International Journal of Security and Networks

ISSN online: 1747-8413 - ISSN print: 1747-8405
https://www.inderscience.com/ijsn

Enhanced phishing URL identification using an integrated attention-based LSTM-CNN with hybrid features

Santosh Kumar Birthriya, Priyanka Ahlawat, Ankit Kumar Jain

DOI: 10.1504/IJSN.2025.10069698

Article History:

Received:	19 November 2023
Last revised:	13 November 2024
Accepted:	13 November 2024
Published online:	17 March 2025

Enhanced phishing URL identification using an integrated attention-based LSTM-CNN with hybrid features

Santosh Kumar Birthriya*, Priyanka Ahlawat and Ankit Kumar Jain

National Institute of Technology,
Kurukshetra-136119, Haryana, India
Email: cs.santosh1b@gmail.com
Email: priyankaahlawat@nitkkr.ac.in
Email: ankitjain@nitkkr.ac.in
*Corresponding author

Abstract: Phishing attacks continue to pose a significant threat to online security, targeting users' personal and financial information through deceptive URLs and websites. This study proposes a robust hybrid deep learning model for phishing URL detection. Our approach follows a multi-step methodology, including URL data pre-processing, advanced feature engineering, and the application of deep learning techniques for precise URL classification. Feature engineering incorporates TF-IDF vectorisation, principal component analysis, and natural language processing based feature extraction, forming a comprehensive hybrid feature set that improves detection accuracy. Experiment results reveal that hybrid features significantly enhance the performance of deep learning models, with the proposed LSTM-CNN with attention model achieving the highest accuracy at 99.92%. This research underscores the potential for advanced hybrid architectures in cybersecurity applications, and efficient solution for real-time phishing detection.

Keywords: phishing; natural language processing; NLP; long short-term memory; LSTM; convolutional neural network; CNN; cybersecurity.

Reference to this paper should be made as follows: Birthriya, S.K., Ahlawat, P. and Jain, A.K. (2025) 'Enhanced phishing URL identification using an integrated attention-based LSTM-CNN with hybrid features', *Int. J. Security and Networks*, Vol. 20, No. 1, pp.8–22.

Biographical notes: Santosh Kumar Birthriya is a PhD scholar in National Institute of Technology, Kurukshetra, India. He has received his MTech in Computer Engineering from National Institute of Technology Kurukshetra, India. His research interests include cyber security, machine learning and deep learning, network and information security.

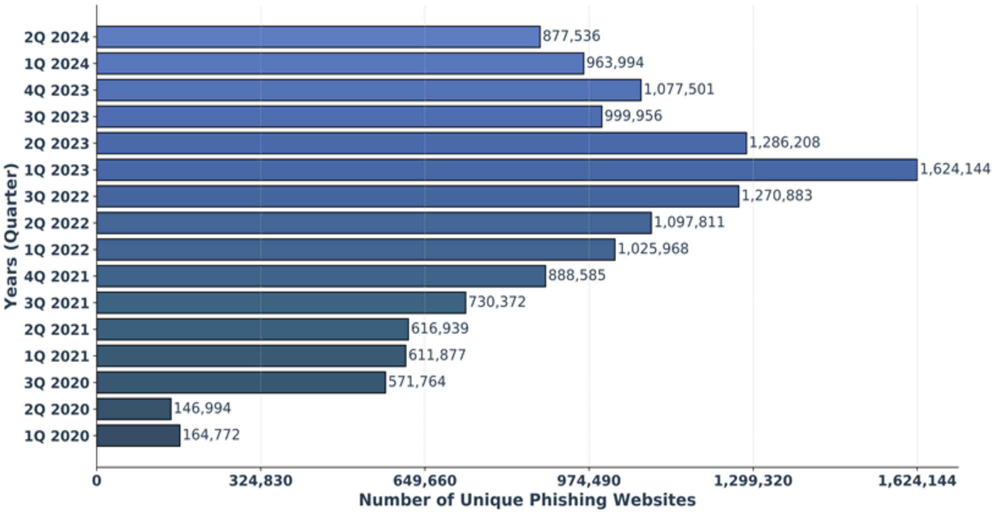
Priyanka Ahlawat received her PhD in Computer Science and Engineering from National Institute of Technology, Kurukshetra, India. She is an Assistant Professor in Computer Engineering Department NIT Kurukshetra Haryana, India. Her interest areas are key management and wireless sensor networks security.

Ankit Kumar Jain is presently working as an Assistant Professor in National Institute of Technology, Kurukshetra, since September 2013. He received his Master of technology from Indian Institute of Information Technology Allahabad (IIIT) India. He received his PhD degree from National Institute of Technology, Kurukshetra in the area of Information and Cyber Security. He has more than 60 research papers in international journals and conferences of high repute including Elsevier, Springer, Taylor & Francis, Inderscience, IEEE, etc. His general research interest is in the area of information and cyber security, phishing website detection, web security, and machine learning.

1 Introduction

In today's fast-evolving digital landscape, individuals can perform numerous tasks through smart devices, including online shopping, education, and obtaining medicines (Kumar, 2023). However, these digital conveniences also bring associated risks. The rapid expansion of online services has opened new channels for cyber attackers to exploit users' personal information (Almomani et al., 2022). Among the most prevalent threats is phishing, which compromises sensitive information such as names, contact details, identification, and credit card data through deceptive e-mails, messages, or URLs (Ahammad et al., 2022).

Figure 1 Statistics of phishing websites 1Q 2020–2Q 2024 (see online version for colours)



Some attackers may cease their efforts after securing financial gain, whereas others pursue further infiltration, breaching organisations for larger datasets. Sophisticated techniques to create deceptive URLs that resemble legitimate sites continuously threaten users, underscoring the urgent need for increased awareness and stringent security measures (Sánchez-Paniagua et al., 2022). Phishing attacks often involve URLs and e-mails in which attackers disguise themselves as trusted entities, such as familiar companies or banks, prompting recipients to click on links or download attachments (Jain and Gupta, 2022). Such actions can lead to the exposure of sensitive information. Phishing schemes commonly replicate the appearance of legitimate company websites or e-mails to deceive users (Kumar Birthriya and Jain, 2022).

1.1 Statistics on phishing websites

Figure 1 illustrates the rapid rise in phishing websites from early 2020 to mid-2024. Starting with relatively low numbers in 2020, there is a clear upward trend, with significant spikes in 2021 and 2022. The number of unique phishing sites peaked dramatically in the second quarter of 2023, reaching over 1.6 million, the highest in the dataset. While there's been a slight dip in 2024, the numbers remain alarmingly high, indicating that phishing threats continue to be a major concern. The data underscores the increasing frequency and scale of phishing attacks over recent years (APWG, 2024).

Traditional machine learning (ML) methods generally require human involvement in feature extraction and selection, handling these tasks separately from classification (Gupta, B. B et al., 2021). Deep learning (DL), in contrast, combines these processes through automated learning and feature extraction, thus reducing the dependence on manual engineering and external services (Birthriya et al., 2024a). The superiority of DL is evident in its high performance and robust problem-solving capabilities, particularly in large-scale applications like speech recognition, image classification, and phishing detection (Opara et al., 2020).

DL plays a foundational role in fields such as autonomous driving, facial recognition, and healthcare. Computers are trained to learn from examples similarly to the human brain, enabling classification tasks directly from extensive datasets, including text, audio, and images. Results from DL models sometimes surpass human capabilities, though training them demands substantial labelled data, significant computational resources, and complex neural network structures (Shrestha and Mahmood, 2019).

Our paper proposes a system that leverages DL for phishing detection, offering a powerful classification tool in the fields of information security and cybersecurity. This work employs an LSTM-CNN architecture with attention, which significantly improves the classification of phishing URLs and contributes to the prevention of financial fraud and cybercrimes.

1.2 Contribution

This work presents several advancements in the LSTM-CNN with attention model:

- Development of a novel LSTM-CNN architecture integrated with an attention mechanism, effectively combining the sequential learning capabilities of LSTM and the feature extraction strengths of CNN to enhance phishing detection.
- Implementation of a robust feature engineering strategy that combines TF-IDF word vectors, PCA for dimensionality reduction, and 40 additional hand-crafted NLP-based features, resulting in a comprehensive hybrid feature set that improves model performance.
- Extensive experimental validation demonstrating that the proposed hybrid model significantly outperforms traditional ML algorithms and other DL models, achieving an accuracy of 99.92% and a low false positive rate.

- The system provides a highly effective phishing detection tool, contributing to the prevention of financial fraud and cybercrimes while offering valuable methodologies to the domains of information security and cybersecurity.

