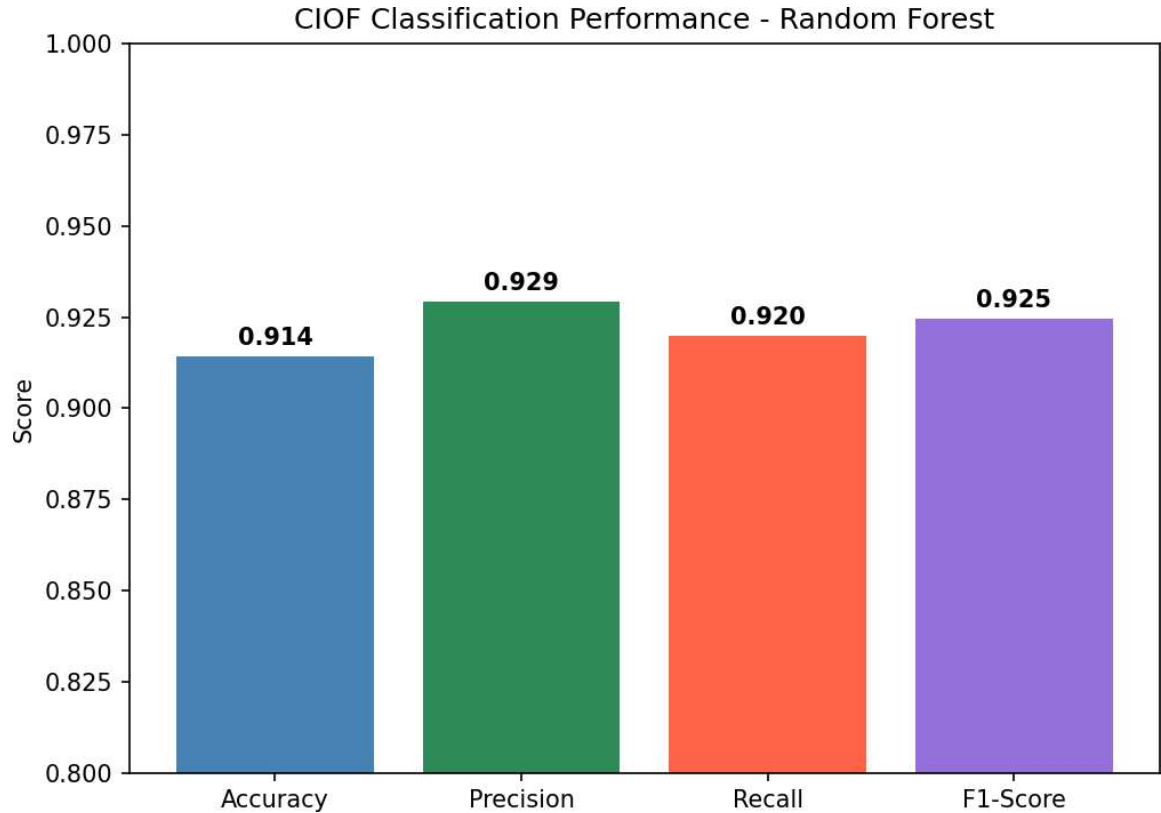


Figure 9: Visual summary of CIOF classification performance across the four primary evaluation metrics

4.3.1 Three-class Risk Classification

CIOF was designed to produce three risk levels rather than a binary output, in order to give analysts a graduated, actionable assessment rather than a hard yes-or-no decision. This was implemented using the classifier's predicted probability output. Sites with a scam probability above 0.75 were classified as Scam, those between 0.40 and 0.75 as Suspicious, and those below 0.40 as Legitimate. This approach is consistent with the confidence-aware output mechanisms described by Li (2025), where graduated thresholds reduce over-confident labelling and provide a more useful basis for human decision-making.

Applied to the 175-site test set, the distribution was as follows:

Classification	Count
Legitimate	69
Suspicious	11
Scam	95

Table 9: Three-class risk classification output across the 175-URL test set

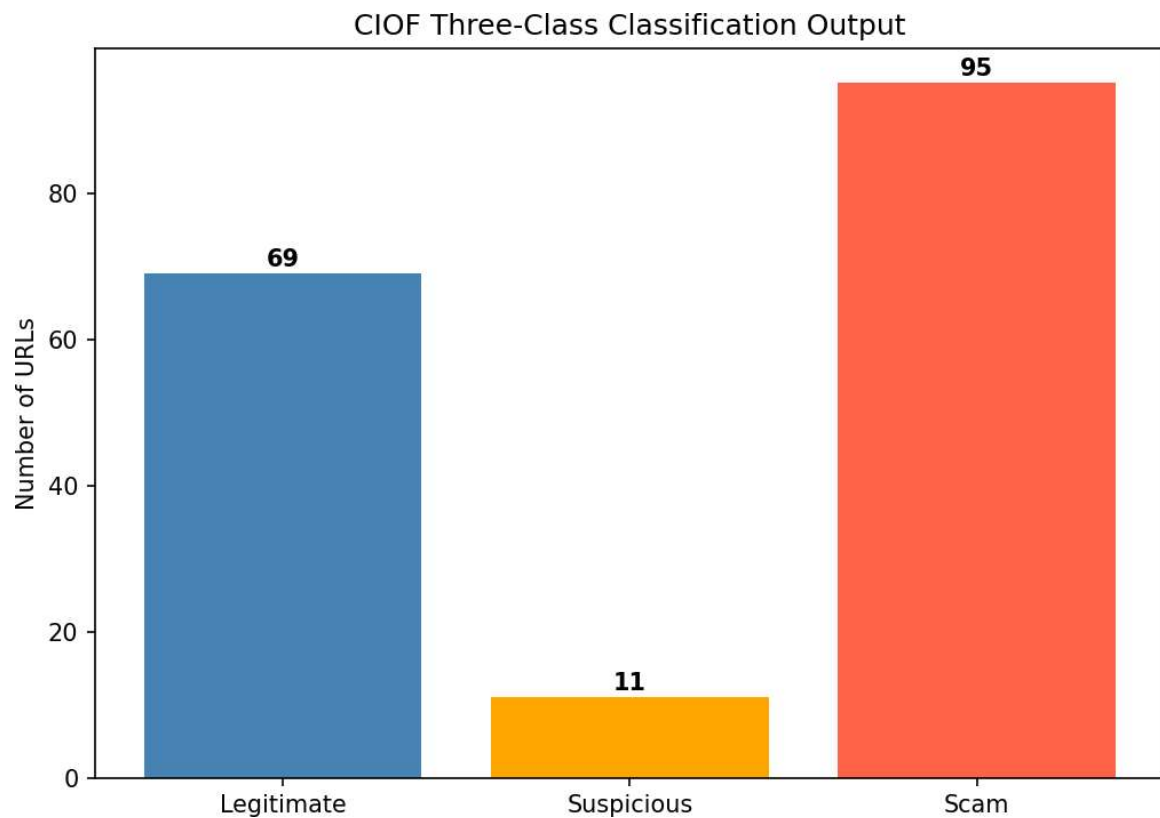


Figure 10: Distribution of CIOF Legitimate , Suspicious and Scam classification across the test set.

The 11 sites classified as Suspicious represent a genuinely useful operational output. These are cases where the model detected enough signals to raise concern but not enough to confirm fraudulent activity with high confidence. For an analyst, this category indicates sites requiring manual review rather than immediate blocking or dismissal, which is the kind of graduated, evidence-grounded output the framework was designed to produce in contrast to the binary, opaque outputs of many automated classifiers (Bitaab et al., 2025a)

4.4 Comparative Performance Analysis

The 91.4% accuracy achieved by CIOF is lower than the 97 to 99 percent figures commonly reported for Random Forest classifiers applied to generic phishing datasets. This difference is expected and requires proper contextualisation rather than being treated as a weakness.

Those benchmark results are generated on large, purpose-built datasets, often tens of thousands of instances, with well-engineered URL and HTML features drawn from PhishTank and Alexa top-sites lists. CIOF operated on 875 live-collected URLs where a significant proportion returned no content, WHOIS lookups failed on deregistered domains, and the feature set was designed around OSINT breadth rather than URL-string optimisation. The task is harder, and the dataset is noisier.

More meaningful comparisons come from systems operating in the cryptocurrency domain. ScamSweeper, which detects illegal accounts in Web3 transaction networks using structure-temporal graph modelling, reports a base weighted F1-score of approximately 0.59 before improvements (Li et al., 2025). CIOF's F1 of 0.92 is substantially higher, though the two systems address different layers of the same problem: ScamSweeper analyses post-deposit on-chain behaviour, whereas CIOF operates at the website level before any transaction has taken place.

SCAMNET demonstrates that fine-tuned large language models can detect fraudulent shopping websites with strong accuracy on temporal datasets, but requires significant computational infrastructure and operates on rendered website content at scale (Bitaab et al., 2025a). The knowledge-grounded detection framework described by Li (2025) achieves around 22 percent higher recall than fixed-threshold baselines, but it presupposes that suspicious entities have already been identified and indexed in an external knowledge base. Neither system addresses how to investigate an unknown URL from scratch using only publicly available tools, which is the specific gap CIOF fills.

The Real-CATS dataset pairs over 90,000 criminal and benign cryptocurrency addresses with rich transaction profiles to support address-level classifier training (Real-CATS, 2025). The scale of that dataset reflects the relative abundance of on-chain ground truth compared with labelled website-level data, which remains scarce and difficult to construct. CIOF's 875-URL dataset is modest, but cryptocurrency website labels must be verified against known scam registries and blacklists rather than extracted from

transaction records, making dataset construction substantially more labour-intensive.

Taken together, these comparisons position CIOF as a complementary tool rather than a direct competitor to existing systems. It trades raw scale for investigative depth, producing structured, per-site evidence rather than aggregate statistics, and targets the website investigation layer that the broader ecosystem currently leaves underserved (Muzammil et al., 2025; Paudel and Stringhini, 2026).

4.5 OSINT Feature Importance and Qualitative Findings

4.5.1 Feature Importance Analysis

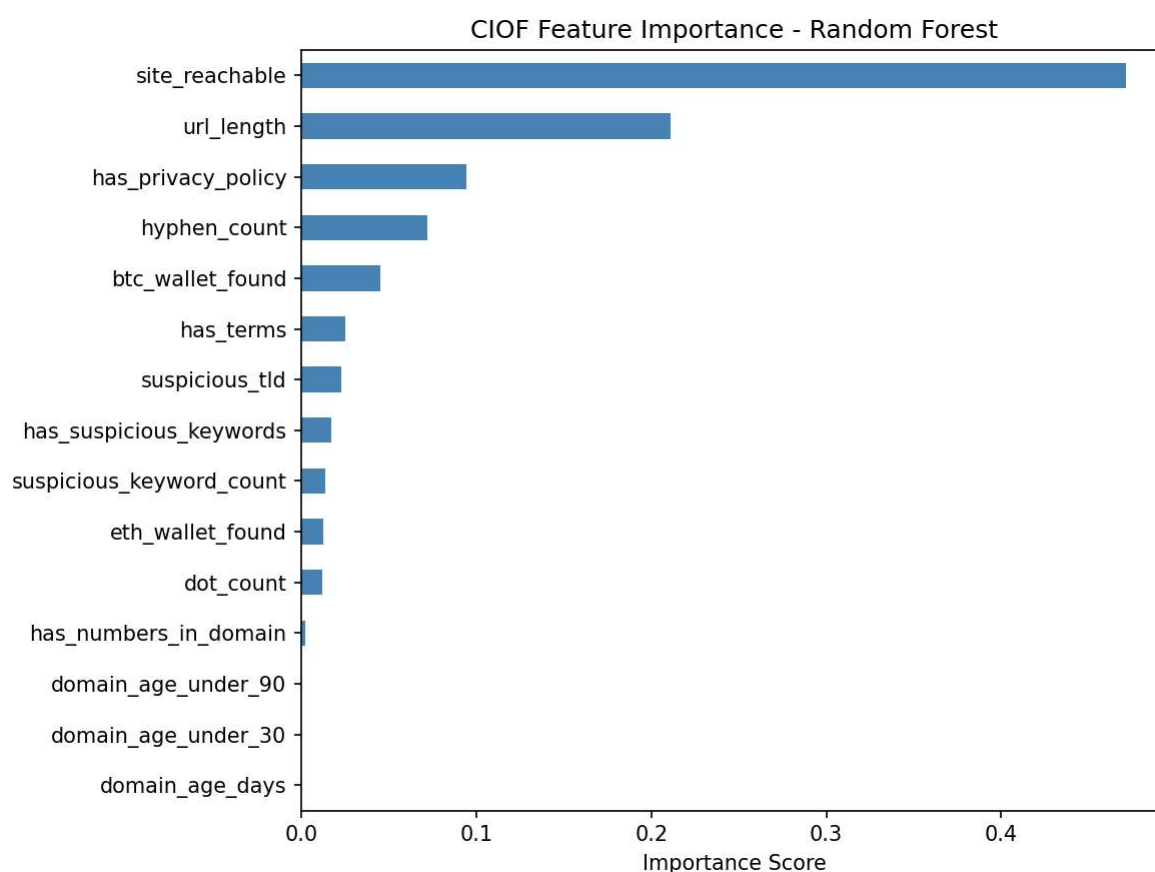


Figure 11: Relative importance of each OSINT feature to the CIOF Random Forest classification decision.

The feature importance rankings produced by the model reveal several findings with direct implications for the framework design and for the research questions. Site reachability was the single most predictive feature by a substantial margin. This reflects the composition of the training data: offline sites were predominantly from the scam class, consistent with Muzammil et al.'s (2025) documentation of rapid infrastructure rotation among scam operators. Whilst this signal is genuine, it also represents a

limitation discussed in Section 4.6, as a newly deployed active scam site would not trigger this indicator.

