Journal of Basrah Researches (Sciences) **52**(1), 170 (2026)
DOI: https://doi.org/10.56714/bjrs.52.1.13

Interpretable Phishing Email Detection based on CNN-Word2vec Models and LIME Explanations

Bakr Thamer Hammoudi[1,*], Omar Z. Akif[2]

[1]Informatics Institute for Postgraduate Studies, University of Information Technology and Communications (UoITC), Baghdad, Iraq.
[2]Department of Computer Science, College of Education for Pure Science (Ibn al-Haitham), University of Baghdad, Iraq.

ARTICLE INFO

Received 25 February 2026
Revised 24 March 2026
Accepted 24 March 2026
Published 30 June 2026

Keywords:
Cybersecurity, Phishing Email Detection, Convolutional Neural Network, Word2Vec, Word Embedding, Fasttext, TF-IDF

Citation: B. Th. Hammoudi, O. Z. Akif., J. Basrah Res. (Sci.) **52**(1), 170 (2026). DOI:https://doi.org/10.56714/bjrs.52.1.13

ABSTRACT

phishing emails are a major cybersecurity threat which primarily attempt to steal private information or lose money. Attackers counterfeit well-known organizations and devise sophisticated tricks to remain unnoticed, so it remains difficult for detection systems even with ample academic research. This study focuses on assessing several methods of vectorizing text and deep learning architecture in classifying emails. The main contribution of this study is to enhance the CNN with Word2vec embedding for phishing email classification. To ensure the robustness, we built a variety of benchmark dataset by merging eight public sources and promoted the feature extraction by some specific modifications of convolutional filters. Extensive preprocessing like data augmentation to handle class imbalance was used. The proposed model was longitudinally compared against TF-IDF, fastText, Word Embedding and traditional machine learning methods. Experimental results show that our proposed model achieved the highest accuracy of 99.05%, outperforming word embedding (98.72%), TF-IDF (98.85%) and fastText (98.99%). The Convolutional Neural Network (CNN) with word2vec proved highly effective in classifying phishing emails. To enhance transparency and trust, the LIME framework was incorporated, offering interpretability via contribution scores. This clarifies the models decision-making process while reducing methodological inconsistencies.

1. Introduction

Phishing is a social engineering cyberattack where cyber criminals use deceptive bulk mail offering links to fake domains and phishing websites through manipulative tactics to divert interfaces that mimic legitimate login pages with the goal to obtain sensitive information illegally [1]. As illustrated in figure 1, this is a phishing scam through which the attacker carries out the theft.

*Corresponding author email: baker.1994thth@gmail.com

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq

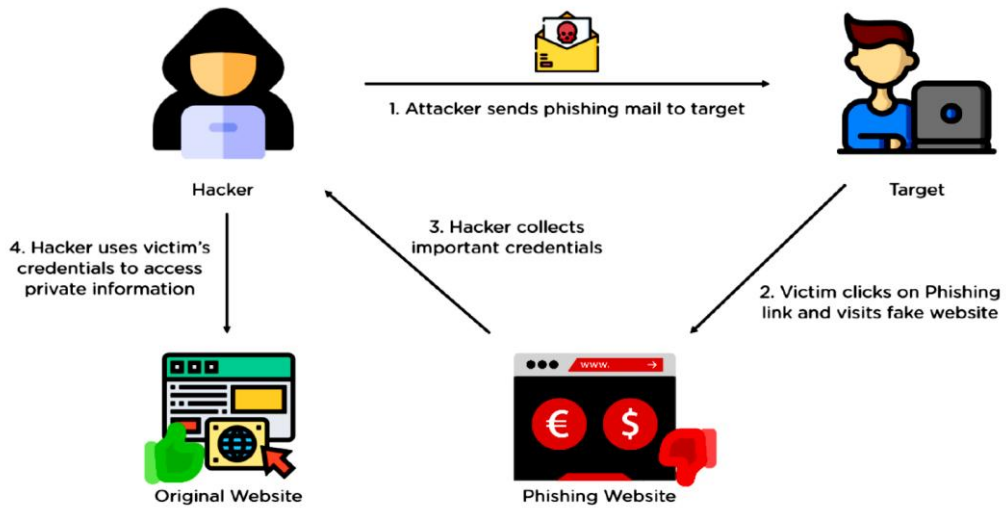

Fig.1. Phishing attack in cybersecurity [2]

According to the APWG.org website, [3] statistics show that in the first quarter of 2025, the number of phishing attacks reached 1,003,92 attacks. In the second quarter of 2025, phishing attacks increased from 1,003,92 attacks to 1,130,393 attacks [4]. Despite rule-based methods being in good reason, their failure to keep up with the diversity of adversary tactics has led to the use of machine learning and deep text representations which have shown substantial gains in terms of accuracy and false alarms [5]. In order to achieve this improvement, it is essential to build a unified processing pipeline, including the ability to clean text and extract features from the title, URL and metadata, and to use contextual models that can grasp changing semantics and styles [6]. However, the real world still faces difficult issues like class imbalance, time distribution change and multilingualism, which in turn require well-designed learning methods and time-sensitive evaluation to achieve stable performance [7]. The model decision interpretability and the compliance with governance and privacy requirements are also essential for model institutional adoption, since they enable the trust development and operational and legal risk mitigations [8]. Based on these motivations, this work proposes a complete architecture including a preprocessing pipeline, contextual modelling with strong training approach and careful temporal evaluation, as well an analysis of both the interpretability and ethical issues, and thus giving the system the foundations for large-scale practical implementation [9].
In this study, the main contributions are constructing a comprehensive phishing dataset by aggregating eight public sources to enhance model generalizability; furthermore, proposing an optimized CNN with word2vec framework with enhanced convolutional filters for superior feature learning; on the other hand, integrating LIME-based Explainable AI to ensure model transparency and interpretability, thereby addressing the black-box limitation of deep learning models.

2. Related work

The section discusses previous studies and processes used to classify email with the use of the artificial intelligence approach such as machine and deep learning.
A comparative study [5] evaluated machine learning and deep learning models, including Naïve Bayes, SVM, Decision Tree, CNN and LSTM, for phishing email classification using one-hot encoding and word embedding. The CNN model utilizing word embedding achieved the highest accuracy of 96.34%, surpassing the same model with one-hot encoding which attained 95.97% accuracy.
In the study [10] investigated a machine learning-based algorithm employing Support Vector Machine (SVM) and Naïve Bayes (NB) models for phishing email prediction. The system reached a high rate of performance with the accuracy of 99.96% and error rate of 0.04%. However, it has two-second prediction time per message to classify messages and is limited by the inability to interpret and an unrepresentative dataset of attack vectors.

In [11], a researcher suggested a phishing detection technique using word embedding and machine learning algorithms; Random Forest and fastText with 99.5% accuracy and decreased the false positive. However, the study does not specify the prediction time required for classifying an email as legitimate or phishing. Furthermore, the proposed system lacks interpretability and relies on a constrained and outdated dataset, which limits the generalizability of its findings.

The research in [7] used a cuckoo search optimized greedy recurrent unit (GRU) model with N-gram features for phishing detection which resulted in an accuracy of 99.72%. However, the robustness of the model is limited by its lack of explainability, reliance on a constrained and outdated single-type dataset and its architecture which limits the ability to extract features comprehensively.

The hybrid model in [12] combines a DNN and a BiLSTM to detect phishing websites through structural URL analysis, attaining 99.21% accuracy. Nevertheless, the way of introducing the semantics of the message content and the limitation of the approach include the message content that is outdated, URLs only.

Researcher in [6] used various deep learning architectures such as convolution neural network (CNN), recurrent neural network (RNN), Long Short- Term Memory (LSTM) and BERT based binary encoding combined with natural language processing technique for extracting features from an email text. Hybrid BERT-LSTM model performed best with an accuracy of 99.61% thus proving that increased datasets make better models. While the BERT got great precision, the LSTM component showed a fair balance between recall and precision. However, the research did not incorporate explainability techniques, relied on a limited dataset comprising only four categories of legacy legitimate and phishing emails and omitted the reporting of per-message prediction time.