The paper is organised as follows: Section 2 provides a review of previous studies, offering relevant background information. Section 3 explains various types of URLs and the tactics employed by attackers. Section 4 introduces a new method, the hybrid LSTM-CNN model with an attention mechanism. Section 5 presents the performance evaluation of the proposed method, while Section 6 provides detailed performance results. Section 7 compares the proposed method with existing solutions for phishing detection. Finally, Section 8 concludes the paper with a summary and recommendations for future research.

2 Related works

Phishing websites continue to pose a significant challenge in cybersecurity, with no definitive solution to eliminate all threats. DL techniques have emerged as a promising approach to address this complex issue, specifically for phishing website detection. This section highlights previous methodologies applied to this problem.

2.1 Long short-term memory (LSTM)

LSTM networks have been utilised for their ability to handle sequential data and capture long-term dependencies. Liang et al. (2020) introduced a novel approach for detecting malicious URLs using a bidirectional long short-term memory (B-LSTM) network. The study also evaluates the performance of the B-LSTM network against traditional ML models and unidirectional LSTM. The findings reveal that the B-LSTM network outperforms the other methods, achieving an impressive F1 score of 95.9% (Jishnu and Arthi, 2023). This study proposes a phishing URL detection method combining RoBERTa for semantic feature extraction and LSTM for classification. RoBERTa captures contextual embeddings of URLs, while LSTM leverages sequential relationships for accurate categorisation. Using a dataset of 300,000 URLs, the system achieves 97.14% accuracy in distinguishing legitimate from phishing URLs.

2.2 Convolutional neural networks (CNN)

CNN have been increasingly applied in phishing URL detection due to their proficiency in extracting spatial hierarchies in data. Aljofey et al. (2020) utilised CNNs to detect phishing URLs at the character level. Analysing URL character sequences with max-pooling and fully connected layers, their model achieved a 95.02% accuracy on a specific dataset, outperforming many existing methods on benchmark datasets. Opara et al. (2020) introduced HTMLPhish, a CNN-based platform that classifies phishing web pages by analysing the semantics of over 50,000 HTML documents. By bypassing manual feature engineering, HTMLPhish achieved over 93% accuracy.

Tajaddodianfar et al. (2020) proposed Texception, a novel DL architecture for analysing URLs at both character and word levels using parallel convolutional layers. Tested on the Microsoft SmartScreen dataset, Texception exhibited a high true positive rate and a low false-positive rate. Wei et al. (2020) presented a method to detect malicious URLs with nearly 100% accuracy using CNNs. Focusing solely on URL text, their method enabled faster detection and identification of zero-day attacks, optimised for efficient use even on mobile devices without compromising performance. Korkmaz et al. (2021) developed a CNN method for phishing detection centred on n-gram feature extraction from URLs. Achieving an 88.90% accuracy rate, their work demonstrated the effectiveness of single-character URL analysis.

2.3 Combining LSTM and CNN

Hybrid models combining LSTM and CNN have been explored to leverage the strengths of both architectures. Adebowale et al. (2020) introduced the intelligent phishing detection system (IPDS), a model combining LSTM and CNN. Trained on a dataset of one million legitimate and phishing URLs from sources like PhishTank and Common Crawl, IPDS achieved an accuracy rate of 93.28%, showcasing the potential of hybrid models. Do et al. (2021) evaluated various architectures, including DNN, CNN, GRU, and LSTM, on a dataset of 11,055 legitimate and phishing URLs, highlighting the superiority of DL models in achieving high-performance metrics. Ozcan et al. (2023) introduced hybrid DL models combining LSTM and deep neural network (DNN) algorithms for phishing URL detection. Leveraging both character embeddings and NLP features, their models captured intricate character-level patterns and uncovered high-level semantic relationships. Experimental results demonstrated that the proposed models outperformed existing methods, achieving superior accuracy.

2.4 Recent advancements

Recent research highlights innovative methods for improving phishing detection accuracy and adaptability. Asiri et al. (2024b) proposed a BiLSTM-based system integrated with a browser extension and Docker container, effectively detecting TinyURLs, browsers in the browser (BiTB), and regular phishing attacks, achieving 99% precision, recall, and F1 score with their weighted average strategy (WeAS) outperforming other methods. Asiri et al. (2024a) further introduced PhishTransformer, a human-in-the-loop system that improved accuracy, precision, recall, and F1 score by 5% through iterative retraining with human feedback. Ujah-Ogbuagu et al. (2024) developed a hybrid CNN-LSTM model, achieving 98.9% and 96.8% accuracy on UCL spoofing website and PhishTank datasets, significantly surpassing standalone CNN and LSTM models. These approaches demonstrate the power of combining advanced DL architectures with human feedback to tackle evolving phishing challenges effectively.

The literature emphasises DL, particularly CNNs, LSTMs, and hybrids, for phishing detection due to their accuracy and adaptability. However, challenges in reducing computational demands and reliance on external data remain. Our research addresses these gaps with a hybrid LSTM-CNN model enhanced by attention mechanisms, offering improved detection without third-party dependencies or manual feature engineering.

Table 1 provides an overview of various studies on phishing detection using DL methodologies. The table summarises the problem statement, the DL techniques employed, the dataset used, the distribution of training and testing instances, and the performance measures reported.

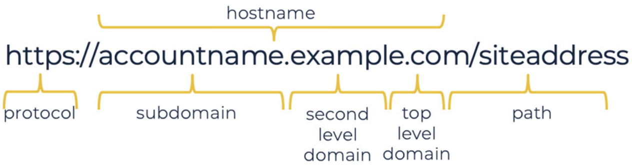
3 URLs and attacker methods

Cybercriminals utilise an array of strategies to circumvent detection by security tools and system administrators. A foundational understanding of URL components, as depicted in Figure 2, is key to grasping these evasion techniques.

A URL is typically composed of four primary elements:

- Protocol: situated at the beginning of the URL, the protocol indicates the method of data transfer. Common protocols include hypertext transfer protocol (HTTP) and HTTPS (HTTP Secure), with HTTPS offering an encrypted and secure connection.
- Subdomain (optional): positioned before the second-level domain (SLD), the subdomain is an optional part of the URL. It is utilised to distinguish different sections or services of a website, with examples like www, blog, or mail.
- Second-level domain: this component often signifies the organisation or purpose of the website. For instance, in www.example.com, 'example' represents the SLD. It plays a vital role in brand identity and remains unique within its specific top-level domain (TLD).
- Top-level domain: following the SLD and marked by the final dot, the TLD categorises domains broadly by type or geographic location. Common examples include .com, .org, .net, and geographically specific endings such as .uk or .jp.

Figure 2 URL components (see online version for colours)



These elements collectively form the structure of a URL, which serves as a distinct address for accessing specific content online (Koppula et al., 2010).

Table 1 Comparative analysis of related works

References	*DL methodology*	*Dataset*	*Accuracy (%)*
Liang et al. (2020)	Bidirectional LSTM	360 NetLab and Alexa	96.40
Jishnu et al. (2023)	RoBERTa for feature extraction, LSTM for classification	Kaggle, Majestic Million, and PhishTan	97.14
Aljofey et al. (2020)	CNN	Alexa, Openphish, Spamhaus, Techhelplist, ISC SANS, PhishTank	95.02
Opara et al. (2020)	CNN	Alexa's top domains, PhishTank	94
Tajaddodianfar et al. (2020)	Parallel convolutional layers	Microsoft SmartScreen browsing telemetry data	-
Wei et al. (2020)	CNN	PhishTank and common crawl	99.98
Korkmaz et al. (2021)	CNN with n-gram feature extraction	PhishTank	88.90
Adebowale et al. (2020)	Hybrid LSTM-CNN	PhishTank and common crawl	93.28
Do et al. (2021)	DNN, CNN, GRU, LSTM	UCI machine learning repository	97.29
Ozcan et al. (2023)	Hybrid LSTM-DNN models	Ebbu2017, PhishTank	98.79
Asiri et al. (2024b)	BiLSTM with weighted average strategy (WeAS)	PhishingArmy and PhishTank, Alexa	99
Asiri et al. (2024a)	PhishTransformer with iterative retraining	PhishingArmy and PhishTank, Alexa	79
Ujah-Ogbuagu et al. (2024)	Hybrid CNN-LSTM	UCL spoofing website and PhishTank datasets	98.9

4 Proposed methodology

Figure 3 illustrates our phishing URL detection method, which integrates a hybrid LSTM-CNN model enhanced with attention. The process begins with data pre-processing to ensure the input is clean and structured. Next, feature engineering extracts meaningful patterns essential for effective detection. The LSTM-CNN model, combined with attention, leverages sequential learning, spatial patterns, and focused feature prioritisation to improve accuracy. Finally, the model is compiled, trained, and used for URL classification, offering a comprehensive and efficient solution for phishing detection.