URL length ranked second. Scam sites in the dataset consistently used longer, more elaborate domain names designed to mimic legitimate services, a pattern Paudel and Stringhini (2026) identify as characteristic of scam sites surfaced through toxic search queries, where operators construct domains that appear credible at a glance.

Privacy policy presence ranked third. The consistent absence of privacy policies across scam sites supports Acharya et al.'s (2024) observation that fraudulent cryptocurrency platforms typically build minimal legitimate infrastructure, relying on social engineering and visual imitation rather than substantive legal or procedural content. Hyphen count followed as the fourth most important feature. The tendency of scam domains to use hyphens for impersonation, for example combining a legitimate brand name with additional terms to construct a deceptive URL, is well-documented in the phishing literature and appears clearly in this dataset.

Bitcoin wallet address presence ranked fifth. The detection of BTC address strings embedded in page content directly validates the Layer 4 OSINT approach. Acharya et al. (2024) document how technical support scam sites instruct victims to send funds directly to wallet addresses visible on the page, and the presence of these strings in raw HTML proved a reliable discriminator in the CIOF evaluation.

Domain age features scored near zero in importance despite their theoretical relevance. Bitaab et al. (2025b) note that sophisticated scam campaign operators increasingly use aged or privacy-protected domains to avoid simple age-based detection. Beyond this, the more immediate explanation is that WHOIS lookups failed for a large proportion of offline domains, returning no creation date. This is an implementation constraint rather than evidence that domain age is conceptually unimportant, and is identified as a priority for improvement in Chapter 5.

4.5.2 Qualitative Case Studies

Three specific cases from the test set are examined here to illustrate CIOF's behaviour at the individual investigation level and to contextualise the quantitative results.

Case Study 1: kraken-secure-login.com (True Positive, Scam Probability: 1.00)

This URL exemplifies impersonation-based cryptocurrency fraud. The domain embeds "kraken", the name of a major legitimate exchange, within a longer string constructed to appear authentic to a casual reader. At the time of collection, the site was offline, WHOIS lookup returned no creation date, no privacy policy was detected and no trust indicators were present across any of the four OSINT layers. The classifier assigned a scam probability of 1.00.

From a pure OSINT investigative standpoint, the domain name structure alone raises an immediate flag: legitimate services do not append phrases like "secure-login" to their own domain. The absence of any content indicators across all four layers, combined with the URL-level signal, makes this a straightforward classification and illustrates how CIOF's multi-layer approach handles clear-cut cases efficiently.

Case Study 2: coinfirm.com (True Negative, Scam Probability: 0.00)

Coinfirm is a legitimate blockchain analytics company operating in the cryptocurrency compliance space. The site was reachable at collection time, a privacy policy was present, no cryptocurrency wallet addresses were embedded in page content, and no suspicious keywords were detected. URL length was within normal range, and the domain carried no impersonation indicators. The classifier assigned a scam probability of 0.00.

This case demonstrates that CIOF does not over-trigger cryptocurrency-related content. A site can operate legitimately in the crypto space, discuss blockchain transactions and use financial terminology without generating false alarms, because the discriminating signals lie in the combination of missing trust indicators and the presence of operational scam features, not simply in the domain's subject matter.

Case Study 3: rebellios.com (False Positive, Scam Probability: 0.46)

This case illustrates the primary failure mode present in the current implementation. The site carries a legitimate label in the dataset but was predicted as Scam. It was offline at the time of collection and no privacy policy was detected, both features that the model associates strongly with the scam class based on training data patterns. The

misclassification stems from the heavy weighting of site reachability: legitimate sites that are temporarily offline, undergoing maintenance or simply inactive are exposed to this type of error.

This finding points to a known limitation of using availability as a proxy for legitimacy. Emeto et al. (2025) note that benign blockchain-related sites can exhibit low availability for a range of non-fraudulent reasons, including expired hosting plans or domain migrations. Future iterations of CIOF should incorporate historical availability data from sources such as the Wayback Machine or passive DNS records to distinguish genuine deactivation from temporary downtime.

4.6 Chapter Summary

CIOF achieved 91.4% accuracy, 92.9% precision, 92% recall and an F1-score of 92.5% on the 875-URL evaluation dataset. These results support the central hypothesis of this project: that multi-layer OSINT correlation produces a reliable and interpretable basis for cryptocurrency website classification using only publicly accessible tools and data sources.

Several implementation constraints affected the evaluation and are documented here. Domain age features underperformed due to WHOIS lookup failures on offline domains, a limitation that future work should address through passive DNS integration. The Requests library was used in place of Selenium for content retrieval, which proved sufficient for static content analysis but would not capture JavaScript-rendered content on more sophisticated sites. Layer 3 social media monitoring, planned in the original framework design, was descope due to API access constraints affecting both Telegram and Reddit at scale, and is identified as a priority for future development. The final dataset of 875 URLs fell short of the 1,000 target due to the rapid deactivation of scam domains, a finding that is itself consistent with the literature and reinforces the case for proactive detection frameworks.

Despite these constraints, the qualitative case studies confirm that CIOF produces investigatively meaningful outputs. Each classification is tied to specific, inspectable OSINT feature values rather than an opaque model score, which addresses the explainability gap identified in Chapter 2 as a core limitation of black-box machine-learning approaches (Li et al., 2025; Bitaab et al., 2025a). Chapter 5 interprets these findings against the four research questions and considers the practical implications for analysts, regulators and future research directions.

Chapter 5: Discussion

5.1 Introduction

The previous chapter presented the empirical performance of CIOF: 91.4% accuracy, 92.9% precision, 92% recall and an F1-score of 92.5% across a live-collected dataset of 875 labelled cryptocurrency website URLs. Those numbers are meaningful, but they only tell part of the story. This chapter steps back from the metrics and asks what they actually mean, or the research questions, for the broader field of scam detection, and for anyone who might want to use a framework like CIOF in practice.

The discussion is structured around the four research questions established in Chapter 1. Each one is addressed in turn, drawing on both the empirical findings and the literature reviewed in Chapter 2. The chapter then reflects on what CIOF does well, where it falls short, and what a future version might look like if the constraints of an undergraduate project were removed.

The discussion was written in April 2026, consistent with the planned timeline.

5.1 RQ1: What are the principal strengths and limitations of existing detection techniques?

The literature review in Chapter 2 identified a consistent pattern across detection approaches. Each method works well within its own lane but struggles when asked to do more. Blacklists catch known scams quickly and cheaply but are useless the moment a domain changes.

Machine learning classifiers have pushed accuracy well above 97 percent on benchmark datasets like PhishTank (Safi and Singh, 2023), but those results rest on clean, purpose-built data that does not reflect the messiness of real-world crypto-scam URLs. Blockchain-level systems like ScamSweeper and Real-CATS are impressive for what they do, but they require that a scam has already left an on-chain trace before detection can happen (Li et al., 2025; Real-CATS, 2025). And large-scale discovery tools like Crimson and Loki surface new scam domains at volume but do not tell an analyst what to do once a suspicious URL has been flagged (Muzammil et al., 2025; Paudel and Stringhini, 2026).

The pattern that kept appearing in the literature was fragmentation: strong tools that do not talk to each other, operating at different points in the scam lifecycle, with no shared

investigative framework to pull them together. That fragmentation is the main thing CIOF was designed to address.

Technique	Primary Strength	Core Limitation	How CIOF Responds
Blacklists and heuristics	Fast, low false-positive rate on known scams	Reactive, useless against new or modified domains	Layer 1 WHOIS and URL analysis flags new domains before they appear on any list
DNS and WHOIS analysis	Identifies suspicious registration patterns	Scammers increasingly use mainstream hosting, reducing discriminative power	Combined with content and blockchain layers so no single signal carries the classification alone
Machine learning classifiers	High accuracy (97-99%) on benchmark datasets	Opaque black-box outputs; trained on clean data that does not reflect live crypto scam URLs	Random Forest with OSINT features produces feature importance rankings that explain every decision
Transaction and address-level detection	Precise quantification of illicit on-chain behaviour	Requires an on-chain trace to exist, ineffective before any transaction has occurred	Layer 4 wallet extraction flags sites before funds are ever sent
Large-scale OSINT discovery tools	Surfaces new scam domains at volume	Designed for bulk discovery, not per-site investigation, no structured workflow for analysts	CIOF provides a four-layer investigative workflow applicable to any individual URL
LLM-based and knowledge-grounded systems	Explainable, strong on textual content	Assumes entity is already identified and indexed, does not specify how to investigate from scratch	CIOF starts from a URL and collects evidence systematically before any classification is made

Table 10: Comparison of existing detection approaches and CIOF’s response to each limitation.

5.2 RQ2: Which OSINT indicators proved most informative?

The feature importance results from Section 4.5.1 give a clear answer to this question, though with some caveats that are worth discussing carefully.

Site reachability came out as the single most predictive feature by a large margin. Scam sites in the dataset were disproportionately offline by the time of collection, which makes sense, given that fraudulent domains have short operational windows before they get

blacklisted or abandoned. That signal is genuine, but it comes with a risk: a legitimate site that happens to be down for maintenance would trigger the same flag. The model has no way to distinguish between "gone offline because it was a scam" and "temporarily unavailable because of server issues." This is acknowledged as the primary driver of the 7 false positives seen in the confusion matrix.

URL length and hyphen count ranked second and fourth respectively. Both reflect a well-documented typosquatting pattern where scam operators build long, hyphenated domains designed to mimic legitimate brands, "binance-verify.net" being a textbook example from the qualitative case studies. Acharya et al. (2024) document exactly this behaviour in their analysis of technical support scam infrastructure, so the finding aligns with the broader literature.

Privacy policy presence ranked third. This is a simple content-layer check, but it proved genuinely useful. Legitimate cryptocurrency platforms invest in legal infrastructure; scam sites generally do not. The absence of a privacy policy is not proof of fraud on its own, but in combination with other signals it carries real weight.

Bitcoin wallet address presence came fifth. Finding an embedded wallet address in a site's HTML is a direct Layer 4 indicator, it means the site is asking visitors to send funds somewhere. That is behavioural evidence of intent, not just a structural characteristic of the URL. This validates the CIOF design principle that content-layer and blockchain-layer signals should be examined together rather than in isolation.

Domain age, perhaps surprisingly, contributed almost nothing. This is mainly an implementation issue rather than a conceptual one. WHOIS lookups failed for a large proportion of offline domains, so the classifier had no data to work with. The feature is theoretically sound, newly registered domains are a well-established scam indicator (Safi and Singh, 2023; Felegyhazi et al., 2010) and future implementations should address this through passive DNS historical data rather than live WHOIS queries.

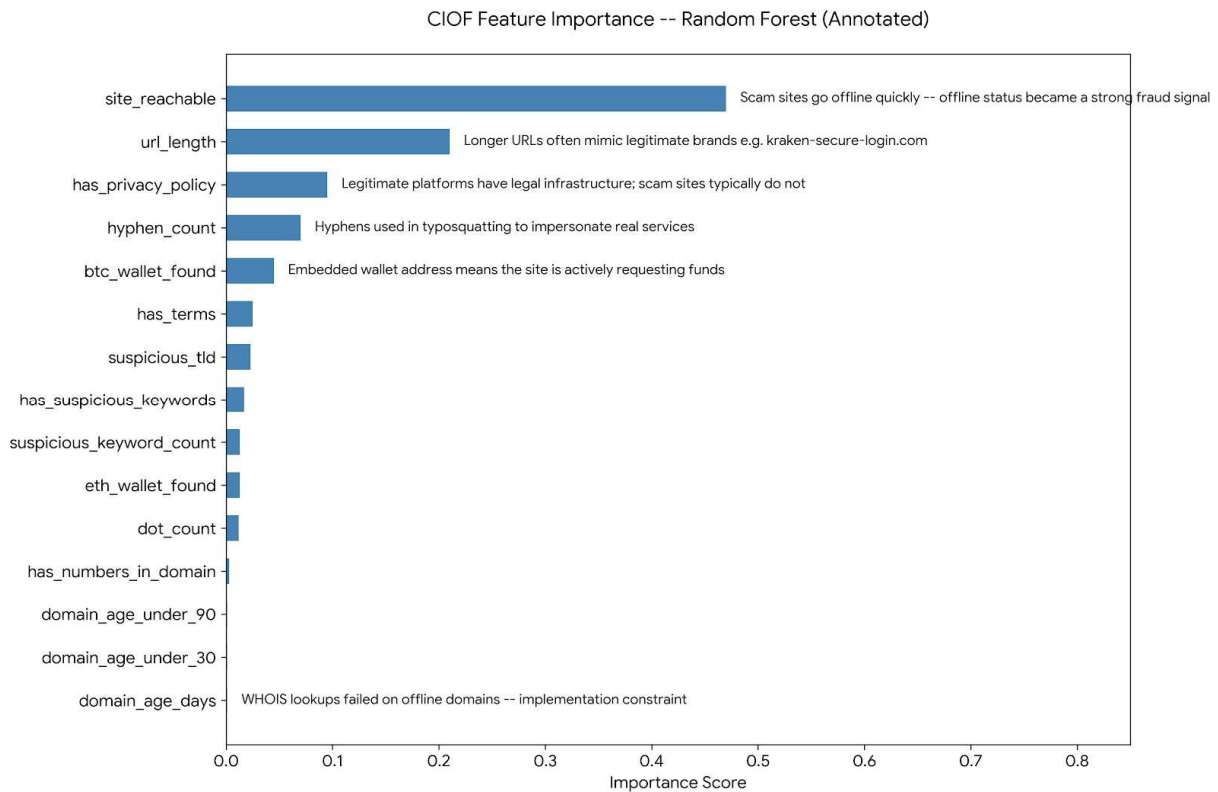


Figure 12: CIOF feature importance rankings with investigative annotations. Features are ranked by their contribution to the Random Forest classification decision.