The research in [13] used TF-IDF and word2vec for textual features extraction and Support vector Machine (SVM) model provided 99.1% accuracy with Explainable AI for interpretability. Despite these strengths, the study is limited by a limited dataset of small volume and varied attack features. Furthermore, the system is not specific about prediction time on email classification and focuses the detection only on textual data and leaves aside the link analysis, because it has a low prediction value.

The study in [9] proposed a hybrid deep learning model using convolutional neural network (CNN) and Bidirectional Gated Recurrent Unit (Bi-GRU) for the detection of phishing emails with an accuracy of 99.68%. However, a deeper model caused overfitting and did not perform well. The research is further mitigated by a small-scale dataset, which lacks contemporary attack vectors and fails to offer model interpretability in classification decisions. Additionally, the lightweight nature of the model is designed for rapid inference in lightweight applications, rather than enterprise deployment applications that require large-scale inference.

The extensive study in [14] tested fourteen machine and deep learning architecture on ten datasets, testing transformer-based architectures such as BERT and RoBERTa to be most accurate (~99%). However, the datasets used are outdated and lack enthusiasm of modern day attack vectors, hindering the detection of modern tactics. The framework also lacks techniques to interpret its classifications and uses separate models focused on comparison instead of an integrated hybrid limitations. These limitations make the models unworkable for practical enterprise deployment.

Employing Bidirectional LSTM (BiLSTM) network architecture and based on character and word level analysis of URLs using embeddings and attention mechanism, [15] achieved 99.32% accuracy and 0.9986 AUC score. While effective, the models reliance on four separate and historical dataset which are not integrated - limits its exposure to modern attack vectors and tactics. Furthermore, the proposed system lacks the interpretability techniques, providing no transparency into the reason behind a messages classification as legitimate or phishing. Its character-based embedding approach has a major limitation of capturing mainly structural URL patterns, and not being able to intrinsically model deeper semantic relationships within its content.

Table 1. Summary of machine learning methods in detecting phishing emails.

Ref	Algorithm	Datasets	Results
[5]	NB	AppRiver, 3,416 phishing and 14,950 legitimate.	Accuracy=75.72%
	SVM		Accuracy=93.88%
	DT		Accuracy=94.26%

[8]	LR RF	Enron dataset, 16563 legitimate and 17188 phishing.	Accuracy=96.16% Accuracy=88.69%
[10]	SVM+NB	2541 phishing and 2500 legitimate.	Accuracy=99.96%
[11]	RF with FastText-CBOW	consists of three main datasets: Dataset 1: 15,430 messages (6,295 legitimate messages, 9,135 phishing messages). Dataset 2: 27,405 messages (18,270 legitimate messages, 9,135 phishing messages). Dataset 3: 27,256 messages (18,270 legitimate messages, 8,986 phishing messages).	DS-1 Accuracy=99.50% DS-2 Accuracy=99.39% DS-3 Accuracy=99.18%
[13]	SVM	Enron, Ling, CEAS, SpamAssassin, Nazario, Nigerian-Fraud 42891 phishing and 39595 legitimate.	Accuracy=99.10%
[14]	NB LR SGD XGBoost DT RF Extra Tree	1-Ling 458 spam and 2401 legitimate emails. 2-Enron 13, 976 spam and 15, 791 ham emails. 3-SpamAssassin 1670 spam and 3928 ham emails. 4- TREC-05 (2005) 20, 643 spam and 31, 842 ham emails. 5-TREC-06 (2006) 3642 spam and 12, 079 ham emails. 6-TREC-07 (2007) 29, 093 spam and 24, 338 ham emails. 7-CEAS (2008) 21, 839 spam and 16, 853 ham emails. 8-Nazario_5 1469 phishing and 1482 legitimate emails. 9-Nigerian_5 1998_2007 contains 1586 scam emails and 2967 legitimate emails. 10-Merged DS 94,376 (45.81%) are phishing emails, and 111,681 (54.19%) are safe emails.	Accuracy=99.23% Accuracy=99.04% Accuracy=99.48% Accuracy=98.80% Accuracy=98.23% Accuracy=99034% Accuracy=99.28%

Following the review of prior research on machine learning techniques, previous studies concerning deep learning methodologies will be examined in this section.

Table 2. Summary of deep learning methods in detecting phishing emails.

Ref	**Algorithm**	**Datasets**	**Results**
[5]	CNN CNN with word embedding	AppRiver, 3,416 phishing and 14,950 legitimate.	Accuracy=95.97% Accuracy=96.34%
[6]	CNN RNN	UCI machine learning repository and SpamAssassin as phishing and Enron as legitimate.	Accuracy=98.74% Accuracy=98.58%
[7]	CNN THEMIS ICSOA-DLPEC	Enron email dataset, 7,781 legitimate and 999 phishing.	Accuracy=97.58% Accuracy=99.34% Accuracy=99.72%

[8]	CNN RNN DNN+BiLSTM	Enron dataset, 16563 legitimate and 17188 phishing.	Accuracy=96.39% Accuracy=87.93% Accuracy=98.69%
[9]	CNN CNN+LSTM CNN+BiLSTM CNN+GRU CNN+BiGRU	Phishing corpus 2,278 phishing + SpamAssassin 4,150 legitimate.	Accuracy=98.87% Accuracy=99.23% Accuracy=99.34% Accuracy=99.01% Accuracy=99.68%
[12]	DNN+LSTM DNN+BiLSTM	PhishTank and Ebbu2017. (36,400 legitimate and 37,175 phishing).	Accuracy=98.79% Accuracy=99.21%
[14]	CNN RNN LSTM	Ling: 458 spam and 2401 legitimate emails. Enron: 13, 976 spam and 15, 791 ham emails. SpamAssassin: 1670 spam and 3928 ham emails. TREC 05/06/07: 53,378 spam and 68,259 Ham emails. CEAS2008: 21, 839 spam and 16, 853 ham emails. Nazario_5: 1469 phishing and 1482 legitimate emails. Nigerian_5: 1998-2007 contains 1586 scam emails and 2967 legitimate emails. Merged Dataset: 94,376 (45.81%) are phishing emails, and 111,681 (54.19%) are safe emails. Balanced Dataset: Balanced between legitimate and fraudulent emails (50%-50%), useful for an unbiased model.	Accuracy=98.09% Accuracy=96.36% Accuracy=97.67%
[15]	BiLSTM+Attention Mechanism	GramBeddings: 799,991 links (50% phishing and 50% legitimate). Malicious and Benign URLs: 450,176 links (23.2% phishing, 76.8% legitimate). Ebbu2017 Phishing: 73,575 links (50.5% phishing, 49.5% legitimate). Mendeley Data web page Phishing: 11,430 links (50% phishing and 50% legitimate) (for testing).	Accuracy=98.24% for GramBeddings dataset. Accuracy=99.32% for Malicious and Benign URLs dataset. Accuracy=98.34% for Ebbu2017 Phishing dataset. Accuracy=90.33% for Mendeley Data Web Page Phishing Detection dataset.

Research shows that, the machine learning techniques improve in speed of execution but fall short in accuracy compared to deep learning techniques which also require much more computational complexity and requirements for resources.

3. Methodology

This part introduces the methodology and practical aspect of the proposed system for phishing email classification. The research specifically focuses on using deep learning models to improve detection, and attempts to find the best model design. The study also addresses if the task involves complex models or if incompetent models suffice. This includes a comprehensive description of the dataset used, the preprocessing procedure applied and the models selected and trained for the task.