4.1 Data pre-processing

Data pre-processing for URL analysis involves understanding the URL structure, where words and special characters like dots ('.') and slashes ('/') signify different sections, such as domains, subdomains, and paths. The process begins with parsing URLs to extract individual words, separating them from special characters. These extracted words are then compared against specified web pages and a list of random words. The goal is to identify brand names, relevant keywords, and non-sensical or random strings within the URL. Correctly distinguishing between these categories is crucial for effective URL classification. The data preparation module supports this by comparing the extracted words with known brand names and keywords, categorising any random or non-sensical strings separately. This results in a structured list of words from the URL, categorised as brand names, keywords, and random strings, which is essential for further stages in URL analysis and classification.

4.2 Feature engineering

The feature engineering stage of our study entails transforming words into vectors, executing principal component analysis (PCA), deriving features using NLP, and conducting feature selection. This results in a composite feature set that enhances the efficiency and accuracy of phishing URL detection.

4.2.1 Word embedding (word2vec)

In the domain of text analysis and mining, vectorisation of words can be performed using the method known as term frequency-inverse document frequency (TF-IDF) (Kabra and Nagar, 2023). Given a set of documents *D*, where each document *d* consists of words, we can define the following:

- Term frequency (TF): for a term *t* in a document *d*, the term frequency is given by equation (1):

$$\mathrm{TF}(t, d) = \frac{\text{Number of times } t \text{ appears in } d}{\text{Total number of terms in } d} \quad (1)$$

- Inverse document frequency (IDF): the IDF for a term *t* across a set of documents *D* is defined by equation (2):

$$\mathrm{IDF}(t, D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right) \quad (2)$$

  where *N* is the total number of documents, and $|\{d \in D$: $t \in d\}|$ is the number of documents containing the term *t*.

- TF-IDF score: the TF-IDF score for a term t in a document d is calculated as shown in equation (3):

$$\text{TF-IDF}(t, d, D) = \mathrm{TF}(t, d) \times \mathrm{IDF}(t, D) \quad (3)$$

The Word2Vec method enables the extraction of approximately 1,700 unique word features from the test dataset.

4.2.2 Principal component analysis

PCA is a statistical procedure that is commonly used for dimensionality reduction. The main idea behind PCA is to transform the original, high-dimensional data into a lower-dimensional space using a new set of variables, termed as principal components, which are uncorrelated and ordered in such a way that they retain the majority of the variability present in the original data (Song et al., 2010).

Given a dataset $X \in \mathbb{R}^{(n\times m)}$, where n represents the number of samples and m represents the number of variables, the goal is to find a transformation matrix $A \in \mathbb{R}^{(m\times k)}$ that transforms the data to a new lower-dimensional space $Y \in \mathbb{R}^{(n\times k)}$, maximising the variance along the new axes. This transformation is defined by equation (4):

$$Y = XA \quad (4)$$

where each row of *A* represents a principal component. These principal components are subject to certain constraints to ensure they capture unique aspects of the data's variance. First, each principal component vector has a length of 1, as indicated by equation (5):

$$\|a_k\| = 1 \text{ for } k = 1, 2, \ldots, k \quad (5)$$

Additionally, different principal components are orthogonal to each other, as shown in equation (6):

$$a_k \cdot a_j = 0 \text{ for } k \neq j \quad (6)$$

The principal components are sorted by the proportion of variance they explain, following the order expressed in equation (7):

$$\mathrm{Var}(Y_1) \geq \mathrm{Var}(Y_2) \geq \cdots \geq \mathrm{Var}(Y_k) \quad (7)$$

Figure 3 Detailed architecture of the proposed methodology (see online version for colours)

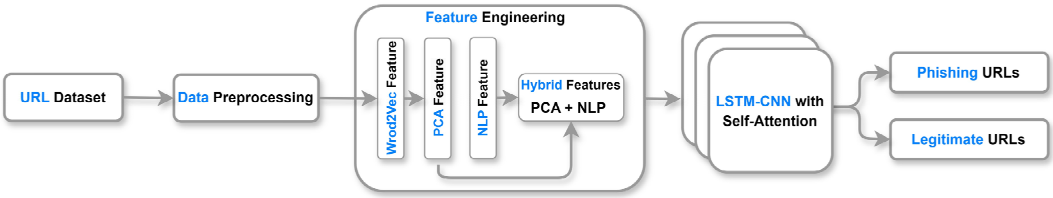


Figure 4 NLP-based 40 features (see online version for colours)

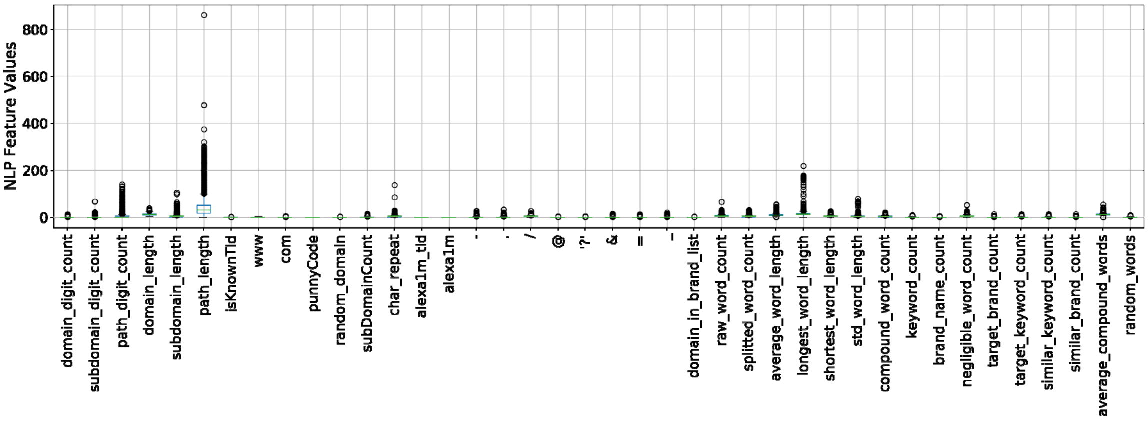
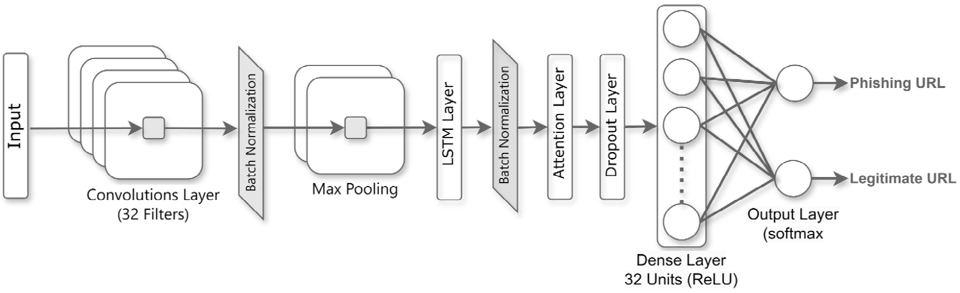


Figure 5 Architecture of the attention-based LSTM-CNN



In our research, we applied PCA to the Word2Vec feature set $X \in \mathbb{R}^{(n \times 1,700)}$, selecting k = 102 principal components. This transformation reduced the dimensionality of the data from 1,700 to 102 while retaining the most significant variance in the dataset.

4.2.3 *Features based on natural language processing (NLP)*

Our approach lies in feature extraction, which is accomplished through pre-processing data using NLP techniques. A pivotal challenge involves deconstructing the URL text into separate words, specifically when dealing with compound text (Young et al., 2018). Recognising that perpetrators frequently use misleading strategies, such as embedding brand names or important keywords within harmful URLs, we suggest an extra 40 NLP-based features we suggest an extra 40 NLP-based features as shown in Figure 4. These features assist in detecting phishing websites and complement the features already acknowledged in previous research.

4.2.4 *Hybrid features*

In our hybrid features approach, we generate word vectors using either CountVectoriser or TF-IDF vectoriser, transforming the dataset D into vectors of size m (where m = 1,700 represents the original Word2Vec feature size). We then apply PCA to these word vectors to obtain the top n components (n = 102), effectively reducing dimensionality. Concurrently, we extract additional NLP features from the dataset using the function NLP: $D \rightarrow \mathbb{R}^k$ (with k = 40 being the number of NLP features). The PCA-reduced word vectors and the NLP features are concatenated (denoted by the symbol ⊕) to form a hybrid feature vector $H \in \mathbb{R}^{n+k}$. This hybrid vector is then input into a DL model $DL : \mathbb{R}^{n+k} \rightarrow \mathbb{R}^{o}$, which maps it to the output space for prediction or classification (where o is the output size of the DL function).

4.3 Proposed LSTM-CNN with self-attention method

In Figure 5, the proposed model for phishing URL identification employs an integrated CNN and LSTM network with an attention mechanism. This hybrid architecture leverages the strengths of CNNs for feature extraction, LSTMs for sequential pattern learning, and attention mechanisms for focusing on critical components within the input sequence, ultimately achieving robust phishing URL detection. The methodology involves a sequential stacking of layers, each with specific functions aimed at improving the model's capacity to identify phishing characteristics.

- *Input processing layer:* the input to the model is a sequence of characters or tokens representing URL components. This input is reshaped to accommodate the requirements of the CNN and LSTM layers that follow, ensuring that each layer can effectively process and extract relevant features.
- *Convolutional layer (Conv1D):* the initial layer in the model architecture is a 1D convolutional layer with 32 filters and a kernel size of 3. This convolutional layer extracts local patterns from the URL sequence, capturing n-gram features that are commonly present in phishing URLs (e.g., repeated substrings or unusual combinations of characters). The layer uses a rectified linear unit (ReLU) activation function to introduce nonlinear transformations, which allows the model to capture complex patterns in the input data. Additionally, ℓ_2 regularisation ($\lambda = 0.01$) is applied to the weights of this layer to mitigate overfitting, especially important due to the complex and varied nature of phishing URLs.
- *Batch normalisation layer:* following the convolutional layer, a batch normalisation layer is applied to normalise the activations, stabilising and accelerating the training process. Batch normalisation adjusts and scales the activations across the mini-batches, helping to reduce the internal covariate shift and allowing for a smoother gradient flow, which improves model convergence and reduces the chances of overfitting.
- *Max pooling layer (MaxPooling1D):* a max pooling layer with a pool size of 2 is then applied to down-sample the feature maps. This layer reduces the dimensionality of the feature representation by retaining only the most salient features from the convolutional output. Max pooling helps focus on the strongest activations, thereby highlighting the key features learned in the previous convolutional layer while reducing the computational burden for subsequent layers.
- *LSTM layer:* following the CNN layers, a 32-unit LSTM layer with return_sequences = True is added to capture the temporal dependencies within the URL structure. LSTM networks are particularly effective for sequential data due to their ability to maintain long-term dependencies and mitigate the vanishing gradient problem. By using the sequence output, this layer ensures that the temporal relationships among URL components are preserved. This LSTM layer is also regularised with ℓ_2 penalty ($\lambda = 0.01$), promoting generalisation by penalising large weights that could lead to overfitting.
- *Attention layer:* to improve the model's focus on key segments of the URL, a custom attention layer is integrated after the LSTM layer. This attention mechanism assigns varying weights to different parts of the input sequence, allowing the model to emphasise parts of the URL that are more likely to indicate phishing behaviour. The attention layer calculates an attention score for each element in the sequence based on learned weights, applies a softmax operation to normalise these scores, and uses them to re-weight the input sequence. The output is then reduced through a summation operation, which effectively captures the weighted context of the sequence, enhancing the model's ability to discriminate between phishing and legitimate URLs.
- *Dropout layer:* a dropout layer with a dropout rate of 0.6 is employed to further reduce the risk of overfitting. Randomly dropping units during the training phase, this layer encourages the model to learn robust features that do not rely on any single node. This step is crucial in ensuring that the model generalises well to unseen URLs, which is essential for real-world phishing detection.
- *Dense layers for classification:* the final layers of the model are fully connected (dense) layers. The first dense layer has 32 units with a ReLU activation function, further refining the extracted features. This dense layer also includes ℓ_2 regularisation to enhance generalisation. The output layer, which is the final layer of the model, has units corresponding to the number of classes, with a softmax activation function for multi-class classification. The softmax function provides a probability distribution over all classes, allowing the model to predict the likelihood of each URL being phishing or legitimate.

The proposed methodology combines CNN, LSTM, and attention mechanisms to capture spatial and sequential patterns in phishing URLs. This hybrid architecture enhances detection accuracy by focusing on high-impact features, making it a robust and effective tool for phishing detection.

4.4 Compilation

The model is compiled using categorical crossentropy as the loss function and 'adam' as the optimiser. The loss function is given by equation (8):

$$L = -\sum \left(y_{\text{actual}} \cdot \log(y_{\text{estimated}}) \right) \tag{8}$$

where y_{actual} and $y_{\text{estimated}}$ are the actual and estimated labels. The Adam optimiser is a popular optimisation technique that handles sparse gradients on complex and noisy challenges, used frequently in DL for computer vision and NLP (Zhang, 2018).

4.5 Overfitting

Overfitting occurs when a model excels on training data but underperforms on unseen data, often due to high model complexity or small dataset size. Early stopping counters this by ending optimisation after a certain point, using a part of training data to assess error rates, and stopping when the error begins to rise. It acts as a regulariser, limiting the parameter space.

5 Performance evaluation

This section provides an in-depth overview of the datasets used, the DL techniques implemented, and the metrics applied to evaluate the model's effectiveness.

5.1 Evaluation metrics

This part details the metrics used to assess the DL techniques applied. Generally, these metrics are employed to evaluate the effectiveness of ML classification and prediction algorithms. In this specific research, the predictive outcomes were evaluated using criteria such as precision, recall, the confusion matrix, and the overall accuracy to assess the system's performance.

- Precision: precision quantifies the number of phishing webpages that were correctly classified as phishing sites among all the predicted ones. It is mathematically represented as shown in equation (9):

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}} \tag{9}$$

- Recall: recall measures the ratio of correct phishing URL predictions relative to all the URLs in the dataset. It is expressed in equation (10):

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \tag{10}$$

- Accuracy: accuracy is a metric that illustrates the percentage of correct classifications compared to the total actual classes in the dataset. It is calculated as in equation (11):

$$\text{Accuracy} = \frac{\text{TruePositive} + \text{TrueNegative}}{\begin{array}{c}\text{TruePositive} + \text{TrueNegative}\\ + \text{FalseNegative} + \text{FalsePositive}\end{array}} \tag{11}$$

- F1-score: the F1-score is the harmonic mean of the model's precision and recall, combining these two aspects into one number. It is formulated as in equation (12):

$$\text{F1-Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{12}$$

This collection of metrics offers a multifaceted evaluation of the model's performance, addressing various facets of precision, recall, accuracy, and overall efficiency in predicting phishing URLs.

5.2 Datasets

In our phishing URL classification research, we focused on using up-to-date datasets to ensure relevant and accurate results. We sourced 4,000 legitimate URLs from the Ebbu (2017) dataset and 2,200 phishing URLs from the PhishTank platform, as referenced in Phishtank (2019). Together, these URLs formed a comprehensive dataset of 6,200 entries, balancing legitimate and phishing URLs for effective classification. We split the dataset into an 80: 20 ratio, allocating 4,960 URLs for training and 1,240 for testing. This split provided ample data for model training while preserving a significant portion for testing, enabling thorough validation and performance evaluation of our phishing detection models.

5.3 Algorithms for URLs classification

Phishing URL classification has been substantially improved through developments in ML and DL technologies. These algorithms leverage extensive datasets to constantly refine their models, achieving more precise and accurate identification of phishing URLs.

5.3.1 ML algorithms

In this study, we evaluated the effectiveness of seven distinct six ML algorithms, comparing their performance across various tasks.

- *Naive Bayes algorithm:* the naive Bayes algorithm is particularly effective in phishing URL detection due to its ability to handle a large number of features efficiently. It classifies URLs by calculating the probability of a URL being phishing or legitimate, based on the features extracted from the URL. Its simplicity and speed make it suitable for real-time phishing detection, despite its assumption of feature independence, which might not always hold true in complex datasets (Aljahdalic et al., 2023).
- *Random forests:* random forests are robust against overfitting, making them suitable for phishing URL classification, where the feature set can be diverse and intricate. By constructing multiple decision trees and averaging their predictions, random forests ensure a more stable and accurate classification compared to individual decision trees. This ensemble method is capable of handling the nonlinear relationships often present in phishing URL datasets (Gupta et al., 2021).

- *K-nearest neighbours (K-NN):* the K-NN algorithm classifies URLs based on the similarity to their nearest neighbours in the feature space. It's particularly effective when there is a clear distinction or pattern in the dataset. However, its performance might be affected by the choice of 'K' and the dimensionality of the feature space, especially in large and complex datasets like those of phishing URLs (Ali et al., 2023).
- *Adaboost:* Adaboost, an adaptive boosting technique, is effective for phishing URL classification as it focuses on difficult-to-classify instances by adjusting the weights of training samples. It combines multiple weak classifiers to form a strong classifier, making it suitable for complex phishing URL detection tasks where simple models might fail (Sharma and Singh, 2022).
- *Decision trees:* decision trees are easy to interpret and can handle both categorical and numerical data, making them a good choice for phishing URL classification. They work by creating a tree-like model of decisions, which is particularly useful for understanding the key features that contribute to a URL being classified as phishing or legitimate. However, they can be prone to overfitting in complex datasets (Ahammad et al., 2022).
- *Support vector machines (SVM):* SVMs are powerful in phishing URL classification due to their ability to create nonlinear decision boundaries. They work by finding the optimal hyperplane that separates the classes in the feature space. SVMs are particularly useful when the dataset has a clear margin of separation, and they can be equipped with different kernel functions to handle the nonlinear nature of phishing URL features (Diki and Muhammad, 2022).

5.3.2 DL algorithms

In our research, we developed and evaluated three models using tailored datasets for phishing URL classification: an LSTM model, a CNN model, and a hybrid LSTM-CNN model with an Attention mechanism.

- *Long short-term memory:* the LSTM network, a type of recurrent neural network, is designed to handle sequential data effectively, making it suitable for processing URLs. An LSTM cell operates through several gates: the forget gate decides which information to discard, the input gate determines which new information to store, and the output gate generates the output based on the updated cell state. Key activation functions include the sigmoid function for gate activations and the hyperbolic tangent (tanh) function to regulate values in the cell state (Roy et al., 2022).
- *Convolutional neural network:* the CNN is structured to process data through multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply filters over input data, capturing spatial features and local patterns. Each convolution operation extracts feature maps that highlight different aspects of the input structure, followed by pooling operations, such as max pooling, to downsample the data and reduce computational complexity. The final output from the CNN layers is flattened and passed to a fully connected layer, which applies a SoftMax function for classification (Tang and Mahmoud 2022; Birthriya et al., 2024b).
- *Hybrid LSTM-CNN with attention mechanism:* the hybrid LSTM-CNN with attention combines the sequential processing capabilities of the LSTM with the spatial feature extraction of CNNs, enhanced by an attention mechanism that assigns weights to significant features. The attention mechanism enables the model to focus on the most relevant parts of the data, refining feature importance and improving classification accuracy. Leveraging both temporal and spatial information in URLs and focusing on critical features, the hybrid model aims to deliver a more robust phishing URL classification.

6 Experiment results

In our research, we employed specially curated datasets to develop and assess various models. We utilised a range of techniques, including ML algorithms, DL algorithms, and a composite approach combining LSTM and CNN with Attention mechanism. The aim was to compare these methods and identify the most effective and accurate model.

6.1 Experiment 1: ML models

In our experiments, we categorised the test features into three groups: PCA-based features, word2vec, and hybrid features, to assess the overall efficiency of the system. Each classification algorithm was tested across these feature types to evaluate their performance in various scenarios. In Table 2, different ML algorithms exhibit varied performances across feature sets. Decision tree and random forest algorithms show strong results, especially with PCA and hybrid features, achieving high precision, recall, F1-score, and accuracy, all above 93%. Adaboost, while performing well with PCA features, sees a notable drop in effectiveness with word2vec features. KNN, SVM, and naive Bayes maintain consistent performance across all features, particularly excelling with PCA and hybrid features. The hybrid approach generally provides a balanced performance across algorithms, often resulting in high accuracy and F1-scores, demonstrating its effectiveness in various ML contexts.

6.2 Experiment 2: DL and hybrid models

In Table 3, the LSTM-CNN with attention achieves the highest performance when using hybrid features, excelling in accuracy, precision, recall, F1-score, and AUC, while also maintaining a remarkably low false positive rate. This highlights the advantage of combining sequential and convolutional architectures with attention mechanisms to effectively capture complex patterns in phishing detection. The hybrid feature set's superior performance can be attributed to its ability to integrate diverse and complementary information, unlike Word2Vec and PCA features, which focus on narrower representations. PCA features provide moderate performance, offering a balance between simplicity and effectiveness, but they fall short of the hybrid feature set in terms of generalisation and robustness. On the other hand, Word2Vec features yield the weakest results across all models, with low accuracy and high false positive rates, reflecting their limited capacity to handle the complexity of phishing detection. Moreover, the training and testing times for hybrid features are consistently lower compared to PCA and Word2Vec in several cases, indicating their computational efficiency alongside their predictive strength. The results highlight the effectiveness of hybrid features and advanced architectures like LSTM-CNN with attention for high-performance phishing detection.

6.2.1 Training and validation analysis

This section examines the training and validation performance of the LSTM, CNN, and attention-based LSTM-CNN models using different feature sets: word2pec, PCA-transformed features, and the proposed hybrid features for phishing URL detection. The aim is to assess how each feature set influences model learning and generalisation, and to compare the effectiveness of these models in accurately classifying phishing URLs.

Figure 6 presents the training and validation accuracy and loss curves for the LSTM model across the three feature sets. The LSTM model shows superior performance when using the hybrid features, achieving a training accuracy exceeding 90% and a validation accuracy around 88%. This indicates strong learning and generalisation capabilities with minimal overfitting. The PCA-based features result in moderate performance, with training accuracy around 85% and validation accuracy close to 82%. Although effective, they are less informative than the hybrid features. In contrast, the word2vec features yield significantly lowers performance, with both training and validation accuracies were remaining around 60%. This suggests that word2vec features alone lack sufficient discriminatory information for effective phishing detection using the LSTM model. The loss trends support these findings: hybrid features achieve the lowest training and validation loss, stabilising near 0.2; PCA features stabilise around 0.3; and Word2Vec features remain high at around 0.6, indicating suboptimal convergence.

Figure 7 shows the training and validation accuracy and loss curves for the CNN model using the different feature sets. Consistent with the LSTM results, the hybrid features yield the highest performance for the CNN model, achieving a training accuracy above 90% and validation accuracy around 88%. This reflects effective learning and strong generalisation. The PCA features perform slightly lower, stabilising around 85% for training accuracy and 82% for validation accuracy. The Word2Vec features continue to show the lowest performance, with both training and validation accuracies around 60%, reflecting insufficient discriminatory capability. The loss plots corroborate these observations: hybrid features achieve the lowest loss values (around 0.2), PCA features stabilise slightly higher (around 0.3), and Word2Vec features remain relatively high (around 0.6), indicating poorer convergence and model fit.

Figure 8 depicts the training and validation accuracy and loss for three feature sets: hybrid features, PCA features, and word2vec features using an LSTM+CNN model with attention over 100 epochs. The hybrid features demonstrate the highest accuracy, stabilising above 90% after around 10 epochs, with minimal and stable loss, reflecting strong performance and generalisation. PCA Features achieve moderate performance, stabilising at about 85% accuracy, with slightly higher loss compared to hybrid features. word2vec Features show the lowest accuracy, levelling out around 60%, and maintain significantly higher loss throughout the training process, indicating limited learning capability. These findings highlight that hybrid features, which integrate diverse feature types, offer a more effective representation for phishing detection, outperforming single-domain feature sets like PCA and word2vec.

Overall, this analysis reinforces that the hybrid feature approach provides the most comprehensive feature representation for phishing detection. The combination of different feature extraction techniques leads to higher accuracy and better convergence, highlighting the importance of comprehensive feature engineering. Future work may focus on refining feature selection and model tuning to reduce false positives and enhance the balance between sensitivity and specificity, ultimately improving the robustness of phishing detection systems.

6.2.2 Confusion matrix analysis

The comparative analysis for confusion matrix of the LSTM, CNN, and LSTM-CNN with Attention models reveals notable variations in performance across different feature sets for phishing URL detection.

In Figure 9, the confusion matrices illustrate the LSTM model's varying performance across feature sets for phishing URL detection. word2vec features show limited precision, with high misclassification rates and few phishing URLs correctly identified. PCA features improve accuracy by capturing more phishing patterns, yet misclassification of legitimate URLs remains significant, leading to a high false positive rate. Hybrid features further enhance phishing detection, accurately identifying more phishing URLs but exacerbating the false positive rate by misclassifying many legitimate URLs. While PCA and hybrid features outperform Word2Vec, the consistently high false positives highlight the need for refined feature engineering and model optimisation to achieve balanced and reliable classification.

Table 2 Performance of ML algorithm (score in %)

Model	*Features*	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Decision tree	PCA	96.4	97.7	97.1	97.02
	Word2Vec	94.4	69.5	80.0	82.48
	Hybrid	93.3	97.3	95.3	*95.14*
Adaboost	PCA	90.8	96.3	93.5	93.24
	Word2Vec	93.6	53.6	68.2	74.74
	Hybrid	91.5	94.0	92.7	*92.53*
KNN	PCA	94.0	97.7	95.8	95.67
	Word2Vec	95.5	69.7	80.6	83.01
	Hybrid	94.6	97.4	96.0	*95.86*
Random forest	PCA	97.0	99.0	98.0	97.98
	Word2Vec	95.8	69.7	80.7	83.14
	Hybrid	95.3	97.6	96.4	*96.36*
SVM	PCA	92.8	97.5	95.1	94.92
	Word2Vec	94.7	69.7	80.3	82.71
	Hybrid	92.3	97.2	94.7	*94.48*
Naive Bayes	PCA	94.0	97.7	95.8	95.67
	Word2Vec	95.5	69.7	80.6	83.01
	Hybrid	94.6	97.4	96.0	*95.86*

Table 3 Performance of DL algorithm (score in %)

Model	*Features*	*Training time*	*Testing time*	*Accuracy*	*Precision*	*Recall*	*F1-score*	*AUC*	*FPR*
LSTM	Word2Vec	38.97	0.36	56.95	71.48	56.50	47.57	0.56	43.49
	PCA	89.38	0.33	89.17	89.29	89.13	89.15	0.89	10.86
	Hybrid	34.42	0.21	*94.50*	94.50	94.50	94.50	0.94	5.49
CNN	Word2Vec	14.33	0.18	56.93	73.56	55.18	45.26	0.55	44.81
	PCA	21.12	0.14	85.67	85.70	85.76	85.67	0.85	14.23
	Hybrid	11.49	0.05	*99.66*	99.75	99.75	99.75	0.99	0.33
LSTM-CNN with attention	Word2Vec	34.38	0.55	56.03	76.67	55.77	45.15	0.55	44.22
	PCA	71.04	0.54	88.92	88.20	88.90	88.90	0.89	11.09
	Hybrid	28.02	0.17	*99.92*	99.92	99.91	99.92	0.99	0.08

Figure 6 LSTM-training and validation accuracy and loss (see online version for colours)

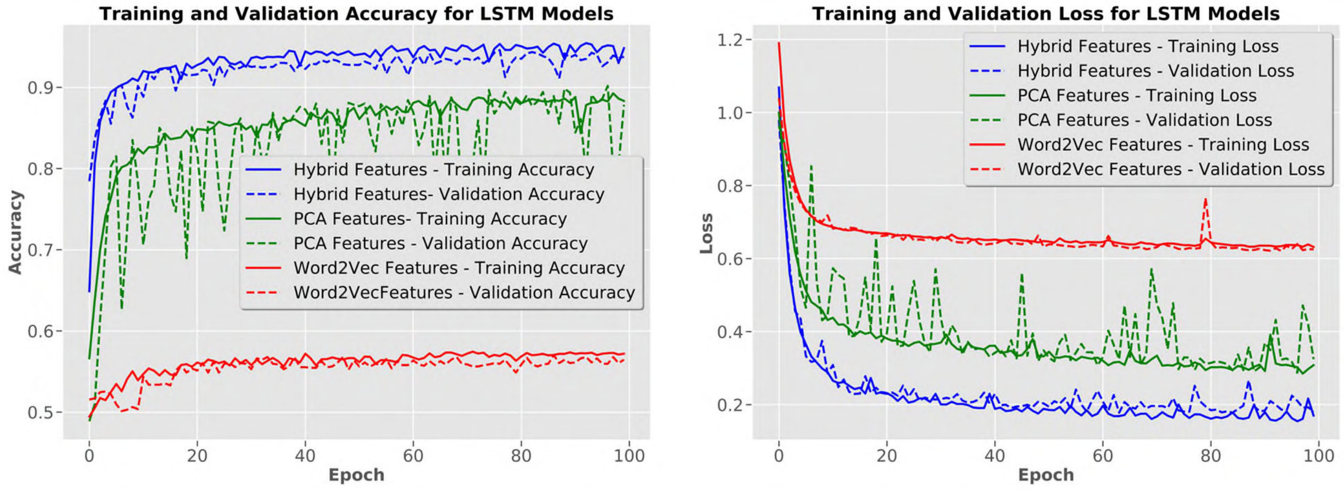


Figure 7 CNN-training and validation accuracy and loss (see online version for colours)

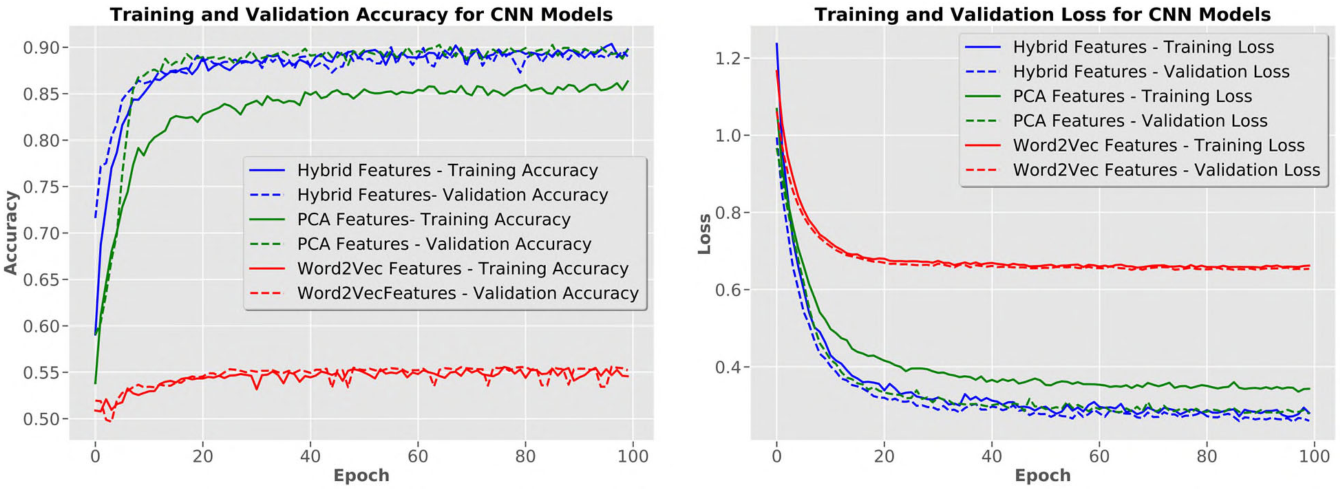


Figure 8 LSTM+CNN with attention-training and validation accuracy and loss (see online version for colours)

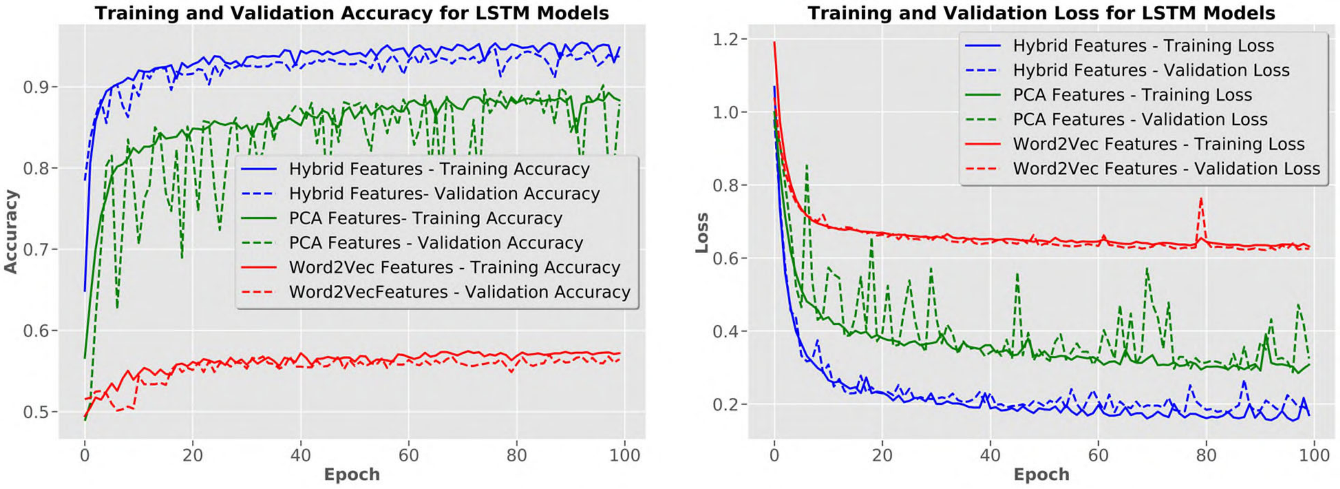


Figure 9 Confusion matrix for LSTM models (see online version for colours)

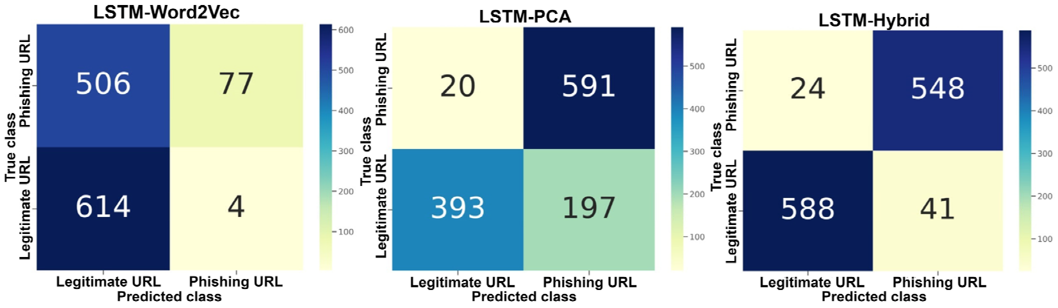


Figure 10 Confusion matrix for CNN models (see online version for colours)

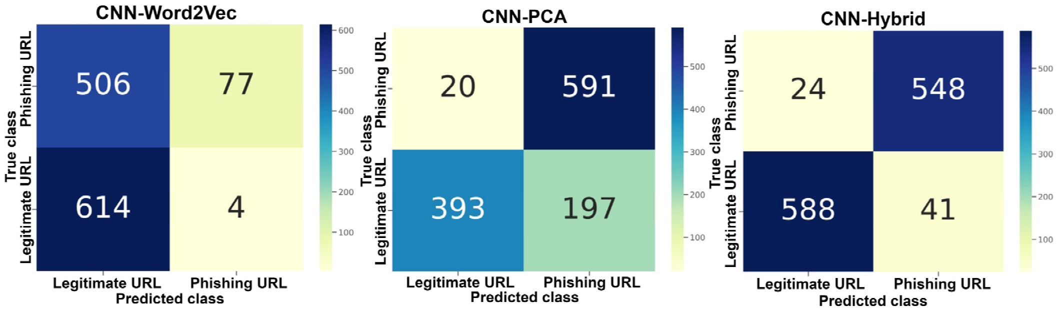


Figure 11 Confusion matrix for LSTM-CNN with attention model (see online version for colours)

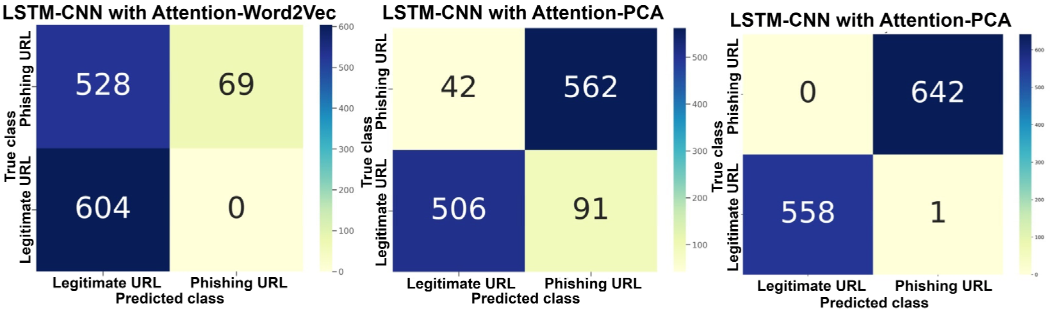


Table 4 Comparative analysis of existing DL approaches for phishing detection

References	*Technique*	*Dataset*	*Feature engineering*	*Testing time (sec)*	*Accuracy (%)*
Adebowale et al. (2020)	LSTM and CNN hybrid	PhishTank, Common Crawl	A combination of image, text, and frame features	24.5	93.28
Janet and Reddy (2020)	LSTM, CNN	PhishTank, VirusTotal	-	-	96
Wei et al. (2020)	CNN	Common Crawl	One-hot character-level encoding	0.002/URL	99.79
Jawade and Ghosh (2021)	CNN	ISCX-URL2016	-	-	99
Tang and Mahmoud (2022)	CNN, CAE	ISCX-URL2016	Character-level embeddings, word level embeddings, contextual embeddings using FastText	-	88
Ozcan et al. (2023)	Hybrid DNN, BiLSTM	Ebbu2017, PhishTank, Marchal2014	Character embedding, hand-crafted NLP, lexical and statistical URL features	-	99.21
Ujah-Ogbuagu et al. (2024)	CNN and LSTM	UCL spoofing website and PhishTank	Word embedding	-	98.9
Proposed method	LSTM, CNN with attention	Phishtank, Ebbu (2017)	Hand-crafted NLP and PCA optimised features	0.21	99.92

In Figure 10, the CNN model's performance exhibits significant variation depending on the feature set chosen for phishing URL detection. Word2Vec features alone result in difficulties for the model in distinguishing phishing from legitimate URLs, leading to frequent misclassifications and indicating an insufficient level of detail for accurate detection. A slight improvement appears through PCA features, as the model captures some phishing patterns; however, a high false positive rate persists, misclassifying many legitimate URLs as phishing. The hybrid feature set enhances phishing detection effectiveness but leads to substantial overestimation, causing most legitimate URLs to be incorrectly labelled as phishing. These findings suggest that, although comprehensive, hybrid features may render the model overly sensitive, emphasising the need for refinement in feature selection or model calibration to achieve a better balance between sensitivity and specificity.

In Figure 11, the confusion matrices for the LSTM-CNN with attention model highlight how different feature sets influence phishing URL detection. The model shows limited effectiveness when employing word2vec features, accurately detecting some phishing URLs but frequently misclassifying legitimate URLs, suggesting insufficient discriminatory capability. Classification performance improves with PCA features, as the model correctly identifies a larger number of phishing URLs; however, many legitimate URLs remain misclassified as phishing, leading to a high false positive rate. The hybrid feature set enhances phishing detection further but lacks balance, as most legitimate URLs are wrongly classified as phishing. These results indicate that although the attention-based LSTM-CNN model with hybrid features exhibits strong sensitivity to phishing, it still faces challenges in clearly distinguishing between phishing and legitimate URLs. This finding underscores the need for further optimisation in feature selection or model tuning to reduce false positives and achieve a balanced classification.

7 Comparison with existing phishing detection methods

Table 4 highlights the progress made in DL techniques for phishing detection, focusing on the strengths and challenges of current methods. Hybrid models such as LSTM-CNN and CNN-LSTM leverage sequential learning for capturing temporal patterns and convolutional layers for local feature extraction, making them effective for detecting complex phishing patterns (Adebowale et al., 2020; Ujah-Ogbuagu et al., 2024). These approaches achieve high accuracy but often vary in efficiency, with some suffering from longer testing times. Techniques like CNN with one-hot encoding have demonstrated exceptional accuracy and computational efficiency, suitable for high-volume URL analysis (Wei et al., 2020). Models integrating character-level embeddings, word-level embeddings, and contextual embeddings show the importance of sophisticated feature engineering in improving detection accuracy (Tang and Mahmoud, 2022). Other approaches employ hybrid feature engineering methods, such as combining lexical, statistical, and handcrafted NLP features, to improve model robustness (Ozcan et al., 2023). Advanced embeddings like FastText further enhance semantic representation, providing significant performance gains. However, a key challenge observed in earlier methods is the lack of emphasis on real-time detection, as higher accuracy often comes with increased testing times.

The proposed method effectively combines LSTM, CNN, and attention mechanisms to enhance phishing detection. The strengths of sequential learning and attention-based feature extraction, it achieves high accuracy while maintaining computational efficiency. Robustness against diverse phishing attempts is ensured through a hybrid feature engineering strategy that incorporates NLP features and PCA-based dimensionality reduction. Testing time is significantly reduced compared to existing methods, making it highly suitable for real-time applications.

8 Conclusions and further research directions

The research presented confirms that integrating LSTM, CNN, and attention mechanisms with a robust hybrid feature set markedly improves phishing URL detection accuracy. The hybrid features, combining TF-IDF word vectors, PCA-reduced components, and additional NLP-based features, enable the model to capture intricate spatial and sequential patterns inherent in phishing URLs. Experimental results show that the proposed LSTM-CNN with attention model achieves superior performance across all evaluated metrics, notably attaining a 99.92% accuracy rate. This performance surpasses that of traditional ML algorithms and other DL models tested, highlighting the efficacy of the hybrid approach. The study underscores the importance of comprehensive feature engineering and the integration of advanced DL architectures in developing effective phishing detection systems. These insights contribute to the advancement of cybersecurity measures aimed at mitigating the risks associated with phishing attacks. Future research could integrate phishing detection models with browser extensions or e-mail clients for automatic, real-time protection. Developing adaptive algorithms to counteract emerging phishing techniques in dynamic online environments is essential. Additionally, leveraging ML to analyse user behaviour patterns could enhance detection accuracy.

References

Adebowale, M., Lwin, K. and Hossain, M. (2020) 'Intelligent phishing detection scheme using deep learning algorithms', *Journal of Enterprise Information Management*, Vol. 36, No. 3, pp.747–766.

Ahammad, S.H., Kale, S.D., Upadhye, G.D., Pande, S.D., Babu, E.V., Dhumane, A.V. and Bahadur, M.D.K.J. (2022) 'Phishing URL detection using machine learning methods', *Advances in Engineering Software*, Vol. 173, p.103288.

Ali, A., Hamraz, M., Gul, N., Khan, D.M., Aldahmani, S. and Khan, Z. (2023) 'A k-nearest neighbour ensemble via extended neighbourhood rule and feature subsets', *Pattern Recognition*, Vol. 142, No. 2023, p.109641.

Aljahdalic, A.O., Banafee, S. and Aljohani, T. (2023) 'URL filtering using machine learning algorithms', *Information Security Journal: A Global Perspective*, Vol. 33, No. 3, pp.193–203.

Aljofey, A., Jiang, Q., Qu, Q., Huang, M. and Niyigena, J. (2020) 'An effective phishing detection model based on character-level convolutional neural network from URL', *Electronics*, Vol. 9, No. 9, p.1514.

Almomani, A., Alauthman, M., Shatnawi, M.T., Alweshah, M., Alrosan, A., Alomoush, W. and Gupta, B.B. (2022) 'Phishing website detection with semantic features based on machine learning classifiers: a comparative study', *International Journal on Semantic Web and Information Systems (IJSWIS)*, Vol. 18, No. 1, pp.1–24.

APWG (2024) *Phishing Activity Trends Report: 1st Quarter 2020 to 2nd Quarter 2024*, Anti-Phishing Working Group.

Asiri, S., Xiao, Y. and Alzahrani, S. (2024a) 'Towards improving phishing detection system using human in the loop deep learning model', in *Proceedings of the 2024 ACM Southeast Conference*, pp.77–85.

Asiri, S., Xiao, Y., Alzahrani, S. and Li, T. (2024b) 'PhishingRTDS: a real-time detection system for phishing attacks using a deep learning model', *Computers & Security*, Vol. 141, No. 2024, p.103843.

Birthriya, S.K., Ahlawat, P. and Jain, A.K. (2024a) 'A comprehensive survey of social engineering attacks: taxonomy of attacks, prevention, and mitigation strategies', *Journal of Applied Security Research*, pp.1–49.

Birthriya, S.K., Ahlawat, P. and Jain, A.K. (2024b) 'Phishing URL detection using deep Q-networks with convolutional neural networks', in *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, IEEE, May, pp.1–6.

Diki, W. and Muhammad, N. (2022) 'Website phishing detection application using support vector machine (SVM)', *Journal of Information Technology and Its Utilization*, Vol. 5, No. 1, pp.18–24.

Do, N., Selamat, A., Krejcar, O., Yokoi, T. and Fujita, H. (2021) 'Phishing webpage classification via deep learning-based algorithms: an empirical study', *Applied Sciences*, Vol. 11, No. 19, p.9210.

Ebbu (2017) *Ebbu 2017 Phishing Dataset*. GitHub Repository [online] https://github.com/ebubekirbbr/pdd/tree/master/input (accessed 24 July 2018.).

Gupta, B.B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A. and Chang, X. (2021) 'A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment', *Computer Communications*, Vol. 175, No. 2021, pp.47–57.

Jain, A.K. and Gupta, B.B. (2022) 'A survey of phishing attack techniques, defence mechanisms and open research challenges', *Enterprise Information Systems*, Vol. 16, No. 4, pp.527–565.

Janet, B. and Reddy, S. (2020) 'Anti-phishing system using LSTM and CNN', in *Proceedings of the 2020 IEEE International Conference for Innovation in Technology (INOCON)*, Bangalore, India, pp.1–5.

Jawade, J.V. and Ghosh, S.N. (2021) 'Phishing website detection using Fast.ai library', in *Proceedings of the 2021 International Conference on Communication Information and Computing Technology (ICCICT)*, Mumbai, India, 25–27 June.

Jishnu, K.S. and Arthi, B. (2023) 'Phishing URL detection by leveraging RoBERTa for feature extraction and LSTM for classification', in *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, IEEE, August, pp.972–977.

Kabra, B. and Nagar, C. (2023) 'Convolutional neural network based sentiment analysis with TF-IDF based vectorization', *Journal of Integrated Science and Technology*, Vol. 11, No. 3, pp.503–503.

Koppula, H.S., Leela, K.P., Agarwal, A., Chitrapura, K.P., Garg, S. and Sasturkar, A. (2010) 'Learning URL patterns for webpage de-duplication', in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, February, pp.381–390.

Korkmaz, M., Kocyigit, E., Sahingoz, O.K. and Diri, B. (2021) 'Phishing web page detection using n-gram features extracted from URLs', in *Proceedings of the 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara, Turkey, pp.1–6.

Kumar Birthriya, S. and Jain, A.K. (2022) 'A comprehensive survey of phishing email detection and protection techniques', *Information Security Journal: A Global Perspective*, Vol. 31, No. 4, pp.411–440.

Kumar, J. (2023) 'Detecting URL phishing using BERT and DistilBERT classifiers', in *International Conference on Soft Computing for Problem-Solving*, August, pp.613–624, Springer Nature Singapore, Singapore.

Liang, Y., Deng, J. and Cui, B. (2020) 'Bidirectional LSTM: an innovative approach for phishing URL identification', in *Innovative Mobile and Internet Services in Ubiquitous Computing: Proceedings of the 13th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2019)*, Springer International Publishing, pp.326–337.

Opara, C., Wei, B. and Chen, Y. (2020) 'HTMLPhish: enabling phishing web page detection by applying deep learning techniques on HTML analysis', in *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, pp.1–8.

Ozcan, A., Catal, C., Donmez, E. and Senturk, B. (2023) 'A hybrid DNN-LSTM model for detecting phishing URLs', *Neural Computing and Applications*, Vol. 35, pp.4957–4973.

Phishtank (2019) *Phishtank Homepage* [online] https://www.phishtank.com (accessed 7 July 2019).

Roy, S.S. et al. (2022) 'Multimodel phishing URL detection using LSTM, bidirectional LSTM, and GRU models', *Future Internet*, Vol. 14, No. 11, p.340.

Sánchez-Paniagua, M. et al. (2022) 'Phishing URL detection: a real-case scenario through login URLs', *IEEE Access*, Vol. 10, No. 2022, pp.42949–42960.

Sharma, B. and Singh, P. (2022) 'An improved anti-phishing model utilizing TF-IDF and AdaBoost', *Concurrency and Computation: Practice and Experience*, Vol. 34, No. 26, p.e7287.

Shrestha, A. and Mahmood, A. (2019) 'Review of deep learning algorithms and architectures', *IEEE Access*, Vol. 7, No. 2019, pp.53040–53065.

Song, F., Guo, Z. and Mei, D. (2010) 'Feature selection using principal component analysis', in *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*, IEEE, Vol. 1.

Tajaddodianfar, F., Stokes, J.W. and Gururajan, A. (2020) 'Texception: a character/word-level deep learning model for phishing URL detection', in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, pp.2857–2861.

Tang, L. and Mahmoud, Q.H. (2022) 'A deep learning-based framework for phishing website detection', *IEEE Access*, Vol. 10, pp.1509–1521.

Ujah-Ogbuagu, B.C., Akande, O.N. and Ogbuju, E. (2024) 'A hybrid deep learning technique for spoofing website URL detection in real-time applications', *Journal of Electrical Systems and Information Technology*, Vol. 11, No. 1, p.7.

Wei, W., Ke, Q., Nowak, J., Korytkowski, M., Scherer, R. and Woźniak, M. (2020) 'Accurate and fast URL phishing detector: a convolutional neural network approach', *Computer Networks*, Vol. 178, No. 2020, p.107275.

Young, T., Hazarika, D., Poria, S. and Cambria, E. (2018) 'Recent trends in deep learning-based natural language processing', *IEEE Computational Intelligence Magazine*, Vol. 13, No. 3, pp.55–75.

Zhang, Z. (2018) 'Improved Adam optimizer for deep neural networks', *in 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, IEEE.