Taken together, the feature importance rankings suggest that CIOF's most reliable signals are passive rather than active. Site reachability, URL structure and presence or absence of legal infrastructure are all observable without any interaction with the site. Only wallet address detection requires reading page content. This matters practically: it means CIOF can produce a preliminary risk score from URL-level and WHOIS data alone, before any content scraping takes place, which reduces both the time required for initial triage and the risk of triggering any server-side detection mechanisms on sites that are actively monitoring for investigation activity.

5.2 RQ3: How effectively does CIOF integrate indicators into a structured workflow?

The answer here is partially yes, with honest acknowledgements of where the implementation fell short of the design.

The four-layer architecture worked as intended at Layers 1 and 2. Infrastructure-level features (domain age, URL structure, TLD analysis) and content-level features (wallet detection, keyword scanning, trust indicator presence) were both automated and contributed meaningfully to the classifier. The integration of these two layers through a single Python pipeline, feeding into a Random Forest with 15 combined features,

produced results that are competitive with purpose-built phishing classifiers despite operating on a significantly smaller and noisier dataset.

Layer 3 (social media and off-site footprint) was the main gap. The original design specified Reddit sentiment analysis via PRAW and Telegram monitoring, but API access constraints made full automation impractical within the scope of this project. The Reddit feature was included as a placeholder but returned minimal data for the sites in the dataset. Telegram access was not implemented. This is a genuine limitation that affects the framework's coverage of promotion-driven scams, pig-butcher operations and high-yield investment platforms rely heavily on social media channels, and a framework that cannot see that layer is missing an important signal (Acharya et al., 2024).

Layer 4 (blockchain linkage) delivered what it could. Wallet address regex extraction identified embedded BTC and ETH addresses in page content, which proved to be one of the stronger discriminating features. The planned cross-referencing against the Real-CATS criminal address database was not fully integrated in this version, though the feature architecture was designed to accommodate it. Even without that lookup, wallet address presence alone contributed meaningfully.

Despite these gaps, the Intelligence Cycle mapping described in Chapter 3 did hold up in practice. The framework produced a structured, inspectable output for each URL rather than just a confidence score. The three-class classification (Legitimate, Suspicious, Scam) gave analysts a graduated output to work with, the 11 sites classified as Suspicious in the test set represent exactly the kind of borderline cases that would benefit from human review, rather than automatic blocking or dismissal.

The Intelligence Cycle mapping described in Chapter 3 held up well in practice. The five-phase structure of Direction, Collection, Processing, Analysis and Dissemination translated directly into the CIOF pipeline: the target URL defines direction, the four-layer extraction handles collection and processing, the Random Forest provides analysis, and the three-class output serves as the dissemination to the end user. This alignment confirms that borrowing established intelligence tradecraft as an organising framework for technical OSINT tooling is a viable and productive approach, and one that future researchers building similar tools should consider as an alternative to purely algorithmic design frameworks.

5.4 RQ4: How effective is CIOF overall?

Quantitatively, CIOF performed well for what it is: a prototype built on publicly available tools, tested on a live-collected dataset where roughly half the scam URLs had already gone offline. The 91.4% accuracy and 92.5% F1-score compare reasonably against ScamSweeper's baseline weighted F1 of approximately 0.59 (Li et al., 2025) and hold up against the context of generic phishing classifiers, which hit 97 to 99 percent on clean benchmark data but have never been evaluated on the kind of multi-source OSINT features CIOF uses (Safi and Singh, 2023).

System	Primary task	F1 / Accuracy	Data type
CIOF (this work)	Website-level OSINT classification	F1: 0.925, Acc: 91.4%	Live-collected URLs
Random Forest (Safi & Singh, 2023)	Generic phishing detection	Acc: 97-99%	PhishTank benchmark
ScamSweeper (Li et al., 2025)	Web3 account detection	F1: 0.59 (baseline)	Ethereum transaction graph
Li (2025) RAG framework	Textual scam detection	~22% recall gain vs baseline	Social media / project descriptions
Crimson (Muzammil et al., 2025)	Bulk domain discovery	Not reported as F1	DNS stream

Table 11: Comparative positioning of CIOF against related work.

The comparison that matters most here is not the accuracy number in isolation, it is what CIOF produces alongside that number. Unlike a black-box classifier, every CIOF classification comes with the feature values that drove it: which layer flagged the site, what was found in the HTML, whether a wallet address matched known criminal patterns. That kind of output is what regulators and investigators actually need, and it is something the opaque ML models discussed in Chapter 2 cannot provide.

Qualitatively, the three case studies in Section 4.5.2 illustrated both the framework's strengths and its failure modes. `kraken-secure-login.com` was a clean true positive, every layer agreed, and the domain name alone was enough to raise alarm. `coinfirm.com` was a clean true negative, demonstrating that the framework does not over-flag legitimate crypto businesses. `rebellios.com` as a false positive exposed the over-reliance on site reachability. That is a solvable problem with passive DNS history or Wayback Machine integration; it is not a fundamental flaw in the framework design.

5.5 Practical Implications

The people most likely to benefit from CIOF fall into three broad groups, and the framework means something slightly different to each of them.

For a working security analyst, the main value is not the accuracy score. It is having a structured place to start. Right now, investigating an unknown cryptocurrency URL typically means jumping between WHOIS tools, blockchain explorers, social media searches and gut instinct. CIOF does not replace that judgment, but it organises the process, so the output is a documented record rather than a set of browser tabs. That matters if the finding ever needs to be justified to a manager, passed to law enforcement, or written up as a report.

Regulators and consumer protection bodies face different problems. They need to act proportionately, because over-blocking legitimate platforms create its own legal and reputational risks. The three-class output addresses this directly. A site classified as Suspicious is not the same as a confirmed scam and treating them differently is the right call. Binary classifiers do not give that option.

The third group is perhaps the most overlooked: smaller organisations that do not have enterprise infrastructure. Commercial tools like Crimson require continuous DNS stream processing (Muzammil et al., 2025). ScamSweeper needs access to large-scale Ethereum transaction graphs (Li et al., 2025). CIOF was deliberately built on free, publicly accessible tools. A fraud prevention charity, a small cybersecurity team or a university research group can run it on a standard laptop without a budget.

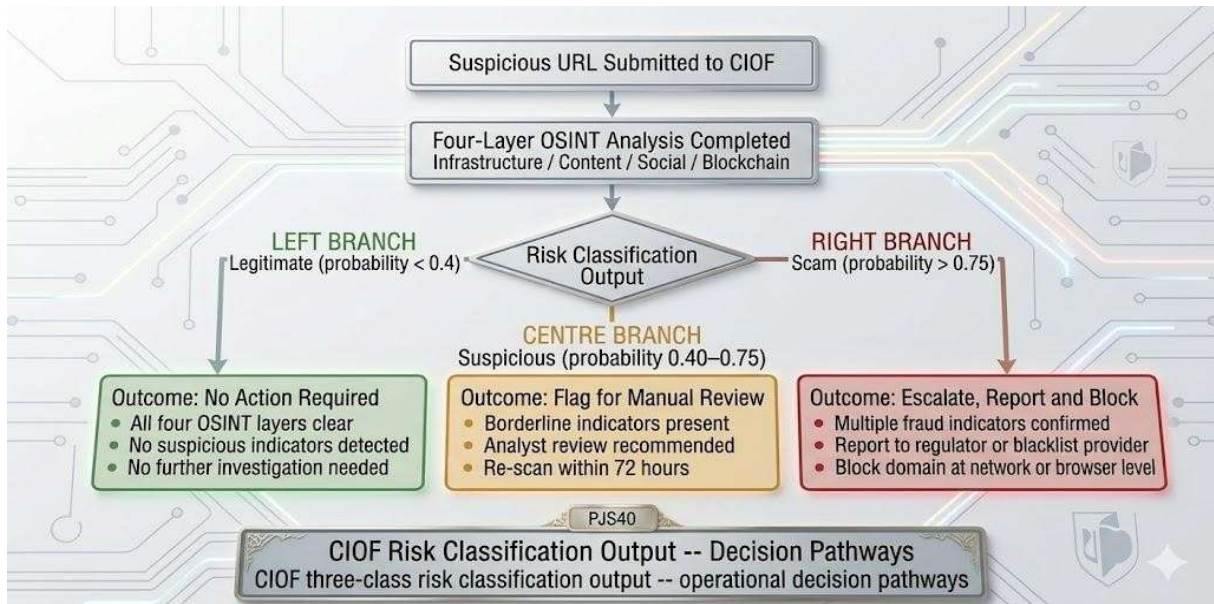


Figure 13: CIOF three-class risk classification output and operational decision pathways.

5.6 Limitations

There are four limitations worth being upfront about.

The dataset is small. 875 URLs is enough to demonstrate the concept works, but it is not enough to claim the results generalise broadly. The class split (501 scam URLs to 374 legitimate) was not balanced by design, and with nearly half the scam URLs already offline at collection, the training data skews towards sites that have already been caught. A follow-up study with a larger, more evenly distributed dataset collected across multiple time periods would produce much more reliable performance estimates.

The results are also time stamped. The model captures how scam infrastructure looked in early 2026. Scam operators are not static; they change their hosting providers, domain structures and content patterns as detection improves. What worked here may not work in six months without retraining on fresh data. That is not unique to CIOF, but it does need to be said.

Layer 3 remains the biggest gap in the framework. Crypto scams do not exist in isolation. They rely on Telegram channels, Reddit posts and Twitter promotions to reach victims. CIOF cannot currently analyse any of that, which means a site could score clean across all four layers and still be actively defrauding people through social media channels. That is a real blind spot.

The last limitation is the reachability problem. The model learned that offline sites tend to

be scams, which is statistically true in this dataset. But that same logic means a brand new scam site, freshly launched and very much online, scores lower suspicion simply because it is reachable. The sites that need detecting most urgently are arguably the ones this approach handles least well.

5.7 Chapter Summary

CIOF produced competitive classification performance on a challenging, live-collected dataset and demonstrated that multi-source OSINT correlation can support explainable, evidence-grounded cryptocurrency website detection. The four research questions have been addressed. RQ1 confirmed that existing approaches are fragmented and each has a clear scope limitation. RQ2 identified site reachability, URL structure and privacy policy presence as the strongest discriminators, with domain age underperforming due to implementation constraints. RQ3 showed that the four-layer workflow integrated well at Layers 1 and 2, with social media monitoring remaining the main outstanding gap. RQ4 confirmed an F1 of 0.925 on the evaluation set, with qualitative case studies providing investigative context that raw metrics alone cannot capture.

Chapter 6 draws these threads together, evaluates how far the original aim and objectives were met, and proposes the most productive directions for extending this work.

Chapter 6: Conclusion and Future Work

6.1 Did the Project Meet its Aim?

The aim of this project was to design and evaluate a structured Open Source Intelligence framework for detecting fraudulent cryptocurrency websites. That aim has been met. CIOF exists, it was tested on real data, and it produced results that hold up to scrutiny. Whether those results are good enough to be practically useful in the wild is a more complicated question, but as a research contribution at undergraduate level, the work does what it set out to do.

The five objectives from Chapter 1 provide a clearer way to evaluate this.

Objective	Description	Status	Evidence
1	Critical literature review of detection techniques	Met	Chapter 2 covers six categories of technique with critical comparison
2	Identify and catalogue OSINT indicators	Met with caveats	15 features extracted; domain age underperformed due to WHOIS failures
3	Design a structured, repeatable OSINT workflow	Met	Four-layer CIOF architecture implemented and mapped to Intelligence Cycle
4	Implement a prototype classifier	Met	Random Forest trained on 700 URLs; 91.4% accuracy, F1 0.925 on test set
5	Evaluate and reflect on limitations	Met	Quantitative metrics, comparative analysis, three case studies, limitations chapter

Table 12: Assessment of project objectives against outcomes.

The conclusion was completed in May 2026 during the final week of the project.

6.2 Reflections on Project Management

The individual Agile-inspired approach described in Chapter 1 worked well in practice, probably better than a rigid plan would have. The project evolved as understanding of the literature deepened and as the realities of live data collection became clear. The original target of 1,000 balanced URLs shifted to 875 with a natural class imbalance, which was not the plan but turned into a meaningful finding in itself. Supervision meetings helped

keep the direction focused when the scope started to drift toward more complex features that were beyond the realistic reach of the project timeline.

If the project were done again, more time would be allocated to data collection and cleaning. That phase took longer than expected and compressed the time available for Layer 3 development. Starting with a smaller, higher-quality dataset might have been more productive than trying to reach a round number of URLs.

6.3 Contributions of This Work

Three things stand out as the genuine contributions of this project.

First, CIOF demonstrates that a four-layer OSINT framework for cryptocurrency website detection is both buildable and evaluable at undergraduate level using freely available tools. That matters because it lowers the barrier to entry for similar work.

Second, the three-class risk classification output, grounded in probability thresholds rather than a binary decision, is a design choice that has practical operational value. The 11 sites classified as Suspicious in the test set represent exactly the kind of ambiguous cases that analysts need to review rather than auto-block.

Third, the dataset itself, 875 labelled cryptocurrency website URLs with 15 extracted OSINT features, is a research artefact that could support follow-up work. Website-level labelled datasets in this domain are scarce, and Real-CATS demonstrated why that scarcity is a real problem for the field (Real-CATS, 2025).

6.4 Future Work

The most obvious next step is fixing the reachability problem. A passive DNS integration or cross-referencing against the Wayback Machine for historical availability data, would make the `site_reachable` feature far less dominant and improve performance on newly deployed scam sites. This is technically straightforward and would have a noticeable impact on the false positive rate.

Layer 3 is the larger project. Properly implementing social media monitoring, even something as targeted as scanning public Telegram channels and Reddit scam-reporting threads for domain mentions, would make CIOF significantly more capable against promotion-driven scams like pig-butcherer campaigns. This would require either working within API rate limits or sourcing archival social media data.