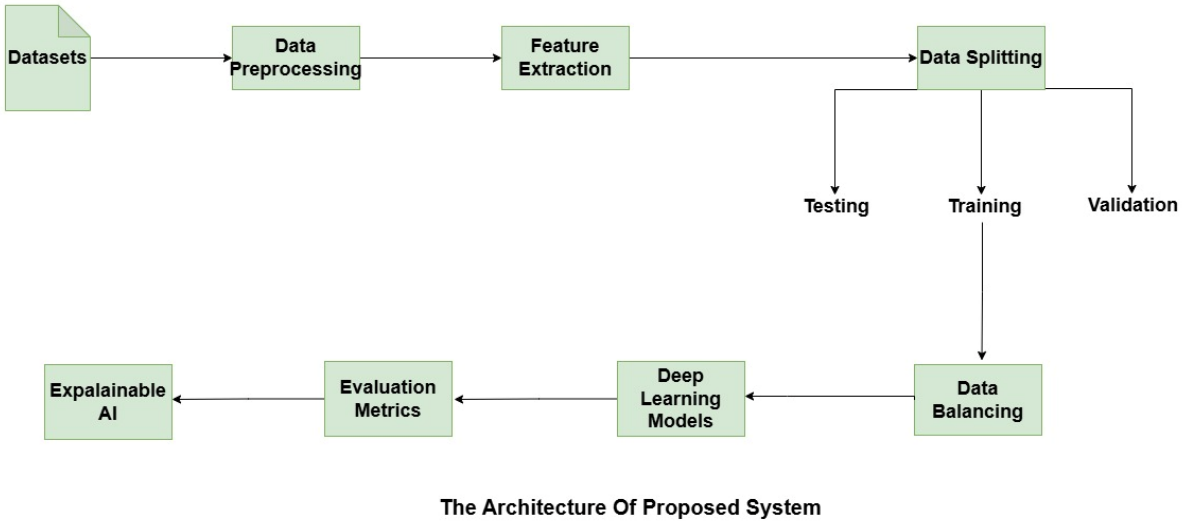

Fig.2. illustrates the architecture of proposed system

3.1 Datasets

In this study, in line with the merging methodology used in [14] for multiple datasets, eight distinct and heterogeneous data sources were consolidated to construct an expansive and integrated dataset. Data integration was performed using a custom script to merge eight datasets using common structural attributes. To maintain uniformity of attributes, the integration strictly required a congruent schema with common fields such as (sender, receiver, date, subject, body, label and URL) and the datasets not meeting such schema were excluded. The initial integration brought 227,511 records (122,859 phishing and 104,562 legitimate). After the duplicates and empty rows were removed, the finalized data contained 218,956 records with (117,414 phishing and 101,542 legitimate messages).

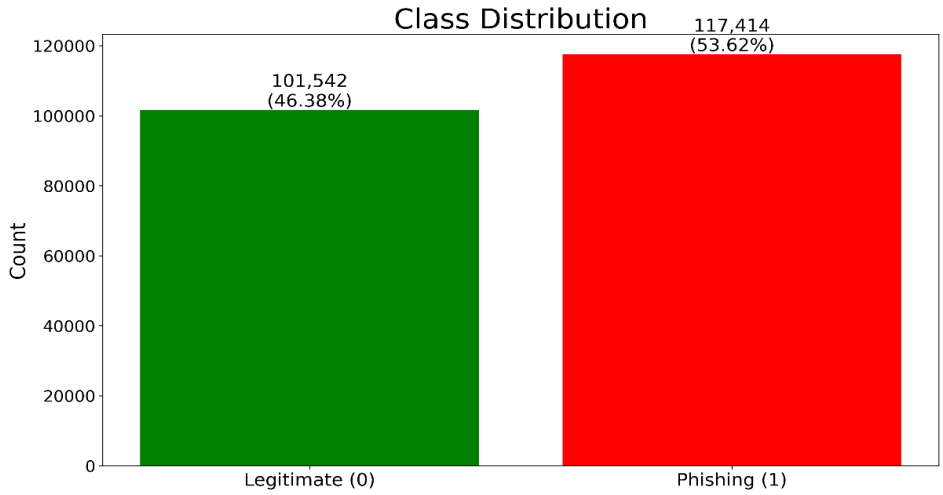

Fig.3. the image illustrates the distribution of data classification with percentage and number

These datasets are as follows:

1- CEAS-08:- It is a well-known dataset that has been used in spam filtering research. It consists of 39,154 messages [16].
2- Nazario: - This is the spam dataset compiled by Jose Nazario, a well-known and important collection in anti-spam research. It consists of 1,565 fraudulent emails [17].
3- Nigerian-Fraud: - is a dataset containing the texts of "Nigerian scam" messages (also known as "419 scams" or "advanced fee fraud"). It consists of 3,332 fraudulent messages [18].

4- SpamAssassin: - This is the official dataset used by the open-source SpamAssassin project to train and test its spam filtering system. It is one of the most popular and widely used spam datasets. It consists of 5,809 fraudulent and legitimate emails [19].
5- Trec05: - is a very large and important dataset used in the 2005 TREC (Text Retrieval Conference) competition to evaluate spam detection systems. It is one of the most famous datasets in the field. It consists of 55,990 email messages [20].
6- Trec06: - This is the set used in the Spam Track competition at TREC 2006. It is a large and important set to complete the evaluation that began at TREC 2005. It consists of 16,457 email messages [21].
7- Trec07: - is the third and final set of spam tracks presented at the TREC conference. It was used to evaluate spam detection systems in light of evolving sender tactics. It consists of 53,757 email messages [22].
8- PhishTank: - It is a vital and continuous source of data on phishing attacks that is compiled collaboratively. It consists of 51,447 malicious links [23].

3.2 Data Preprocessing

This section outlines the preprocessing pipeline implemented on the research data, containing subsequent stages of data loading, normalization, textual cleansing, partitioning and advanced processing. In figure 4, shows the data preprocessing and cleaning processes and methods used in this study.

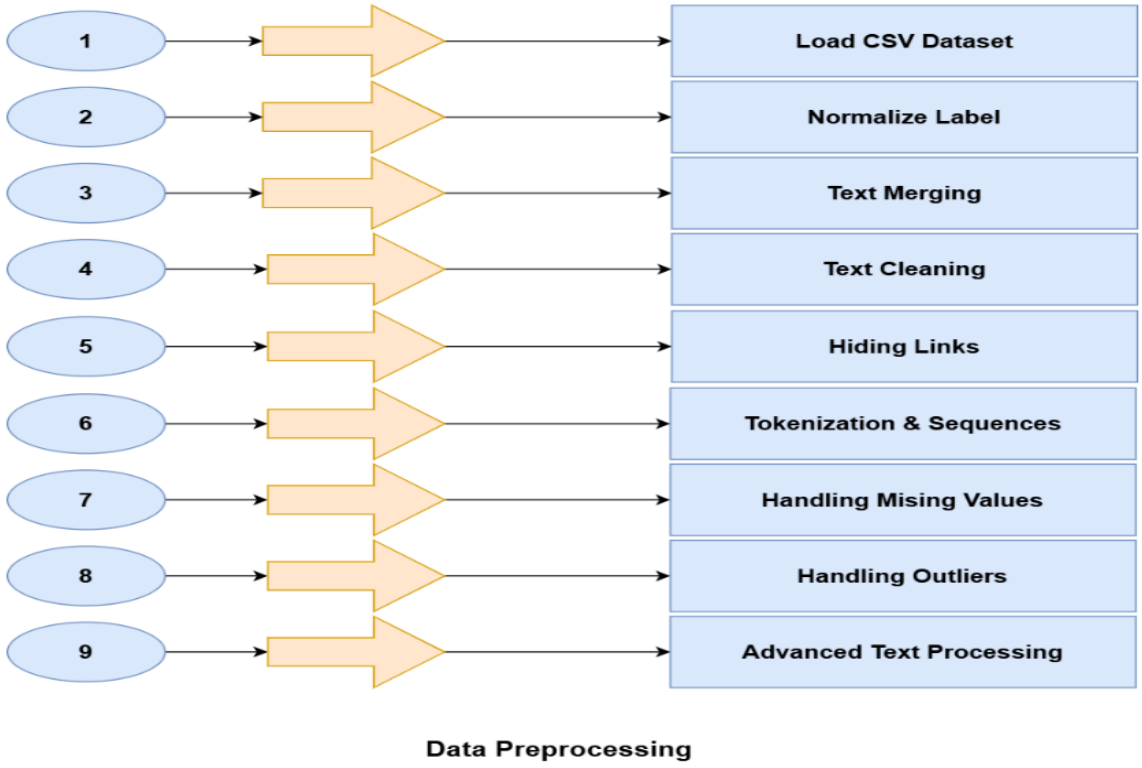

Fig.4. illustrates the data preprocessing

In the preprocessing data, several procedures were carried out as shown in the figure 4, consisting of 9 stages as follows: -

1- Load CSV: - The process of uploading the original data after it has been merged with it rows, as explained at the beginning of the section.
2- Normalize Label: - converts any value in a label to text, removes spaces and then converts it to lowercase. The goal is to standardize values before matching (benign/malicious/ham/spam/legitimate/phishing/0/1/0.0/1.0) and then convert them to 0 and 1.

