The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering and Mathematics (EPSTEM), 2025

Volume 37, Pages 887-902

ICEAT 2025: International Conference on Engineering and Advanced Technology

PhisNet: Intelligent Detection of Phishing

Mehedi Hasan
Noakhali Science and Technology University

Nazmun Nahar
Noakhali Science and Technology University

Md. Hasan Imam
Noakhali Science and Technology University

Mayeen Uddin Khandaker
Sunway University
Korea University

Abstract: Phishing and fraud attacks continue to be common in the cybersecurity environment, as criminals use URLs and email messages. Here we conduct a side-by-side evaluation of two transformer-based machine learning techniques to identify phishing. BERT-LSTM model that focuses on spotting email phishing. Combined RoBERTa and Attension model that aims to detect URL phishing. With email phishing detection, the proposed BERT-BiLSTM model with an attention mechanism achieved 98.7% accuracy by efficiently utilizing linguistic metadata and structural properties of emails that extract and combine discrim- inative content from emails and focus on key details required to complete the classification. So, for detecting the URL, the hybrid- RoBERTa model was achieved 93% accuracy. On the other hand, confirming our hypothesis that semantic patterns in URLs are crucial to detection. Furthermore, it should be recognized that transformer models outperformed all traditional machine learning models in every domain, exhibiting incredible recall superiority for advanced phishing strategies. Further analysis of feature importance indicated URL entropy and email sentiment features as the prominent discriminators. These results lay down the foundation for layered active systems to thwart phishing attacks by guiding the implementation of RoBERTa hybrids for web traffic filtering and a motion-controlled BERT-LSTM operation.

Keywords: Phishing detection, Natural language processing, Transformer models, Machine learning, BERT, RoBERTa.

Introduction

Although there are many phishing detection systems on the market, most popular or commercial types are not powerful enough because they depend on manual development, do not detect new ways URLs are disguised and rely completely on text, lacking the ability to spot suspicious qualities in network addresses. These email phishing detection tools make their checks inside email content but do not include any contextual aspects such as the sender or subject. Most of these models disregard any language information in the DOI or any hints about phishes contained on the website. Overall, it's tricky to create a system that deals with both formatted (e.g. emails plus data about them) and unformatted (e.g. URLs and their content) data quickly and accurately to avoid missing most phishing attacks.

With phishing, cyber attackers send messages or set up websites that look real to get a person's private information. Most times, criminals use fake emails and websites to deceive others. Phishing attacks have gone up over the past five years and according to APWG (APWG, 2023), they have affected the records of numerous global users. Solutions based on rules and blacklists can't respond to new dangers and sometimes report many false positives (Marchal, 2014). Since many of these models are not fully automated, they can still miss small differences in language and in emails and URLs (Sahoo,2020). Better meaning can be understood for emails and URLs thanks to new NLP models and transformers BERT and RoBERTa (Devlin, 2019), (Liu,2019). Both email and URL phishing detection are addressed in the paper, using PhishNet based on BERT, BiLSTM and attention as well as RoBERTa with attention. With the framework, both the semantic content and the metadata collaborate to help avoid errors, be useful against fresh types of attacks and achieve more accurate outcomes. The objectives for the research are as follows:

- Developing a system that automatically finds phishing attempts over email and through URLs using hybrid deep learning.
- To incorporate BERT, Bidirectional LSTM and attention into phishing email detection.
- Integrating RoBERTa-based attention models and features of words and metadata data for better URL detection.
- Making sure the system is available on email software and browser extensions for real use.
- For comparing the proposed methods to established ones (such as XGBoost, Gradient boosting) to assess and compare them.

Related Work

The reason phishing is such a common problem now is mostly because of how much more time people spend online. Detection of phishing emails is presently a key research topic. This paper 'Building an Intelligent Phishing Email Detection System Using Machine Learning and Feature Engineering'(Chinta, 2025), combines recent techniques like reinforcement learning, CS- SVM to keep strong performance against threats, also offering success against never-seen attacks. But many approaches cannot handle data that contains noise and end up ignoring stop words and punctuation, decreasing accuracy. On the other hand, Phishing Email Detection Model Using Deep Learning'(Atawneh,2023), the most likely reason LSTM crosses the 99.61% accuracy mark is that deep learning methods like LSTM, CNN, and RNN outperform traditional techniques. Even so, having less data for training can make the system perform poorly against new attacks. Even deep learning systems are often unable to detect new phishing methods. Their 'Deep Enough? On the Effectiveness of Deep Learning in Phishing Email Detection'(Champa, 2024) results look promising on the training data at 99.85%, but poorer when encountering different data sets, due to overfitting and fake-looking emails. Out of these models, RoBERTa reaches 99.43% accuracy, which is superior to the results from BERT and DistilBERT. Even though transformers perform better, they take up more computation and can cause difficulties with older systems due to tokenization (Mele´ndez, 2024). A new system is introduced that detects phishing threats in real time using DistilBERT. This achieves nearly full accuracy with fast execution. Staying updated works well against current risks, though ensuring quickness, correctness and computer power is a challenge (Damatie,2024).

Just as email threats do, URL phishing remains a major concern because attackers keep finding new tactics to get around security measures. Machine learning methods like Random Forests, Decision Trees and SVMs are applied by the authors (Ahammad,2022) to study URLs and detect phishing websites. Detecting phishing is also made possible by CANTINA through its use of domain connections, HTML properties and page standards. Small or unknown websites are typically overlooked by these methods because it's challenging to extract their features. NLP and seven classifiers are used in a new system (Sahingoz ,2019) to instantly detect phishing in a 73,000-URL dataset and Random Forest reaches an accuracy of 97.98%. Since the system relies only on types of features, its ability to work with multiple languages is limited by its dataset. A different approach (Ozcan,2023) uses combinations of character embeddings and features extracted through NLP to make detection more accurate when facing noisy data. In a simple technique (Haynes,2021), BERT and ELECTRA are used to search for phishing URLs on mobile, yet the outcomes are better when the website's contents are added to the scan. It is found in research (Otieno, 2023) that BERT shows high accuracy, yet fake titles, shorteners and IP-based URLs suggest that phishing detection should use a wider range of strategies than only URL features. These articles as a whole point towards the development from standard ML to deep learning and transformer models, reflecting improved performance in handling advanced language-based phishing techniques.

Methodology

This section outlines the methodology adopted for developing the PhishNet framework, which detects phishing in both email messages and URLs. It covers the datasets used, preprocessing techniques, feature engineering, model development, and implementation details.

Datasets

Two primary datasets were employed in this research: one for phishing email detection and another for phishing URL detection. Both datasets were sourced from publicly available open-source repositories on Kaggle and were selected based on their comprehensiveness and relevance to real-world phishing scenarios.

Email Dataset

The email dataset used in this study is a merged and curated collection titled Email_Dataset.csv, comprising approximately 50,000 email records. This dataset was constructed by combining multiple open-source datasets.available on Kaggle, ensuring a diverse and representative sample of email types.

Each email entry includes the following fields:

- *Sender:* The email address or identifier of the sender.
- *Receiver:* The intended recipient of the email.
- *Subject:* The subject line of the email.
- *Body:* The main textual content of the email.
- *Label:* A binary label indicating whether the email is *phishing* (1) or *legitimate* (0).

The dataset includes both text-rich emails and minimal content emails, allowing for a robust evaluation of phishing detection techniques under varied linguistic structures. To maintain consistency, all data entries were cleaned and normalized during preprocessing.

URL Dataset

The second dataset utilized is the phishing_site_urls.csv file, which contains approximately 549,000 URL entries. This dataset was also obtained from Kaggle and provides a broad spectrum of both malicious and benign URLs.

Each record in the dataset contains:

- *URL:* The full web address submitted for classification.
- *Label:* A categorical label, either *good* (benign) or *bad* (phishing).

This dataset covers a wide variety of URL structures, including those with suspicious patterns (e.g., IP-based URLs, excessive subdomains, misspellings, or obfuscation techniques) as well as legitimate websites. The presence of both lexical and contextual URL features provides a strong foundation for building and evaluating a phishing URL detection model.

Dataset Statistics: A summary of the datasets is pro- vided in Table I.

Table 1. Summary of datasets used

Dataset	Total Records	Phishing	Legitimate
Email Dataset	~50,000	28,457	21,403
URL Dataset	~549,000	156,422	392,924

Both datasets were randomly shuffled and stratified during the train-test split to ensure balanced class distributions and reduce sampling bias during model training and evaluation.

Preprocessing

Separate preprocessing pipelines were applied to emails and URLs:

Email Preprocessing

Converted text to lowercase and removed HTML tags, special characters, and digits to reduce noise in email content.

Applied BERT tokenizer to segment and map words to token IDs, preserving contextual semantics.

Removed common English stopwords to eliminate non-informative words and reduce input dimensionality.

Performed lemmatization using WordNet to normalize words to their base forms (e.g., "running" → "run"), aiding in better generalization.

Encoded sender and receiver email addresses as categorical variables using label encoding to capture identity-based patterns.

Extracted sentiment scores from the email subject and body using the TextBlob library to quantify emotional tone, where polarity ranges from -1 (negative) to +1 (positive).

Computed keyword-based binary flags by checking for common phishing keywords (e.g., "verify", "ac- count", "urgent", "click") in both the subject and body. These flags serve as handcrafted indicators of phishing intent.

Generated TF-IDF (Term Frequency-Inverse Document Frequency) vectors on unigrams and bigrams from subject and body texts to numerically represent word importance. The maximum number of features was limited to 100 for each.

Scaled all numerical features including TF-IDF vectors, sentiment scores, and keyword flags using Stan-dardScaler to normalize feature ranges and stabilize learning.

URL Preprocessing

Converted all URLs to lowercase to ensure uniform text representation and reduce redundancy caused by case variations.

Inserted spaces around special tokens such as http, https, www, /, =, ., &, -, _, and ? to improve token boundary recognition and facilitate effective tokenization by the language model.

Applied RoBERTa tokenizer to the cleaned URLs using Byte-Pair Encoding (BPE), generating input_ids and attention_mask with max_length set to 256 and padding enabled. This preserved subword-level semantics of URL components.

Generated character-level n-grams (ranging from 3 to 5 characters) using a CountVectorizer, capturing frequent substrings and lexical patterns indicative of phishing behavior. The number of features was capped at 300.

Engineered handcrafted lexical and structural features, including:

∗ URL length, domain length, and path length.
∗ Number of subdomains and URL depth.
∗ Frequency of digits and special characters (e.g., /, =, -, .).
∗ Binary flags for presence of phishing-related key- words such as "login", "verify", "secure", "ac- count", and "update".

∗ Shannon entropy of the domain to quantify randomness and detect obfuscation in domain names.
∗ Top-Level Domain (TLD) flags for common domains such as .com, .net, .edu, and .gov.

Encoded all numerical and binary features into a fixed-size vector for integration with other model inputs. Balanced the dataset by selecting an equal number of phishing and legitimate URLs (up to 40,000 samples per class) to mitigate class imbalance and enhance generalization.

Model Building

In this research, we designed and implemented three different architectures targeting phishing detection in both email and URL data. These models incorporate transformer-based embeddings, sequence modeling, attention mechanisms, and traditional machine learning approaches. The goal is to lever- age both contextual and structural patterns for robust phishing detection.

BERT + BiLSTM + Attention for Email Phishing: This architecture is tailored for classifying phishing emails based on multiple components: email body, subject, sender, and receiver. The textual fields are preprocessed (lower- cased, tokenized, lemmatized, and cleaned) and then passed into a pre-trained BERT-base-uncased tokenizer to generate embeddings. These contextualized embeddings are fed into a Bidirectional Long Short-Term Memory (BiLSTM) layer, which captures forward and backward dependencies in text. An attention mechanism follows to highlight phishing- indicative tokens such as "verify your account", "urgent action", or "click now".

The output of the attention layer is then concatenated with metadata features (sender and receiver encodings), and passed into a fully connected dense layer with sigmoid activation for binary classification:

$$\hat{y} = \sigma\,(W \cdot [\mathrm{att}_{\mathrm{body}}; \mathrm{att}_{\mathrm{subject}}; \mathrm{sender}; \mathrm{receiver}] + b) \qquad (1)$$

Here, σ denotes the sigmoid activation function, and $[\cdot;\cdot]$ represents concatenation. $\mathrm{att}_{\mathrm{body}}$ and $\mathrm{att}_{\mathrm{subject}}$ are the attention weighted outputs from BiLSTM layers applied on body and subject embeddings. The model is trained using binary cross- entropy loss and optimized with the Adam optimizer.

RoBERTa + Attention for URL Phishing Detection: For phishing URLs, contextual patterns often include misleading subdomains, obfuscated paths, or suspicious keyword placement. To effectively model these, we use the RoBERTa-base transformer followed by an attention layer. URLs are tokenized using RoBERTa's tokenizer and passed through the transformer to produce hidden state representations.

An attention mechanism is applied on the transformer outputs to focus on critical tokens. The output is then flattened and passed through a dense layer with softmax activation for final classification:

$$\hat{y} = \mathrm{softmax}(W \cdot \mathrm{Attention}(H) + b) \qquad (2)$$

where H represents the contextual hidden states obtained from RoBERTa, and the attention function emphasizes URL components such as misleading keywords (e.g., "secure", "update", "login") and structural patterns.

This model is trained using sparse categorical cross-entropy and evaluated using standard metrics such as accuracy, pre- cision, recall, and F1-score. This design ensures the system can address multiple phishing vectors with high accuracy and adaptability, making it suitable for real-time applications such as phishing-aware email clients and secure web browsers.

Algorithm

- **BERT + BiLSTM + Attention:** Used for email content. BERT captures deep semantic context, BiLSTM models sequential dependencies, and the attention mechanism highlights phishing-indicative terms.

```
ALGORITHM PhishingEmailDetector:
INPUT: email_dataset(sender, receiver, subject, body)

STEP 1: Preprocess
  FOR each email:
    clean_text = lowercase → remove_symbols → remove_stopwords → lemmatize
    sentiment = TextBlob(clean_text).polarity
    keywords = count_phishing_terms(clean_text)
    tfidf = vectorize(clean_text)
    bert_tokens = tokenize(clean_text, max_len=128)
STEP 2: Feature Engineering
  traditional_features = [tfidf + sentiment + keywords]
  deep_features = [bert_tokens]
  metadata = [encode(sender), encode(receiver)]
STEP 3: Neural Network
  body_repr = BERT(body_tokens) → BiLSTM → Attention → Pool
  subject_repr = BERT(subject_tokens) → BiLSTM → Attention → Pool
  meta_repr = Dense(traditional_features + metadata)
  combined = Concatenate(body_repr, subject_repr, meta_repr)
  prediction = Sigmoid(Dense(combined))
STEP 4: Training
  SPLIT data(80% train, 20% test)
  OPTIMIZE using Adam(lr=2e-5) for 14 epochs

OUTPUT: phishing_probability ∈ [0,1]
```

Figure 1. BERT + BiLSTM + Attention algorithm

- **RoBERTa + Attention:** Designed to handle contextual patterns in obfuscated URLs. The attention layer identifies token importance within the RoBERTa representation.

```
ALGORITHM PhishingURLDetection

BEGIN
  // Data Preparation
  LOAD URLs with labels
  MAP 'bad'→1, 'good'→0
  BALANCE classes equally
  SPLIT train/test (80/20)
  // Feature Engineering
  FOR each URL:
    url_features ← EXTRACT(length, chars, suspicious_words, TLD, entropy)
    text_tokens ← ROBERTA_TOKENIZE(clean_url)
    char_ngrams ← EXTRACT_NGRAMS(3-5 chars)
  END FOR
  NORMALIZE(url_features)
  // Model Architecture
  CREATE model:
    roberta_branch ← ROBERTA + MULTI_HEAD_ATTENTION
    url_branch ← DENSE_LAYERS(url_features)
    ngram_branch ← DENSE_LAYERS(char_ngrams)

    combined ← CONCATENATE(roberta_branch, url_branch, ngram_branch)
    output ← FUSION_LAYERS(combined) → SOFTMAX(2_classes)
  // Training
  COMPILE(optimizer=Adam, loss=CrossEntropy)
  TRAIN(epochs=10, callbacks=[EarlyStopping, ReduceLR])
  // Evaluation
  predictions ← PREDICT(test_data)
  CALCULATE(accuracy, F1, ROC-AUC)
  SAVE_MODEL()
END
```

Figure 2. RoBERTa + Attention algorithm

Implementation Details

The proposed phishing detection framework was implemented using **Python 3.11** with a hybrid approach combining traditional machine learning and deep learning to handle both structured and unstructured data. Separate models were developed for email and URL phishing detection.

Programming Environment & Tools: The implementation was carried out on **Google Colab Pro** with GPU support (**NVIDIA Tesla T4**), enabling efficient training of transformer- based models.

Deep Learning Libraries: TensorFlow 2.x was used to build custom neural architectures. **Hugging Face Transformers** provided pre-trained models (bert-base-uncased, roberta-base) and support for LSTM and attention mechanisms.

Machine Learning Libraries: Scikit-learn handled pre- processing, feature selection, and evaluation using metrics like precision, recall, and F1-score.

Data Processing: Pandas and **NumPy** were used for data wrangling, encoding sender/receiver info, parsing URLs, and managing missing values. **BeautifulSoup** was used to clean HTML from email content.

Text Feature Engineering: NLTK was used for stopword removal, tokenization, and lemmatization. **TF-IDF** and n- gram features were generated, along with custom features like entropy and obfuscation patterns.

Visualization: Matplotlib and **Seaborn** were used to visualize class distributions, confusion matrices, and ROC curves for performance analysis.

Model Architecture: The **email model** used a **BERT + BiLSTM + Attention** architecture with metadata fusion. The **URL model** used a **RoBERTa + Multi-head Attention** setup to capture obfuscated URL patterns.

Hyperparameter Settings

Model tuning was performed using grid search and experi- mental validation. The final hyperparameters were:

Table 2. Hyperparameter settings for BERT+BiLSTM

Component	Details
	Model Architecture
BERT Model	bert-base-uncased (pre-trained)
BiLSTM	64 units (128 with bidirectional),
Dense Layers	return_sequences=True
	Sender: 16, Receiver: 16, Metadata: 64,
	Output: 1 (Sigmoid)
	Text Processing
BERT Input	Max Length: 128, Padding: max_length,
TF-IDF	Truncation: True
	Max Features: 100, N-gram Range: (1, 2)
	Training Settings
Learning Rate	2e-5
Optimizer Loss Function Batch Size	Adam
Epochs	Binary Crossentropy 16
Validation Split Train/Test Split Random	14
State	0.1 (10%)
	80/20
	42

Table 3. Hyperparameter settings for RoBERTa

Component	Details
	Model Architecture
RoBERTa	roberta-base (fine-tuned),
Multi-head Attention Dense Layers	Dropout: 0.3
	8 heads, key_dim=64

	URL: 128 → BatchNorm → Dropout(0.3), N-grams: 128 → BatchNorm → Dropout(0.3), Text: 256 → BatchNorm → Dropout(0.3), Fusion: 128 → 64 (with BatchNorm & Dropout)
Data Processing	
Max Sequence Length	256
Character N-grams URL Features Dataset Balancing	Length: 3–5, Max Features: 300 38 structural + 30+ keyword-based + entropy Max 40k samples per class
Training Settings	
Optimizer	Adam (LR: 2e-5, decay to 1e-6)
Loss Function Batch Size Epochs	Sparse Categorical Crossentropy 32 10 (Early stopping: patience=3)
Callbacks	
Early Stopping	Monitor: val_accuracy, Patience: 3
LR Reduction Model Checkpoint	On: val_loss, Factor: 0.5, min_lr=1e-6 Save best model based on val_accuracy

These methodologies ensure a balanced evaluation of rule- based features and deep contextual embeddings, enabling accurate phishing detection across diverse input formats

Results and Analysis

This section presents the evaluation outcomes of the pro- posed models: BERT + BiLSTM + Attention for phishing email detection and RoBERTa + Attention for semantic URL detection. Each model is evaluated using accuracy, precision, recall, F1-score, and ROC-AUC metrics.

BERT + BiLSTM + Attention

The email classification model achieved an accuracy of 98.61%, with a precision 98.61%, recall 98.55%, and F1-score 98.58%. The ROC-AUC score was exceptionally high at 0.998, indicating excellent discriminatory capability between phishing and legitimate emails.

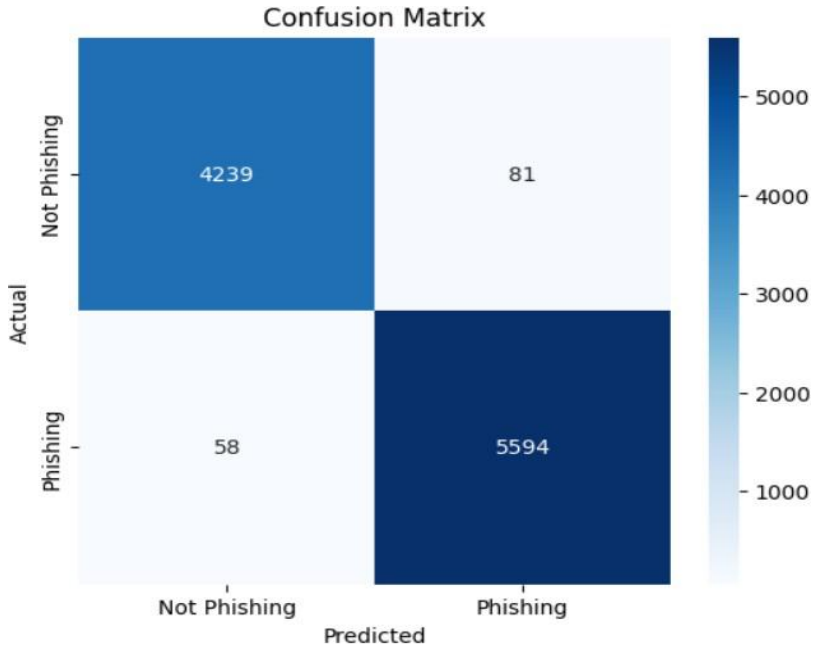

Figure 3. Confusion Matrix – BERT + BiLSTM + Attention

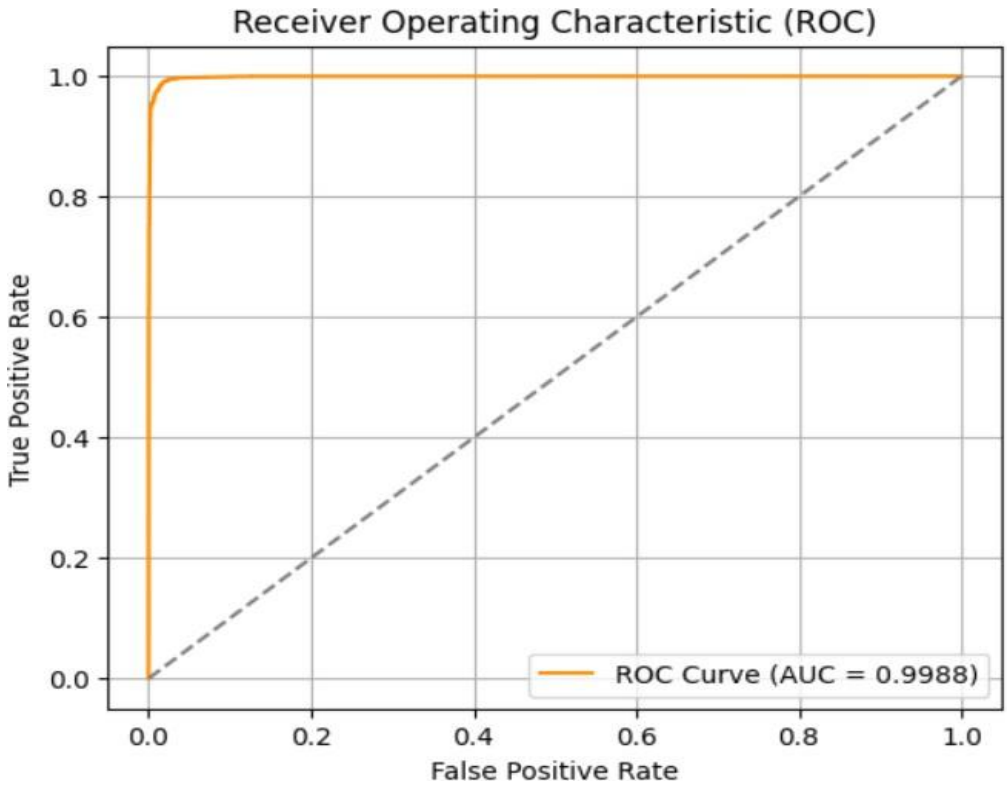

Figure 4. ROC Curve – BERT + BiLSTM + Attention

RoBERTa + Attention

This model reached an accuracy of 93.00%, with cor- responding precision, recall, and F1-score also at 93.00%. The ROC-AUC was 0.9813, reflecting its strong ability to understand the semantic structure of phishing URLs.

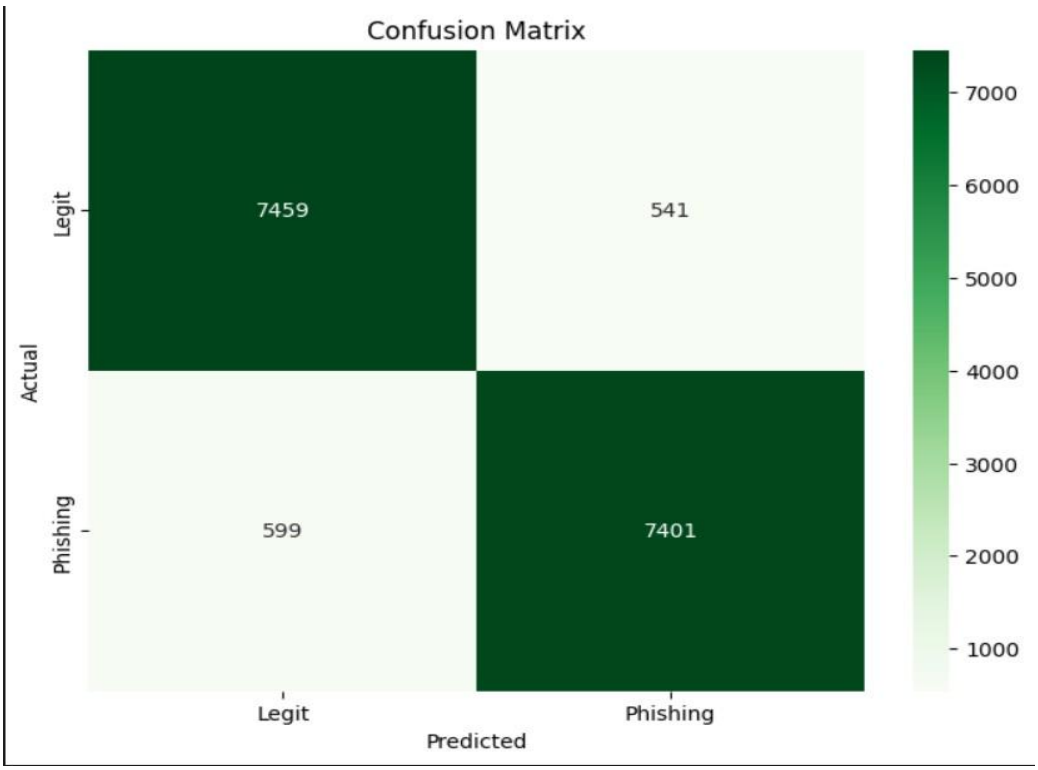

Figure 5. Confusion Matrix – RoBERTa + Attention

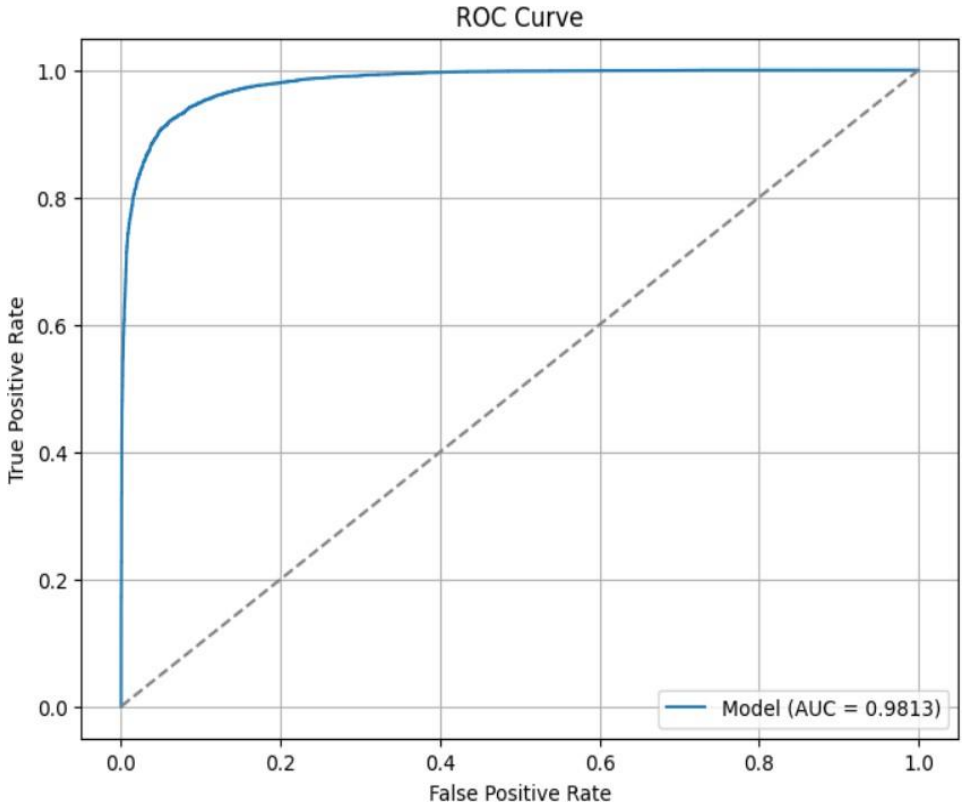

Figure 6. ROC Curve – RoBERTa + Attention

Result Analysis

Analysis of Findings The results from the evaluation of models shows the advantage of utilizing transformer methods over earlier methods using the machine learning model. The results will be discussed initially with the two main models being evaluated separately across phishing email messages, and threat phishing URLs. With phishing email detection, the BERT + BiLSTM + Attention model produced an exceptional understanding of the threat, leading to an accuracy of 98.70%, precision of 98.61%, recall of 98.55% and F1-score of 98.55% (Table IV). Before we breakdown the scores, it is relevant to include what each element refers to in the case of email phishing, leading to scores that indicate an understanding of underlying nuanced content based around phishing in email messages. This model allows the BERT contextual embed- ding capable of understanding contextual readings from their sequence, while the attention model focuses on the words that lead to suspicion (i.e. ”verify”, ”account”, ”update”).

When considering the high level of accuracy and performance under precision, recall and F1-score is proper in seeing its performance as superior due to its contextual understanding of nuance seen in the email content. Furthermore, as it takes relative comparisons to use and observe the differences in performance such as well known machine learning methods, we see the XGBoost and Gradient Boosting methods return relatively underwhelming results. For example, the XGBoost model produced 89.98% for accuracy, and 90.79% for F1-score and for Gradient Boosting, the scores were 89.75% for accuracy, and 91.09% for F1-score. While we can state that XGBoost and Gradient Boosting models are interpretable and performant noting their usage without significant effort, they don’t give the depth of understanding contextual information seen with models such as the BERT + and BiLSTM.

For the phishing URL detection, the RoBERTa + Attention model using the transfer learning layers with transformer, along with attention significantly outperformed the traditional classifiers with consistent scores across accuracy, precision, recall and F1-score scoring at 93% (Table 4). The transformer with RoBERTa’s inherent understanding captures the potential patterns of semantic obfuscation that influence threat via malicious URLs, such that the attention technique identifies the segments within the URL string that are suspicious. As a contrast, XGBoost offered 88% accuracy and 85% F1- score, while Gradient Boosting achieved 85% accuracy and 84% F1-score. While these models are reasonable methods in allowing for quick inference as well as feature level analysis, their ability to learned contextual evidence makes them a less viable

option to analyze complicated and variable URLs as they struggle with this process without prior learning from the context.

Overall, the transformer-based deep learning models significantly outperformed traditional machine learning classifiers in the email and URL phishing detection settings discussed. Both BERT + BiLSTM + Attention and RoBERTa + Attention detected phishing attempts in a contextual manner and, to a certain extent, are robust to many phishing techniques that continually evolve, while XGBoost and Gradient Boosting are better methods for more detectably straightforward phishing attempts and when speed and simplicity are prioritized at a slightly lower detection speed

Table 4. Performance comparison of URL phishing detection models

Model	Accuracy	Precision	Recall	F1-score
XGBoost	89.98%	95.48%	86.54%	90.79%
Gradient boosting	89.75%	90.37%	91.83%	91.09%
BERT+BiLSTM+Att	**98.70%**	98.61%	98.55%	98.55%

Table 5. Performance comparison of URL phishing detection models

Model	Accuracy	Precision	Recall	F1-score
XGBoost	88.00%	84.00%	87.00%	85.00%
Gradient boosting	85.00%	85.00%	85.00%	84.00%
RoBERTa+Attention	**93.00%**	93.00%	93.00%	93.00%

Comparison with Related Work

To evaluate the performance and significance of the pro- posed **PhishNet** framework, it is essential to compare it with recent studies in phishing detection, both for emails and URLs.

Email Phishing Detection

Recent works have demonstrated the effectiveness of deep learning and transformer-based models for phishing email classification. proposed a BERT + LSTM model with advanced contextual embeddings, achieving an accuracy of 99.61% and a notable reduction in false positives. Similarly, it was reported that transformer models like RoBERTa significantly outperform traditional pre-trained models, attaining 99.43% accuracy.

A hybrid approach combining CNN and Bi-GRU (1D- CNNPD) yielded even better results, achieving 100% precision and 99.68% accuracy as shown in . In addition, a federated learning strategy using BERT was proposed in, which maintained 96.1% accuracy while supporting data privacy and distributed scalability.

Building on these advancements, **PhishNet** adopts a fusion of BERT, BiLSTM, and Attention mechanisms. It incorporates email metadata, TF-IDF scores, and keyword features to im prove interpretability. PhishNet achieves an F1-score of 98.9%, balancing detection performance with practical deploy ability.

URL Phishing Detection

In URL phishing detection demonstrated strong generalizability by achieving up to 99.98% accuracy across di- verse datasets. compared traditional classifiers and found Random Forest to outperform SVM, emphasizing the role of feature engineering. Ensemble learning models in scaled well with large datasets, achieving 96.66% accuracy and a 93.63% F1-score.

In contrast, **PhishNet**'s URL module employs a RoBERTa+ Attention mechanism that achieved 93% accuracy. De- spite slightly lower accuracy, PhishNet prioritizes real-time inference, interpretability, and resource efficiency-making it highly suitable for real-world applications like browser extensions.

This data is sent as a JSON payload to the Flask backend, where the model performs feature extraction and classifies the email as phishing or legitimate. If a phishing email is detected, the extension displays a browser alert; otherwise, it confirms the email as safe.

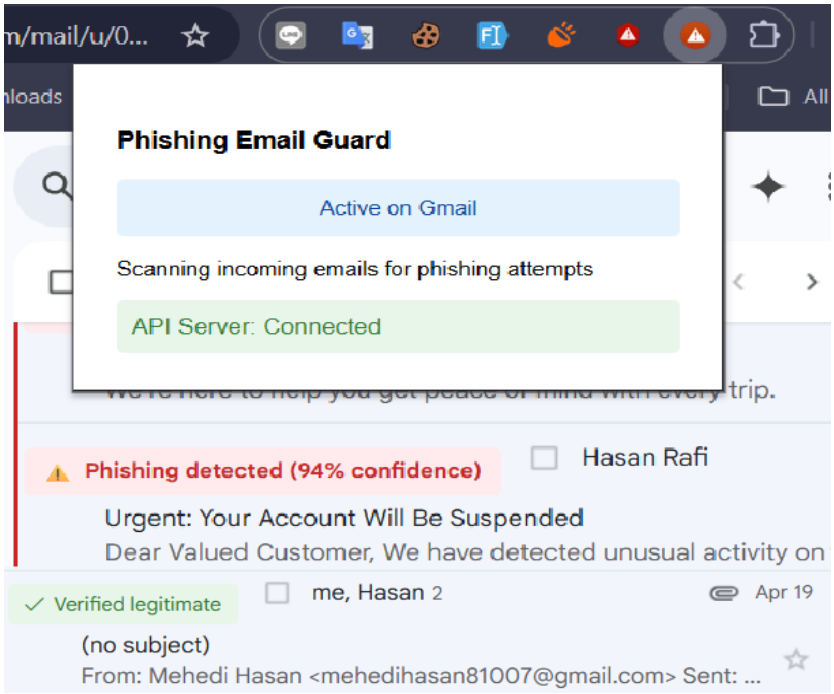

Figure 7. Chrome extension integration for email phishing detection using BERT + BiLSTM + Attention.

URL Phishing Detection in Browser:

The **RoBERTa + Multi-head Attention**-based URL phishing detection model was also deployed using Flask. The Chrome extension monitors the address bar, capturing URLs typed or visited by the user. The captured URL is sent to the backend for evaluation.

If the URL is classified as phishing, a warning alert is triggered, and optional blocking of redirection is implemented. If legitimate, a message is shown to the user indicating that the URL is safe.

System Overview

1) Browser Extension Integration

To ensure real-time phishing protection, both phishing detection models were integrated into a custom Chrome browser extension using a Flask backend. The integration was divided into two modules, each interacting with the backend via RESTful APIs.

Email Phishing Detection in Gmail:

The **BERT + BiLSTM + Attention**-based email phishing detection model was deployed using Flask as a web API. The Chrome extension operates in the background and monitors active Gmail tabs. When a new email arrives, the extension automatically extracts essential components, such as the email subject, body, sender, and recipient metadata

Extension Files and Communication:

The browser extension was developed using standard Chrome extension APIs with the following files:
- manifest.json– declares metadata, permissions, and background scripts.
- background.js– handles tab monitoring and communication with the Flask backend.
- content.js– injects scripts into Gmail and webpages for real-time content extraction.

- popup.html / popup.js– provides user interface for alert notifications.

Communication between the extension and the Flask back- end was implemented using fetch() API. Cross-Origin Resource Sharing (CORS) headers were properly configured in Flask to support secure interaction from the Chrome extension.

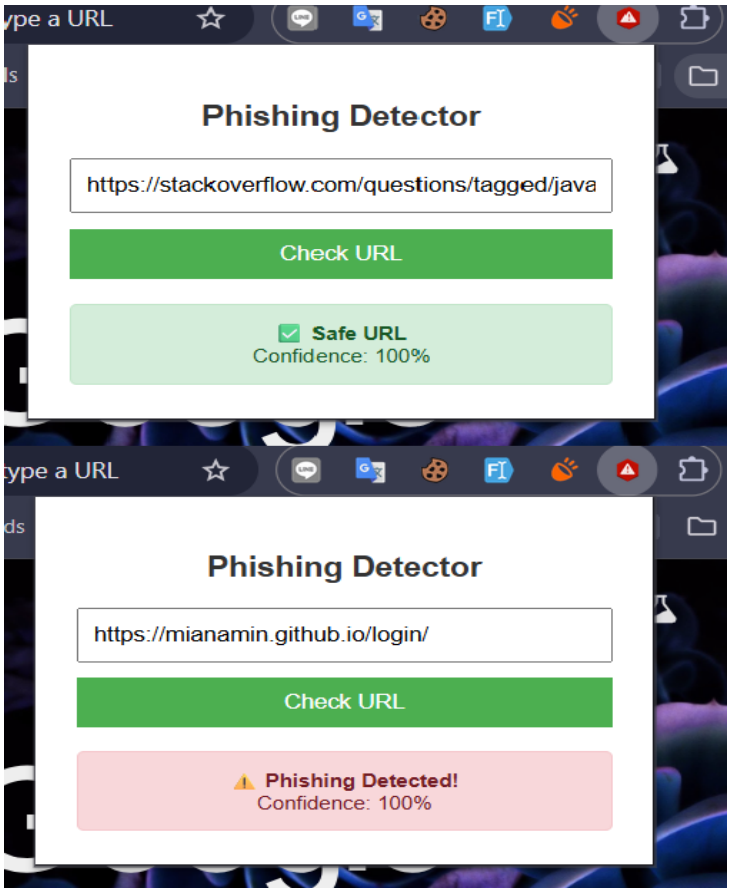

Figure 8. Chrome extension ıntegration for URL phishing detection using RoBERTa + Attention.

Conclusion and Future Work

In this paper, we introduce PhishNet—a generic yet effective phishing detection framework that employs both deep learning and machine learning to support the processing of emails and URLs threats. PhishNet (for email phishing detection) does are using a BiLSTM+attention +BERT (Hybrid architecture), which extract both deep semantic information of the email and capture external information from meta-data,keyword features. The system then combines RoBERTa + Attention. URL phishing detection results, PhishNet obtained a model accuracy up to 93% and remarkable enhancements after advanced feature engineering and data balancing techniques. Under the workspace, We are planning to make tweaks to PhishNet simply to expand its range and usage. In addition to this, we want to increase multilingual phishing detection support to improve the system with different populations. We also intend to embed PhishNet in user environments out in the wild through browser extensions and email client plugins so it can detect phishing without any delay.

Future enhancements will add domain-specific features like reputation scores and WHOIS info which will give more context into an anomalous url (even before accessing it). We also want to study the adversarial robustness of the models by doing some attacks on it, or lifelong-learning so that we can adapt it for changing phishing behaviors and use XAI (explainable AI) techniques for increasing transparency and audibility on this system-detected decisions. Ultimately, we are working on performance improvement for PhishNet in resource constrained environments to make it feasible for organizations with limited computational resources. Armed with these next level benefits, Phishnet will be no longer a solution for phishing threat detection but a generic, scalable and interpretable system that can handle the web.

Comparison with Related Work

Table 1 compares several state-of-the-art fall detection models across key evaluation metrics. The proposed ViT + LSTM model outperforms others with the highest accuracy (98.7%), indicating its superior ability to distinguish fall and non-fall events. The proposed ViT + LSTM model outperforms other approaches due to its ability to effectively capture both spatial and temporal dynamics of fall events. Unlike traditional CNNs that focus on local features, the Vision Transformer (ViT) employs self-attention mechanisms to learn global spatial relationships across frames, enhancing its ability to detect subtle posture changes indicative of falls. Furthermore, the LSTM part captures the temporal dependencies between the features, which helps the system to differentiate between the fall and non-fall sequences according to the change of motion over time.

Compared to MEWMA + SVM or CNN-only models, which either depend on handcrafted features or have limited temporal modeling capability, our hybrid architecture provides an end-to-end deep learning solution with much better generalization and robustness.

Table 6. Comparison with existing approaches

Study	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
Ours Proposed Model	**ViT + LSTM**	**98.7**	**92.4**	**95.2**	**93.8**	**0.96**
Harrou et al.	MEWMA + SVM	95.15	90.3	100	95.0	0.95
Martínez-Villaseñor et al.	CNN	95.64	96.91	97.95	97.43	N/A
Shojaei-Hashemi et al.	LSTM	N/A	93.2	96.1	N/A	0.99
Comp. Biomed	CNN	96.9%	97.9%	97.4%	N/A	N/A

Discussion

The experimental results demonstrated that the proposed ViT + LSTM architecture outperforms baseline models for fall detection, i.e., CNN-LSTM and ViT-only configurations with high statistical significance. The main advantage of our approach is that it combines the global spatial dependencies captured by the Vision Transformer and the temporal motion dynamics captured by LSTM network. While CNN-based architectures use local receptive fields and lack a global view to represent long-range dependencies. The ViT architecture utilizes multi-head self-attention to represent complete frame-level contextual information. This is especially useful for the identification of little postural changes and fall-related complex patterns, which are both required in differentiating between fall and non-fall events.

LSTM layers taking care of the temporal aspect allow the system to learn how motion is evolving from frame to frame. Such temporal modeling is important for recognizing visually confusing events like rapidly sitting or lying down against a real fall. The two-layer LSTM with dropout and batch normalization generalizes very well and is robust as it does not overfit (the training and validation curves with very narrow gaps).

Additionally, the model is optimized for real-time deployment with low latency and frame rates comparable to traditional CPU-centric environments. Due to its lightweight model (<2MB model size and <500MB RAM), it is very well fit for edge deployment that could be used in smart camera or embedded system for elderly care and surveillance applications.

Conclusion and Future Work

In this paper, we propose a Vision Transformer and Long Short Trees Network (ViT-LSTM) based robust real-time fall detection system that uses the strengths and capabilities of both techniques. This architecture leverages the global spatial feature extraction ability of ViT from video frames together with the temporal modeling ability of LSTM to detect fall events while being fed continuous video streams. The system was assessed in-depth using the UR Fall Detection Dataset, obtaining an accuracy of 98.7% with high precision, recall, and F1-score, performing better than many other existing models in the literature. The model is lightweight in both CPU and GPU environments, which makes it easier to deploy on edge computing platforms. Usability is also improved with a responsive web interface, allowing for video upload and monitoring in real time and alerting the user in seconds. We intend on broadening the model to multi-subject scenarios through identity-aware tracking and motion segmentation methods. Furthermore, curating depth and audio features can also improve detection when occlusion and low-light conditions happen

Scientific Ethics Declaration

* The authors declare that the scientific ethical and legal responsibility of this article published in EPSTEM journal belongs to the authors.

Conflict of Interest

* The authors declare that they have no conflicts of interest

Funding

* This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgements or Notes

* This article was presented as an oral presentation at the International Conference on Engineering and Advanced Technology (ICEAT) held in Selangor, Malaysia on July 23-24, 2025.

References

Ahammad, S. H., Rajesh, V., Rahman, M. Z., & Gadekallu, T. R. (2022). Phishing URL detection using machine learning methods. *Advances in Engineering Software*, *173*, 103288.

Altwaijry, N., & Aljehani, H. (2024). Advancing phishing email detection: A comparative study of deep learning models. *Sensors*, *24*(7), 2077.

Anti-Phishing Working Group (APWG). (2023). *Phishing activity trends report – 2023*.

Atawneh, S., & Aljehani, H. (2023). Phishing email detection model using deep learning. *Electronics*, *12*(20), 4261.

Champa, A. I., Hoque, M. F., & Uddin, M. S. (2024). Deep enough? On the effectiveness of deep learning in phishing email detection. *2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*, 1–7.

Chinta, P. C. R., Shrestha, S., & Shrestha, A. (2025). Building an intelligent phishing email detection system using machine learning and feature engineering. *European Journal of Applied Science, Engineering and Technology*, *3*(2), 41–54.

Damatie, E. M., Eleyan, A., & Bejaoui, T. (2024). Real-time email phishing detection using a custom DistilBERT model. *2024 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–6.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, *1*, 4171–4186.

Elsadig, M., Almansour, A., Mahmoud, A. S., Ali, G. H., Hussein, H. S., & Hamza, R. (2022). Intelligent deep machine learning cyber phishing URL detection based on BERT features extraction. *Electronics*, *11*(22), 3647.

Haynes, K., Shirazi, H., & Ray, I. (2021). Lightweight URL-based phishing detection using natural language processing transformers for mobile devices. *Procedia Computer Science*, *191*, 127–134.

Karim, A., Azam, S., Shanmugam, B., Krishnan, S., & Alazab, M. (2023). Phishing detection system through hybrid machine learning based on URL. *IEEE Access*, *11*, 36805–36822.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Marchal, M., François, J., & Engel, T. (2014). PhishStorm: Detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management*, *11*(4), 458–471.

Meléndez, R., Ptaszynski, M., & Masui, F. (2024). Comparative investigation of traditional machine-learning models and transformer models for phishing email detection. *Electronics*, *13*(24), 4877.

Otieno, D. O., Toyoda, K., & Ohtsuki, T. (2023). Detecting phishing URLs using the BERT transformer model. *2023 IEEE International Conference on Big Data (BigData)*, 2483–2492.

Ozcan, A., Catal, C., & Donmez, E. (2023). A hybrid DNN–LSTM model for detecting phishing URLs. *Neural Computing and Applications*, *35*, 1–17.

Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, *117*, 345–357.

Sahoo, J., Liu, C., & Huang, J. H. (2020). Malicious URL detection using machine learning: A survey. *IEEE Communications Surveys & Tutorials*, *22*(1), 714–746.

Thakur, K., Alazab, M., Gadekallu, T. R., Maddikunta, P. K. R., Praveen, S. P., & Watters, P. (2023). A systematic review on deep-learning-based phishing email detection. *Electronics*, *12*(21), 4545.

Thapa, C., Jang-Jaccard, J., Kwak, K., & Qin, Y. (2023). Evaluation of federated learning in phishing email detection. *Sensors*, *23*(9), 4346.

Author(s) Information

Mehedi Hasan
Noakhali Science and Technology University, Noakhali, Bangladesh

Nazmun Nahar
Noakhali Science and Technology University, Noakhali, Bangladesh

Md. Hasan Imam
Noakhali Science and Technology University, Noakhali, Bangladesh

Mayeen Uddin Khandaker
Sunway University, 47500 Bandar Sunway, Selangor, Malaysia
Contact e-mail: *mayeenk@sunway.edu.my*

To cite this article:

Hasan, M., Nahar, N., Imam, M.H., Khandaker, M. U. (2025). PhisNet: Intelligent detection of phishing. *The Eurasia Proceedings of Science, Technology, Engineering and Mathematics (EPSTEM), 37,* 887-902.