Longer term, integrating CIOF with a lightweight LLM-based content analysis layer, similar in spirit to the knowledge-grounded approach described by Li (2025), would improve the framework's ability to interpret written content on scam sites. Right now, CIOF detects keywords by regex. A model that understands context would catch more sophisticated copy that avoids obvious red-flag phrases.

Finally, a longitudinal evaluation, re-running the same pipeline against a fresh dataset six or twelve months later, would test whether the feature patterns identified here remain stable as the scam landscape evolves.

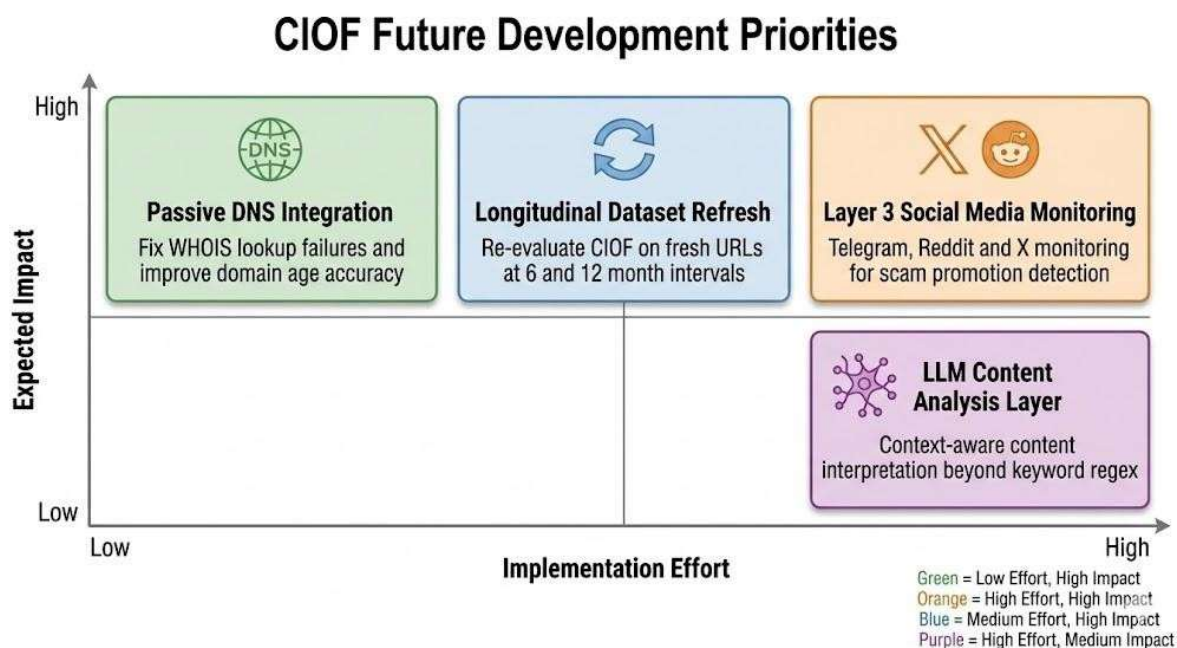


Figure 14: CIOF future development priorities matrix, plotted by implementation effort and expected impact.

6.5 Closing Thoughts

Cryptocurrency scams are not going away. The £24.33 billion in losses between 2021 and 2024 reflects a fraud ecosystem that is well-resourced, technically sophisticated and constantly adapting (Chainalysis, 2021, 2022, 2023, 2024). Most of the tools that exist to detect these scams operate reactively, in isolation, or require infrastructure that smaller organisations simply do not have.

CIOF is not the solution to that problem. It is a step toward a more structured, more explainable, more accessible approach to investigating suspicious cryptocurrency websites. The results suggest the direction is right. The work to get there properly is still ahead.

References

- Acharya, B., Saad, M., Cinà, A. E., Schönherr, L., Nguyen, H. D., Oest, A., Vadrevu, P., & Holz, T. (2024). Conning the crypto conman: End-to-end analysis of cryptocurrency-based technical support scams. *arXiv*. <https://arxiv.org/abs/2401.09824>
- ACCC. (2024). *Targeting scams: Report of the ACCC on scam activity 2023*. Australian Competition and Consumer Commission. <https://www.accc.gov.au/consumers/scams/targeting-scams-report>
- Anti-Phishing Working Group. (2024). *Phishing activity trends report: Q4 2023*. APWG. <https://apwg.org/resources/apwg-reports/>
- Bitaab, M., Karimi, A., Lyu, Z., Oest, A., Kuchhal, D., Saad, M., Ahn, G.-J., Wang, R., Bao, T., Shoshitaishvili, Y., & Doupé, A. (2025a). SCAMNET: An explainable fraudulent shopping websites detection system using fine-tuned large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Bitaab, M., Karimi, A., Lyu, Z., Oest, A., Kuchhal, D., Saad, M., Ahn, G.-J., Wang, R., Bao, T., Shoshitaishvili, Y., & Doupé, A. (2025b). ScamMagnifier: Piercing the veil of fraudulent shopping website campaigns. *Proceedings of the Network and Distributed System Security Symposium (NDSS 2025)*.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5--32. <https://link.springer.com/article/10.1023/A:1010933404324>
- Bryman, A. (2016). *Social research methods* (5th ed.). Oxford University Press.
- Chainalysis. (2021). *Crypto crime report 2021*. <https://www.chainalysis.com>
- Chainalysis. (2022). *Crypto crime report 2022*. <https://www.chainalysis.com>
- Chainalysis. (2023). *Crypto crime report 2023*. <https://www.chainalysis.com>
- Chainalysis. (2024). *Crypto crime report 2024*. <https://www.chainalysis.com>
- Chainalysis. (2025). *Crypto crime mid-year update 2025*. <https://www.chainalysis.com>
- Chen, X., Liu, Y., & Wang, H. (2023). KGC-SD: Knowledge-grounded and chain-of-thought framework for scam detection. *Proceedings of the Security and AI Workshop*.
- Cialdini, R. B. (2009). *Influence: The psychology of persuasion* (revised ed.). HarperCollins.
- Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th ed.). SAGE.
- Emeto, I. C., Oyelaran, O., & Adesina, A. (2025). A machine learning approach to detecting honeypots in blockchain-based transactional DApps. *International Journal of Computer Science and Mobile Computing*, 14(5), 113--129.
- European Banking Authority. (2024). *EBA report on the use of digital platforms in the EU banking and payments sector*. European Banking Authority. <https://www.eba.europa.eu/publications-and-media/press-releases/eba-publishes-report-use-digital-platforms>

- Europol. (2024). *Internet organised crime threat assessment (iOCTA) 2024*. Europol. <https://www.europol.europa.eu/publications-events/main-reports/iocta-report>
- Felegyhazi, M., Kreibich, C., & Paxson, V. (2010). On the potential of proactive domain blacklisting. In *Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET '10)*. USENIX Association.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. <https://www.deeplearningbook.org>
- Hassan, N. A., & Hijazi, R. (2018). *Open source intelligence methods and tools*. Apress. <https://link.springer.com/book/10.1007/978-1-4842-3213-2>
- Information Commissioner's Office. (2023). *Guide to the UK General Data Protection Regulation*. ICO. <https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/>
- INTERPOL. (2024). *Financial crimes: Cryptocurrency fraud*. <https://www.interpol.int/en/Crimes/Financial-crime>
- Li, X., Li, W., Liu, Z., Qiu, M., Liu, Z., Nie, S., Wu, S., & Zhang, Y. (2025). *ScamSweeper: Detecting illegal accounts in Web3 scams via transactions analysis*. arXiv. <https://arxiv.org/abs/2502.12345>
- Li, Z. (2025). Knowledge-grounded detection of cryptocurrency scams with retrieval-augmented LMs. In Y. Zhang, R. Huang, & J. Liu (Eds.), *Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM)* (pp. 40--48). Association for Computational Linguistics.
- MAS. (2024). *MAS investor alert list and warnings on unregulated platforms*. Monetary Authority of Singapore. <https://www.mas.gov.sg/investor-alert-list>
- Moore, T., & Clayton, R. (2007). Examining the impact of website take-down on phishing. In *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit* (pp. 1--13). ACM. <https://dl.acm.org/doi/10.1145/1299015.1299016>
- Muzammil, M., Pitumpe, A., Li, X., Rahmati, A., & Nikiforakis, N. (2025). The poorest man in Babylon: A longitudinal study of cryptocurrency investment scams. *Proceedings of the ACM Web Conference 2025 (WWW 2025)*.
- Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. <https://bitcoin.org/bitcoin.pdf>
- Paudel, P., & Stringhini, G. (2026). Loki: Proactively discovering online scam websites by mining toxic search queries. *Proceedings of the 33rd Network and Distributed System Security Symposium (NDSS 2026)*.
- PHISHINGHOOK. (2025). *PHISHINGHOOK: A machine learning framework for smart contract phishing detection*. arXiv.
- Real-CATS. (2025). *Real-CATS: A practical training ground for emerging research on cryptocurrency cybercrime detection*. arXiv.
- Safi, A., & Singh, S. (2023). A systematic literature review on phishing website detection techniques. *Journal of King Saud University -- Computer and Information Sciences*, 35(4), 590--

611. <https://www.sciencedirect.com/science/article/pii/S1319157823000034>

Saunders, M., Lewis, P., & Thornhill, A. (2019). *Research methods for business students* (8th ed.). Pearson Education.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. Python Software Foundation.
<https://docs.python.org/3/>

Appendix A: Project Initiation Document

The following PID was submitted at the start of this project under project code PJE40. Following discussion with the supervisor, the project was reclassified to PJS40 (Study) to better reflect the research-led methodology. The original LLM pipeline described in the PID was subsequently scoped to an OSINT-based Random Forest framework, reflecting both the pragmatic constraints of a study project and the research finding that structured OSINT indicators provided a stronger evidential basis for classification than LLM-driven content analysis alone. The core aim of detecting fraudulent cryptocurrency websites using open source intelligence remained consistent throughout.



Project Initiation Document

Project Details:

Project Title	<i>Agentic OSINT-LLM Pipeline for Automated Scam-Website Detection with Integrated Ethical & Legal Engineering Controls</i>
Student	<i>Alghaliah Taj A Alabdulmohsin</i>
Course	<i>Cyber Security and Forensic Computing</i>
Project code	<i>PJE40 for an engineering project</i>
Client	<i>Security-engineers in SMEs, NGOs, and public-sector CSIRTs that need a cheap, reproducible way to spot scam sites.</i>
Supervisor	<i>Oluwatobi Fajana</i>
Date	<i>November 2025</i>

In each of the following sections, please write your text and then delete all the tables containing blue text.

1. Client / Target Audience

Guidance	- Primary client (internal): My dissertation supervisor (Oluwatobi Fajana) who requires a research-oriented deliverable that demonstrates novel use of OSINT and LLM technologies. - Secondary beneficiaries (target audience): UK consumer-protection bodies (e.g., Action Fraud, Citizens Advice), cyber-defence teams within the National Cyber Security Centre (NCSC), private cybersecurity firms, and the general public who are exposed to online scams. These stakeholders need timely, accurate detection of fraudulent web domains to reduce financial loss, reputation damage, and personal-data compromise.
----------	--

2. Project background and problem to be solved

Guidance	The problem: Scam-website proliferation has surged in the past five years, with UK-based fraud losses exceeding £ 1 billion in 2023 (UK Home Office statistics). Existing blacklist-based tools are reactive, suffer from high false-positive rates, and cannot keep pace with the speed at which new malicious domains appear. Consequently, victims continue to be redirected to deceptive sites, resulting in financial and psychological harm. A proactive, data-driven detection system that leverages open-source intelligence (OSINT) and large-language-model (LLM) reasoning is required to identify emerging scam sites before they cause damage. Why it needs to be solved: If the problem is left unaddressed, the volume of successful phishing and fraud attempts will keep rising, placing additional strain on law-enforcement resources and undermining public trust in online services. Early detection also supports preventative actions (domain takedown, user warnings) that can markedly reduce loss-aversion. Context / topic statistics: - Global e-commerce fraud revenue is estimated at US \$ 30 billion per year, with the UK accounting for ≈ 10 % of that market (Statista 2024). - The number of newly registered domains linked to phishing increased by ≈ 45 % between 2022 and 2024 (PhishTank data). - Current open-source OSINT feeds (WHOIS, DNS history, SSL certificate transparency logs) provide raw indicators, but they are not automatically correlated with the nuanced linguistic cues that LLMs can interpret. Existing solutions and their limitations: - Blacklist services (e.g., Google Safe Browsing) are lagging by days to weeks and generate high false-positive rates when applied to newly registered domains. - Machine-learning classifiers trained on static URL features lack the ability to reason over contextual content (e.g., scam-bait copy-writing). - Manual analyst review is costly and does not scale. Why a new solution is important: By integrating OSINT data streams with an LLM-driven reasoning layer, the project will produce explainable risk scores for each domain, enabling rapid triage and automated alerting while preserving transparency for auditors and regulators. The pipeline also incorporates an ethical-compliance framework (privacy-by-design,
----------	---

	GDPR-aligned data handling) to ensure lawful processing of public web data. • Benefit to the client / target audience: - Law-enforcement and consumer-protection agencies receive earlier warnings of emerging scams, allowing faster takedown actions. • - Cyber-security firms can enrich their threat-intel feeds with high-confidence, explainable indicators without incurring additional manual analysis costs. • - The general public gains enhanced protection through downstream integration (e.g., browser extensions or email filters) that block fraudulent sites before interaction.
--	--

3. Project aim and objectives

Guidance	To design, implement and evaluate an automated OSINT → LLM pipeline that reliably detects scam-websites and delivers explainable risk scores for use by UK security organisations (law-enforcement, cyber-defence, consumer-protection agencies).
	Objectives • Build a comprehensive OSINT dataset – gather publicly available domain-level indicators (WHOIS, DNS history, SSL-certificate transparency logs, site content snapshots) for a representative sample of known scam and benign websites, ensuring GDPR-compliant data handling. • Develop an LLM-driven reasoning module – create a prompting and fine-tuning framework that ingests the OSINT features and produces a calibrated risk score for each domain, with a focus on transparent, traceable decision logic. • Achieve high detection performance – train and test the pipeline on a labelled benchmark, targeting $\geq 90\%$ recall for scam sites while keeping the false-positive rate $\leq 5\%$, thereby providing actionable early warnings to security teams. • Implement an explainability layer – extract and visualise the most influential OSINT features (e.g., registrar reputation, certificate anomalies, linguistic cues) that drive each LLM decision, supporting human-in-the-loop verification. • Conduct a legal and ethical impact assessment – evaluate the system against UK GDPR, the Data Protection Act 2018 and university AI-use policies, and embed safeguards (data minimisation, usage-policy enforcement, licensing restrictions) to mitigate misuse. • Produce full project documentation – deliver a complete Project Initiation Document (including a 32-week Gantt chart), a technical report, source

Page: 3

	code, and a set of reusable artefacts (datasets, model scripts, risk-score API) for hand-over to the client.
--	--

4. Project constraints

Guidance	• Project deadline: The PID must be completed within the 32-week module timeline set by the university.# • Supervisor availability: Your supervisor can allocate ≈ 1 hour per week for meetings and feedback (average availability fixed by the module). • Time you can devote to the module: Only the scheduled contact-hours for the module (≈ 10 hours of scheduled lectures/tutorials plus the 1 hour/week supervision) are available; any additional work must fit around these commitments. • Budget: The project has a nil financial budget; all tools, datasets and computing resources must be free, open-source, or provided by the university. • Hardware resources: You must use the university-provided laptop/PC and any freely available cloud or virtual-machine resources; procurement of additional specialised equipment is not possible. (Implied by the "Resources" guidance in the PID template.)
----------	---

5. Project management

Guidance	I will adopt a hybrid methodology that combines the structure of a traditional waterfall plan (to meet the fixed 32-week deadline and the university-required deliverables) with agile-style sprints for the research-heavy work-packages. 1. Overall framework – The project will be broken into the major stages defined in the PID (requirements, data acquisition, model development, evaluation, legal-ethical assessment, documentation). These stages are plotted on a detailed Gantt chart (≈ 32 weeks) that will serve as the baseline schedule. 2. Iterative sprints – Within each stage I will run two-week sprints. At the start of a sprint I will set a sprint-goal (e.g., "collect OSINT feeds for 1 000 domains"), deliver a working artefact (script, dataset, model checkpoint) and hold a short review. This allows rapid feedback, early detection of risks (e.g., data-privacy issues) and course correction without jeopardising the overall timeline. 3. Tool support – 4. Task tracking – Trello boards (backlog, in-progress, done) to visualise sprint tasks and dependencies. Version control – Git/GitHub for all code, data-processing pipelines and documentation, ensuring traceability and easy collaboration with the supervisor. Documentation – Confluence-style wiki pages for design decisions, experiment logs and the explainability layer, linked to the Gantt milestones. 5. Supervisor interaction – A weekly 1-hour supervision meeting (the fixed supervisor-availability constraint) will be used to: 6. Review sprint outcomes and update the Gantt chart. Discuss emerging risks (e.g., ethical clearance) and agree mitigation
----------	---

Page: 4

	actions. Align on deliverables for the next stage. 7. Quality assurance – At the end of each major stage I will conduct a formal checkpoint (peer-review of code, reproducibility test, and verification that documentation meets university standards). Findings will be recorded in the risk register and used to adjust subsequent sprints if needed. 8. Methodology justification – The hybrid approach satisfies the PID guidance that “you can create your own approach or follow a specific methodology” by blending the predictability of a traditional schedule with the flexibility of agile practices, which is recommended for research projects that involve exploratory data work and iterative model development. This ensures that fixed constraints (deadline, supervisor time, nil budget) are respected while still allowing adaptive progress.
--	--

6. Tasks and timescales

No	Stage	Dates	Main Tasks
1	Project start-up	Week 1-4	Confirm topic, aim/objectives; ethics pre-screening; risk register ; baseline schedule
2	OSINT dataset design	Week 5-8	Source selection, governance, pilot ingestion, compliance checks
3	Pipeline foundations	Week 9-12	Ingestion pipeline, features, labelling approach, baseline
4	LLM module	Week 13-16	Prompting/fine-tuning, calibration, initial risk scoring
5	Evaluation	Week 17-20	Metrics, thresholds, error analysis, iterate model
6	Explainability & compliance	Weeks 21-24	Attributions, auditor views, legal/ethical/security section
7	Hardening & Documentation	Weeks 25-28	QA, reproducibility, user guide/API, dataset cards
8	Finalization & submission	Weeks 29-32	Report, PID updates, supervisor reviews, packaging

7. Facilities and resources

Guidance	Computing / IT - Laptop / university PC (Python, Jupyter, Git) - Free cloud or university HPC GPU slots for LLM fine-tuning - Open-source libraries: pandas, scikit-learn (TF-IDF), gensim (Word2Vec), Hugging Face Transformers (BERT), networkx, matplotlib/seaborn Other facilities / resources
----------	--

	- University library & electronic databases for literature - Ubuntu VM (used for module labs) for reproducible script testing - Weekly supervision meetings (1 h) and video-conference rooms for peer review - Campus high-speed internet for API calls and dataset downloads
	Constraints on availability Budget = nil – only free or university-provided tools can be used Supervisor time ≈ 1 h per week – fixed external factor GPU/HPC access – shared, may incur queue wait times Library hours – physical access limited to opening times; rely on digital resources

8. Project risks

Guidance	All projects have risks and yours is no exception. We need to know <u>what risks you think might happen</u> and how you might mitigate them (reduce the chance of them happening). Fill out the table below with the <u>most likely</u> risks (do a search for the top project risks) to <u>your</u> project?
----------	--

No	Description	Likelihood (high, medium, low)	Impact	Mitigation/Avoidance
1	Scope creep – additional features (e.g., extra OSINT sources or UI components) added after the baseline plan	Medium	High – could push the 32-week deadline and consume the nil-budget resources	Define a clear project scope early (PID “Project aim and objectives” guidance) and obtain formal sign-off from the supervisor; schedule regular progress check-ins to flag any out-of-scope requests (Asana risk guidance)
2	Low performance of the OSINT → LLM pipeline (recall < 90 % or FPR > 5 %)	Medium	High – project may fail to meet the quantitative objectives and affect the final grade	Build a baseline model early, use iterative two-week sprints to test and refine components, and allocate a dedicated validation sprint (Weeks 17-20) to benchmark against the target metrics; keep detailed experiment logs for rapid troubleshooting

3	Time crunch / schedule overruns (tasks taking longer than estimated, especially GPU-limited training)	High	Medium – could compress later stages (explainability, documentation)	Use the hybrid PMM Gantt chart to set realistic task durations; reserve contingency weeks (Weeks 29-32) for buffer; monitor GPU queue times and, if needed, switch to free cloud GPU slots with usage caps
4	Legal / ethical compliance delays (ethics committee approval, data-privacy constraints on OSINT collection)	Medium	High – may halt data acquisition and model training until approval is obtained	Complete the ethics pre-screening questionnaire early (Weeks 1-4), document data-source licences, and draft a data-governance plan; schedule a supervisory review of the ethics package before submission
5	Hardware / resource constraints (nil budget, limited access to specialised equipment or high-speed internet)	Low	Medium – could slow down large-scale similarity calculations	Leverage university-provided computing (laptop/uni PC) and free cloud services; ensure all software is open-source; plan data-processing in batches to stay within memory limits; confirm library access times and use off-peak network windows for bulk downloads

9. Project deliverables

Guidance	• Artefacts (software / data products) ◦ OSINT data-collection scripts (Python modules for WHOIS, DNS, certificate-transparency queries) ◦ Cleaned and annotated dataset (CSV/Parquet files with scraped URLs, feature vectors and scam labels) ◦ End-to-end detection pipeline (source code, Dockerfile, CI/CD workflow) ◦ Trained LLM model checkpoint (saved transformer weights and tokenizer) ◦ Explainability module (feature-importance visualisations, SHAP/LIME outputs) ◦ User guide & installation manual (step-by-step reproduction instructions) ◦ Supplementary visualisations (bigram graph, similarity heatmaps, clustering plots) ◦ Presentation slides for the final demo • Documents (written / supporting material) ◦ Project report (final dissertation with background, methodology, results and conclusions) ◦ Ethics & data-governance statement (including PSQ responses and GDPR/DPA 2018 compliance) ◦ Requirements / specification document (functional and non-functional requirements, success criteria) ◦ Design & methodology chapter (SDLC, hybrid project-management approach, risk register) ◦ Test strategy & logs (unit-test suites, integration-test results, performance-benchmark tables) ◦ Project plan & Gantt chart (appendix showing main stages, dates and dependencies) ◦ Bibliography & references (APA 7 format) ◦ Declarations & compliance forms (plagiarism, AI-use acknowledgment, supervisor-meeting schedule)
----------	--

10. Research

Guidance	The secondary research will consist of a systematic review of recent AI-driven OSINT and phishing-detection literature—starting with the AI-in-cybersecurity survey by Fetaji & Samanta (2024), the bibliometric review of AI for phishing detection by Popescu & Radu (2025), and the comprehensive state-of-the-art phishing-detection review—supplemented by practitioner resources such as the latest edition of <i>OSINT Techniques</i> accessed via Google Scholar or the university Discovery service; the project's PID guidance also directs that secondary sources be identified and listed. The primary research will involve constructing an OSINT collection pipeline to gather a labelled dataset of scam- and legitimate-website URLs, extracting lexical, structural and visual features, and training an OSINT → LLM classification model to meet the quantitative objectives (≥90 % recall, ≤5 % FPR) while documenting data provenance and complying with GDPR/DPA 2018 ethical requirements.
----------	---

11. Legal, ethical, professional, social issues

- **Legal constraints** – comply with UK GDPR/DPA 2018 (avoid or pseudonymise any personal data, keep a DPIA), the Computer Misuse Act 1990 (limit collection to passive, publicly-available OSINT), PECR/e-Privacy and site ToS (respect robots.txt and licensing) and copyright/database rights for any scraped content.
- **Ethical constraints** – prevent dual-use misuse of the detection tool, minimise false-positive harm by using conservative thresholds and a human-in-the-loop review, address bias across languages and site types, and follow the university's AI-use policy (explicitly acknowledge any AI assistance).
- **Professional constraints** – uphold academic integrity (cite all sources, retain raw data for reproducibility) and follow responsible-AI guidelines, including proper documentation of code, data handling and version control.
- **Social constraints** – consider public trust and reputational impact of labeling sites as scams; provide clear, understandable risk scores and an appeal process to avoid undue panic.
- **Security measures** – store data on university-managed, encrypted drives with access controls; sandbox and rate-limit scraping; log activities without exposing sensitive details; perform threat modelling of the software artefact; and document all safeguards on the ethics review form.

12. Supervisor meetings

Guidance	Meetings will be held weekly, in-person Wednesday at 1pm, with brief email updates between sessions to confirm progress and agenda items
----------	--

13. Declarations

Please tick the following declarations:

Usage (optional)	
☐	I give permission for this document to be made available to other students as examples of previous work.
Authenticity of work	
☐	I confirm that I have read the University rules in respect of plagiarism and student misconduct.
☐	I understand that if I use work from an external source, I must reference and cite the source in any work that I produce.
Ethics	

☐	I understand that ethical approval will be needed for this project regardless of whether I carry out primary research or not.
☐	I understand that if I do conduct primary research, I must include all raw data and research documentation in my final project report.
☐	I understand that not including the raw data can mean potentially failing the project.
☐	I understand that not obtaining ethical approval is grounds for academic misconduct and possible project failure.
Use of AI Artificial Intelligence (AI), such as ChatGPT, can be used as a tool to help assist and inform the initial development of your work but it is not a replacement for your own critical thinking and analysis. We require that your work is your own original content, demonstrating your knowledge, skills, and critical thinking abilities. To that end, AI-generated content must not be included in <u>any work</u> you submit for assessment unless suitably referenced. Not doing so is a breach of the University's Academic Regulations as outlined in the Student Conduct Policy and constitutes an assessment offence.	
☐	I have read the University AI policy and if relevant, I will acknowledge the use of AI in any work I produce for this project with appropriate referencing.
☐	I understand that not appropriately referencing the use of AI is an assessment offence equal to plagiarism and may result in a penalty which remains on my student record.

Name: Alghaliah Taj A AIAbdulgohsin

Date: 11/11/2025

Appendix A: Gantt chart

[Draft gantt chart](#)

Appendix B: Ethics Certificate



Certificate of Ethics Review

Project title: Agentic OSINT-LLM Pipeline for Automated Scam-Website Detection with Integrated Ethical & Legal Engineering Controls

Name:	AlGhaliah AlAbdulmohsin	User ID:	UP21157 89	Application date:	26/11/2025 16:18:38	ER Number:	TETHIC-2025-112405
--------------	----------------------------	-----------------	---------------	--------------------------	------------------------	-------------------	--------------------

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative(s) for the **School of Computing** is/are [Elisavet Andrikopoulou, Kirsten Smith](#)

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- [University Policy](#)
- [Safety on Geological Fieldwork](#)

It is also your responsibility to follow University guidance on Data Protection Policy:

- [General guidance for all data protection issues](#)
- [University Data Protection Policy](#)

Which school/department do you belong to?: **School of Computing**

What is your primary role at the University?: **Undergraduate Student**

What is the name of the member of staff who is responsible for supervising your project?: **Oluwatobi Fajana**

Is the study likely to involve human subjects (observation) or participants?: No

Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?: No

Are there risks of significant damage to physical and/or ecological environmental features?: No

Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)? No

Does the project involve animals in any way?: No

Could the research outputs potentially be harmful to third parties?: No

Could your research/artefact be adapted and be misused?: No

Will your project or project deliverables be relevant to defence, the military, police or other security organisations and/or in addition, could it be used by others to threaten UK security?: No

Please read and confirm that you agree with the following statements: I confirm that I have considered the implications for data collection and use, taking into consideration legal requirements (UK GDPR, Data Protection Act 2018 etc.), I confirm that I have considered the impact of this work and and taken any reasonable action to mitigate potential misuse of the project outputs, I confirm that I will act ethically and honestly throughout this project.

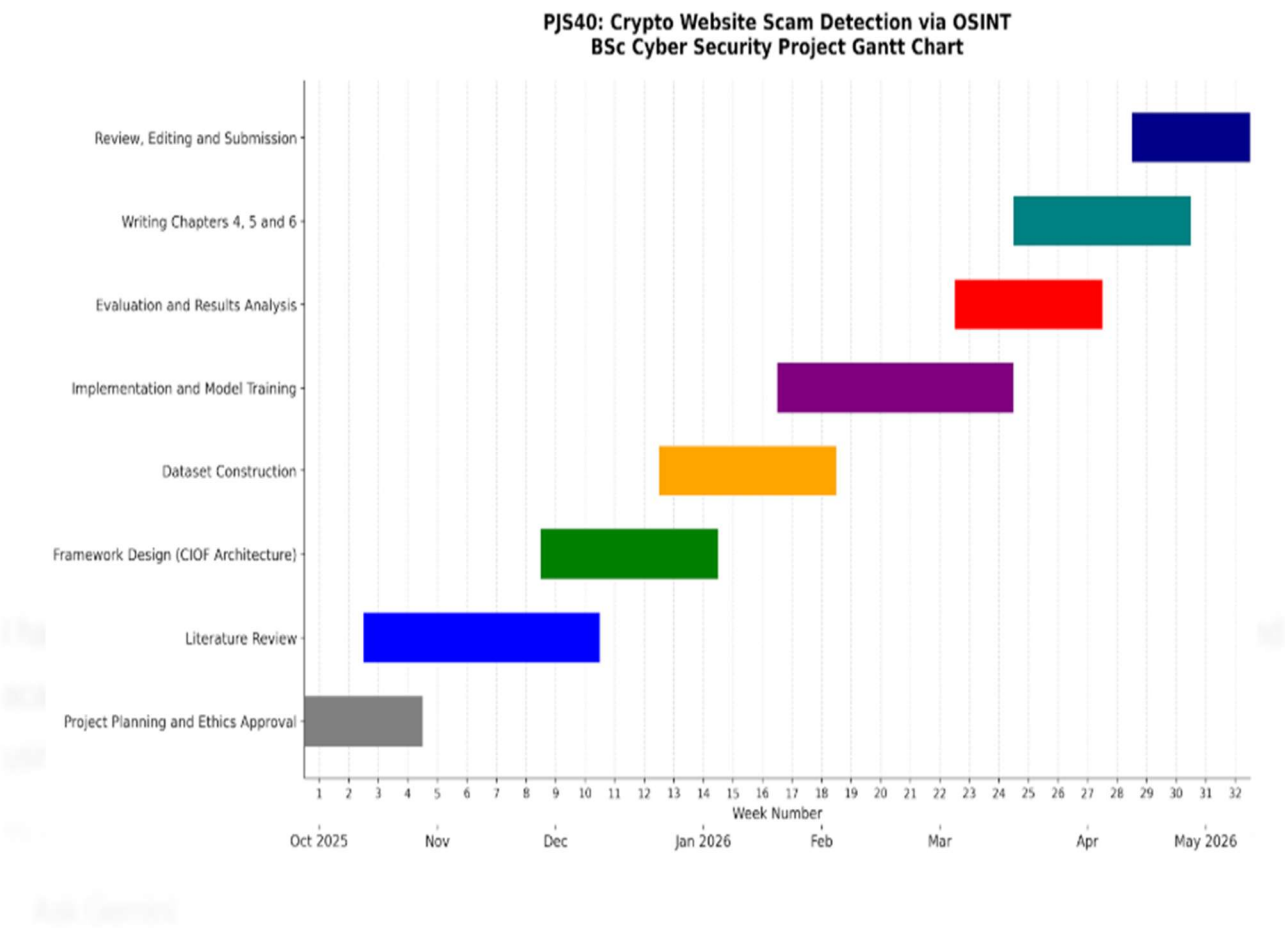
Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy.

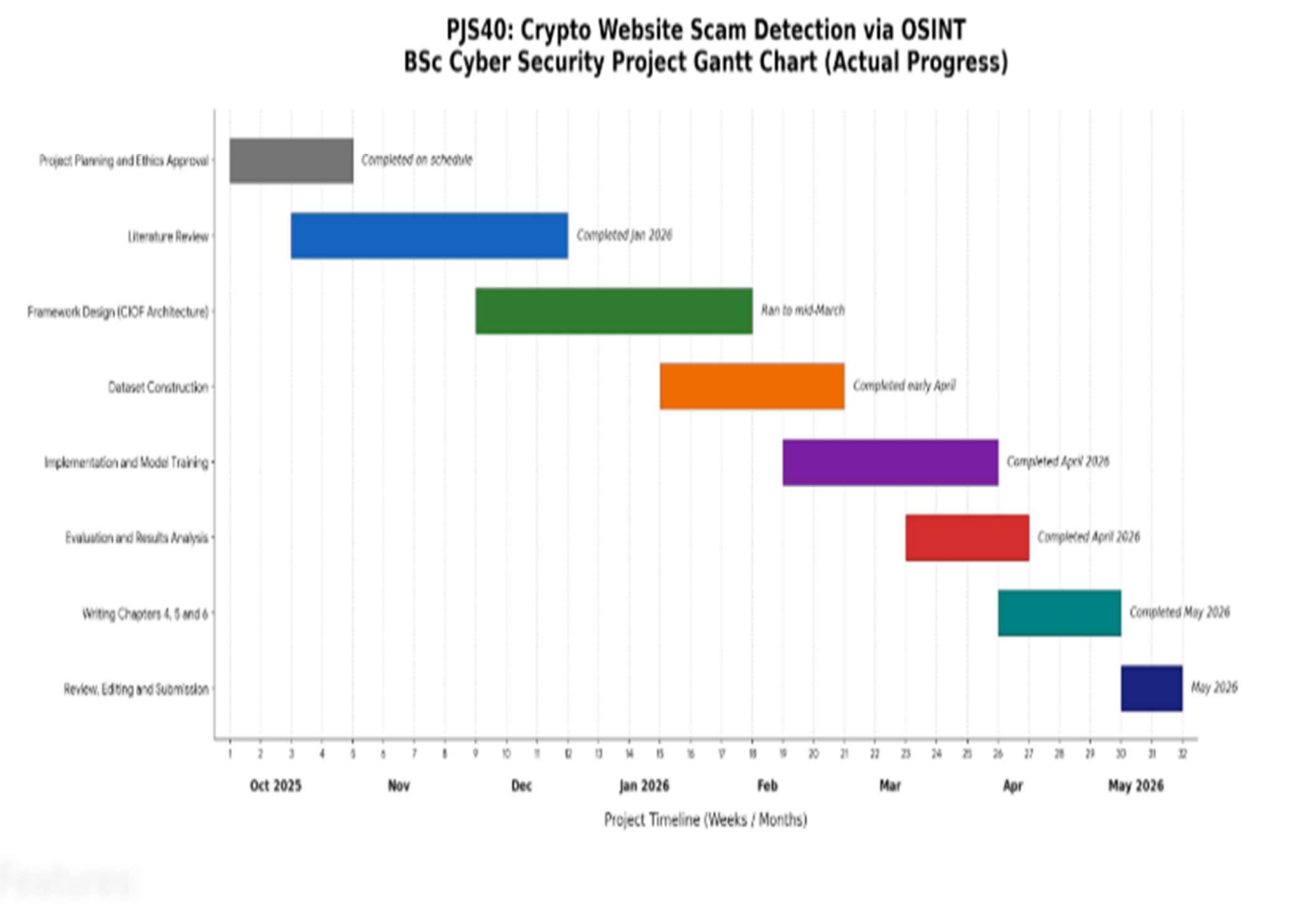
Supervisor comments:

Supervisor's Digital Signature: **oluwatobi.fajana@port.ac.uk** Date: **27/11/2025**

Appendix C: Gantt Chart



Appendix D: Actual Gantt Chart



Appendix E: Jupyter Notebook Source Code

In [1]:

```
!pip install pandas scikit-learn python-whois requests beautifulsoup4 matplotlib se

Requirement already satisfied: pandas in c:\users\algha\anaconda3\lib\site-packages
(2.2.3)
Requirement already satisfied: scikit-learn in c:\users\algha\anaconda3\lib\site-pa
ckages (1.6.1)
Collecting python-whois
  Using cached python_whois-0.9.6-py3-none-any.whl.metadata (3.0 kB)
Requirement already satisfied: requests in c:\users\algha\anaconda3\lib\site-packag
es (2.32.3)
Requirement already satisfied: beautifulsoup4 in c:\users\algha\anaconda3\lib\site-
packages (4.12.3)
Requirement already satisfied: matplotlib in c:\users\algha\anaconda3\lib\site-pack
ages (3.10.0)
Requirement already satisfied: seaborn in c:\users\algha\anaconda3\lib\site-package
s (0.13.2)
Requirement already satisfied: numpy>=1.26.0 in c:\users\algha\anaconda3\lib\site-p
ackages (from pandas) (2.1.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\algha\anaconda3\l
ib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\algha\anaconda3\lib\site-pa
ckages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\algha\anaconda3\lib\site-
packages (from pandas) (2025.2)
Requirement already satisfied: scipy>=1.6.0 in c:\users\algha\anaconda3\lib\site-pa
ckages (from scikit-learn) (1.15.3)
Requirement already satisfied: joblib>=1.2.0 in c:\users\algha\anaconda3\lib\site-p
ackages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\algha\anaconda3\lib
\site-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\algha\anaconda3
\lib\site-packages (from requests) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\algha\anaconda3\lib\site-pa
ckages (from requests) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\algha\anaconda3\lib\s
ite-packages (from requests) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\algha\anaconda3\lib\s
ite-packages (from requests) (2025.10.5)
Requirement already satisfied: soupsieve>1.2 in c:\users\algha\anaconda3\lib\site-p
```

In [4]:

```
import pandas as pd

df = pd.read_csv('dataset.csv')
print(df.shape)
print(df['label'].value_counts())
print(df.head())
```

(875, 2)

label

1 501

0 374

Name: count, dtype: int64

	url	label
0	Vintarioinvest.com	1
1	Tiomarkets.eu	1
2	Okrex.com	1
3	Tasttytrade.com	1
4	Nevixen.com	1

In [5]:

```
import re
import requests
from bs4 import BeautifulSoup
import whois
from datetime import datetime
import time

# Wallet address patterns
BTC_PATTERN = re.compile(r'\b(bc1|[13])[a-zA-HJ-NP-Z0-9]{25,62}\b')
ETH_PATTERN = re.compile(r'\b0x[a-fA-F0-9]{40}\b')

# Suspicious keywords
SUSPICIOUS_KEYWORDS = [
    'guaranteed profit', 'guaranteed return', 'double your',
    'get rich', 'risk free', '1000%', 'send crypto',
    'airdrop', 'claim your', 'free bitcoin', 'seed phrase', 'private key'
]

print("Libraries loaded successfully.")
```

Libraries loaded successfully.

In [6]:

```
# function to get the domain age using whois lookup
def get_domain_age(domain):
    try:
        w = whois.whois(domain)
        creation = w.creation_date
        if isinstance(creation, list):
            creation = creation[0]
        if creation:
            age = (datetime.now() - creation).days
            return age
    except:
        pass
    return -1 # couldnt retrieve it, just return -1

# function to grab the raw html content of a site
def get_site_content(url):
    headers = {'User-Agent': 'Mozilla/5.0'}
    try:
        r = requests.get('https://' + url, headers=headers, timeout=8)
        return r.text
    except:
        try:
            r = requests.get('http://' + url, headers=headers, timeout=8)
            return r.text
        except:
            return ""

# main function that pulls all features for one url
def extract_features(url):
    features = {}
    domain = url.replace('http://', '').replace('https://', '').split('/')[0]

    # Layer 1 - domain and infrastructure info
    age = get_domain_age(domain)
    features['domain_age_days'] = age
    features['domain_age_under_30'] = 1 if (age >= 0 and age < 30) else 0
    features['domain_age_under_90'] = 1 if (age >= 0 and age < 90) else 0
    features['url_length'] = len(url)
    features['has_numbers_in_domain'] = 1 if re.search(r'\d', domain) else 0
    features['hyphen_count'] = domain.count('-')
    features['dot_count'] = domain.count('.')

```

```

features['suspicious_tld'] = 1 if domain.endswith(
    ('.xyz', '.top', '.club', '.review', '.wiki',
     '.online', '.live', '.info', '.cc', '.vip')) else 0

# layer 2 - whats actually on the site
content = get_site_content(url)
if content:
    features['site_reachable'] = 1
    soup = BeautifulSoup(content, 'html.parser')
    text = soup.get_text().lower()
    features['btc_wallet_found'] = 1 if BTC_PATTERN.search(content) else 0
    features['eth_wallet_found'] = 1 if ETH_PATTERN.search(content) else 0
    keyword_hits = sum(1 for kw in SUSPICIOUS_KEYWORDS if kw in text)
    features['suspicious_keyword_count'] = keyword_hits
    features['has_suspicious_keywords'] = 1 if keyword_hits > 0 else 0
    features['has_privacy_policy'] = 1 if 'privacy policy' in text else 0
    features['has_terms'] = 1 if 'terms of service' in text or 'terms and condi
else:
    features['site_reachable'] = 0
    features['btc_wallet_found'] = 0
    features['eth_wallet_found'] = 0
    features['suspicious_keyword_count'] = 0
    features['has_suspicious_keywords'] = 0
    features['has_privacy_policy'] = 0
    features['has_terms'] = 0

time.sleep(1)
return features

print("Functions defined successfully.")