3- Text merging: Merge the subject and body fields into one text with the addition of help symbols such as: [domain: example.com]. [Link] if links exist. [Source: PhishTank].
4- Text cleaning: Processing text to standardize its appearance and remove noise by removing HTML markup (such as
,), standardizing to convert all characters to lowercase, removing punctuation except for colons and underscores (_), removing diacritics (vowel marks) from Arabic text, and adjusting spacing to remove redundancies.
5- Hiding links: Replace all links in the text with [URL], with an optional option to retain the underlying domain name for specific analytical purposes.
6- Advanced text processing: Removal of non-alphabetic symbols (English and Arabic) and splitting of text into words while retaining words of two or more letters.
7- Tokenization & sequences: Convert the list of words into a numeric sequence using a custom Tokenizer, then pad or trim all the sequences to a uniform length (e.g., 250 tokens) to ensure that the input dimensions of the model are consistent.
8- Handling Missing Values: - The process of handling missing or undefined (NaN) values in a dataset where the Listwise method was used to delete any row that did not contain any type of data.
9- Handling Outliers: - The process of handling outliers in textual data by truncating words or sentences that exceed predefined standard length, with the aim of preventing model distortion and improving processing efficiency.

3.3 Feature Extraction

Transform textual data in raw form to formats that deep learning models can process. Using textual, structural, temporal and behavioral feature. The present study presents a multi-modal feature-engineering framework that fuses features. This is an all-inclusive method that enhances detection accuracy whilst maintaining computational efficiency thus improving phishing email classification. As illustrated in figure.5, the process of extracting features from the model is shown.

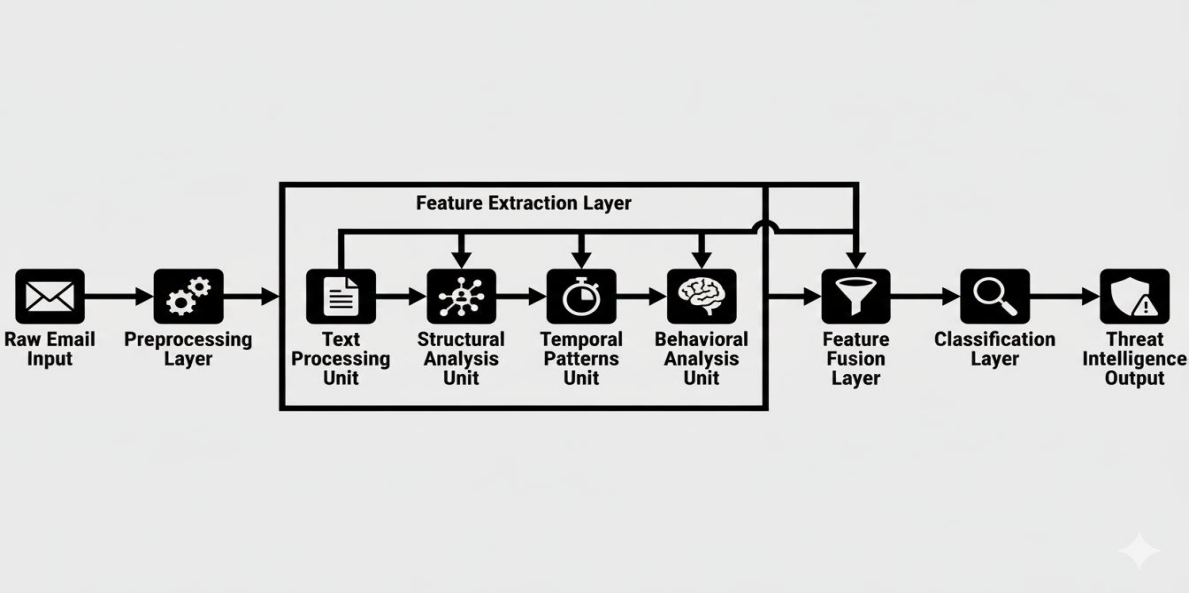

Fig.5. proposed system of the feature extraction process

Feature extraction can be seen from the architecture in figure.5. After preprocessing, the email is sent through a feature extraction layer with four units, including text processing, structural analysis, temporal pattern analysis and behavioral analysis. Textual information of subject and body are converted into the Word2vec embeddings for semantic representation. Structural features like domain tokens and presence of URL, etc. are embedding in the text. The representation that are extracted are then combined and sent to the CNN-based classification layer for phishing detection.

Table 3. Below shows all the features extracted from the dataset

Seq	Categories	Features	Description
1	Textual	combined_text subject_clean body_clean sender_clean receiver_clean	Analyzing the textual content of the message and title using natural language processing to detect suspicious patterns and words.
2	Structural	_has_any_url _in_pt _domain_token att_has_attachments att_attachment_count att_suspicious_extensions att_has_exe att_has_zip att_has_doc att_has_pdf att_has_script url_url_count url_suspicious_domains url_ip_urls url_short_urls url_typosquatting url_has_https url_has_http url_has_port url_avg_url_length url_max_url_length url_has_suspicious_words url_has_brand_impersonatio n hdr_has_spf hdr_has_dkim hdr_has_dmarc hdr_header_anomalies hdr_has_reply_to hdr_sender_mismatch hdr_has_received_headers hdr_received_count hdr_has_authentication_resu lts hdr_has_x_priority hdr_is_urgent	Analyzing the external structure of the message: links, attachments, and email header to detect phishing patterns.
3	Temporal	temp_hour temp_day_of_week temp_day_of_month temp_month temp_is_night temp_is_weekend temp_is_holiday temp_sender_frequency temp_unusual_time	Analyzing the timing and sending of the message to discover unusual behaviors and time-based attack patterns.

		temp_unusual_day temp_pattern_deviation	
4	Behavioral	Body_length subject_length body_word_count subject_word_count has_urgent_keywords has_financial_keywords has_suspicious_words sender_length receiver_length has_sender has_receiver url_count suspicious_url_count avg_url_length day month year weekday	Analyzing digital indicators and statistical keywords to detect phishing patterns in message content.

3.4 Data Splitting and Balancing

The dataset was divided into 70% training set with the remaining 30% split equally into validation and testing subsets. To mitigate class imbalance, text data augmentation including (back-translation, synonym replacement, deletion and swap) was applied solely to the training data. This approach maintained the real data distribution for realistic evaluation, at the same time, we achieved a balance and enriched training corpus data, which improved the robustness and generalizability.

4. Selected Deep Learning Models

In the final stage of the methodology, different models are created and learned for the pre-processed and splitted data.

4.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) represent a basic type of deep learning architecture and are popularly used in fields such as image recognition, video analysis and Natural Language Processing (NLP) [24]. The proposed of CNN is design to discover hierarchical features independently through the use of input layer, convolutional layers and output layer. In figure.6, the proposed CNN architecture, word2vec embeddings are concatenated before training, which gives the model the ability to exploit both general and task-specific language knowledge. This is succeeded by 6 successive convolutional blocks of filter sizes of 96, 192 where Conv1D, Batchnormalization and ReLU activation are used to derive hierarchical patterns in the email text.

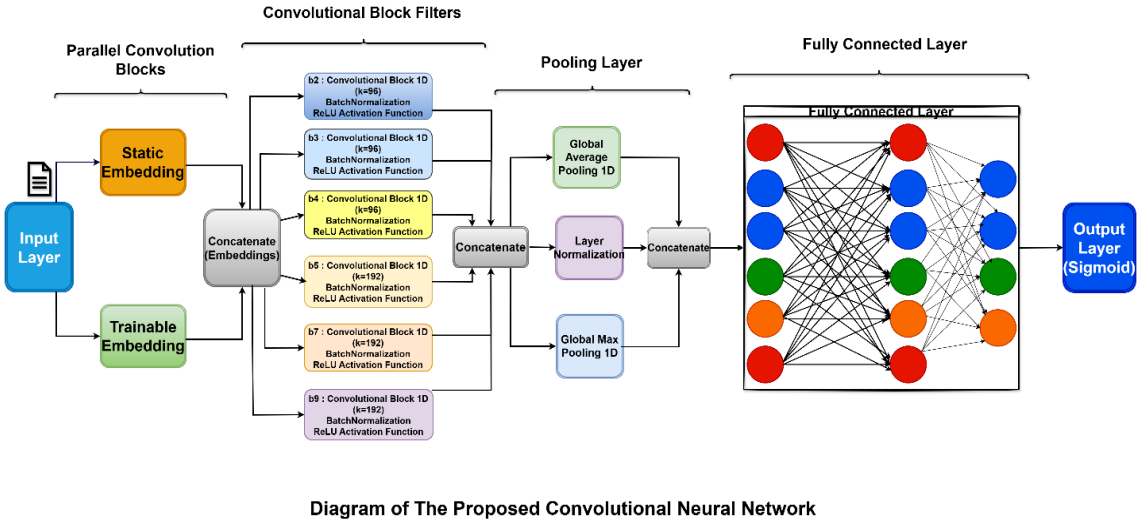

Fig.6. Diagram of the proposed CNN

Following extraction of the features, parallel global average and max pooling layers are utilized to extract complementary salient features and fixed by layer normalization. They are trained in a fully connected layer and binary classification output with sigmoid activation function. The model contains 62,675,009 parameters (30,001,984 non-trainable parameters (pre-trained word2vec) and 32,673,025 trainable parameters optimized from training) that can be effectively trained on misleading linguistic patterns of NLP-based phishing detection.

4.1.1 TF-IDF (Term Frequency – Inverse Document Frequency)

TF-IDF method works by checking how often words show up in bunch of documents. TF spot the big words on one document and IDF highlight the ones that are special on whole collection [11]. It converts the text of emails into numbers based on the weight of words according to their commonness and uncommonness. Then with it normalizes the numbers so that longer letters don't give everything a bias. Those numbers are what the machine learning models use in order to determine whether an email is legitimate or a phishing. Therefore: -

$$\text{TF (t , d)} = \frac{x}{y} \tag{1}$$

Where x represents the number of times t appears in document d, and y represents the number of words in document d. The IDF index measures the uniqueness of a word across a set of texts.

$$\text{IDF (t , d)} = \log(\frac{N}{n}) \tag{2}$$

Where N is the total number of documents in the collection corpus and n is the number of documents in which the term t appears.

$$\text{TF-IDF} = TF * IDF \tag{3}$$

4.1.2 Word2vec

Is a neural network model that learns continuous vector represents of word that analyze where words appear together in a big collection of text so that it picks up on meaning stuff [11]. Word2vec uses neural nets to turn words into dense vector embeddings that capture meaning inside email text. Trained on huge corpora using CBoW or SkipGram, the embeddings provide the classifiers with tons of features in order to tell if an email is phishing or legitimate based on its context. Word2vec has two algorithms are Continuous Bag of Word (CBoW) and SkipGram (SG).

4.1.2.1 SkipGarm: - vectorize a lexical dictionary in a way that can handle on-rare terms better. Word2vec employs a cosine function to make embeddings over a training corpus with parameters such as dimensions, window size, worker threads and minimum count [11]. By fine-tuning those parameters the vectors become more effective.

4.1.2.2 Continuous Bag of Words: - it works in the opposite way of SkipGram and predicts target words based on context. Despite being faster and more effective in frequent terms, it has lower accuracy. It uses the same parameters as SkipGram to create the word vectors in real values, where dimensionality, etc, are what define the end word embeddings [11].

4.1.3 FastText

Is an open source library developed by Facebook AI Research (FAIR) [25], generates word embeddings through unsupervised or supervised learning algorithm, CBoW and SkipGram. fastTexts character-level n-gram decomposition which captures subword patterns, for strong representation of obfuscated vocabulary in phishing emails. This approach helps to improve the feature discrimination which directly increases the binary classification accuracy and generalization of the models.

4.1.4 Word Embedding

Is a natural language processing model based on principles analogous to artificial neural networks. It converts text into a digital representation of words in the form of vectors which embody semantic meaning and allow textual information to be rendered as numbers whilst maintaining contextual linkages between words [15]. Word embedding model transform textual data to numerical vectors that provide the semantic relationships. These representations act as discriminative features for binary classification models which helps in discrimination between phishing and legitimate emails accurately.

5.Results

This section introduces empirical results of the suggested phishing detection models. It explains the set-up of the performance measures, the results obtained and sent interpretive insights with Explainable AI (xAI). It details the experimental configurations, performance metrics, obtained results and provides interpretive insights through Explainable AI (XAI).

5.1 Experimental Settings

The suggested system was built with Keras and TensorFlow, an open-source library for machine learning. All the experiments were conducted in the Pycharm integrated development environment. The dataset was divided into 70% dataset to train the model and the remaining 30% was equally divided between validation and testing set (each 15% of the total data). This was done to test the proposed models.

5.2 Evaluation Metrics

The phishing email classification models were tested using standard evaluation measures of accuracy, precision, recall and F1_score. In order to assess the effectiveness of the oversampling technique used for the class imbalance problem, the metric ROC-AUC was calculated. The definition and calculation for each metric is given as follows:
Accuracy: It is the ratio of correct predictions of legitimate emails and phishing emails to the total number of emails [14].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

Precision: It is the percentage of phishing emails identified as actually phishing emails [14].

$$\text{Precision} = \frac{TP}{TP+FP} \tag{5}$$

Recall: It is the percentage of actual positive phishing emails that were correctly identified as phishing emails [14].

$$\text{Recall} = \frac{TP}{TP+FN} \tag{6}$$

F1_Score: It is a measure of the balance between precision and recall [14].

$$\text{F1\_Score} = 2 * \frac{(Precision*Recall)}{Precision+Recall} \tag{7}$$

The evaluation metrics are defined as follows: True Positives (TP) and True Negatives (TN) represents the number of (phishing and legitimate) emails correctly classified respectively. False Positives (FP) and False Negatives (FN) are misclassified. The ROC-AUC curve evaluates the classifiers ability to rate phishing emails above legitimate emails and is calculated as the area under the curve.

5.3 Experimental Results

Several methods were used in the convolutional neural network, as explained above, and the best result was achieved by classifying phishing emails using the Word2vec method, as follows:-
The proposed framework processing email data using the preprocessing, feature extraction, feature fusion and classification stages in a structured pipeline mechanism. At first, the raw emails undergo cleaning and normalization and the subject and body are combined to create the textual input. The feature extraction layer analyses with the help of 4 complementary units: text processing, structural analysis, temporal patterns analysis and behavioral analysis. Textual information is converted into semantic vectors based on word2vec embeddings; meanwhile structural information (domain tokens, presence of URL) in embedded in textual information. The extracted features are then combined and inputted to a CNN-based classification model which would use multiple convolution filters to extract context pattern. The resulting representations are aggregated by using pooling layers and fed to fully connected layers in order to carry out phishing email classification. The proposed CNN-word2vec based phishing email detection model is described in the algorithm 1.

Input: Email dataset D = {(e_i, y_i), y_i ∈ {0,1}},
Output: Predicted label ŷ and explanation Exp.

1. Preprocess Emails: T_i ← clean(subject_i + body_i) Mask URLs → [URL]
2. Split Dataset: D → D_train, , D_val , D_test
3. Balance Training Data: D_train ← Augment(minority class)
4. Tokenize Text: S_i ← tokenize(T_i)
5. Convert to Sequences: X_i ← pad_sequences(S_i, max_len)
6. Train Word2Vec Embeddings: W ← Word2Vec(S_train)
7. Build Embedding Matrix: E ← map(vocabulary, W)
8. Construct CNN Model: X_i → Embedding_static(E) || Embedding_trainable(E)
9. Concatenate Embeddings
10. Feature Extraction: Conv1D(k ∈ {2,3,4,5,7,9})
11. Feature Aggregation: GlobalMaxPooling + GlobalAvgPooling
12. Classification: Dense → Sigmoid → ŷ
13. Train Model: Minimize Binary Cross-Entropy
14. Explain Prediction: Exp ← LIME(model, email)
15. Evaluate Model: Accuracy, Precision, Recall, F1-score, ROC-AUC

Algorithm 1. Proposed CNN-Word2Vec Phishing Email Detection with xAI

The model combines dual embeddings (static and trainable), multi-kernel convolution layers to extract features and when interpreting model predictions uses LIME-based Explainable AI. The hyperparameter values used in the convolutional neural network are summarized in table 4.

Table 4. Summary of hyperparameters of CNN model

Hyperparameters	Values
Seed	42
Loss Function	Binary_crossentropy
Number of Layers	6 Layers
Kernel Size	2,3,4,5,7 and 9
Filters	96,96,96,192,192 and 192
Activation Function	ReLU
FC Layer Activation Function	Sigmoid
Dropout	0.5
Kernel Regularizer	L2 (1e-4)
Pooling Layer	Average pooling + Max pooling
Optimizer	AdamW, LR=1e-4, WD=1e-4
Early Stopping	Monitor=val_loss , patience=5
Batch_Size	128
Epochs	50

As shown in Table 5, the experiments are described with instruments used in each experiments and results.

Table 5. Illustrates the experiments of word2vec

Approach	Number of layers	Filters size	Dropout	Pooling layer	Results				
					Acc	Prec	Rec	F1	AUC
Word2vec	4	3,5,7,9	0.4	4 MaxPooling	98.44%	99.01%	98.38%	98.70%	99.85%
	4	3,5,7,9	0.5	4 MaxPooling	95.71%	93.53%	98.13%	95.81%	97.81%
	4	3,5,7,10	0.55	4 MaxPooling	98.45%	99.01%	98.41%	98.71%	99.85%
	6	2,3,4,5,7,10	0.45	6 MaxPooling	98.74%	98.89%	99.02%	98.96%	99.90%
	6	2,3,4,5,7,9	0.5	6 MaxPooling	98.84%	99.07%	98.95%	99.01%	99.92%
	10	1,2,3,4,5,7,9,11,13,15	0.6	Dual pooling (Avg+Max)	99.02%	98.99%	99.35%	99.17%	99.93%
	6	2,3,4,5,7,10	0.5	6 MaxPooling	98.21%	98.63%	98.39%	98.51%	99.82%
	6	2,3,4,5,7,9	0.6	6 MaxPooling	97.72%	98.13%	98.08%	98.10%	99.76%
	6	2,3,4,5,7,10	0.6	6 MaxPooling	98.85%	99.07%	99.02%	99.05%	99.92%
	6	2,3,4,5,7,10	0.45	6 MaxPooling	99.01%	99.12%	99.20%	99.16%	99.92%
	6	2,3,4,5,7,9	0.50	Dual pooling (Avg+Max)	99.05%	98.85%	99.44%	99.20%	99.93%

Table 5 shows that the best accuracy score obtained by using word2vec method was achieved from testing with dual channels with accuracy of 99.05% and 99.93% ROC-AUC value for distinguishing phishing messages and legitimate messages, respectively, from the unseen data. Table 6 presents the results of other methods used in detecting phishing, the tools used with the results, as follows:

Table 6. Illustrates the experiments of word embedding

Approach	Number of layers	Filters size	Dropout	Pooling layer	Results				
					Acc	Prec	Rec	F1	AUC
Word Embedding	3	3,5,7	0.6	3 MaxPooling	95.67%	92.74%	99.09%	95.81%	97.97%
	3	3,5,7	0.3	3 MaxPooling	95.71%	93.58%	98.16%	95.81%	97.81%
	3	3,5,7	0.6	3 MaxPooling	96.05%	97.74%	93.35%	95.49%	98.32%
	3	3,7,10	0.5	3 MaxPooling	97.85%	97.99%	97.20%	97.60%	99.31%

3	3,5,7	0.7	3 MaxPooling	96.84%	97.28%	96.38%	96.83%	99.28%
3	3,5,7	0.6	3 MaxPooling	98.72%	98.92%	98.91%	98.91%	99.90%

Table 6 indicates that the word embedding technique attained a highest accuracy of 98.72% in phishing email detection. The results of the FastText method are presented in Table 7:

Table 7. Illustrates the experiments of fastText

Approach	Number of layers	Filters size	Dropout	Pooling layer	Results				
					Acc	Prec	Rec	F1	AUC
FastText	6	2,3,4,5,7,9	0.5	6 MaxPooling	98.93%	98.99%	99.20%	99.09%	99.92%
	6	2,3,4,5,7,9	0.4	6 MaxPooling	98.97%	99.03%	99.22%	99.12%	99.93%
	6	2,3,4,5,7,9	0.6	6 MaxPooling	98.99%	99.09%	99.20%	99.14%	99.92%
	8	1,2,3,4,5,7,9,11	0.7	8 MaxPooling	98.89%	98.76%	99.35%	99.06%	99.92%
	8	1,2,3,4,5,7,9,11	0.4	8 MaxPooling	98.82%	99.03%	98.96%	98.99%	99.92%

Table 7 indicates that the fastText method attained its peak experimental accuracy of 98.99% in the detection of phishing emails. In the final approaches of CNN experiments is TF-IDF method, the following was shown in Table 8:

Table 8. Illustrates the experiments of TF-IDF

Approach	Number of layers	Filters size	Dropout	Pooling layer	Results				
					Acc	Prec	Rec	F1	AUC
TF-IDF	6	2,3,4,5,7,9	0.5	6 MaxPooling	67.45%	93.80%	49.08%	64.44%	81.92%
	4	3,5,7,9	0.5	4 MaxPooling	73.28%	90.52%	56.04%	69.22%	80.91%
	2	3,5	0.6	2 MaxPooling	97.65%	97.56%	98.07%	97.81%	99.24%
	2	3,5	0.5	2 MaxPooling	97.76%	98.07%	97.74%	97.91%	99.59%
	6	1,3,5,7,9,10	0.5	6 MaxPooling	98.85%	98.88%	99.17%	99.02%	99.84%

The table 8 demonstrates that the TF-IDF method achieved its highest experimental accuracy of 98.85% in detecting phishing emails. Based on the methods used and the results obtained, it is clear that the word2vec method has proven its efficiency and accuracy in detecting phishing messages and distinguishing between phishing and legitimate ones. Table 9 illustrates the difference between the methods used in the convolutional neural network, along with the prediction and training time for each method and for each sample of test messages.

Table.9 illustrates the difference between the approaches

Seq	Methods	Results	Training Time	Prediction Time
1	TF-IDF	98.85%	1764.15 min	0.007864 s/sample
2	Word2vec	99.05%	472 min	0.001729 s/sample
3	FastText	98.99%	190.16 min	0.107825 s/sample
4	Word Embedding	98.72%	386.41 min	0.003458 s/sample

Figure 7 displays the training and validation results obtained during the training process and the loss after 50 epochs, thus showing the models ability to learn new patterns without overfitting.

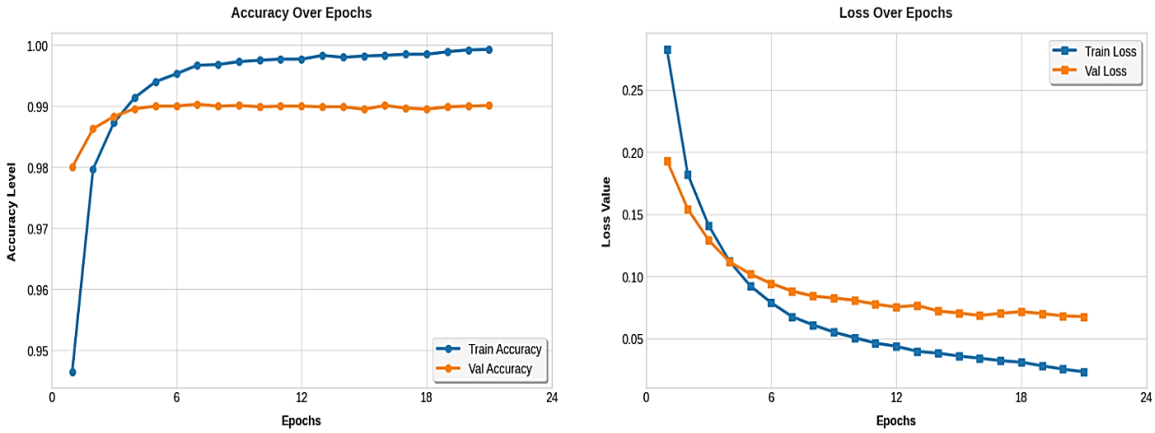

Fig.7. Illustrates the training and testing accuracy and loss for word2vec

The model of figure 8 provides a binary classification (legitimate / phishing) with a confidence score. The model had an accuracy of 99.05% on the test set, which correctly recognizes 13,342 legitimate messages and 19,191 phishing messages and made only 311 errors with 32,844 test cases (an error rate of 0.95%).

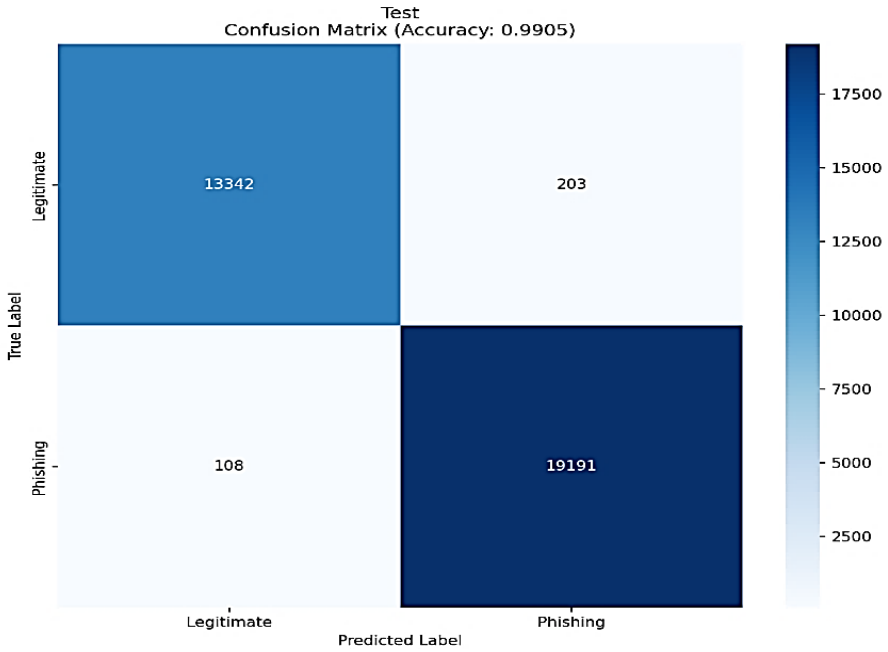

Fig.8. illustrates the confusion matrix of word2vec model

The results in Table 10 shows the improved performance of our model (CNN with word2vec).

Table.10. Compared between our proposed models with other deep learning approaches

Algorithms	Datasets	Results
CNN one-hot encoding [5]	AppRiver, 3,416 phishing and 14,950 legitimate.	Accuracy=96.34%
CNN [6]	UCI machine learning repository and SpamAssassin as phishing and Enron as legitimate.	Accuracy=98.74%
CNN [7]	Enron email dataset, 7,781 legitimate and 999 phishing.	Accuracy=97.58%

CNN [8]	Enron dataset, 16563 legitimate and 17188 phishing.	Accuracy=96.39%
1D-CNNPD [9]	Phishing corpus 2,278 phishing + SpamAssassin 4,150 legitimate.	Accuracy=98.87%
CNN [14]	Balanced Merged Dataset	Accuracy=98.09%
CNN one-hot encoding [26]	AppRiver, 3,416 phishing and 14,949 legitimate.	Accuracy=96.52%
CNN character embedding [27]	PhishTank, 51,316 legitimate and 36,173 phishing.	Accuracy=97.17%
Our Proposed System	Merged Dataset, 101,542 legitimate and 117,414 phishing.	Accuracy=99.05%

5.4 Explainable AI

Although high predictive performance is a requirement, transparency would be needed to deploy deep learning in cybersecurity to address the built-in black box limitation. In this regard, we combined LIME (Local Interpretable Model-agnostic Explanations) into our framework. The proposed model proved to be very efficient with 100% accuracy when evaluated on external test lifetime as shown in figure 9, prediction fidelity of 99.69% and a speedy inference period of 0.065 seconds.

Fig.9. illustrates the explainable of external test messages

To interpret these decision, LIME assigns importance weights to the words, and it can be seen that the words contribute to either pushing the predictions towards either the phishing (positive weights) or the legitimate (negative weights) classes. Visualizations validate that the model is based on correct semantics settings. E.g., a "Bank Fraud Alert" provides malicious indicators to predict the phishing (e.g., action: 0.0077, fraud: 0.0068). On the other hand, legitimate communications were rightfully categorized using benign terms (e.g., conference: -0.0018). It also showed that mixed signals worked well with the model, an IT maintenance Notice with helpdesk being suspicious (0.1778) and scheduled

being strong legitimate indicators (-0.2373). Finally, the addition of LIME gives actionable intelligence regarding rapid incident triage and also helps to verify that the model is learning generalized semantic features, which justifies its deployment in cybersecurity practice.

6. Discussion

The superior performance of the proposed CNN-Word2vec model is achieved by successfully integrating semantic word embeddings with convolution extract feature. Unlike the traditional methods of vectorization like TF-IDF and FastText due to the sparse frequency based representations and neglecting semantic relation among words, word2vec produces dense vector representation that captures the contextual and semantic similarities. This helps the model to detect variations on phishing language even when attackers use different words to communicate similar evil aims behind. Furthermore, the CNN architecture is efficient in detecting discriminating local patterns and N-Grams structure contained in the email text by convolutional filters of different kernel sizes that enables the model to identify subtle phishing indicators which exist in different parts of the flow message. Compared with sequential architecture like LSTM, CNN will give faster train time with less computational complexity and sufficient learning ability of feature. In order to guarantee strong evaluation, a complete benchmark dataset is built in this study by fusing eight phishing email datasets from 2005-2025, enabling great diversity in attack patterns. Additionally, Explainable AI (xAI) techniques are implemented, specifically LIME aimed at searching the words that have a strong influence on the classification decision, enhancing transparency, interpretability and trustfulness of the phishing detection systems. In table.10, our suggested model is obviously superior to the previous studies all the comparisons have more detection accuracy. This comes as a result of building a bigger and more varied dataset by incorporating eight sources and qualitative refinements to the convolutional neural network filters and addition of Word2vec. The findings validate that our approach is more generalized and phishing detection is more reliable than previous researchers.

7. Conclusion and Future Work

Email phishing remains a major threat to cybersecurity because it specifically targets the information through increasingly sophisticated attacks. The implementation of advanced methods of artificial intelligence is required for precise detection. This study provides a comparative framework for evaluation of phishing email detection models with different datasets. Results show that Neural Convolutional Network (CNN) model with Word2vec embedding has shown better performance. Additionally, the fastText model was found to be a computationally efficient alternative and as a result, was suitable for deployment within the context of a real-time system. Future research should focus on optimizing the models for real-world implementation, and also multiply validation methods to multilingual phishing datasets to increase the global validity of machine and deep learning based solutions.

References

[1]A. Ozcan, C. Catal, E. Donmez, and B. Senturk, “A hybrid DNN–LSTM model for detecting phishing URLs,” *Neural Comput. Appl.*, vol. 35, no. 7, pp. 4957–4973, 2023, DOI: 10.1007/s00521-021-06401-z.

[2]Available:https://www.simplilearn.com/ice9/free_resources_article_thumb/phishing_working_2-What_Is_Phishing.PNG

[3]Anti-Phishing Working Group, “Phishing Activity Trends Report 2nd Quarter 2025,” Most. Accessed: 30. Aug, 2025. [online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q1_2025.pdf

[4]Anti-Phishing Working Group, “Phishing Activity Trends Report 2nd Quarter 2025,” Most. Accessed: 1. Sep, 2025. [online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q2_2025.pdf

[5]S. Bagui, D. Nandi, S. Bagui, and R. J. White, “Machine Learning and Deep Learning for Phishing Email Classification using One-Hot Encoding,” J. Comput. Sci., vol. 17, no. 7, pp. 610–623, 2021, DOI: 10.3844/jcssp.2021.610.623.

[6]S. Atawneh and H. Aljehani, “Phishing Email Detection Model Using Deep Learning,” Electron., vol. 12, no. 20, 2023, DOI: 10.3390/electronics12204261.
[7]R. Brindha, S. Nandagopal, H. Azath, V. Sathana, G. P. Joshi, and S. W. Kim, “Intelligent Deep Learning Based Cybersecurity Phishing Email Detection and Classification,” Comput. Mater. Contin., vol. 74, no. 3, pp. 5901–5914, 2023, DOI: 10.32604/cmc.2023.030784.
[8]P. Krishnamoorthy, M. Sathiyanarayanan, and H. P. Proença, “A novel and secured email classification and emotion detection using hybrid deep neural network,” Int. J. Cogn. Comput. Eng., vol. 5, no. December 2023, pp. 44–57, 2024, DOI: 10.1016/j.ijcce.2024.01.002.
[9]N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, “Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models,” Sensors, vol. 24, no. 7, pp. 1–19, 2024, DOI: 10.3390/s24072077.
[10] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, “A predictive model for phishing detection,” J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 2, pp. 232–247, 2022, DOI: 10.1016/j.jksuci.2019.12.005.
[11] M. Somesha and A. R. Pais, “Classification of Phishing Email Using Word Embedding and Machine Learning Techniques,” J. Cyber Secur. Mobil., vol. 11, no. 3, pp. 279–320, 2022, DOI: 10.13052/jcsm2245-1439.1131.
[12] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, “A hybrid DNN–LSTM model for detecting phishing URLs,” Neural Comput. Appl., vol. 35, no. 7, pp. 4957–4973, 2023, DOI: 10.1007/s00521-021-06401-z.
[13] A. Al-Subaiey, M. Al-Thani, N. Abdullah Alam, K. F. Antora, A. Khandakar, and S. A. Uz Zaman, “Novel interpretable and robust web-based AI platform for phishing email detection,” Comput. Electr. Eng., vol. 120, no. Ml, 2024, DOI: 10.1016/j.compeleceng.2024.109625.
[14] A. Alhuzali, A. Alloqmani, M. Aljabri, and F. Alharbi, “In-Depth Analysis of Phishing Email Detection: Evaluating the Performance of Machine Learning and Deep Learning Models Across Multiple Datasets,” Appl. Sci., vol. 15, no. 6, pp. 1–30, 2025, DOI: 10.3390/app15063396.
[15] Ö. Ş. Akçam, A. Tekerek, and M. Tekerek, “Development of BiLSTM deep learning model to detect URL-based phishing attacks,” Comput. Electr. Eng., vol. 123, no. September 2024, 2025, DOI: 10.1016/j.compeleceng.2025.110212.
[16] CEAS 2008 Spam Corpus, “CEAS 2008 Spam Corpus,” Conference on Email and Anti-Spam (CEAS). Accessed: Aug. 20, 2025. [Online]. Available: https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset?select=CEAS_08.csv
[17] J. Nazario, “Nazario Spam and Phishing Corpus,” University of California, Irvine. Accessed: Aug. 20, 2025. [Online]. Available: https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset?select=Nazario.csv
[18] Nigerian Fraud, “Nigerian Fraud Email Dataset (419 Scam Corpus).” Accessed: Aug. 20, 2025. [Online]. Available: https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset?select=Nigerian_Fraud.csv
[19] The Apache Software Foundation, “SpamAssassin Public Mail Corpus,” The Apache Software Foundation. Accessed: Aug. 20, 2025. [Online]. Available: https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset?select=SpamAssasin.csv
[20] T. R. Cormack, Gordon V. and Lynam, “TREC 2005 Public Spam Corpus,” National Institute of Standards and Technology (NIST). Accessed: Aug. 20, 2025. [Online]. Available: https://zenodo.org/records/8339691/files/TREC_05.csv?download=1
[21] T. R. Cormack, Gordon V. and Lynam, “TREC 2006 Public Corpus,” National Institute of Standards and Technology (NIST). Accessed: Aug. 20, 2025. [Online]. Available: https://zenodo.org/records/8339691/files/TREC_06.csv?download=1
[22] T. R. Cormack, Gordon V. and Lynam, “TREC 2007 Public Corpus,” National Institute of Standards and Technology (NIST). Accessed: Aug. 20, 2025. [Online]. Available: https://zenodo.org/records/8339691/files/TREC_07.csv?download=1
[23] OpenDNS, “PhishTank: An anti-phishing site,” Cisco Systems, Inc. Accessed: Sep. 18, 2025. [Online]. Available: https://phishtank.org/developer_info.php%0A

[24] J. Koushik, “Understanding Convolutional Neural Networks,” no. 3, pp. 1–6, 2016.
[25] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” 1996.
[26] R. Eckhardt, “Convolutional Neural Networks and Long Short Term Memory for Phishing Email Classification,” vol. 19, no. 5, pp. 27–35, 2021.
[27] M. Korkmaz, E. Kocyigit, O. K. Sahingoz, and B. Diri, “A Hybrid Phishing Detection System Using Deep Learning-based URL and Content Analysis,” 2022.

Journal of Basrah Researches (Sciences) **52**(1), 170 (2026)
DOI: https://doi.org/10.56714/bjrs.52.1.13

كشف رسائل البريد الالكتروني التصيدية القابلة للتفسير بإستخدام نماذج الشبكة العصبية الألتفافية القائم على word2ve وتفسيرات LIME

بكر ثامر حمودي[1]، * , عمر زياد عاكف[2]

[1]معهد المعلوماتية للدراسات العليا، جامعة تكنولوجيا المعلومات والاتصالات، العراق، بغداد.
[2]جامعة بغداد، كلية التربية للعلوم الصرفة (إبن الهيثم)، قسم الحاسبات.

معلومات البحث

الاستلام	25 شباط 2026
المراجعة	24 اذار 2026
القبول	24 اذار 2026
النشر	30 حزيران 2026

الكلمات المفتاحية

الأمن السيبراني, كشف رسائل التصيد, الشبكات العصبية الألتفافية, Word2Vec, تمثيل الكلمات, FastText, TF-IDF

الملخص

تشكل رسائل التصيد الاحتيالي تهديداً كبيراً للأمن السيبراني, إذ تهدف اساساً الى سرقة المعلومات الخاصة أو التسبب بخسائر مالية. يقوم المهاجمون بإنتحال هويات جهات معروفة ويبتكرون أساليب متطورة للتخفي, مما يجعل اكتشافهم صعباً رغم وفرة الأبحاث الأكاديمية. تُركز هذه الدراسة على تقييم عدة طرق لتمثيل النص وبُنى التعلم العميق في تصنيف الرسائل الألكترونية. تتمثل المساهمة الأساسية في تعزيز الشبكة العصبية الألتفافية بإستخدام تمثيل Word2Vec لتصنيف رسائل التصيد. ولضمان القوة, جرى إنشاء مجموعة بيانات معيارية متنوعة بدمج ثمانية مصادر عامة, مع تحسين إستخراج الميزات من خلال تعديلات خاصة على مرشحات الألتفاف. وتم إعتماد معالجة موسعة للبيانات مثل زيادة العينات لمعالجة عدم توازن الفئات. تمت مقارنة النموذج المقترح مع عدة نماذج مثل TF-IDF, FastText وWord Embedding واساليب التعلم التقليدية. أظهرت النتائج إن النموذج المقترح حقق اعلى دقة بلغت 99.05% متفوقاً على تمثيل الكلمات (98.72%) و TF-IDF (98.85%) و (98.99%) FastText. أثبتت الشبكة العصبية الألتفافية مع نموذج word2ve فعاليتها العالية في تصنيف رسائل التصيد. ولتعزيز الشفافية, تم دمج اطار LIME لتقديم تفسير يعتمد على درجات المساهمة, مما يوضح طريقة إتخاذ النموذج للقرار ويحد من التباينات المنهجية.

Citation: B. Th. Hammoudi, O. Z. Akif., J. Basrah Res. (Sci.) **52**(1), 170 (2026). DOI:https://doi.org/10.56714/bjrs.52.1.13

***Corresponding author email:** baker.1994thth@gmail.com

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq