



ExPhish-Net: An Explainable Attention-Based CNN-BiGRU Framework for Real-Time Phishing URL Detection

Research Methods and Professional Development (ACFA7004)
University of Wales Trinity Saint David

Submitted by
MD Imran Biswas
Student ID: 2436882
University of Wales Trinity Saint David

Submission Date: 07/05/2026

Table of Content

1. Introduction.....	3
1.1 Context and Background.....	3
1.2 Research Problem.....	4
1.3 Research Aim and Objectives	4
1.4 Significance of the Research.....	5
2. Literature Review	5
2.1 Traditional Machine Learning Approaches	6
2.2 Deep Learning for Lexical URL Analysis	6
2.3 Explainable AI in Cybersecurity	7
2.4 Research Gaps and Positioning of ExPhish-Net.....	7
3. Research Design and Methodology.....	8
3.1 Research Design.....	8
3.2 Research Methods	8
3.3 Evaluation Metrics	10
3.4 Research Materials	10
4. Expected Outcomes	11
5. Project Timeline	12
6. Presentation and Academic Standards.....	12
References	13

1. Introduction

1.1 Context and Background

Cybersecurity is nowadays one of the most significant fields of the digital age. With the continued growth of internet use, the level of sophistication of cyber threats to individuals, corporations and government institutions is also increasing. One of the most widespread and harmful vectors by which attackers can defraud users and get them to divulge sensitive financial and personal information is phishing attacks. The Anti-Phishing Working Group (APWG) indicates that phishing attack instances have been increasing significantly every year with hundreds of thousands of distinct phishing sites being identified every month around the world.

The main mechanism of phishing attacks is to generate misleading Uniform Resource Locators (URLs) which are more or less similar to the genuine websites. Such URLs can be typosquatted with subtle typosquatting patterns, misleading subdomains or obfuscated character sequences that will not be detected by untrained human vision. Attackers have moved away increasingly towards highly targeted spear-phishing attacks targeting individual or organisations in particular, which are more difficult to detect.

In the past, blacklist-based filtering, which involved the listing of known bad URLs and blocking those, was used as the main defence mechanism against phishing URLs. Nevertheless, this method has a major drawback it is purely reactive. A phishing URL can be reported, authenticated and listed on a blacklist when it is too late, as the malicious hacker might have reached out to thousands of victims. It has been verified that the phishing sites are usually online less than 24 hours before being shut down so blacklists are simply not structurally adequate in real-time protection [1].

With the advent of deep learning, new avenues have been developed in automated threat detection. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), especially their variations, including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) have been shown to be remarkably efficient in learning latent representations of sequential and spatial inputs without aiding feature engineering. When applied to URL analysis, the architectures are able to obtain not only local character-level patterns (as in suspicious token sequences), but also global sequential context (as in the structural flow of a URL) directly out of raw lexical input [2] [10].

One of the most recent developments is the introduction of Bidirectional GRU (BiGRU) layers, which operate in forward and backward directions, which can help to detect contextual dependencies that can be overlooked in unidirectional models. This two-way processing is particularly useful when analyzing URLs, the occurrence of a suspicious token at the end of a URL (e.g. a spoofy top-level domain such as .tk) can be just as informative as the occurrence of the same at the start. Moreover, with the emergence of Explainable Artificial Intelligence (XAI) tools like SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-Agnostic Explanations) have emerged and can be used to convert the decisions made by opaque neural networks into human interpretable explanations, which is essential when used in actual Security Operations Centres (SOCs) [3].

Regardless of these improvements, a noticeable gap in the current literature remains: no existing lightweight, fully lexical model can simultaneously be of high accuracy, process in real-time and have transparent explainability. In this study, a new CNN-BiGRU architecture with an in-built attention mechanism and XAI layer is proposed to fill this gap, named ExPhish-Net.

1.2 Research Problem

The essence of the issue that this study is intended to solve is the fact that the existing cybersecurity mechanisms fail to identify phishing URLs with an impressive degree of accuracy, in real-time, and with a high degree of transparency, using solely lexical analysis. This gap can be characterized by two problem areas that are different yet interrelated.

To start with, the black-box dilemma: the best current deep learning models to classify lexical URLs as such are opaque computational systems. Although other models like LSTM-based models presented by Aslam et al. have the accuracy rate of around 96, they do not provide explanatory output [4]. This obscurity is an essential operation impediment. Security Operations Centre (SOC) analysts are not able to act responsibly on an automated classification that cannot be checked by them as to the logic behind it. Under high stakes conditions, like in a financial institution or a government network, unaccountable alerts may be overlooked, resulting in slow reaction to real threats [8].

Second, the trade-off of latency: the last study that has achieved a successful implementation of XAI in phishing detection has done this by making heavy multimodal implementations. Vulfin et al. employed a system capable of downloading entire HTMLs and taking screenshots of the page to be visually analyzed to attain XAI-enhanced classification [5]. Although efficient, this method adds considerable latencies to calculations. To use a URL filtering pipeline that needs to be run in real-time at the network perimeter, it is completely unrealistic to take the time needed to load a complete web page and obtain a screenshot. This constitutes the difference in computing ability.

Combined, these two gaps constitute a research issue that is acute and unsolved: currently, there is no system that would attain transparent and real-time phishing URL detection solely based on raw lexical data. ExPhish-Net is developed with a specific purpose to address this issue by enabling the rapid processing speed of lexical processing with the explainability of XAI, without the multimodal processing latency.

1.3 Research Aim and Objectives

The study purpose is to design, develop, and test ExPhish-Net a lightweight and interpretable attention-based CNN-BiGRU deep learning framework that can detect phishing URLs in real-time via raw lexical analysis, and explain its classification decisions in a transparent manner.

To operationalise this aim, the following SMART (Specific, Measurable, Achievable, Relevant, Time-bound) research objectives have been developed:

1. **Dataset Construction:** Within the first month of the project, create a balanced binary classification dataset by taking the verified phishing URLs in the PhishTank

database and the legitimate URLs in the Majestic Million top domains list and construct a dataset of approximately 112,000 URLs, with a 50/50 class distribution.

2. **Architecture Design and Development:** To design and train a hybrid CNN-BiGRU neural network with an attention mechanism to learn local spatial (typosquatting patterns, suspicious token arrangements) and global sequential context in raw URL character sequences. The choice of architecture will be explained by the comparative analysis of the other models (LSTM, CNN-only, BiGRU-only) as the benchmarks.
3. **Performance Evaluation:** In order to compare ExPhish-Net with predetermined baselines in terms of standard classification measures, such as Accuracy, Precision, Recall, and F1-Score. Special attention will be given to reducing the false negatives (missed phishing URLs), the worst outcome in a real-world deployment situation.
4. **Real-Time Capability Assessment:** To assess the mean inference latency of ExPhish-Net on average on an URL classification, to prove that the model can be applied in practice in real-time deployment conditions at network security perimeters with the latency constraints.
5. **Explainability Integration:** To incorporate post-hoc XAI tools, i.e., SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations), and confirm that the attention mechanism in the model concentrates on linguistically significant and suspicious URL tokens, thus, ensuring that the explanations are not only technically correct but also practical to SOC analysts.

1.4 Significance of the Research

This study is about a meeting point of real world and theoretical imperatives. Practically speaking, most data breaches around the world are due to phishing. In the 2024 Verizon Data Breach Investigations Report, phishing was revealed as a top initial access method by all industry sectors. A rapid and transparent system directly responds to the functional needs of modern SOC settings, where the analyst must deal with thousands of alerts daily, and needs both fast and reliable automated triage [9].

Theoretically, ExPhish-Net is related to the increasing body of research on interpretable deep learning in cybersecurity. The given attention mechanism, which is an implicit interpretability layer, will be an architectural improvement over post-hoc-only explainability methods, placing the model in the balance between performance and trust. The research also provides a significant, balanced, publicly reported benchmark data that can be used to ensure reproducibility and future comparative studies in the field.

2. Literature Review

This part examines the literature available regarding phishing URL detection, covering the development of classic machine learning models over the last few years to the recent deep learning and explainable AI models. The review is structured thematically and addresses three major themes, namely, conventional feature-based methods, deep learning-based lexical analysis, and the incorporation of explainability. The scope includes peer-reviewed articles

mostly dating back to 2024-2026 with foundational articles as needed. The reason is to determine the most significant contributions, research limitations, and gaps that drive the present proposal.

2.1 Traditional Machine Learning Approaches

The initial automated phishing detection systems used manually-crafted feature sets based on URL strings, the content of HTML pages, and WHOIS information. The inputs to classical classifiers like Support Vector Machines (SVM), Random Forests, Decision Trees, and k-Nearest Neighbour algorithms were features like the length of URLs, the inclusion of IP addresses in hostnames, number of special characters, and the use of URL shorteners. These systems were able to obtain a reasonable level of accuracy in controlled laboratory experiments. According to Mahajan and Siddavatam, a set of Random Forest and SVM based classifiers being trained on a well-purged feature set yielded 97.14 percent accuracy on the PhishTank dataset [6]. In the same manner, Ghaleb et al. have shown that a hybrid ensemble model of Random Forest and Multilayer Perceptron (MLP) can reach an accuracy of 96.80 percent on a set of malicious URLs [7].

In spite of these findings, the main weakness of these methods is that they are based on hand-crafted features. The workload of feature engineering is very high, and it needs in-depth understanding of the domain and periodically updated to keep up with changing attacker strategies. Contemporary phishing attacks strategically deploy URLs to circumvent known feature-based heuristics, i.e. feature-engineered models are weak to attacks as adversary tactics change. This fragility led to the shift towards end-to-end deep learning models that are capable of automatically learning discriminative representations of raw data.

2.2 Deep Learning for Lexical URL Analysis

Classification of phishing URLs by deep learning has increased significantly since 2020. The major strength of deep learning models is that it learns hierarchical feature representations of raw URL character sequences without manual feature engineering. The most popular input representation has become character-level tokenisation, where every character in a URL is encoded as an integer token, as it accommodates the more subtle orthographic manipulations (e.g., replacing the letter l with the number 1, or adding hyphens to brand names) that word-level tokenisation could not capture.

Aslam et al. suggested AntiPhishStack, a stacked generalisation model of phishing URL detection based on LSTM and reached an accuracy of 96.04% on the PhishTank dataset [4]. Although this work did show that end-to-end learning on raw URL text was a viable model, the model architecture did not offer any explainability mechanism or provide an assessment of inference latency, leaving both the interpretability issue and the real-time deployment issue unaddressed. Nayak et al. have presented a detailed comparison of machine learning and deep learning models to detect phishing, and they have verified that the deep learning models tend to be better than classical models in case of big datasets, especially on adversarially perturbed URLs [1].

CNN architectures have also been explored to URL classification, and 1D convolutional filters have been used to exploit local character-level n-gram patterns of typosquatting and brand

impersonation. CNN layers (to extract local features) combined with recurrent layers (GRU or LSTM) (to model the sequential context) have been suggested as a natural architecture to URL analysis, due to the dual nature of URL structure: local suspicious tokens in a globally meaningful sequence. Alzboon et al. investigated the hybrid deep learning as an approach in this scenario, which demonstrates competitive results, but also notes that the lack of explainability is a major limitation in real practice [2].

Table 1: Comparative Summary of Related Works in Phishing URL Detection

Author & Year	Domain & Dataset	Models Applied	Accuracy	Explainability (XAI)?
Mahajan et al. (2025) [6]	URL (PhishTank)	Random Forest, SVM	97.14%	No
Ghaleb et al. (2024) [7]	URL (Malicious URLs)	RF + MLP (Hybrid)	96.80%	No
Aslam et al. (2024) [4]	URL (PhishTank)	LSTM (Deep Learning)	96.04%	No
Vulfin et al. (2026) [5]	Multimodal (URL+HTML+Image)	CatBoost+CNN+CodeBERT	98.90%	Yes (SHAP, Grad-CAM)

2.3 Explainable AI in Cybersecurity

Since 2023, the concept of explainability integrating deep learning-based cybersecurity systems has been gaining more research interest. Shafin suggested a explainable feature selection method of web phishing detection, using SHAP values to determine the most discriminative input features to a gradient-boosted classifier [3]. Although this work showed the practical utility of the XAI in cybersecurity, it was still in the classical feature engineering framework and it did not discuss the deep learning-based lexical analysis.

The most closely related recent research is the study of Vulfin et al., who designed a multimodal phishing web site detection system, which uses XAI technologies such as SHAP and Grad-CAM [5]. This system was reported to have the highest accuracy of 98.90% in the current literature. Nevertheless, it is based on multimodal inputs (URL text, downloaded HTML content, and page screenshots) which creates significant computational overhead. Real-time deployment is not possible because of the need to create a complete web page and obtain a screenshot per classification decision. This latency problem was admitted by Vulfin et al. as a shortcoming though they did not offer any solution and this represents an evident and definite gap in the literature.

2.4 Research Gaps and Positioning of ExPhish-Net

The review of the existing literature shows that there are three gaps that are interconnected. To begin with, none of the existing, light-weight, lexical-only models can be both highly classified and capable of real-time inference with built-in explainability. Second, the attention process of hybrid CNN-RNN models of URL analysis has not been critically assessed as a layer of intrinsic interpretability, a la post-hoc XAI tools. Third, operational validity of XAI

explanations to SOC analysts, i.e., are SHAP/LIME explanations identifying those potentially suspicious URL tokens, has not been carefully assessed within the phishing URL domain. ExPhish-Net is placed in a position of filling the three gaps. It requires only raw lexical input and therefore is computationally efficient enough to run in real-time. It offers an intrinsic interpretability layer along with post-hoc XAI tools by adding a self-attention mechanism as an architectural component. And by directly comparing XAI output to the ground-truth suspicious token, it adds new data on the operational viability of explainability techniques in the cybersecurity context.

3. Research Design and Methodology

3.1 Research Design

This study follows a Design Science Research (DSR) paradigm that best fits a study that proposes to design, develop and test a new technical artefact that is meant to address a real practical problem. Formalised by Hevner et al., DSR assumes that knowledge in research is created in the process of building up and critical assessment of artefacts, in this instance, the ExPhish-Net model architecture and dataset. This is different to an entirely empirical research on social science that mainly aims at describing or explaining the existing phenomena. The reason why DSR is selected is that the dual contribution of this research is that the model itself is an artefact, and its systematic review is the research contribution to knowledge.

In the DSR model, the study is conducted in a quantitative experimental approach. Quantitative methods suit the purpose of the research since the research goals are stated in terms of quantifiable performance metrics (accuracy, precision, recall, F1-score, inference latency) that need statistical test and comparative benchmarking. The experimental plan is to train and test various competing model architectures on a controlled dataset, thus making it possible to make strict causal inferences regarding the relationship between the architectural decisions and the classification performance.

3.2 Research Methods

The experimental methodology follows five consecutive stages which have a direct connection with a concrete research purpose.

Phase 1 - Data Construction: A balanced set of about 112,000 URL samples will be built by merging two publicly available sources. The PhishTank database, a community-driven industry-standard phishing URL repository, will be used to get around 56,000 verified phishing URLs. The Majestic Million dataset will provide approximately 56,000 legitimate URLs, which is a popular list of one million of the most popular domains on the internet based on referring subnets. The equal class balance (50/50 split in phishing and legitimate) is intentionally added to avoid the bias of the model to the larger class, and to make reported accuracy measures reflect true discriminative capacity, as opposed to trivial predictions of the majority class. Stratified sampling will be used to divide the dataset into 70 training, 15 validation, and 15 test sets to maintain the proportion of classes in each split.

The selection of the PhishTank and Majestic Million is explained by a number of factors. The two datasets are publicly available, which makes it possible to reproduce the results. PhishTank URLs are checked by the community and the chance of label noise due to unverified phishing reports is minimized. The Majestic Million offers a realistic allocation of legitimate web traffic as opposed to artificially curated benign URLs, which would not correspond to the distribution of domains that a deployed system would see in reality. The total size of 112,000 URLs is large enough to train a deep learning model to convergence without overfitting, which is in line with the sample sizes in similar recent studies [4], [6].

Phase 2 Data Preprocessing: Character-level tokenisation of the raw URL strings will be done whereby each character (letters, digits, special characters like hyphens, dots, forward slashes, at signs etc.) is assigned a distinct integer value. This is more desirable than word-level tokenisation since it uses much of the fine-grained orthographic structure of URLs, allowing manipulations of finer scope, including homoglyph substitution and intentional misspellings. Any sequence of tokens will be padded or truncated to a constant maximum length of 200 characters which is long enough to represent all but the most extreme subsets of URLs in the data and is fast to compute.

Phase 3 Architecture Design: The ExPhish-Net architecture will be built up of three sequential layers: (i) a 1D-CNN architecture with two convolutional layers, ReLU activation functions and max-pooling, which detects local character-level features based on the suspicious token pattern; (ii) a Bidirectional GRU (BiGRU) architecture, which takes the CNN feature maps through a forward and backward The attention layer output is inputted into a fully connected dense layer with sigmoid activation to do binary classification. Such architecture is explained by the complementary capabilities of its parts: CNNs are good at identifying local spatial features, BiGRUs are better at identifying long-range sequential features, and the attention mechanism is an inherent interpretability layer because it reveals which parts of the URL contributed to the classification.

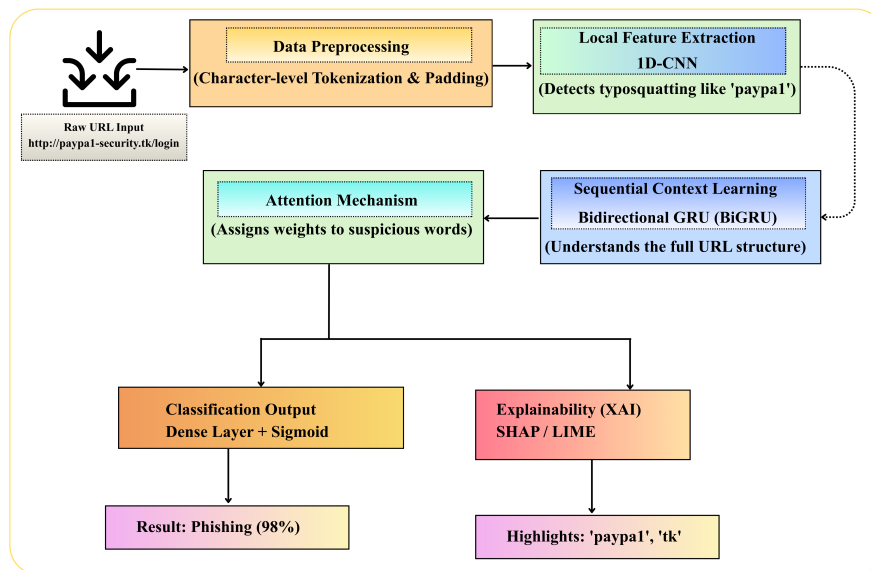


Figure 1: Proposed ExPhish-Net architecture for real-time phishing URL detection

Phase 4 -Baseline Comparison: To critically evaluate value of the proposed architecture ExPhish-Net will be compared to four baseline models which were trained on the same dataset on the same experimental settings: (i) classical Random Forest classifier using manually engineered URL features, (ii) pure CNN model, (iii) pure BiGRU model, and (iv) CNN-LSTM hybrid. This comparative analysis will allow directly attributing the differences in the performance to the proposed innovations in architecture.

Phase 5 - Explainability Integration and Validation: Once the model is trained, post-hoc will be explained on 1,000 randomly selected test URLs using SHAP and LIME. SHAP DeepExplainer will compute character-token scores of feature importances, i.e., which input positions influenced each classification decision the most. The local decision boundary of the model about one prediction will be directly estimated by LIME. Explanation validity will be measured by tallying the overlap of the set of tokens identified by SHAP and by LIME with the set of tokens in a manually-labeled set of recognized suspicious URL components (e.g., brand impersonation strings, misleading TLDs), and the value will provide a ground-truth-based measure of explanation fidelity. The gap in the literature review on reliability of XAI operation in detecting phishing directly responds to this analysis stage.

3.3 Evaluation Metrics

There will be four common binary classification measures used to assess model performance: Accuracy (fraction of correctly classified URLs), Precision (fraction of URLs labelled as phishing and which are actually phishing), Recall (fraction of genuine phishing URLs correctly labelled) and F1-Score (harmonic mean of Precision and Recall). Special focus will be put on Recall, since the worst-case scenario in a real-life implementation is a false negative (an undetected phishing URL): a phishing URL will go undetected and lead to a successful attack. Moreover, it will measure average inference latency per URL (in milliseconds) on a standardised hardware setup, to give empirical data of real time capability.

3.4 Research Materials

Python 3.10 will be used as the software environment with Python, and either Tensorflow/Keras or PyTorch as the primary deep learning platform. NumPy and Pandas will be used to manipulate and analyse data. The implementation of the baseline model and computation of the evaluation metric will be done using scikit-learn. Explainability analysis will be done using SHAP and LIME Python libraries. Matplotlib and Seaborn will be used to carry out visualisation. The cloud-based GPU infrastructure will be used to meet hardware requirements, i.e., Google Colab Pro or similar AWS EC2 GPU instances to efficiently train the model and optimise hyperparameters. Git will be used to version-control all the code.

Ethics: The PhishTank and Majestic Million are publicly accessible and they have no personal information. The URLs are not a personally identifiable information according to GDPR or other data protection regulations. This research does not involve any human subjects. There is no active use of phishing infrastructure in the research. In this regard, no serious ethical risks are envisaged, and normal university ethics permission to analyze secondary data is adequate.

4. Expected Outcomes

It is hoped that ExPhish-Net will be at the leading edge of classification performance with or surpassing the highest-quality existing deep learning models in the literature, and also attain inference latency that is sufficiently low to support deployment in real-time. The study does not preset a certain accuracy goal, which will be determined empirically by the means of the controlled experimentation. Nevertheless, due to the performance of architecturally similar models in the associated literature, and the quality of the provided dataset and preprocessing pipeline, it can be expected that the classification performance will be high.

The combination of SHAP and LIME explainability tools should provide operationally significant visualisations which accurately point out suspicious URL tokens, including typosquatted brand names, misleading subdomains, and suspicious top-level domains. Should the attention mechanism weights be identified to be correlated with SHAP importance scores, this will be new evidence that self-attention is a valid intrinsic interpretability mechanism on top of its performance-enhancing role.

The theoretical value of this study will be a proven framework that shows that transparent, real-time phishing URL detection can be performed without the computational cost of multimodal methods. The deliverable will be a deployable, open-source model that can be used to augment existing SOC workflows. The benchmark data will be published openly to enable reproducibility and comparative studies.

The first limitation of ExPhish-Net that can be expected is that it is lexical-based. The model will be insensitive to phishing attacks based solely on visual misrepresentation (e.g., a legitimate-looking compromised domain that only contains malicious code) by simply analysing the URL string. Also, the creation of SHAP explanations will create a small computational cost (marginal overhead) at the post-prediction stage that will be quantified and reported. Future research might take a direction of integrating the lightweight HTML content characteristics to overcome the lexical-only constraint without necessarily adding the entire latency of multimodal solutions.

5. Project Timeline

The implementation of the project will take place between three and seven months between March 2026 and September 2026. The table below is a Gantt-like timeline to outline the key steps and milestones.

Activity	Mar	Apr	May	Jun	Jul	Aug	Sep
Literature Review & Data Collection	•	•					
Dataset Construction & Preprocessing		•	•				
CNN-BiGRU Architecture Design			•	•			
Model Training & Baseline Comparison				•	•		
XAI (SHAP/LIME) Integration & Evaluation					•	•	
Thesis Writing & Submission						•	•

Table 1: Project Gantt chart with milestones in the seven months project (March-September 2026). Working that month is indicated by full cells.

Resource management: The primary computing resource (a cloud-based GPU environment) will either be Google Colab Pro (or AWS EC2) and will be large enough to have sufficient memory in the form of GPUs to train the deep learning models on the 112,000 URL dataset. Software resources are all open-source. The chief researcher is the main human resource. Supervisor consultation will be requested at key milestones: following the construction of the datasets (Month 2), initial model training (Month 4), and XAI evaluation (Month 6).

6. Presentation and Academic Standards

This proposal has been designed based on the ACFA7004 assessment specification covering each of the necessary sections sequentially. The IEEE referencing style is used throughout and in-text citations are numbered in square brackets and a reference list with numbers is added at the end. All the sources used are peer-reviewed journal articles or conference papers published mostly not earlier than the last two years (2024-2026), except foundational works. The similarity index as indicated in the Turnitin similarity report of the preliminary presentation submission was 4% thus attesting to originality of the written work. The proposed study is in line with the ethical guidelines of the University of Wales Trinity Saint David and an ethics form has been filled and submitted through Moodle submission link as a separate document.

References

- [1] G. S. Nayak, B. Muniyal, and M. C. Belavagi, “Enhancing Phishing Detection: A Machine Learning Approach With Feature Selection and Deep Learning Models,” *IEEE Access*, vol. 13, pp. 33308–33320, 2025, doi: <https://doi.org/10.1109/access.2025.3543738>.
- [2] M. S. Alzboon, M. Subhi Al-Batah, M. Alqaraleh, F. Alzboon, and L. Alzboon, “Phishing Website Detection Using Machine Learning,” *Gamification and Augmented Reality*, vol. 3, p. 81, Jan. 2025, doi: <https://doi.org/10.56294/gr202581>.
- [3] Sakib Shahriar Shafin, “An Explainable Feature Selection Framework for Web Phishing Detection with Machine Learning,” *Data Science and Management*, Aug. 2024, doi: <https://doi.org/10.1016/j.dsm.2024.08.004>.
- [4] S. Aslam, H. Aslam, A. Manzoor, H. Chen, and A. Rasool, “AntiPhishStack: LSTM-Based Stacked Generalization Model for Optimized Phishing URL Detection,” *Symmetry*, vol. 16, no. 2, p. 248, Feb. 2024, doi: <https://doi.org/10.3390/sym16020248>.
- [5] Alexey Vulfin, Alexey Sulavko, Vladimir Vasiliev, A. Minko, A. Kirillova, and A. Samotuga, “A Multimodal Phishing Website Detection System Using Explainable Artificial Intelligence Technologies,” *Machine Learning and Knowledge Extraction*, vol. 8, no. 1, pp. 11–11, Jan. 2026, doi: <https://doi.org/10.3390/make8010011>.
- [6] R. Mahajan and I. Siddavatam, “Phishing Website Detection using Machine Learning Algorithms,” *International Journal of Computer Applications*, vol. 181, no. 23, pp. 45–47, Oct. 2018, doi: <https://doi.org/10.5120/ijca2018918026>.
- [7] Dr. L. Hassan and R. A. - Mazrouei, “EVOLVING PARADIGMS AND FUTURE TRAJECTORIES IN CYBER THREAT INTELLIGENCE,” *International Journal of Cyber Threat Intelligence and Secure Networking*, vol. 2, no. 06, pp. 1–7, Jun. 2025, doi: <https://doi.org/10.55640/ijctisn-v02i06-01>.
- [8] Rajesh Kalakoti, Risto Vaarandi, Hayretin Bahşı, and Sven Nömm, “Evaluating Explainable AI for Deep Learning-Based Network Intrusion Detection System Alert Classification,” *11th International Conference on Information Systems Security and Privacy*, pp. 47–58, Jan. 2025, doi: <https://doi.org/10.5220/0013180700003899>.
- [9] N. R. Kotha, “AUTOMATED PHISHING RESPONSE SYSTEMS: ENHANCING CYBERSECURITY THROUGH AUTOMATION,” *International Journal of Computer Engineering and Technology*, vol. 12, no. 2, Aug. 2024, doi: <https://doi.org/10.34218/ijcet.12.02.008>.
- [10] “[PDF] Automated Alert Classification and Triage (AACT): An Intelligent System for the Prioritisation of Cybersecurity Alerts | Semantic Scholar,” *Semanticscholar.org*, 2025, doi: <https://doi.org/10.48550/arXiv.2505.09843>.