```

Functions defined successfully.

In [7]:

```
results = []

for i, row in df.iterrows():
    print(f"Processing {i+1}/{len(df)}: {row['url']}")
    features = extract_features(row['url'])
    features['url'] = row['url']
    features['label'] = row['label']
    results.append(features)

    # save progress every 50 urls just in case it crashes
    if (i + 1) % 50 == 0:
        pd.DataFrame(results).to_csv('features_progress.csv', index=False)
        print(f"--- progress saved at {i+1} urls ---")

# save the final completed file
features_df = pd.DataFrame(results)
features_df.to_csv('features_final.csv', index=False)
print("DONE. All features saved to features_final.csv")

Processing 1/875: Vintarioinvest.com
Processing 2/875: Tiomarkets.eu
Processing 3/875: Okrex.com
Processing 4/875: Tasttytrade.com
Processing 5/875: Nevixen.com
Processing 6/875: Coinmatr.com
Processing 7/875: Ascendexe.com
Processing 8/875: Giantwhale.com
Processing 9/875: CryptovaultNC.com
Processing 10/875: Fundedbullfx.com
Processing 11/875: Bullontrade.com
Processing 12/875: Konmines.co
Processing 13/875: Auro-dex.com
Error trying to connect to socket: closing socket - timed out
Processing 14/875: Pxbt.com
Processing 15/875: Veltris-capital.com
Processing 16/875: Capex.com
Processing 17/875: Egscapltd.com
Processing 18/875: Moonbitx.com
Error trying to connect to socket: closing socket - timed out
Processing 19/875: Alphapepe.io
Processing 20/875: Ecmarkets.com
Processing 21/875: Engicapital.com
```

```

Processing 152/875: nmyetlerwallet.com
Processing 153/875: eth-drop.com
Processing 154/875: nmyetlerwallet.com
Processing 155/875: origirprotocol.com
Processing 156/875: classicetherwallt.com
Processing 157/875: eth-airdrop.live
Processing 158/875: get-ether.fund
Processing 159/875: ico-ton.org
Processing 160/875: gether-gift.com
Processing 161/875: poloniex-withdrawals.bitballoon.com
Processing 162/875: ethpromoverge.com
Processing 163/875: dav-network.io
Processing 164/875: offpromoeth.com
Processing 165/875: bonus.eth-gifts.com
Processing 166/875: shapeskift.io
Processing 167/875: shapeshift.io
Processing 168/875: get5000eth.com
Processing 169/875: ethgiveaway.org
Processing 170/875: ethgiveaway.live
Processing 171/875: eth-givingaway.com
Processing 172/875: bonus.ether-gift.com
Error trying to connect to socket: closing socket - [WinError 10054] An existing co
nnection was forcibly closed by the remote host
Processing 173/875: ethereum.org-giveaway.live
Processing 174/875: getyoureth.com
Processing 175/875: nmyetlerwallet.com
Processing 176/875: get-ethereum.cc
Processing 177/875: freecoiners.com
Processing 178/875: clo-airdrop.info
Processing 179/875: myetherwallet-ht.com
Processing 180/875: myetherwallet0.com
Processing 181/875: polomeix.co
Processing 182/875: poloneix.co
Processing 183/875: hitlbt.com
Processing 184/875: hittbte.com
Processing 185/875: myetherwallet-xn1.com
Processing 186/875: pyapay.co
Processing 187/875: daostack-token.io
Processing 188/875: gift5000-eth.com
Processing 189/875: quarkchain-token.io
Processing 190/875: bonus.ethersgift.com
Processing 191/875: tron-network.org

```

```
Processing 837/875: fameex.com
Processing 838/875: tapbit.com
Processing 839/875: coinw.com
Processing 840/875: deepcoin.com
Processing 841/875: btcc.com
Processing 842/875: bitmart.com
Processing 843/875: digifinex.com
Processing 844/875: ascendex.com
Processing 845/875: probit.com
Processing 846/875: coinstore.com
Processing 847/875: toobit.com
Processing 848/875: p2pb2b.com
Processing 849/875: latoken.com
Processing 850/875: azbit.com
--- progress saved at 850 urls ---
Processing 851/875: coinsbit.io
Processing 852/875: dex-trade.com
Processing 853/875: vindax.com
Processing 854/875: coincheck.com
Processing 855/875: coins.ph
Processing 856/875: buda.com
Processing 857/875: cryptology.com
Processing 858/875: coinmama.com
Processing 859/875: paybis.com
Processing 860/875: bitpanda.com
Processing 861/875: Trustee.plus
Processing 862/875: spectrocoin.com
Processing 863/875: now-wallet.io
Processing 864/875: holytransaction.com
Processing 865/875: litevault.net
Processing 866/875: brd.com
Processing 867/875: indiesquare.me
Processing 868/875: jaxx.io
Processing 869/875: heliowallet.com
Processing 870/875: unstoppable.money
Processing 871/875: ether.li
Processing 872/875: etherwall.org
Processing 873/875: omniwallet.org
Processing 874/875: coinpayments.net
Processing 875/875: mycrypto.com
DONE. All features saved to features_final.csv
```

In [8]:

```
# Load the extracted features and check everything looks right
features_df = pd.read_csv('features_final.csv')
print(f"Total URLs processed: {len(features_df)}")
print(f"Sites that were reachable: {features_df['site_reachable'].sum()}")
print(f"Sites that were offline: {(features_df['site_reachable'] == 0).sum()}")
print(f"\nScam sites (1): {(features_df['label'] == 1).sum()}")
print(f"Legit sites (0): {(features_df['label'] == 0).sum()}")
```

Total URLs processed: 875

Sites that were reachable: 430

Sites that were offline: 445

Scam sites (1): 501

Legit sites (0): 374

In [10]:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import (classification_report, confusion_matrix,
                             accuracy_score, precision_score,
                             recall_score, f1_score)

import matplotlib.pyplot as plt
import seaborn as sns

# fill any missing values with 0 and define my features
features_df = features_df.fillna(0)

feature_columns = [
    'domain_age_days', 'domain_age_under_30', 'domain_age_under_90',
    'url_length', 'has_numbers_in_domain', 'hyphen_count', 'dot_count',
    'suspicious_tld', 'site_reachable', 'btc_wallet_found', 'eth_wallet_found',
    'suspicious_keyword_count', 'has_suspicious_keywords',
    'has_privacy_policy', 'has_terms'
]

X = features_df[feature_columns]
y = features_df['label']

# 80% train, 20% test - standard split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

```
# train the model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

print("Model trained successfully.")
print(f"Training set size: {len(X_train)}")
print(f"Test set size: {len(X_test)}")
```

Model trained successfully.
 Training set size: 700
 Test set size: 175

In [12]:

```
# get my predictions and print all the performance metrics
y_pred = model.predict(X_test)

print("=== CIOF MODEL RESULTS ===\n")
print(f"Accuracy: {accuracy_score(y_test, y_pred):.4f}")
print(f"Precision: {precision_score(y_test, y_pred):.4f}")
print(f"Recall: {recall_score(y_test, y_pred):.4f}")
print(f"F1-Score: {f1_score(y_test, y_pred):.4f}")
print("\nFull breakdown:")
print(classification_report(y_test, y_pred, target_names=['Legitimate', 'Scam']))
```

=== CIOF MODEL RESULTS ===

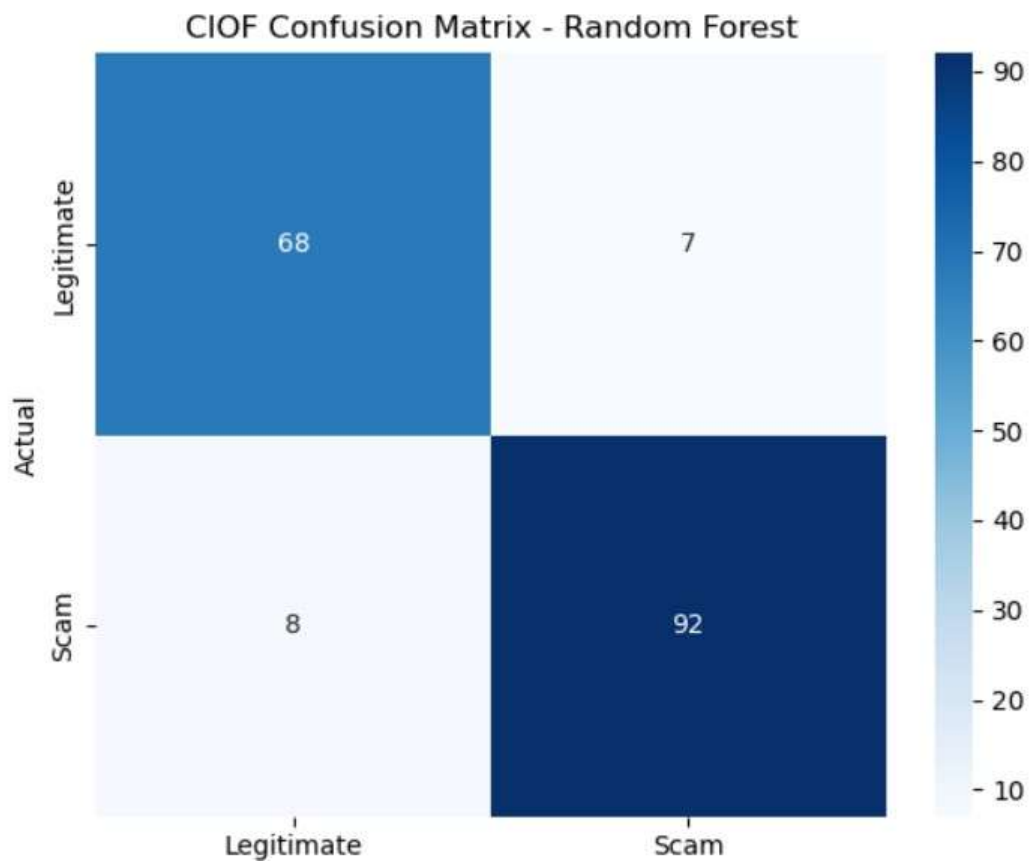
Accuracy: 0.9143
 Precision: 0.9293
 Recall: 0.9200
 F1-Score: 0.9246

Full breakdown:

	precision	recall	f1-score	support
Legitimate	0.89	0.91	0.90	75
Scam	0.93	0.92	0.92	100
accuracy			0.91	175
macro avg	0.91	0.91	0.91	175
weighted avg	0.91	0.91	0.91	175

In [13]:

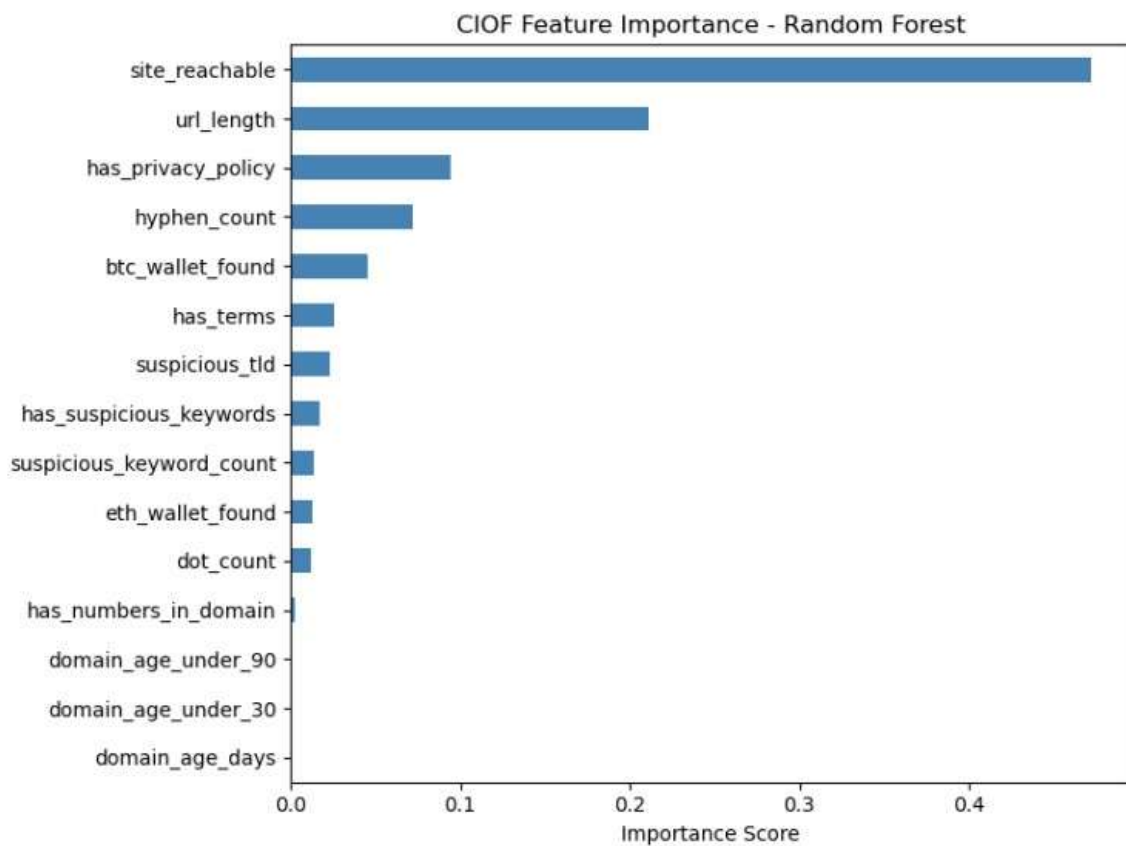
```
# confusion matrix - shows exactly where the model got it right and wrong
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Legitimate', 'Scam'],
            yticklabels=['Legitimate', 'Scam'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('CIOF Confusion Matrix - Random Forest')
plt.tight_layout()
plt.savefig('confusion_matrix.png', dpi=150)
plt.show()
print("Saved.")
```



In [14]:

```
# which features mattered most to the model
importances = pd.Series(model.feature_importances_, index=feature_columns)
importances_sorted = importances.sort_values(ascending=True)

plt.figure(figsize=(8, 6))
importances_sorted.plot(kind='barh', color='steelblue')
plt.title('CIOF Feature Importance - Random Forest')
plt.xlabel('Importance Score')
plt.tight_layout()
plt.savefig('feature_importance.png', dpi=150)
plt.show()
print("Saved.")
```

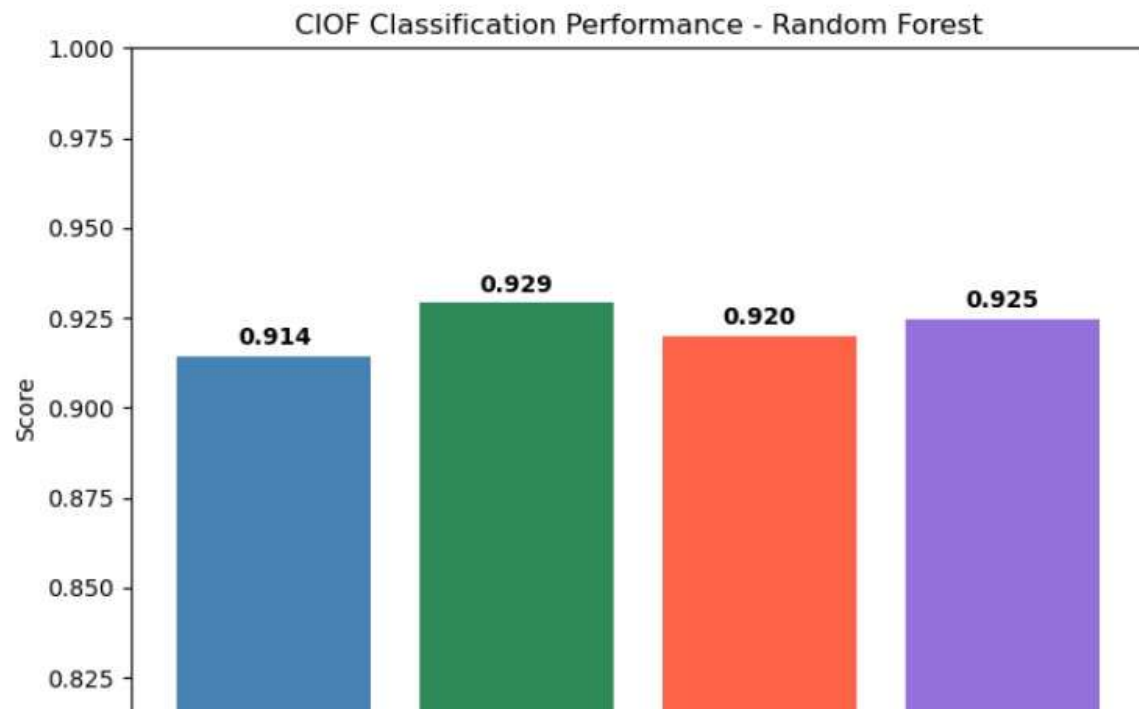


Saved.

In [15]:

```
# bar chart of my model performance metrics
metrics = {
    'Accuracy': accuracy_score(y_test, y_pred),
    'Precision': precision_score(y_test, y_pred),
    'Recall': recall_score(y_test, y_pred),
    'F1-Score': f1_score(y_test, y_pred)
}

plt.figure(figsize=(7, 5))
plt.bar(metrics.keys(), metrics.values(), color=['steelblue', 'seagreen', 'tomato',
plt.ylim(0.8, 1.0)
plt.title('CIOF Classification Performance - Random Forest')
plt.ylabel('Score')
for i, (k, v) in enumerate(metrics.items()):
    plt.text(i, v + 0.003, f'{v:.3f}', ha='center', fontweight='bold')
plt.tight_layout()
plt.savefig('performance_metrics.png', dpi=150)
plt.show()
print("Saved.")
```



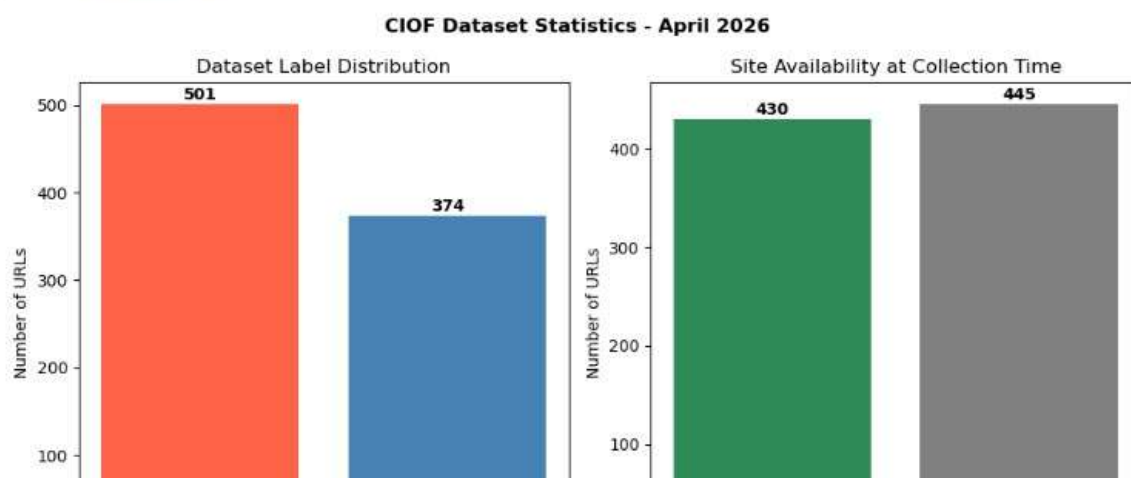
In [16]:

```
# dataset composition chart - shows the breakdown of my dataset
fig, axes = plt.subplots(1, 2, figsize=(10, 5))

# left chart - scam vs legitimate split
axes[0].bar(['Scam (1)', 'Legitimate (0)'],
            [501, 374],
            color=['tomato', 'steelblue'])
axes[0].set_title('Dataset Label Distribution')
axes[0].set_ylabel('Number of URLs')
for i, v in enumerate([501, 374]):
    axes[0].text(i, v + 5, str(v), ha='center', fontweight='bold')

# right chart - reachable vs offline
axes[1].bar(['Reachable', 'Offline'],
            [430, 445],
            color=['seagreen', 'grey'])
axes[1].set_title('Site Availability at Collection Time')
axes[1].set_ylabel('Number of URLs')
for i, v in enumerate([430, 445]):
    axes[1].text(i, v + 5, str(v), ha='center', fontweight='bold')

plt.suptitle('CIOF Dataset Statistics - April 2026', fontweight='bold')
plt.tight_layout()
plt.savefig('dataset_stats.png', dpi=150)
plt.show()
print("Saved.")
```



In [17]:

```
# adding a three-class output - legitimate, suspicious, or scam
# using the model's confidence probability instead of just yes/no

y_proba = model.predict_proba(X_test)[:, 1] # probability of being scam

def classify_three_class(prob):
    if prob >= 0.75:
        return 'Scam'
    elif prob >= 0.40:
        return 'Suspicious'
    else:
        return 'Legitimate'

three_class_predictions = [classify_three_class(p) for p in y_proba]

# count how many fell into each category
from collections import Counter
counts = Counter(three_class_predictions)
print("Three-class classification results:")
print(f"Legitimate: {counts['Legitimate']}")
print(f"Suspicious: {counts['Suspicious']}")
print(f"Scam: {counts['Scam']}")
print(f"Total: {sum(counts.values())}")
```

Three-class classification results:

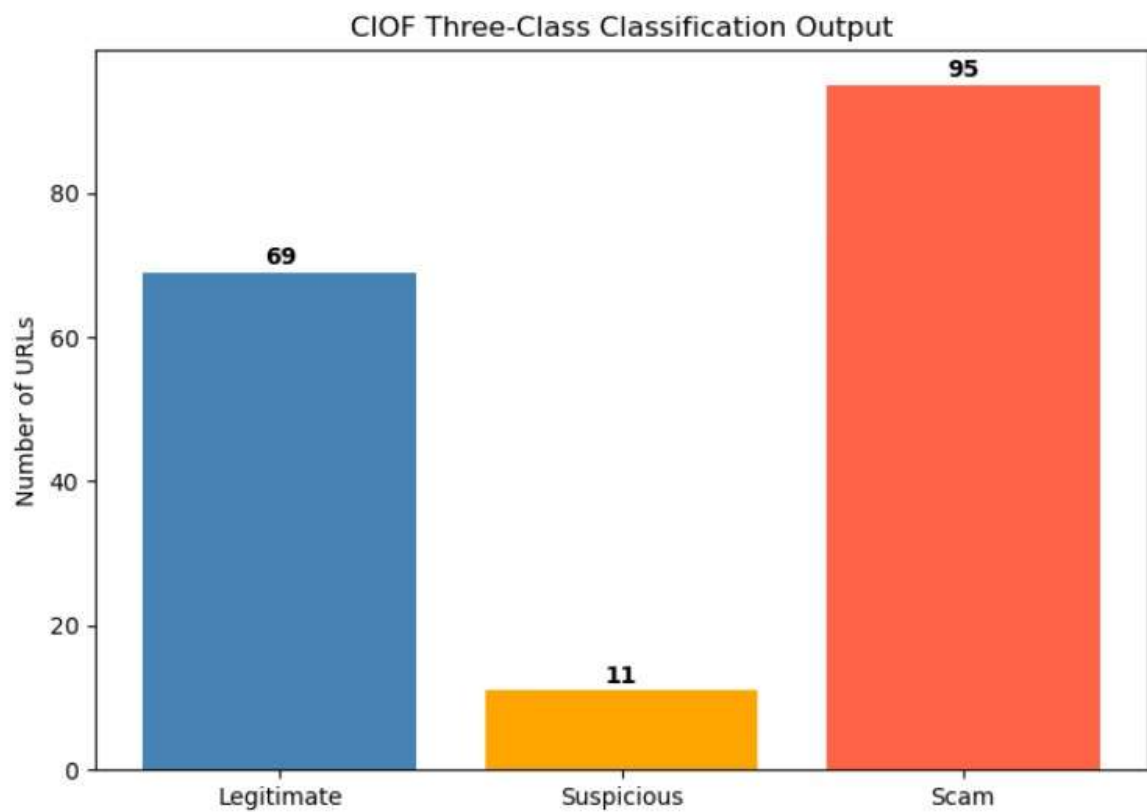
```
Legitimate: 69
Suspicious: 11
Scam: 95
Total: 175
```

Out[17]:

In [18]:

```
# bar chart showing the three-class distribution
categories = ['Legitimate', 'Suspicious', 'Scam']
values = [counts['Legitimate'], counts['Suspicious'], counts['Scam']]
colors = ['steelblue', 'orange', 'tomato']

plt.figure(figsize=(7, 5))
plt.bar(categories, values, color=colors)
plt.title('CIOF Three-Class Classification Output')
plt.ylabel('Number of URLs')
for i, v in enumerate(values):
    plt.text(i, v + 1, str(v), ha='center', fontweight='bold')
plt.tight_layout()
plt.savefig('three_class_output.png', dpi=150)
plt.show()
print("Saved.")
```



In [19]:

```
# find good case study examples from the test set
X_test_copy = X_test.copy()
X_test_copy['url'] = features_df.loc[X_test.index, 'url'].values
X_test_copy['actual_label'] = y_test.values
X_test_copy['predicted_label'] = y_pred
X_test_copy['scam_probability'] = y_proba

# case 1 - clearest scam the model got right (highest scam probability)
clearest_scam = X_test_copy[X_test_copy['actual_label'] == 1].sort_values('scam_pro
print("CASE 1 - Clearest Scam:")
print(clearest_scam[['url', 'domain_age_days', 'btc_wallet_found',
                    'suspicious_keyword_count', 'has_privacy_policy',
                    'site_reachable', 'scam_probability']].to_string())

print("\n")

# case 2 - clearest legitimate site
clearest_legit = X_test_copy[X_test_copy['actual_label'] == 0].sort_values('scam_pr
print("CASE 2 - Clearest Legitimate:")
print(clearest_legit[['url', 'domain_age_days', 'btc_wallet_found',
                    'suspicious_keyword_count', 'has_privacy_policy',
                    'site_reachable', 'scam_probability']].to_string())

print("\n")

# case 3 - one the model got wrong (misclassified)
misclassified = X_test_copy[X_test_copy['actual_label'] != X_test_copy['predicted_l
print("CASE 3 - Misclassified Example:")
print(misclassified[['url', 'domain_age_days', 'btc_wallet_found',
                    'suspicious_keyword_count', 'has_privacy_policy',
                    'site_reachable', 'actual_label', 'predicted_label',
                    'scam_probability']].to_string())
```

CASE 1 - Clearest Scam:

	url	domain_age_days	btc_wallet_found	suspicious_keyword_count	has_privacy_policy	site_reachable	scam_probability
50	kraken-secure-login.com		-1			0	
0		0					1.0

CASE 2 - Clearest Legitimate:

	url	domain_age_days	btc_wallet_found	suspicious_keyword_count	has_privacy_policy	site_reachable	scam_probability
804	coinfirm.com		-1			0	1
1		1					0.0

CASE 3 - Misclassified Example:

	url	domain_age_days	btc_wallet_found	suspicious_keyword_count	has_privacy_policy	site_reachable	actual_label	predicted_label	scam_probability
274	rebellios.com		-1			0			0
0		1							0.458469

In []: