

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

Research Methods and Professional Development

Presented by: MD Imran Biswas

University of Wales Trinity Saint David

Contents



- **Research Context and Background**
- **Research Problem**
- **Research Aim and Objectives**
- **A Brief Literature Review and Summary**
- **Research Methodology and Justification**
- **Data Collection and Evaluation Process**
- **Project Planning and Resource Management**
- **Project Impact and Limitations**
- **References**

Research Context and Background



1. **Threat Landscape:** Phishing remains one of the most serious cyber threats on the international level. Hackers are becoming more advanced and tricky in their URLs manipulation strategy to garner users into sharing sensitive financial and personal data [1].
2. **Detection Mechanisms:** Traditional blacklist and manual feature-based systems are becoming ineffective against modern, dynamic phishing attacks. [2]
3. **Deep Learning Advances:** CNNs and RNNs are now used to automatically identify hidden semantic patterns directly from raw URL text without human aid. [1]
4. **Shift to Explainability:** While deep learning is powerful, it remains a "black box." Integrating Explainable AI (XAI) is critical for transparent and credible threat detection. [3]

Research Problem



The Identified Problem: Based on the literature gap, the core problem in current cybersecurity systems is the inability to perform transparent, highly accurate, and real-time phishing URL detection using raw lexical analysis.

Key Problem Areas:

1.The "Black-Box" Dilemma: Highly accurate deep learning models for lexical URL analysis currently operate opaquely. Without explainability, Security Operations Center (SOC) analysts cannot trust or verify the automated decisions [2].

2.The Latency Trade-off: Attempting to solve the transparency issue with current XAI frameworks requires heavy multimodal architectures (rendering full web pages and taking screenshots) [4]. This computational overhead makes them entirely impractical for the rapid filtering of incoming URLs.

Research Aim and Objectives



Research Aim: It is proposed to design, develop and test ExPhish-Net, a **lightweight** and **interpretable** attention-based CNN-BiGRU deep learning model that can be applied to the real time detection and with high accuracy of phishing URLs in raw lexical analysis.

Research Objectives (SMART) :

Data Formulation: To create a balanced dataset of 112, 000 URLs by combining PhishTank with Majestic Million datasets.

Architecture Design: To adopt a hybrid CNN-BiGRU based architecture with an Attention mechanism in order to identify local and sequential context of URLs.

Performance Evaluation: In order to measure the model effectiveness in terms of standard measures with the intention of accuracy of above 97%.

Explainability Integration: To incorporate XAI tools (SHAP and LIME) to give clear and understandable explanations of the lexical characteristics that lead the model to make its decisions.

A Brief Literature Review and Summary



Recent Advancements & Limitations:

1. **Traditional Machine Learning (2024-2025):** Studies by Mahajan et al. [1] achieved high accuracy (97.14%) using Random Forest and SVM, but they heavily rely on time-consuming manual feature engineering.
2. **Deep Learning (2024-2025):** Researchers like Aslam et al. [3] successfully applied deep learning (LSTM) for raw text analysis, completely eliminating manual feature extraction. However, these models act as "black boxes" with zero explainability.
- 3.**Explainable AI (2026):** The most recent work by Vulfin et al. [4] successfully integrated XAI (SHAP, Grad-CAM). However, they used a heavy multimodal approach (downloading HTML + Images), which introduces significant latency.

Literature Summary & Gap Analysis:

Author & Year	Domain & Dataset	Models Applied	Accuracy	Explainability (XAI)?
Mahajan et al. (2025) [1]	URL (PhishTank)	Random Forest, SVM	97.14%	No
Ghaleb et al. (2024) [2]	URL (Malicious URLs)	RF + MLP (Hybrid)	96.80%	No
Aslam et al. (2024) [3]	URL (PhishTank)	LSTM (Deep Learning)	96.04%	No
Vulfin et al. (2026) [4]	Multimodal (URL+HTML+Image)	CatBoost+CNN+CodeBERT	98.90%	Yes (SHAP, Grad-CAM)
ExPhish-Net (Proposed)	URL (PhishTank + Majestic)	CNN-BiGRU + Attention	Target: >97%	Yes (SHAP/LIME)

Research Methodology and Justification

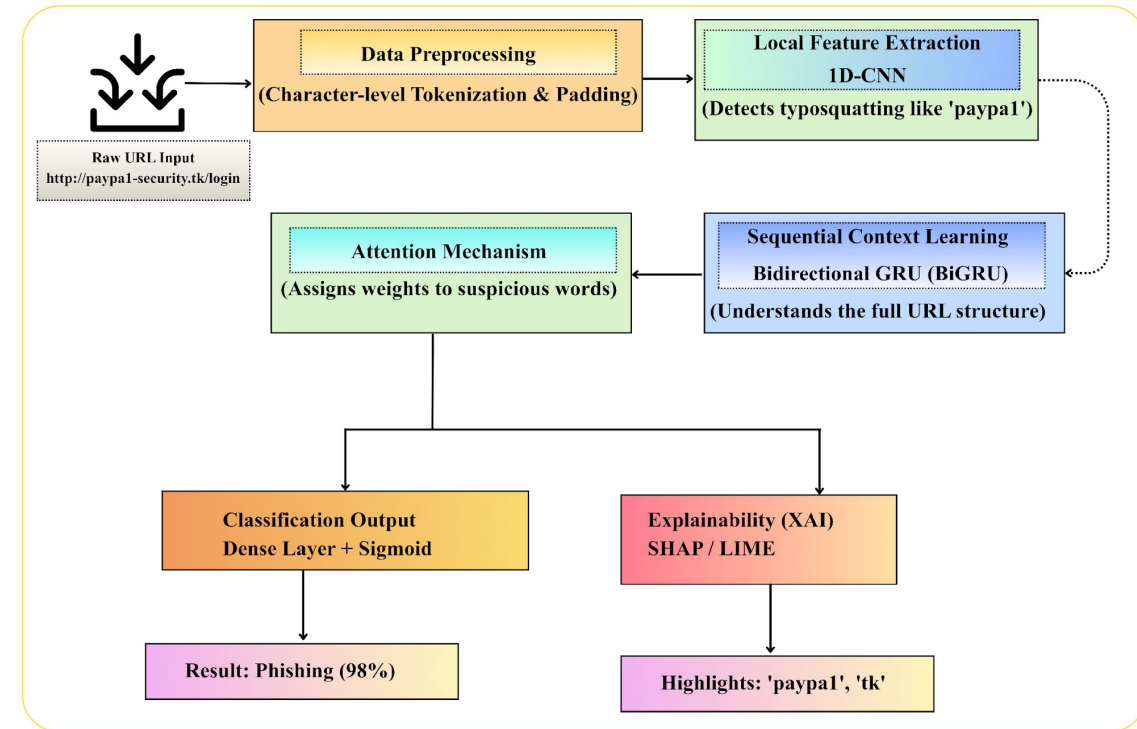


Strategy: Raw lexical analysis using quantitative and experimental deep learning.

ExPhish-Net Architecture:

- **1D-CNN:** Finds local spatial characteristics (detects typosquatting).
- **BiGRU:** Trains two way sequencing of the URL.
- **Attention:** Weights malicious tokens for better explainability.

Justification: No manual feature engineering; lightweight to do real-time detection.



Data Collection and Evaluation Process



Dataset Formulation:

1. **Data Sources:** The dataset will combine approximately 56,000 verified phishing URLs from the PhishTank database and 56,000 legitimate URLs from the Majestic Million top domains list.
2. **Class Balance:** An exact 50/50 split will be maintained to prevent model bias toward the majority class.
3. **Data Splitting:** The final dataset of 112,000 URLs will be divided into 70% Training, 15% Validation, and 15% Testing sets.

Data Preparation:

1. **Tokenization:** URLs will not use standard word embeddings. Instead, character-level tokenization will be applied to capture raw structural anomalies.
2. **Padding:** All URL sequences will be padded or truncated to a fixed uniform length for neural network processing.

Evaluation and Validation:

1. **Performance Metrics:** The framework will be evaluated using standard metrics: Accuracy, Precision, F1-Score, and a specific focus on Recall to strictly minimize false negatives (missed phishing attacks).
2. **XAI Validation:** Post-hoc explanation tools like SHAP and LIME will be tested on the validation set to visually confirm that the model's Attention mechanism is focusing on the correct malicious tokens.

Project Planning and Resource Management



GANTT CHART

Activity	March	April	May	June	July	August	Sept
Literature Review & Data Collection							
Preprocessing & Tokenization							
Preprocessing & Pretraining							
CNN-BiGRU Architecture Design							
Model Training & Tuning							
XAI (SHAP/LIME) Integration							
Documentation & Submission							

Resource Requirements:

- Hardware:** Cloud-based GPU infrastructure (e.g., Google Colab Pro or AWS) for faster deep learning model training.
- Software:** Python 3, TensorFlow/Keras or PyTorch framework, Scikit-learn, and SHAP/LIME libraries.
- Data Sources:** Open-access PhishTank API and Majestic Million CSV datasets.

Project Impact and Limitations



Potential Impact:

1. **Real-Time SOC Support:** Provides Security Operations Center (SOC) analysts with a high-speed, automated URL filtering system that does not suffer from multimodal latency.
2. **Building Trust:** By using XAI, it translates opaque deep learning decisions into human-readable lexical triggers, bridging the gap between AI accuracy and human trust.

Project Limitations:

1. **Lexical Dependency:** Because the model strictly analyzes raw URL text for speed, it might miss highly sophisticated phishing attacks that rely entirely on visual deception (like fake logos) on legitimate-looking compromised domains.
2. **Computational Overhead of XAI:** While much faster than rendering HTML or taking screenshots, generating SHAP/LIME explanations still adds a slight processing overhead during the post-prediction phase.

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, 2025.
- [2] M. S. Alzboon et al., "Phishing Website Detection Using Machine Learning," Gamification and Augmented Reality, vol. 3, p. 81, 2025.
- [3] S. S. Shafin, "An explainable feature selection framework for web phishing detection with machine learning," Data Science and Management, vol. 8, pp. 127-136, 2025.
- [4] R. Mahajan and I. Siddavatam, "Phishing Website Detection using Machine Learning Algorithms," Int. J. Comput. Appl., vol. 181, pp. 45-47, 2025.
- [5] F. A. Ghaleb et al., "Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning," Sensors, vol. 22, 2024.
- [6] S. Aslam et al., "AntiPhishStack: LSTM-Based Stacked Generalization Model for Optimized Phishing URL Detection," Symmetry, vol. 16, 2024.
- [7] A. Vulfin et al., "A Multimodal Phishing Website Detection System Using Explainable Artificial Intelligence Technologies," Mach. Learn. Knowl. Extr., vol. 8, 2026.

Any Question



