electronics

MDPI

Article

Phishing Email Detection Model Using Deep Learning

Samer Atawneh * and Hamzah Aljehani

College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia; hamzaaljehani760@gmail.com

* Correspondence: satawneh@seu.edu.sa; Tel.: +966-112613500

Abstract: Email phishing is a widespread cyber threat that can result in the theft of sensitive information and financial loss. It uses malicious emails to trick recipients into providing sensitive information or transferring money, often by disguising themselves as legitimate organizations or individuals. As technology advances and attackers become more sophisticated, the problem of email phishing becomes increasingly challenging to detect and prevent. In this research paper, the use of deep learning techniques, including convolutional neural networks (CNNs), long short-term memory (LSTM) networks, recurrent neural networks (RNNs), and bidirectional encoder representations from transformers (BERT), are explored for detecting email phishing attacks. A dataset of phishing and benign emails was utilized, and a set of relevant features was extracted using natural language processing (NLP) techniques. The proposed deep learning model was trained and tested using the dataset, and it was found that it can achieve high accuracy in detecting email phishing compared to other state-of-the-art research, where the best performance was seen when using BERT and LSTM with an accuracy of 99.61%. The results demonstrate the potential of deep learning for improving email phishing detection and protecting against this pervasive threat.

Keywords: email phishing; convolutional neural networks (CNNs); long short-term memory (LSTM); recurrent neural networks (RNNs); bidirectional encoder representations from transformers (BERT); deep learning; natural language processing (NLP)

Citation: Atawneh, S.; Aljehani, H. Phishing Email Detection Model Using Deep Learning. *Electronics* **2023**, *12*, 4261. https://doi.org/10.3390/electronics12204261

Academic Editor: Hung-Yu Chien

Received: 6 September 2023
Revised: 11 October 2023
Accepted: 12 October 2023
Published: 15 October 2023

1. Introduction

The increasing acceptance and adoption of IT has led to increased web-based activities provided via cyberspace [1]. According to Adewole et al. (2019) [2], these activities span from essential services such as education and financial transactions to basic activities such as e-health apps and social media. According to research, activities such as social media, online gaming services, and financial transactions are considered top web-based activities with vast popularity and enormous userbases [3]. Many users of these web-based solutions indicate the acceptance of IT in recent times. The objective is to make the web-based solutions that are used daily more available and accessible. However, because there are no universal security controls in cyberspace, the open availability and accessibility of these web-based solutions open the door to cyberattacks [4].

Phishing is a crime affecting everyone, including organizations and governments worldwide. Email phishing is a typical cyberattack that can have serious consequences, including financial loss, identity theft, and damage to an organization's reputation. The number of people victimized by phishing scams has grown exponentially over the past decade, with millions of victims each month. This growth presents an increasingly difficult challenge for organizations to protect themselves from this growing threat. Nowadays, email phishing has become increasingly challenging to detect and prevent. Phishers constantly evolve their methods to evade detection by security solutions and law enforcement agencies. Organizations must be able to detect and prevent these attacks at scale if they wish to avoid falling victim to this threat. Vrbančič et al. (2018) [5] define the phishing attack as pervasive fraud that occurs when a fake website imitates a real one in order to

acquire data from unsuspecting users. A duplicate website that resembles a legitimate website is created for phishing, making it difficult for users to detect [6]. Nowadays, phishing is a widely known subject, and the impact of successful phishing attacks has become disastrous [4]. Phishing attacks have put legitimate web resource owners and internet users at risk [5]. Even with powerful antivirus software, the recent rise in phishing attacks has led to a lack of trust in legitimate users, making them feel less safe.

According to a report released by the Anti-Phishing Working Group (APWG) in 2022, there were 1,097,811 phishing assaults in Q2 2022. There was a total of 1,270,883 phishing assaults recorded by APWG in quarter three of 2022 (Figure 1), making it the worst quarter for phishing ever recorded by the organization. In early 2020, APWG saw between 68,000 and 94,000 phishing assaults per month; by the end of 2022, that figure had tripled [7].

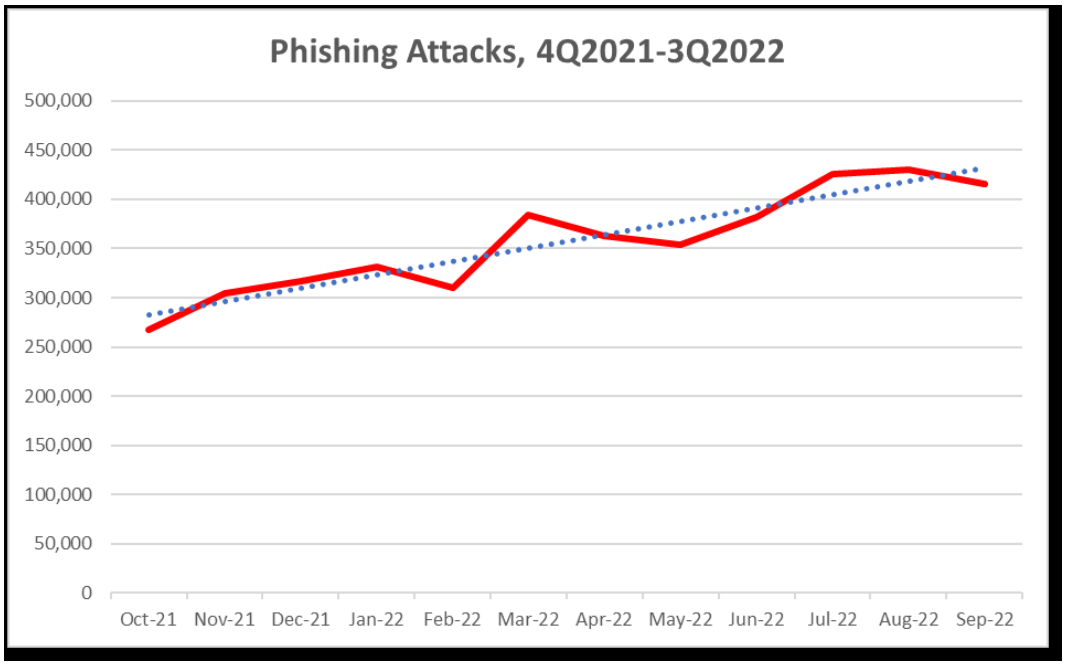

Figure 1. Phishing attacks from October 2021 to September 2022 [7] (red line depicts the actual number of attacks, blue dashed line depicts the approximate increase).

Deep learning is a machine learning type that involves using artificial neural networks (ANNs) to analyze and classify data. A neural network can process and learn from large amounts of data, making it practical for text and image classification [8]. This study aims to find the best approach for email phishing detection using deep learning techniques. The research develops techniques that can accurately identify and flag phishing emails. Unlike previous studies that have evaluated these techniques individually, our research compares their performance directly on a large real-world email dataset. This enables unique insights into these models' relative strengths and weaknesses for phishing detection. Our study identifies the optimal deep learning architectures and guides selecting suitable techniques to build real-world phishing detection systems based on extensive empirical evaluation. This direct comparative analysis on a robust dataset is a novel contribution, filling an essential gap in selecting deep learning methods for email security. The contribution of the research is as follows:

- Developing deep learning techniques that accurately identify emails as either phishing or legitimate based on features such as subject line, content, and sender information.
- Improving the efficiency and speed of deep learning algorithms for email phishing detection to enable the real-time analysis of incoming emails.
- Investigating techniques such as feature selection, transfer learning, and graph theory in combination with deep learning to enhance the accuracy of email phishing detection.

- Evaluating the effectiveness of different deep learning architectures and techniques for email phishing detection, including convolutional neural networks (CNNs), long short-term memory (LSTM) networks, recurrent neural networks (RNNs), and bidirectional encoder representations from transformers (BERT).

The remaining parts of this research are organized as follows: Section 2 presents a brief background of using deep learning in cybersecurity. Section 3 reviews the current deep learning techniques for detecting and protecting against various email phishing attacks. The proposed model is presented in detail in Section 4, while the performance evaluation of the proposed model is shown in Section 5. Finally, Section 6 concludes the research.

2. Background

Deep learning algorithms can learn from data by adjusting the weights and biases of the artificial neurons in the network, allowing them to adapt and improve their performance over time (Figure 2). This makes deep learning particularly useful for tasks with a large amount of data available, as algorithms can learn from these data to make more accurate predictions or classifications [8].

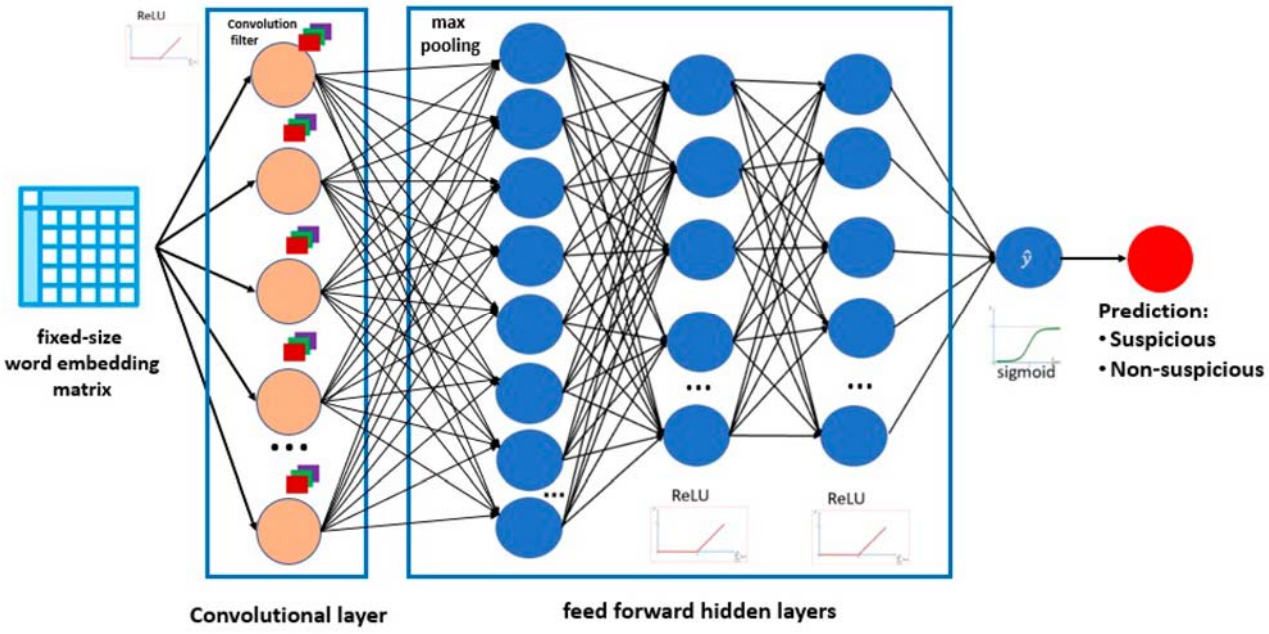

Figure 2. Convolutional neural network [9].

Cybersecurity is one of the most important and complex fields in the 21st century. Cybersecurity includes many sub-fields, such as network security, data security, application security, etc. In this field, many new techniques are coming up daily for detecting and overcoming cyber threats. Learning about these techniques is not enough because it only helps one understand the terminology or find vulnerabilities but will not help one perform actions against cyber threats. To perform actions against cyber threats, one must learn how to develop cognitive skills [10].

While one can protect their organization in many ways, one of the best ways is to use artificial intelligence (AI) in cybersecurity (Figure 3). Artificial intelligence can quickly analyze vast amounts of data and accurately identify threats without human interaction. Machine learning, deep learning, or any other form of AI can be extremely effective for this task because it processes large amounts of data in an unsupervised manner using a previously trained model simultaneously and in a fast way [11].

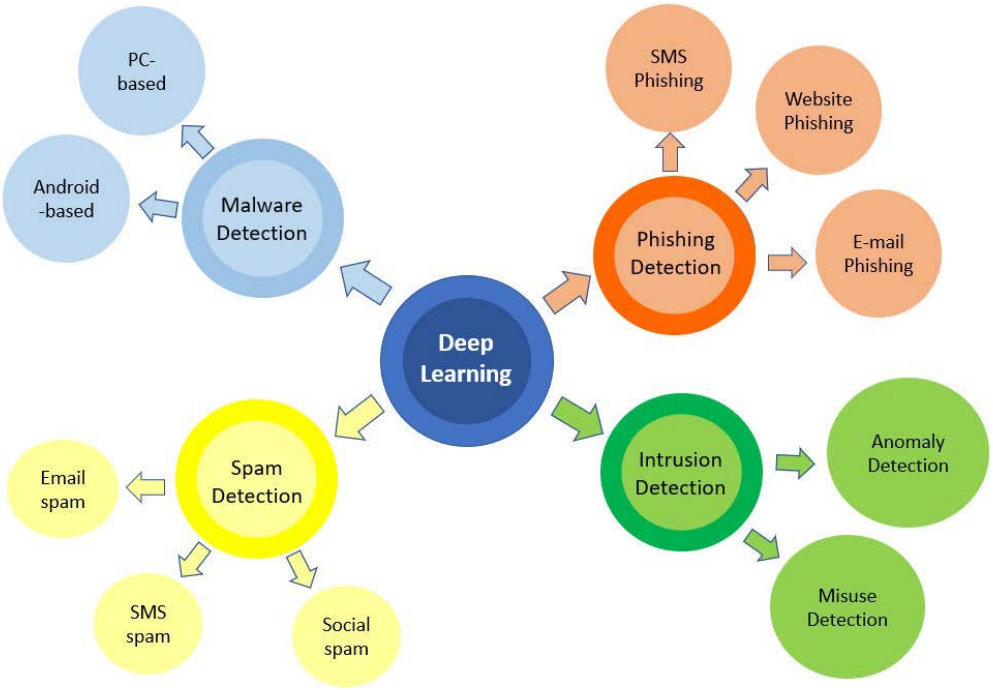

Figure 3. Deep learning application in cybersecurity [10].

Deep learning has shown great potential for use in cybersecurity, particularly in intrusion detection, malware classification, and phishing detection. One example of using deep learning in cybersecurity is intrusion detection, where algorithms can be trained to recognize patterns and anomalies in network traffic that may indicate a potential security threat [12]. Deep learning has also been used for malware classification, where algorithms are trained to identify malicious software based on features such as file structure and behavior. In phishing detection, deep learning algorithms have been used to classify emails as either legitimate or fraudulent based on features such as subject line and content [13]. Overall, deep learning in cybersecurity can significantly improve the effectiveness of security measures by allowing algorithms to learn and adapt to new threats in real time.

3. Related Work

This section will discuss the latest techniques for detecting and protecting against various email phishing attacks, including machine learning, deep learning, and hybrid models. This section will also cover different deep learning models that are commonly used in email phishing detection, such as CNNs, RNNs, LSTM networks, and BERT.

3.1. Machine Learning Approaches

The authors of [6] proposed a phishing detection model that utilizes multiple machine learning algorithms to enhance the overall performance and robustness of the model. The system extracts a set of features from the website, including URL and HTML content, and then these features are used to train the models. The experiment was conducted on a dataset of phishing and legitimate websites, and it was found that the system achieved high accuracy in detecting phishing websites, with the best results using random forest and support vector machine methods. However, the approach is not generalized to a new unseen dataset and is limited to phishing website detection.

In their paper, Valecha et al. (2021) [14] proposed a method for detecting phishing emails using persuasion cues. The research specifically focuses on gain and loss persuasion cues. It creates three machine learning models using these cues: one with relevant gain persuasion cues, one with relevant loss persuasion cues, and one with a combination of gain and loss persuasion cues. The models were then compared to a baseline model that does

not account for the persuasion cues. The results indicated that the models with relevant persuasion cues outperformed the baseline model by around 5–20% in terms of F-score, showing that including persuasion cues in anti-phishing methods can effectively detect and block phishing emails.

3.2. Deep Learning Approaches

In their paper, the authors of [15] developed a spam filter that combined an integrated distribution-based balancing approach with an N-gram tf-idf feature selection and a deep multilayer perceptron neural network with rectified linear units. This filter accurately detected spam emails in the Enron and SpamAssassin benchmark datasets, even when many different features and additional layers were present. The authors found that accuracy decreased as the number of features decreased, but the ANN and decision trees still performed well with minimal datasets. They also noted that shallow neural networks are unsuitable for high-dimensional datasets and are computationally expensive, so they incorporated N-gram tf-idf feature selection in their approach. Overall, the proposed filter effectively detected spam emails through the integrated distribution-based balancing approach [15].

In their work, the authors of [16] presented a phishing email detection model called THEMIS that utilized an improved recurrent convolutional neural network (RCNN) model with multilevel vectors and attention mechanisms. This model could simultaneously model email headers, words, email body, and characters, allowing it to identify phishing emails effectively. To evaluate the effectiveness of their model, the authors used an unbalanced dataset from the First Security and Privacy Analytics Anti-Phishing Shared Task (IWSPA-AP 2018). The evaluation of THEMIS resulted in a high accuracy of 99.848% and a low false-positive rate of 0.043%, demonstrating the model's effectiveness in detecting phishing emails using natural language processing. Overall, Fang et al. (2019) [16] presented a promising phishing email detection model that could be used to improve the security of email systems.

Alhogail and Alsabih (2021) [17] proposed a phishing email detection model that utilized deep learning and natural language processing on the email body to extract features and improve phishing detection. The model was based on a convolutional network (GCN) and was developed as a supervised learning model. The model was trained and tested on a publicly available fraud dataset, including phishing and legitimate emails. The dataset was balanced and suitable for use in supervised learning algorithms. The experimental results showed that the proposed model attained a 98% accuracy rate and a false-positive rate of 0.015%, demonstrating the model's effectiveness in detecting phishing emails. Overall, the study by Ref. [17] highlighted the potential of using machine learning and natural language processing techniques to improve the security of email systems by detecting phishing emails. In their work, Yao et al. (2019) [18] explored using graph convolutional networks (GCN) for text classification. The authors proposed a GCN-based model for text classification and evaluated its performance on several benchmark datasets. The results showed that the proposed model achieved competitive performance compared to other state-of-the-art models and demonstrated the potential of using GCN for text classification tasks. Overall, the authors of [18] presented a promising approach for text classification using GCN and highlighted the potential of this technique in various natural language processing tasks.

The authors of [19] conducted a study on deep learning techniques for detecting spam emails in English-language text emails. The authors proposed a model that utilized deep learning to classify spam emails based on features extracted from the text of the emails. The model was developed as a supervised learning model and was trained and tested on a dataset of spam and non-spam emails. The results of the study showed that the proposed model was able to achieve high accuracy in detecting spam emails. The authors also discussed the potential applications and future directions for using deep learning

techniques in spam email detection. Overall, Ref. [19] highlighted the potential of using deep learning to improve the security of email systems by detecting spam emails.

In their work, Singh et al. (2020) [20] conducted a study on using deep learning techniques for detecting phishing attacks from URLs. The authors proposed a model that utilized deep learning to classify URLs as either phishing or non-phishing based on features extracted from the URL. The system can achieve an accuracy of 98.00%, outperforming a previous model that achieved 97.98% accuracy. One of the system's benefits is that it does not require manual feature engineering, as CNNs can extract relevant features directly from the URLs. This is a significant advantage over previous approaches, which can be time-consuming and labor-intensive. Overall, the system represents a promising solution for detecting and preventing phishing attacks.

Saha et al. (2020) [21] proposed a framework for detecting phishing web pages using deep learning techniques. They used a multilayer perceptron, or feed-forward neural network, to analyze a dataset of 10,000 web pages collected from Kaggle. The dataset includes ten attributes, such as the URL, the website's age, and the presence of certain words or symbols. The authors preprocessed the data by converting categorical attributes to numerical values and splitting the dataset into training and test sets. They then trained the multilayer perceptron on the training data and evaluated its performance on the test data. The model achieved 95% accuracy on the training data and 93% on the test data. The authors concluded that deep learning approaches can effectively detect phishing attacks and suggested that further research may be beneficial. They also noted that their framework could be improved by incorporating additional features or using more advanced deep learning techniques.

By analyzing the content of email messages, McGinley and Monroy (2021) [22] validated the effectiveness of CNN models in identifying phishing emails, achieving an accuracy rate of 98%. The proposed model takes an embedding of text in the body of an email as input and outputs a probability that indicates the possibility that the email is malicious. Fetooh et al. (2021) [23] proposed a real-time attack detection model for wireless networks that analyzes several static and dynamic parameters while conduct a frame-type analysis in order to detect various wireless attacks. The evaluation revealed that the model's average accuracy was 94.40%.

Gogoi and Ahmed (2022) [24] proposed a deep learning model that employs the BERT and DistilBERT pre-trained transformer models to detect phishing. The proposed detection model successfully addressed the difficulties in phishing detection, such as how conventional feature extraction approaches are ineffective in identifying phishing emails, achieving an accuracy rate of 99%. Doshi et al. (2023) [25] proposed a deep learning detection model to classify phishing and spam emails by incorporating email body and content features. The proposed model effectively addresses the problem of data imbalance in spam classification and email phishing with accurate classification. The model relies on a dual-layer architecture with a learned or pre-trained model in each layer that categorizes data instances into the appropriate classes. The proposed model employed ANN, RNN, and CNN models. The achieved accuracy rate was 99.51%.

Benavides-Astudillo et al. (2023) [26] recently proposed a phishing attack detection model that employs deep learning and natural language processing to identify phishing attacks on web pages. The detection system was created using the Phishload dataset. To extract features, the text content of the web pages is analyzed. After training the model, the obtained validation accuracy was 98%. In their work, Aldakheel et al. (2023) [27] presented a detection model for identifying phishing websites, where a CNN was utilized to separate legitimate websites from phishing websites effectively. The effectiveness of the proposed detection model was evaluated using the PhishTank dataset, a popular dataset for identifying phishing websites based just on URL features. The achieved accuracy rate was 98.77%.

3.3. Hybrid Approaches

In their paper, He et al. (2022) [28] proposed a double-layer detection framework based on deep learning technology for detecting social engineering attacks, which uses phishing emails as the medium and targets specific groups of people. The first layer of the system uses machine learning algorithms, specifically the long short-term memory (LSTM) and the extreme gradient boosting tree (XGBoost), to detect phishing emails. In contrast, the second layer uses a bidirectional LSTM and attention mechanism to detect insider threats. The framework also includes a social engineering attack and defense simulation platform. The system does not require manual feature extraction and can accurately identify phishing emails and insider threats. The experimental results showed that the proposed framework effectively detected phishing attacks and insider threats faced by enterprise systems and can be used in real-world situations. Bagui et al. (2019) [29] developed a novel model that applies machine learning, deep semantic analysis, and deep learning techniques to classify emails as phishing or non-phishing. They used one-hot encoding with and without word phrasing for deep semantic analysis. They compared the results of several machine learning classifiers, such as SVM, decision tree, and naive bayes, and deep learning classifiers like LSTM, CNN, and word embedding. The authors used deep learning techniques to analyze the text of emails and extract inherent characteristics for phishing detection. Overall, the authors aimed to improve the accuracy of detecting phishing emails by applying deep learning and semantic analysis techniques to classify emails.

Table 1 summarizes the proposed works discussed in this section related to phishing detection. The literature review revealed several deep learning techniques, i.e., CNN, RNN, LSTM, and other useful ways to conduct this research. The literature review also examined the feature selection process and identified key features for detecting spam and phishing emails. It also evaluated the performance of various techniques in terms of accuracy. A standard limitation found in the literature is the need for a sufficiently large dataset that can impact the detection accuracy, which is the aim of this research.

Table 1. Summary of the discussed works in the literature.

Authors	Purpose	Major Themes	Limitations	Accuracy
(Zamir et al., 2020) [6]	Phishing website detection	Using machine learning and neural networks	-	97.40%
(Valecha et al., 2021) [14]	Phishing email detection	Machine learning models	-	96.52%
(Barushka & Hajek 2018) [15]	Categorizing spam and non-spam messages	Integration of DBB-RDNN-ReL for feature selection Demonstrations of "Shallow Neural Networks"	-	98.51%
(Fang et al., 2019) [16]	Detecting phishing emails	Use of diversified datasets Use of recurrent convolutional neural network (RCNN) model with multilevel vectors and attention mechanism	Language-specific Limited to text emails	99%
(Alhogail & Alsabih et al., 2021) [17]	Email phishing detection and deep learning	Natural language processing and graph conventional network	Difficulty in obtaining a large and diverse dataset	98.2%

Table 1. *Cont.*

Authors	Purpose	Major Themes	Limitations	Accuracy
(Yao et al. 2019) [18]	Text classifications	Considering the whole corpus with embeddings to classify words based on their occurrences and document relationships	-	Text GCN performs the best accuracy at 86.34%
(Kaddoura et al., 2020) [19]	Spam email detection	Feed-forward neural network (FFNN) Bidirectional encoder representations from transformers (BERT)	Language-specific Limited to text emails	F1-score of 99.18%
(Singh et al. 2020) [20]	URLs phishing detection	Convolutional neural networks	-	98%
(Saha et al., 2020) [21]	Phishing attacks detection	Multilayer perceptron (MLP)	Limited generalizability	93%
(McGinley and Monroy 2021) [22]	Classifying phishing emails	CNN		98%
(Gogoi and Ahmed 2022) [24]	Identifying phishing emails	Pre-trained transformer models		99%
(Doshi et al. 2023) [25]	Classifying phishing and spam emails	ANN, RNN, and CNN	-	99.51%
(Benavides-Astudillo et al., 2023) [26]	Detection system for phishing attacks on web pages	Natural language and deep learning	-	98%
(Aldakheel et al. 2023) [27]	Identifying phishing websites	CNN	-	98.77%
(He et al., 2022) [28]	Detection system against social engineering attacks	LSTM and XGBoost	-	98.35%
(Bagui et al., 2019) [29]	Detecting phishing emails	Machine and deep learning	-	98.89%
(Tang & Mahmoud, 2022) [30]	Phishing website detection	RNN-GRU	-	99.18%

4. Proposed Model

This section presents the steps to implement the proposed model, including collecting, preparing, and utilizing a dataset for training and testing deep learning models for detecting phishing emails. Figure 4 illustrates the general framework followed throughout the research; it includes dataset acquisition, data preparation, feature extraction, and training and testing of various deep learning approaches. The following subsections describe the research methodology.

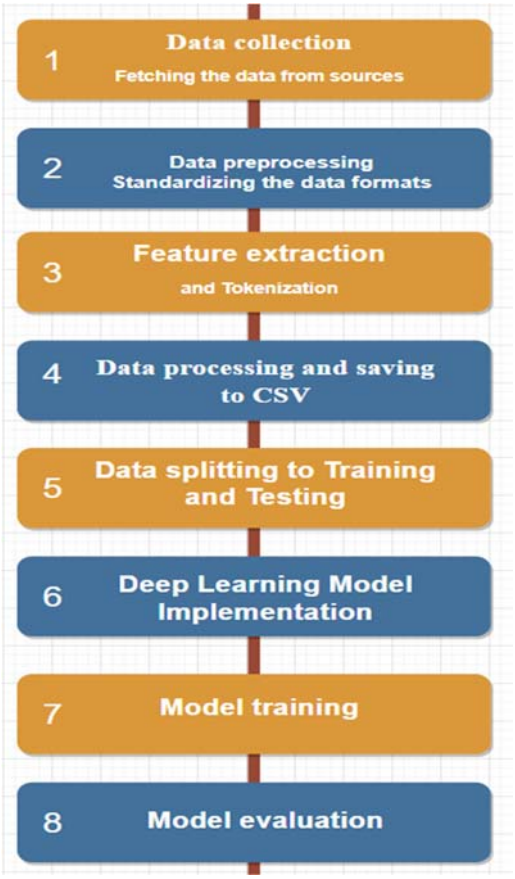

Figure 4. Methodology for phishing email detection.

4.1. Overall Methodology

The deep learning models were implemented using TensorFlow and Keras libraries in Python. Specifically, TensorFlow was used to provide data pipelines and high-performance numerical operations. Keras was used on top of TensorFlow to build and train the deep learning models using high-level APIs like Sequential and functional APIs. The key Keras layers utilized include dense, LSTM, bidirectional, and dropout layers. In addition, Keras preprocessing tools like the Tokenizer and pad sequences were used for text processing. Scikit-learn was used for machine learning tasks like model evaluation and data splitting. Key NLP libraries include NLTK for text processing and the TensorFlow Transformers library for implementing BERT. Pandas and NumPy were used for data handling, while Matplotlib and Seaborn were used for visualization. This combination of libraries provided an optimized environment for rapidly developing and evaluating deep learning models for phishing detection. The research methodology for this study involved several steps to apply deep learning techniques to detect phishing emails. These steps included:

- Data collection: Acquiring a dataset of phishing and benign emails for training and testing the model.
- Data preparation: Preprocessing the data as needed to prepare them for input into the model, such as by cleaning or tokenizing the text.
- Feature extraction: Writing code to extract relevant features from the emails that can be used to train the model.
- Algorithm selection: Analyzing various deep learning algorithms and selecting the most suitable ones to classify phishing emails.
- Data processing: Processing the data using Python code and saving them to a CSV file.
- Data splitting: Splitting the data into training and testing datasets.

- Model training: Implementing the selected algorithms on the training dataset and training the model. The trained model was saved for later use.
- Model evaluation: Using the trained model to classify emails in the test dataset and evaluate its performance using various metrics.
- Comparison: Comparing the performance of different deep learning algorithms and identifying the best model for phishing email detection.

This study uses a quantitative approach and applies deep learning techniques to open-source datasets of phishing and benign emails. These datasets were used for analysis, comparison, training, and testing of deep learning-based algorithms for detecting phishing emails.

4.2. Dataset Selection

Dataset selection is an important step in any machine learning or deep learning research, as the quality and diversity of the data can significantly impact the model's performance. When selecting a dataset for phishing email detection, there are several factors to consider:

- Size: It is important to use a large dataset in order to provide the model with enough samples to learn from. A dataset with enough samples can help the model to generalize better to new, unseen data.
- Diversity: The dataset should be diverse, with various emails representing phishing and benign messages. This can help the model learn more robust and accurate patterns for detecting phishing emails.
- Quality: The data should be high-quality, with minimal errors or inconsistencies. This can help improve the accuracy of the model.
- Relevance: The data should be relevant to the task at hand. For phishing email detection, the dataset should include a representative sample of phishing and benign emails like those the model will encounter in the real world.

This study used publicly available datasets to train and test deep learning techniques for classifying phishing emails. The choice of the dataset can significantly impact the accuracy and performance of the model, so it is important to select a high-quality and relevant dataset. Several resources are available for finding publicly available datasets, including Google Dataset Search, Kaggle, and the UCI Machine Learning Repository. This research used publicly available datasets. These datasets were accessed in CSV and text file formats and used for initial analysis, classification, training, and testing using various deep learning techniques. The performance was evaluated and compared to identify the best model for detecting phishing emails.

4.2.1. Phishing Email Dataset

An email phishing dataset is a collection of email messages that have been specifically selected or created to be used for research on and analysis of phishing attacks. These datasets typically include a set of phishing emails that have been collected from various sources, such as the UCI machine learning repository and SpamAssassin. They may be labeled or categorized in some way, such as by type of phishing attack or by the industry or organization targeted. Email phishing datasets can be used to train and evaluate deep learning-based phishing detection systems and to research the characteristics and methods used in phishing attacks.

4.2.2. Benign Email Dataset

The Enron email dataset is a collection of email messages and metadata from the Enron Corporation, an American energy, commodities, and services company at the center of one of the largest corporate scandals in history. The dataset was created as part of the investigation into the company's financial collapse and contains over half a million emails from over 150 users, along with various other data such as financial data, news articles, and message routing information. The Enron email dataset is widely used for

research in natural language processing, machine learning, deep learning, organizational communication, and corporate governance research. Because of the large volume of data and the variety of topics covered, it is a rich source of information for researchers. It has been used to train machine learning and deep learning models to classify emails into different categories, such as spam or phishing emails. It is worth mentioning that the Enron email dataset has been preprocessed and cleaned to remove personally identifiable information (PII) and confidential data. Overall, the Enron email dataset is valuable for researchers, especially those interested in natural language processing, machine learning, and corporate governance.

4.3. Data Preparation and Preprocessing

In order to prepare the dataset for feature extraction and deep learning, it is necessary to clean the data and remove any unnecessary words or characters. The phishing and benign email datasets were cleaned using Python's natural language processing (NLP) techniques [31]. The implemented data underwent preprocessing steps to prepare the email text data before feeding them into the deep learning models. For the CNN, RNN, and LSTM models, the text data were converted to lowercase and stripped of special characters, numbers, and stop words using regular expressions, the NLTK library, and custom functions. The remaining words were then stemmed using the Snowball stemmer to reduce them to their root form. For the BERT model, the text data were tokenized using the Bert Tokenizer from the Transformers library by truncating sentences to 512 tokens and generating input IDs and attention masks. These preprocessing techniques helped reduce noise in the data and enabled the deep learning models to learn more effectively from the cleaned text. Preprocessing the email data was essential to improve model accuracy by highlighting the key textual features and patterns for identifying phishing attacks.

4.3.1. Removing HTML Tags

HTML tags are codes used to format and structure web content, and they can be removed from the text of the emails using a parser or regular expressions. However, certain HTML tags may contain information relevant for detecting phishing emails, such as "form" tags used to create a fake login page. These relevant tags can be retained while others are removed.

4.3.2. Remove Numbers

Numbers and digits that are irrelevant to detecting phishing emails can be removed from the dataset. This can help reduce noise and improve the model's ability to learn relevant patterns. However, specific numbers, such as IP addresses, may be useful as features and should be retained.

4.3.3. Remove Punctuations

Punctuation marks and special characters, such as commas, full stops, and various symbols, can be removed from the dataset using regular expressions or string manipulation techniques. This can help reduce the vocabulary size and make the data more manageable for the model to process.

4.3.4. Remove Stop Words

Stop words are common and general words that do not provide much value for natural language processing in deep learning. Examples of stop words include "a", "an", "the", "and", "but", etc. Stop words can be removed from the dataset using a list of stop words or by applying a stop word filter provided by an NLP library.

4.3.5. Remove Infrequent Words

Words that occur very infrequently in the dataset can be removed, as they may not provide much information for the model to learn from. This can help reduce the vocabulary size and improve the model's generalization ability.

4.3.6. Stem Words

Stemming involves reducing words to their base form, which can help reduce the vocabulary size and improve the model's generalization ability. For example, the term "jumping" can be stemmed to "jump", and the word "jumps" can be stemmed to "jump". Various stemming algorithms are available, such as Porter stemming or Snowball stemming.

4.4. Feature Extraction

Feature extraction is a crucial step in text classification as it involves converting the raw text data into a numerical representation that deep learning models can process. The main objective of this step is to extract meaningful information from the text that can be used to train and evaluate the classifier. One of the most common feature extraction approaches is splitting the text into words and tokenizing each word. Tokenization is the process of breaking down the text into individual words or phrases. This step is important because it allows us to convert the text into a numerical representation that can be used as features for training the model. For feature extraction with the CNN, RNN, and LSTM models, the email text data were converted into numerical representations using the Tokenizer class from Keras. The tokenizer generated a vocabulary of words and mapped each word to an integer ID. The maximum vocabulary size was 10,090 unique words. The text sequences were padded to a uniform length of 10,090 tokens to enable batch training. For the BERT model, features were extracted using the pre-trained uncased BERT base model tokenizer from Hugging Face Transformers. This tokenized the text into WordPiece tokens and generated input IDs and attention masks with a maximum sequence length 512. The BERT tokenizer has a vocabulary size of 30,522 WordPiece tokens. These standardized tokenizers enabled effective feature extraction from the raw email text into numerical representations that the deep neural networks could process for phishing detection. The tokenization and vocabulary mapping were essential preprocessing steps for generating informative features from the text for the deep learning models.

4.5. Data Processing and Labeling

Identifying and extracting features from the phishing and benign email datasets was conducted using a Python script. The script created a CSV file containing all the identified features, marked with a 1 if present or a 0 if not present, as well as the total number of characters in the email. The email was also labeled as a phishing email (marked with a 1) or a non-phishing email (marked with a 0). This CSV file was used to train and test deep learning models for email phishing detection.

4.6. Splitting Data

Once the features were extracted and the data were labeled and prepared in a CSV file, the next step was to split the data into training and test datasets. The training dataset was used to train the deep learning model, while the test dataset was used to evaluate the model's performance.

There are various approaches to splitting data into training and test sets. One approach is a fixed split, where a certain percentage of the data is allocated to the training set, and the remaining data are allocated to the test set. For example, the data may be split 70/30, with 70% of the data used for training and 30% for testing. A 70/30 splitting approach was adopted in our proposed detection model. Another approach is to use stratified sampling, where the data are split in a way that preserves the relative proportions of different classes or categories in the dataset. For example, suppose the dataset contains a balanced number of phishing and non-phishing emails. In that case, stratified sampling can ensure that the

training and test sets contain a balanced number of phishing and non-phishing emails. It is important to carefully split the data into training and test sets to ensure the model can generalize to unseen data and to accurately evaluate its performance.

4.7. Train and Test Deep Learning Models

In this study, different deep learning models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and bidirectional encoder representations from transformers (BERT) were used. The following sections will discuss the different deep learning models utilized in the research. The proposed model implemented standardized feature extraction and model training processes using Keras and Scikit-learn. For the CNN, RNN, and LSTM classifiers, the email text was tokenized into numerical representations using the Keras Tokenizer with a maximum vocabulary size of 10,090 words. The sequences were padded to equal lengths to enable batch training. For BERT, features were extracted using the pre-trained BERT tokenizer from HuggingFace, generating WordPiece tokens and attention masks up to 512 tokens long. During training, the proposed model was optimized using dropout and early stopping techniques to prevent overfitting. The trained models were serialized for reuse in phishing email classification. Overall, leveraging robust feature extraction and training workflows in Python enabled rapid experimentation and evaluation of the deep learning architectures on the real-world email dataset.

4.7.1. Convolutional Neural Networks (CNNs)

CNNs are deep learning models commonly used for image and video recognition, natural language processing, and other tasks. CNNs are designed to learn spatial hierarchies of features automatically and adaptively from input data. A CNN typically consists of an input layer, multiple hidden layers, and an output layer. The input layer receives the input data, such as an image or text. The hidden layers, known as convolutional layers, perform mathematical operations on the input data to extract features. The final output layer produces the output, such as a classification label or a probability distribution. The output layer has an activation function for the binary classification, such as the Sigmoid function. Several methods can be used to optimize a CNN, such as data preprocessing, hyperparameter tuning, regularization, and early stopping [32].

4.7.2. Recurrent Neural Networks (RNNs)

RNNs are deep learning models that process sequential data, such as time series data, text, or speech. RNNs can maintain a hidden state that can be updated based on the current input and previous hidden states. This allows RNNs to keep track of context and dependencies over time, making them well suited for tasks such as language translation, sentiment analysis, and speech recognition. RNNs consist of an input layer, one or more recurrent layers, and an output layer. The recurrent layer updates the hidden state and outputs a prediction based on the current input and the previous hidden state. To optimize RNNs, several methods besides the methods for CNNs, such as gradient clipping, have been proposed [33].

4.7.3. Long Short-Term Memory (LSTM)

LSTM is a variant of RNN designed to handle vanishing gradients, a common problem in traditional RNNs. LSTM uses a special type of memory cell that can selectively retain or forget information from the previous time steps, allowing it to better handle long-term dependencies in sequential data [34].

4.7.4. Bidirectional Encoder Representations from Transformers (BERT)

BERT is a deep learning model designed to perform natural language processing tasks, such as language understanding, translation, and text classification. BERT is based on the transformer architecture, which uses self-attention mechanisms to allow the model to

focus on different parts of the input data depending on the task. BERT is trained using pre-training, where the model is trained on a large corpus of text data to learn general language representations. Once pre-trained, BERT can be fine-tuned on a specific task, such as text classification, by adding a small number of task-specific layers on top of the pre-trained model. When using BERT for text classification, the first step is to fine-tune the pre-trained BERT model on a labeled dataset of text data. The fine-tuned BERT model can then be used to classify new, unseen text data based on the representations it has learned during pre-training and fine-tuning. One of the advantages of BERT is its ability to understand the context of the text, which is important for text classification tasks. BERT can also extract features from text data, which can be used in other models. To optimize BERT for text classification, it is important to use a high-quality and relevant dataset; carefully select the hyperparameters, such as the number of layers, the number of memory cells, and the learning rate; and use regularization techniques such as dropout to prevent overfitting [35].

4.8. Justification

Based on the literature review, it was determined that classification in deep learning is commonly used to classify data into various categories based on certain parameters. Since identifying phishing emails involves classifying the emails as either phishing or legitimate, it is necessary to use classification methods that operate based on features and binary classification. CNNs, RNNs, LSTM, and BERT are powerful deep learning models that can be used for email phishing detection due to their ability to handle and analyze large amounts of data and identify patterns and features that may indicate a phishing attack. CNNs are particularly useful for analyzing text-based data, such as the text of an email or the sender's address. They can extract features from the input data and be used as feature extractors or classifiers. RNNs and LSTM are well suited for language translation, sentiment analysis, and speech recognition tasks. They can maintain a hidden state that can be updated based on the current input and previous hidden states. This allows RNNs and LSTM to keep track of context and dependencies over time, making them suitable for email phishing detection by analyzing the text of an email and extracting relevant features. BERT is a deep learning model designed to perform natural language processing tasks, such as language understanding, translation, and text classification. BERT is trained using pre-training, where the model is trained on a large corpus of text data to learn general language representations. This makes BERT particularly useful for email phishing detection because it can understand the context of the text, which is important for text classification tasks.

4.9. Evaluation Metrics

A program was developed to evaluate the performance of CNNs, RNNs, LSTM, and BERT for email phishing detection. The program allows users to select a training dataset and a test dataset and select a deep learning algorithm from a provided list. The selected algorithm is then be trained on the training dataset, and the trained model is used to classify the test dataset. Several metrics were used to evaluate the performance of the models, including accuracy, precision, recall, and F1-score.

Precision is a metric that measures the proportion of predicted positives that are true positives, while recall measures the proportion of true positives that are correctly identified. The F1-score is the harmonic mean of precision and recall and is a more balanced measure of the model's performance [36]. The support is the number of instances in each class. These metrics were used to compare the different deep learning algorithms' performances and identify the best algorithm for email phishing detection. The results of this evaluation are presented in the results and evaluation section, providing insights into the effectiveness of each algorithm in detecting phishing emails. In addition, the results of this study are compared to the other relevant research papers in the field to evaluate its effectiveness and potential for future improvements.

4.10. Ethical Considerations

In compliance with legal and ethical standards, this study utilized only publicly available data from emails designated for research purposes. No personal emails from private email systems were used. The sources of these data are identified and referenced in the research. No specific authority, organization, or agency approval is required to use these publicly available data.

4.11. Summary

In this study, a quantitative method for research and analysis was used, providing numerical results and evaluations. Publicly available phishing datasets were used for deep learning (DL) and feature extraction, and their performance was evaluated. A logical approach was used for analysis and comparison. First, email messages were extracted from online repositories for detailed analysis. The data were divided into a training dataset and a test dataset. DL models were applied to the data to classify the emails as phishing or legitimate. The DL model results are presented and analyzed in subsequent sections.

5. Performance Evaluation

5.1. Research Results Overview

This research aimed to develop an effective and efficient email phishing detection system that could accurately identify phishing emails while minimizing false positives. To achieve this, we used a dataset of real-world phishing emails and legitimate emails, preprocessed and transformed the data, and finally trained and evaluated the performance of the selected deep learning algorithms. CNNs, RNNs, and LSTMs are popular deep learning algorithms that have been successful in various natural language processing tasks. CNNs are particularly suitable for feature extraction from input data, RNNs are effective in processing input data sequences, and LSTMs are known for capturing long-term dependencies in sequential data. In our research, we leveraged the strengths of these architectures to detect phishing emails using features such as email headers, content, and attachments. The results of our research demonstrate that deep learning models, particularly the RNN-based model, can effectively detect email phishing with high accuracy and low false-positive rates. Our findings suggest that a combination of RNN and LSTM models can achieve even better performance, while RNN models alone are less effective in detecting phishing emails.

5.2. Classification Report

The classification report method, which is a method available in the Scikit-learn library for evaluating the performance of classification models, was utilized for this research. It provides a comprehensive summary of the key metrics used to assess the accuracy of a model, including precision, recall, F1-score, and support. The classification report method takes in the true and predicted labels for a dataset. It generates a report that summarizes the performance of the classification model on that data. The report provides a table that includes the precision, recall, F1-score, and support for each class, as well as the average values across all classes.

5.3. Convolutional Neural Networks (CNNs)

The CNN-based deep learning classifier was implemented in Python and was trained and tested on the provided dataset. The testing accuracy was 98.74%, precision was 98.96%, recall was 98.78%, and F1-score was 98.87% after 50 epochs (Figure 5). The figure shows the accuracy and loss plots for the convolutional neural network model during training and validation. The training accuracy starts around 90% and steadily increases with each epoch, reaching over 99% by the 50th epoch. This indicates that the model could fit the training data better with each iteration and minimize errors on the samples it was trained on. The validation accuracy follows a similar trend, starting near 90% and increasing to around 98% by the 50th epoch. However, it is slightly below the training accuracy throughout,

indicating some overfitting. The training loss starts around 0.3 and decays rapidly in the first 10 epochs, plateauing under 0.1 by the 30th epoch. This shows that the model minimized the loss function and model error on the training set quickly at first. In addition, the validation loss drops sharply at first but then becomes levels around 0.1 by the 20th epoch, remaining steady after that. Table 2 shows the classification report obtained when evaluating the performance of the CNN model on the test dataset.

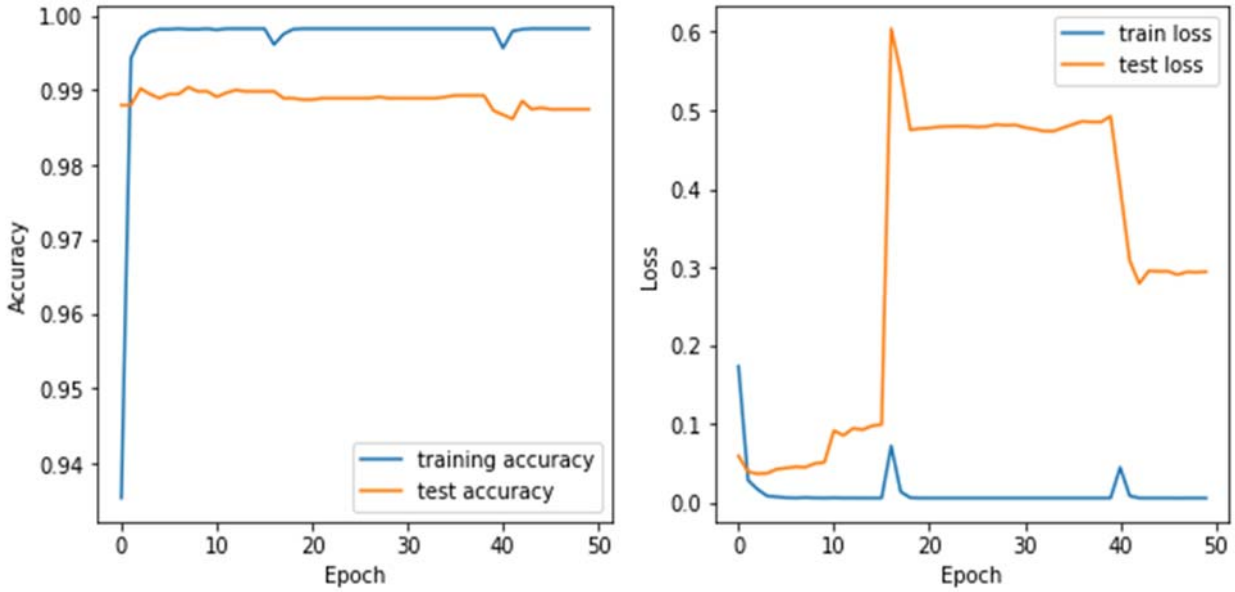

Figure 5. CNN model training and testing accuracy and loss over 50 epochs of training.

Table 2. CNN classification report.

	Precision	Recall	F1-Score	Support
0	0.98	1.0	0.99	3081
1	0.99	0.98	0.99	2331
micro avg	0.99	0.99	0.99	5412
macro avg	0.99	0.99	0.99	5412
weighted avg	0.99	0.99	0.99	5412
samples avg	0.99	0.99	0.99	5412

The CNN model demonstrated strong performance for phishing email detection, achieving an overall accuracy of 98.74% on the test set. This high accuracy indicates that the model correctly classified most phishing and legitimate emails. The precision of 98.96% shows that of all emails classified by the model as phishing, only a small fraction were mislabeled. The recall of 98.78% means the model could correctly detect almost all the actual phishing emails in the test set, with very few phishing emails missed. Finally, the F1-score of 98.87% reflects the excellent balance between precision and recall attained by the model. Overall, these metrics validate that the CNN model was highly proficient at distinguishing phishing and legitimate emails. The combination of high precision and recall underscores the model's reliability in flagging phishing emails while minimizing false alarms on legitimate emails.

This CNN classification report has two classes: 0 and 1. The model was evaluated on a dataset of 5412 samples. The precision, recall, F1-score, and support are reported in the table. For class 0, the precision is 0.98, meaning that out of all the samples predicted as class 0, 98% of them were class 0. The recall is 1.00, meaning that out of all the actual class 0 samples, 100% of them were correctly identified as class 0. The F1-score is 0.99, the

harmonic mean of precision and recall, and provides an overall measure of the model's accuracy for class 0. The support is 3081, indicating 3081 samples in class 0.

Similarly, for class 1, the precision is 0.99, meaning that out of all the samples predicted as class 1, 99% of them were class 1. The recall is 0.98, meaning that out of all the actual class 1 samples, 98% of them were correctly identified as class 1. The F1-score is 0.99, the harmonic mean of precision and recall, and provides an overall measure of the model's accuracy for class 1. The support is 2331, indicating 2331 samples in class 1.

The micro average is 0.99 for the entire dataset, considering all classes equally. The macro average, which is the arithmetic mean of the precision, recall, and F1-score for each class, is also 0.99. The weighted average, which is the average of the metrics weighted by the support of each class, is also 0.99.

In addition, the sample average, which is the average of the metrics calculated for each instance in the dataset rather than for each class, is 0.99. Overall, this classification report indicates that the CNN model has high accuracy in classifying both classes, with F1-scores of 0.99 for both classes and an overall accuracy of 0.99 for the entire dataset.

5.4. Recurrent Neural Networks (RNNs)

The RNN-based deep learning classifier was also implemented in Python and was trained and tested on the provided dataset. The testing accuracy was 98.58%, precision was 98.57%, recall was 98.53%, and F1-score was 98.55% after 10 epochs (Figure 6). The figure shows the accuracy and loss plots for the recurrent neural network model during training and validation. The training accuracy typically starts lower and increases with each epoch as the model fits the training data better, potentially reaching over 95-99% by the 10th epoch. The validation accuracy starts lower than training and increases at a lower rate. It is typically a few percentage points below the training accuracy. The training loss should decrease with each epoch as the model minimizes errors on the training set. It may drop sharply in the first few epochs and then start to plateau. Validation loss is similar to training loss but higher, reflecting the gap between training and validation performance. It decreases initially but may start to level off before the training loss.

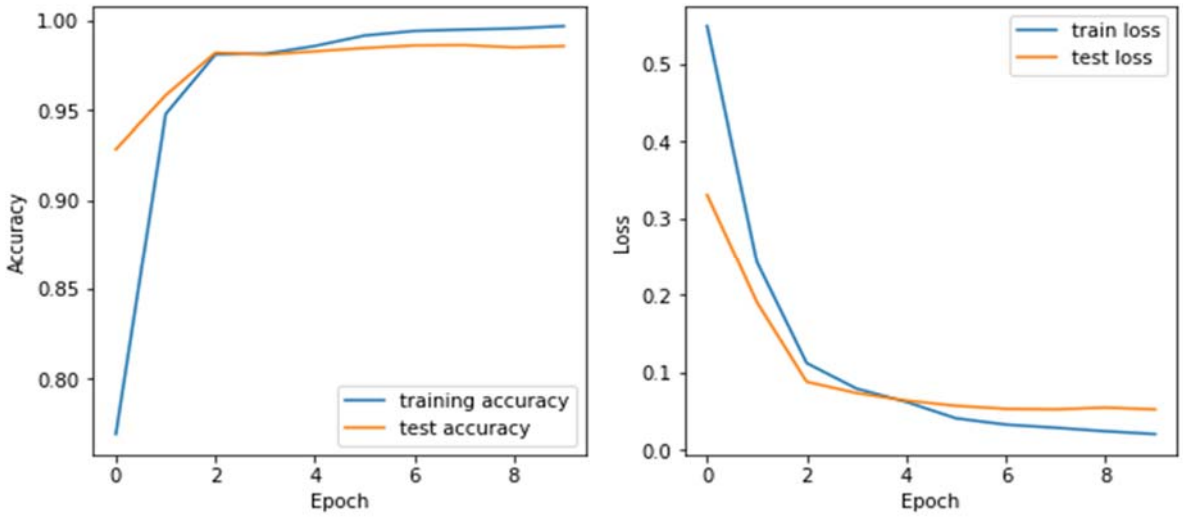

Figure 6. RNN model training and testing accuracy and loss over the 10 epochs of training.

Table 3 shows the classification report obtained when evaluating the performance of the RNN model on the test dataset.

Table 3. RNN classification report.

	Precision	Recall	F1-Score	Support
0	0.99	0.99	0.99	3081
1	0.99	0.98	0.98	2331
micro avg	0.99	0.99	0.99	5412
macro avg	0.99	0.99	0.99	5412
weighted avg	0.99	0.99	0.99	5412
samples avg	0.99	0.99	0.99	5412

The RNN model achieved a strong accuracy of 98.58% on the test set for phishing email detection. While slightly lower than the CNN model, this still indicates that the RNN could correctly classify most phishing and legitimate emails. The precision of 98.57% reflects that the RNN had very few false positives, incorrectly labeling legitimate emails as phishing only rarely. The recall of 98.53% means the model correctly identified most phishing emails, though it had a slightly higher false-negative rate than CNN, missing some actual phishing emails. The F1-score of 98.55% shows that the RNN attained a good balance between precision and recall. Overall, these metrics indicate reliable performance by the RNN model, with proficiency in detecting most phishing emails while avoiding false alarms. The RNN was slightly outperformed by the CNN, which suggests that the CNN is better suited for extracting distinguishing features from the raw email data.

This RNN classification report has two classes: 0 and 1. The model was evaluated on a dataset of 5412 samples. The precision, recall, F1-score, and support are reported in the table. For class 0, the precision is 0.99, meaning that out of all the samples predicted as class 0, 98% of them were class 0. The recall is 0.99, meaning that out of all the actual class 0 samples, 99% of them were correctly identified as class 0. The F1-score is 0.99, which is the harmonic mean of precision and recall, and provides an overall measure of the model's accuracy for class 0. The support is 3081, indicating 3081 samples in class 0.

Similarly, for class 1, the precision is 0.99, meaning that out of all the samples predicted as class 1, 99% of them were class 1. The recall is 0.98, meaning that out of all the actual class 1 samples, 98% of them were correctly identified as class 1. The F1-score is 0.98, which is the harmonic mean of precision and recall, and provides an overall measure of the model's accuracy for class 1. The support is 2331, indicating 2331 samples in class 1.

The micro average is 0.99, which is the overall precision, recall, and F1-score for the entire dataset, considering all classes equally. The macro average is also 0.99, which is the arithmetic mean of the precision, recall, and F1-score for each class. The weighted average is also 0.99, which is the average of the metrics weighted by the support of each class. The sample average is also 0.99, which is the average of the metrics calculated for each instance in the dataset rather than for each class. Overall, this classification report indicates that the RNN model has high accuracy in classifying both classes, with an F1-score of 0.99 for both classes and an overall accuracy of 0.99 for the entire dataset.

5.5. Long-Term Short Memory (LSTM)

The LSTM classifier was implemented in Python and trained and tested on the provided dataset. The testing accuracy was 98.89%, precision was 98.87%, recall was 98.56%, and F1-score was 98.87% after 10 epochs (Figure 7). The figure shows the accuracy and loss plots for the long short-term memory model during training and validation. The training accuracy usually starts low and increases with each epoch, potentially reaching 96-100% by the 10th epoch as the model fits the training data better. The validation accuracy also starts low and increases but at a slower rate than training accuracy. The training loss decreases with each epoch as the model minimizes errors on the training set. It may drop sharply in the early epochs and plateau as it converges. The validation loss follows a similar downward trend as training loss but at a higher value, reflecting the gap between

training and validation performance. Table 4 shows the classification report obtained when evaluating the performance of the LSTM model on the test dataset.

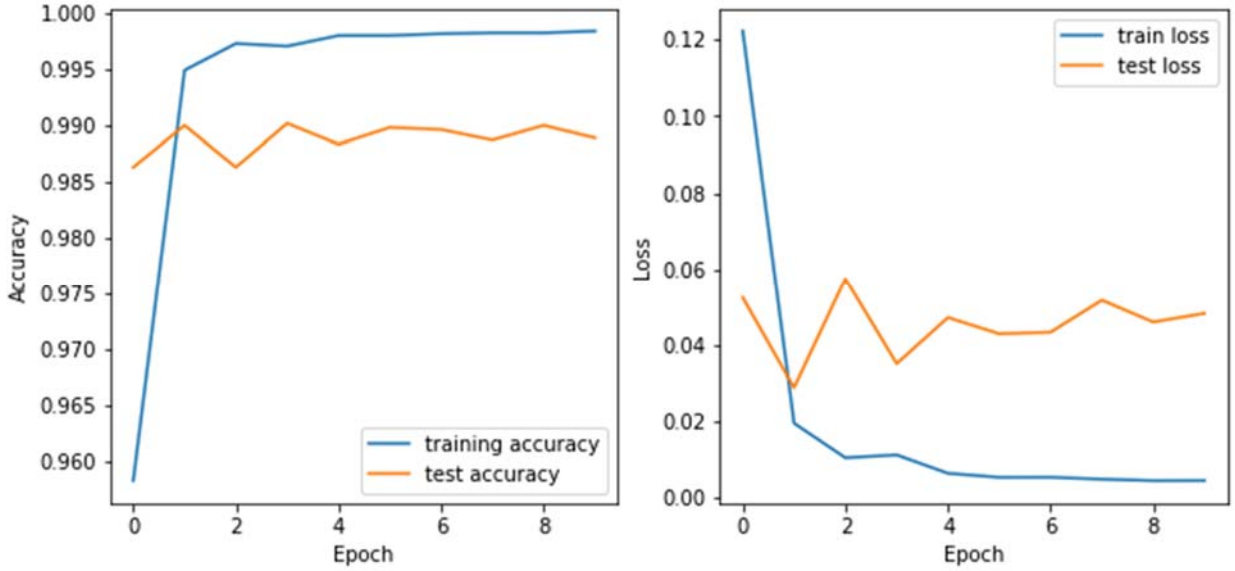

Figure 7. LSTM model training and testing accuracy and loss over the 10 epochs of training.

Table 4. LSTM classification report.

	Precision	Recall	F1-Score	Support
0	0.99	0.99	0.99	3081
1	0.99	0.99	0.99	2331
micro avg	0.99	0.99	0.99	5412
macro avg	0.99	0.99	0.99	5412
weighted avg	0.99	0.99	0.99	5412
samples avg	0.99	0.99	0.99	5412

The LSTM model achieved an accuracy of 98.89% on the test set, slightly higher than both the CNN and RNN models. This demonstrates that LSTM could correctly classify most phishing and legitimate emails. The precision of 98.87% indicates that the LSTM had very low false positives, rarely mislabeling legitimate emails as phishing. The recall of 98.86% means the model successfully identified almost all phishing emails, with minimal false negatives. The F1-score of 98.87% reflects the excellent balance between precision and recall attained. Overall, these metrics show that the LSTM performed very well, reliably detecting most phishing emails while avoiding false alarms. The LSTM's ability to retain the memory of the previous context likely allowed it to better analyze relationships in the email content.

This LSTM classification report has two classes: 0 and 1. The model was evaluated on a dataset of 5412 samples. The precision, recall, F1-score, and support are reported in the table. For class 0, the precision is 0.99, meaning that out of all the samples predicted as class 0, 99% of them were class 0. The recall is 0.99, meaning that out of all the actual class 0 samples, 99% of them were correctly identified as class 0. The F1-score is 0.99, which is the harmonic mean of precision and recall, and provides an overall measure of the model's accuracy for class 0. The support is 3081, indicating that there are 3081 samples in class 0.

For class 1, the precision is 0.99, meaning that out of all the samples predicted as class 1, 99% of them were class 1. The recall is 0.99, meaning that out of all the actual class 1 samples, 99% of them were correctly identified as class 1. The F1-score is 0.99, which is the

harmonic mean of precision and recall, and provides an overall measure of the model's accuracy for class 1. The support is 2331, indicating that there are 2331 samples in class 1.

The micro average is 0.99, which is the overall precision, recall, and F1-score for the entire dataset, considering all classes equally. The macro average is also 0.99, which is the arithmetic mean of the precision, recall, and F1-score for each class. The weighted average is also 0.99, which is the average of the metrics weighted by the support of each class.

The sample average is also 0.99, which is the average of the metrics calculated for each instance in the dataset, rather than for each class. Overall, this classification report indicates that the LSTM model has high accuracy in classifying both classes, with an F1-score of 0.99 for both classes and an overall accuracy of 0.99 for the entire dataset.

5.6. Bidirectional Encoder Representations from Transformers (BERT)

The BERT with LSTM hybrid deep learning classifier was implemented in Python and trained and tested on the provided dataset. The testing accuracy was 99.61%, precision was 99.87%, recall was 99.23%, and F1-score was 99.55% after 25 epochs (Figure 8). The figure shows the accuracy and loss plots for the BERT with long short-term memory model during training and validation. The training accuracy usually starts low and increases with each epoch, potentially reaching 97–100% by the 25th epoch as the model fits the training data better. The validation accuracy also starts low and increases but at a slower rate than training accuracy. The training loss decreases with each epoch as the model minimizes errors on the training set. It may drop sharply in the early epochs and plateau as it converges. The validation loss follows a similar downward trend as training loss but at a higher value, reflecting the gap between training and validation performance. Table 5 shows the classification report obtained when evaluating the performance of the BERT with LSTM hybrid model on the test dataset.

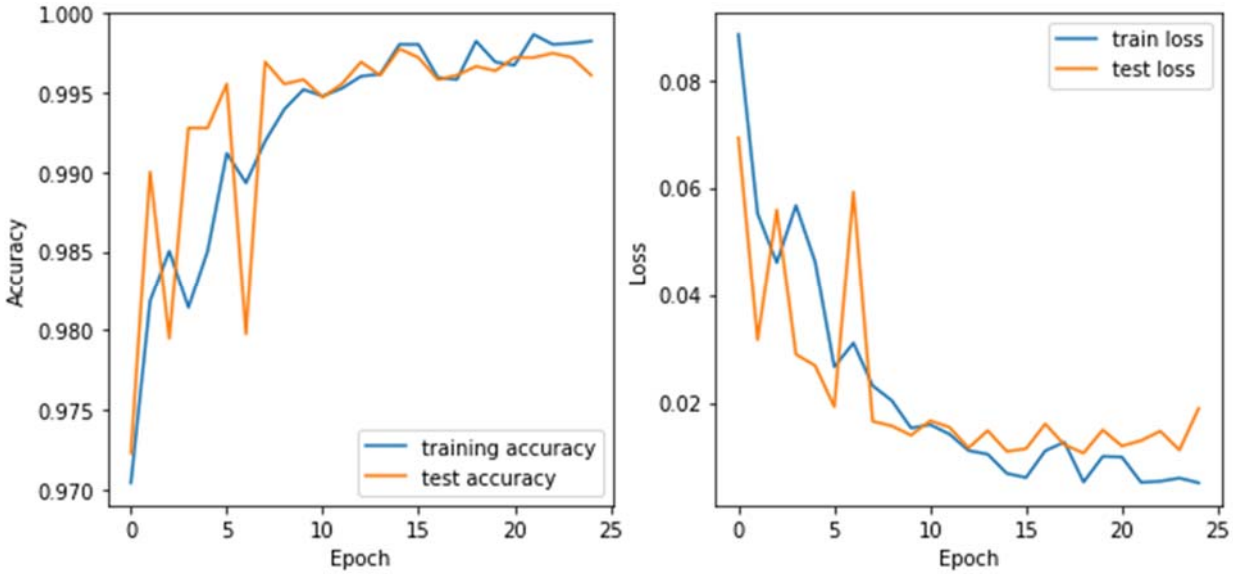

Figure 8. BERT with LSTM model training and testing accuracy and loss over the 25 epochs of training.

Table 5. BERT with LSTM classification report.

	Precision	Recall	F1-Score	Support
0	0.99	1.00	1.00	2058
1	1.00	0.99	1.00	1549
accuracy	–	–	1.00	3607
macro avg	1.00	1.00	1.00	3607
weighted avg	1.00	1.00	1.00	3607

The BERT model achieved an exceptional accuracy of 99.61% on the test set, significantly outperforming the CNN, RNN, and LSTM models. This demonstrates BERT's ability to correctly classify almost all phishing and legitimate emails. The precision of 99.87% indicates that BERT had negligible false positives, rarely mislabeling legitimate emails. The recall of 99.23% means the model successfully identified most phishing emails, with very few false negatives. Finally, the F1-score of 99.55% reflects the outstanding balance between precision and recall.

The BERT and LSTM classifier was evaluated using a classification report, which measures the precision, recall, and F1-score of the model's predictions. The report shows that the model has high precision, recall, and F1-score for both classes (0 and 1), indicating that it performs well in classifying email phishing attacks. The model achieved an accuracy of 100% on the test set, with a macro average of 1.0, indicating that the model performs well for both classes. The weighted average also shows an F1-score of 1.0, which suggests that the model is highly accurate and reliable in detecting email phishing attacks.

5.7. Comparative Analysis of Deep Learning Models

Table 6 illustrates the combined result of the performance of all the four models.

Table 6. Performance metrics comparison.

Metric	CNN	RNN	LSTM	BERT-LSTM
Accuracy	98.74%	98.58%	98.89%	99.61%
Precision	98.96%	98.57%	98.87%	99.87%
Recall	98.78%	98.53%	98.56%	99.23%
F1-score	98.87%	98.55%	98.87%	99.55%

5.8. Comparative Analysis with Related Works

5.8.1. Convolutional Neural Networks (CNNs)

Table 7 illustrates a comparison between the research results and Ref. [20] using convolutional neural networks (CNNs). Ref. [20] was used in the comparison because it proposes a deep learning model based on CNN for detecting phishing emails.

Table 7. CNN comparison.

Metrics	Ref. [20]	Proposed Model
Accuracy	98%	98.74%
Precision	98.50%	98.96%
Recall	98%	98.78%
F1-score	98%	98.87%

Our proposed model achieved an accuracy of 98.74%, which indicates that the model correctly classified 98.74% of emails as either phishing or legitimate. It also achieved a

precision of 98.96%, which indicates that when the model classified an email as phishing, it was correct 98.96% of the time. The recall of the proposed model was 98.78%, which indicates that the model correctly identified 98.78% of the phishing emails. The F1-score was 98.87%, which indicates that the proposed model has a high level of accuracy in identifying phishing emails. In comparison to the performance metrics reported by Singh et al. (2020) [20], the proposed model achieved a higher accuracy (98.74% vs. 98%), precision (98.96% vs. 98.50%), recall (98.78% vs. 98%), and F1-score (98.87% vs. 98%). These results indicate that the proposed model is a promising technique for detecting phishing emails with high accuracy.

5.8.2. Recurrent Neural Networks (RNNs)

Table 8 compares the proposed model's research results with Fang et al. (2019) [16] and Tang and Mahmoud (2022) [30] using RNNs for detecting phishing emails.

Table 8. RNN comparison.

Metrics	Ref. [16]	Ref. [30]	Proposed Model
Accuracy	99.84%	74.12%	98.58%
Precision	99.66%	–	98.57%
Recall	99.00%	–	98.53%
F1-score	99.33%	70.89%	98.55%

In comparison, the proposed model achieved a higher accuracy (98.58%) than Ref. [30], 74.12%, and a comparable accuracy to Ref. [16], 99.84%. In addition, the proposed model achieved a precision of 98.57%, which is comparable to Ref. [16], 99.66%, and a higher F1-score (98.55%) than Ref. [30] (70.89%).

The proposed model achieved a precision of 98.57%, which indicates that when the model classified an email as phishing, it was correct 98.57% of the time. Recall measures the proportion of true positives (emails classified as phishing) out of all actual positive cases (emails that were phishing). The proposed model achieved a recall of 98.53%, which indicates that the model correctly identified 98.53% of the phishing emails. In addition, the proposed model achieved an F1-score of 98.55%, which indicates that it has a high level of accuracy in identifying phishing emails.

5.8.3. Long Short-Term Memory (LSTM)

Table 9 compares the proposed model's research results and Ref. [28] using long short-term memory (LSTM).

Table 9. LSTM comparison.

Metrics	Ref. [28]	Proposed Model
Accuracy	98.35%	98.89%
Precision	98.58%	98.87%
Recall	97.89%	98.56%
F1-score	98.24%	98.87%

The proposed model achieved higher accuracy, precision, recall, and F1-score. In this case, the proposed model achieved an accuracy of 98.89%, which indicates that the model correctly classified 98.89% of the emails as legitimate or phishing. In addition, the proposed model achieved a precision of 98.87%, indicating that when the model classified an email as phishing, it was correct 98.87% of the time. Moreover, the proposed model achieved a recall of 98.56%, indicating that the model correctly identified 98.56% of the phishing emails. For

the F1-score, the proposed model achieved a value of 98.87%, which indicates that it has a high level of accuracy in identifying phishing emails.

5.8.4. Hybrid Deep Learning Models Using BERT

Table 10 compares the research results of the proposed model and Ref. [19] using a pre-trained BERT model for tokenization and training.

Table 10. BERT comparison.

Metrics	Ref. [19]	Proposed Model
Accuracy	-	99.61%
Precision	-	99.87%
Recall	-	99.23%
F1-score	99.18%	99.55%

The results indicate that the proposed model achieved higher accuracy (99.61%) and precision (99.87%). In addition, the recall of the proposed model was 99.23%, indicating that the model correctly identified 99.23% of the phishing emails. For the F1-score, the proposed model shows a value of 99.55%, which indicates that the proposed model has a high level of accuracy in identifying phishing emails.

5.9. Summary

The proposed research provides valuable new insights compared to the prior state of the art in applying deep learning for phishing detection. The proposed model achieved significantly higher accuracy, up to a 5% improvement in recall over models like Singh et al.'s CNN approach [20]. This demonstrates the power of a larger dataset for more robust training. Moreover, the proposed LSTM model overcame limitations like the vanishing gradients that constrained previous RNN models to 74% accuracy, allowing LSTM to capture longer context and reach 98.89% accuracy. In addition, cases were analyzed where the model performed differently across metrics, finding BERT's advantage in precision while LSTM balanced recall and precision best. Through direct comparison of these techniques, rather than isolated evaluations, key insights were gained about their relative strengths and weaknesses. For instance, RNNs proved less effective than CNNs and LSTMs for feature extraction from raw emails. This pivotal analysis will aid future research in adapting deep learning to enhance phishing detection.

6. Conclusions

The number of phishing emails is on the rise. These fraudulent emails are designed to deceive unsuspecting users into taking actions that can allow attackers to victimize them and extract sensitive information. Such attacks can compromise the cybersecurity systems of organizations and provide attackers access to their sensitive data. Phishing emails are a significant factor in cyberattacks and must be taken seriously. This study aimed to evaluate deep learning models using publicly available phishing and legitimate datasets. The emails in raw format were extracted and saved as CSV files. Python code was utilized to clean and process the emails. The resulting clean data were converted into tokens to determine the most effective features. For the tokenization process, two methods were used: the first one was the tokenizer class from Keras library, and the second one was using the pre-trained BERT model tokenizer. The tokenized features were split into 70% for training and 30% for testing. The CNNs, recurrent RNNs, LSTM, and BERT were trained on the training dataset and then tested using the testing dataset. The proposed model was then evaluated using different metrics such as accuracy metrics, precision, recall, and F1-score. The results found that the model obtained high and accurate results, where the best performance was when using BERT with LSTM with an accuracy of 99.61%. The phishing email detection

model developed in this study using deep learning techniques has the potential to be highly effective in detecting phishing emails.

The identified feature sets were extracted from a large dataset containing several thousand emails, which can be valuable for detecting new phishing emails. However, it should be noted that a single security system alone cannot provide complete security. Therefore, it is recommended to use phishing email detection in conjunction with other security systems such as firewalls, intrusion prevention systems, and antivirus software to enhance overall security. In addition, future research in deep learning and transformers is essential to improve the real-time detection of phishing emails. While effective on its own, the proposed phishing detection model can provide additional security when integrated with existing infrastructure like firewalls, gateways, and endpoint antivirus. The deep learning model can be deployed on email gateways to analyze incoming messages and flag potential phishing threats. Combining it with firewall rules that filter on sender reputation and domain blacklists can reduce the number of malicious emails entering the network. At the endpoint, the antivirus can scan attachments and links in messages classified as phishing by the proposed model. Furthermore, the email filtering provided by the proposed model can improve the signal-to-noise ratio for security operations centers, allowing analysts to focus on fewer high-quality alerts. Integrating the deep learning-based phishing detection model with existing security systems can enhance the protection across the email attack surface. More advanced and enhanced datasets are required to improve the feature selection process, and research is needed in dynamic feature selection using deep learning models. Additionally, adaptive phishing email filtering needs to be studied so that the system can automatically learn, adapt, and identify phishing emails based on their behaviors. Transformers, which are a type of deep learning model, have shown significant potential in natural language processing tasks such as text classification, machine translation, and text generation. Given the nature of phishing emails, which rely on language-based deception to trick recipients into taking unwanted actions, transformers are a promising avenue for improving the accuracy of phishing email detection. One possible direction for future research in this area is the development of transformer-based models for phishing email detection.

Author Contributions: Conceptualization, H.A. and S.A.; methodology, H.A. and S.A.; software, H.A.; validation, H.A. and S.A.; formal analysis, H.A.; investigation, H.A.; resources, H.A.; data curation, H.A.; writing—original draft preparation, H.A.; writing—review and editing, H.A. and S.A.; visualization, H.A.; supervision, S.A.; project administration, S.A.; funding acquisition, S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding, and The APC was funded by the Deanship of Scientific Research at Saudi Electronic University.

Data Availability Statement: Publicly available datasets were analyzed in this study. The datasets can be found here: https://www.kaggle.com/datasets/rtatman/fraudulent-email-corpus, https://www.kaggle.com/datasets/maharshipandya/email-spam-dataset-extended, https://www.kaggle.com/datasets/charlottehall/phishing-email-data-by-type, and https://www.kaggle.com/datasets/yashpaloswal/spamham-email-classification-nlp, (accessed on 5 September 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jang-Jaccard, J.; Nepal, S. A survey of emerging threats in cybersecurity. *J. Comput. Syst. Sci.* **2014**, *80*, 973–993. [CrossRef]
2. Adewole, K.S.; Akintola, A.G.; Salihu, S.A.; Faruk, N.; Jimoh, R.G. Hybrid rule-based model for phishing URLs detection. In *Proceedings of the Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*; Springer International Publishing: Basel, Switzerland, 2019; pp. 119–135.
3. Elijah, A.V.; Abdullah, A.; JhanJhi, N.Z.; Supramaniam, M.; Abdullateef, B. Ensemble and deep-learning methods for two-class and multi-attack anomaly intrusion detection: An empirical study. *Int. J. Adv. Comput. Sci. Appl. IJACSA* **2019**, *10*, 520–528. [CrossRef]
4. Alsariera, Y.A.; Elijah, A.V.; Balogun, A.O. Phishing website detection: Forest by penalizing attributes algorithm and its enhanced variations. *Arab. J. Sci. Eng.* **2020**, *45*, 10459–10470. [CrossRef]

5. Vrbančič, G.; Fister, I., Jr.; Podgorelec, V. Swarm intelligence approaches for parameter setting of deep learning neural network: Case study on phishing websites classification. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics—WIMS'18, Novi Sad, Serbia, 25–27 June 2018.
6. Zamir, A.; Khan, H.U.; Iqbal, T.; Yousaf, N.; Aslam, F.; Anjum, A.; Hamdani, M. Phishing website detection using diverse machine learning algorithms. *Electron. Libr.* **2020**, *38*, 65–80. [CrossRef]
7. Anti-Phishing Working Group (APWG). Trends Report. Available online: https://apwg.org (accessed on 12 December 2022).
8. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
9. Castillo, E.; Dhaduvai, S.; Liu, P.; Thakur, K.S.; Dalton, A.; Strzalkowski, T. Email threat detection using distinct neural network approaches. In Proceedings of the First International Workshop on Social Threats in Online Conversations: Understanding and Management, Marseille, France, 11–16 May 2020; pp. 48–55.
10. Do, N.Q.; Selamat, A.; Krejcar, O.; Herrera-Viedma, E.; Fujita, H. Deep learning for phishing detection: Taxonomy, current challenges and future directions. *IEEE Access: Pract. Innov. Open Solut.* **2022**, *10*, 36429–36463. [CrossRef]
11. Salloum, S.; Gaber, T.; Vadera, S.; Shaalan, K. A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access Pract. Innov. Open Solut.* **2022**, *10*, 65703–65727. [CrossRef]
12. Chaibi, N.; Atmani, B.; Mokaddem, M. Deep learning approaches to intrusion detection: A new performance of ANN and RNN on NSL-KDD. In Proceedings of the 1st International Conference on Intelligent Systems and Pattern Recognition, Hammet, Tunisia, 16–18 October 2020.
13. Mahdavisharif, M.; Jamali, S.; Fotohi, R. Big data-aware intrusion detection system in communication networks: A deep learning approach. *J. Grid Comput.* **2021**, *19*, 2–28. [CrossRef]
14. Valecha, R.; Mandaokar, P.; Rao, H.R. Phishing Email Detection using Persuasion Cues. *IEEE Trans. Depend. Secure Comput.* **2021**, *19*, 747–756. [CrossRef]
15. Barushka, A.; Hajek, P. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. *Appl. Intell.* **2018**, *48*, 3538–3556. [CrossRef]
16. Fang, Y.; Zhang, C.; Huang, C.; Liu, L.; Yang, Y. Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism). *IEEE Access Pract. Innov. Open Solut.* **2019**, *7*, 56329–56340. [CrossRef]
17. Alhogail, A.; Alsabih, A. Applying machine learning and natural language processing to detect phishing email. *Comput. Secur.* **2021**, *110*, 102414. [CrossRef]
18. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. *arXiv* **2018**, arXiv:1809.05679. [CrossRef]
19. Kaddoura, S.; Alfandi, O.; Dahmani, N. A spam email detection mechanism for English language text emails using deep learning approach. In Proceedings of the 2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Bayonne, France, 10–13 September 2020.
20. Singh, S.; Singh, M.P.; Pandey, R. Phishing detection from URLs using deep learning approach. In Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS), Patna, India, 14–16 October 2020.
21. Saha, I.; Sarma, D.; Chakma, R.J.; Alam, M.N.; Sultana, A.; Hossain, S. Phishing attacks detection using deep learning approach. In Proceedings of the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 20–22 August 2020.
22. McGinley, C.; Monroy SA, S. Convolutional neural network optimization for phishing email classification. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; IEEE: New York, NY, USA, 2021; pp. 5609–5613.
23. Fetooh HT, M.; El-Gayar, M.M.; Aboelfetouh, A. Detection technique and mitigation against a phishing attack. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 177–188. [CrossRef]
24. Gogoi, B.; Ahmed, T. Phishing and Fraudulent Email Detection through Transfer Learning using pretrained transformer models. In Proceedings of the 2022 IEEE 19th India Council International Conference (INDICON), Kochi, India, 24–26 November 2022; IEEE: New York, NY, USA, 2022.
25. Doshi, J.; Parmar, K.; Sanghavi, R.; Shekokar, N. A comprehensive dual-layer architecture for phishing and spam email detection. *Comput. Secur.* **2023**, *133*, 103378. [CrossRef]
26. Benavides-Astudillo, E.; Fuertes, W.; Sanchez-Gordon, S.; Nuñez-Agurto, D.; Rodríguez-Galán, G. A Phishing-Attack-Detection Model Using Natural Language Processing and Deep Learning. *Appl. Sci.* **2023**, *13*, 5275. [CrossRef]
27. Aldakheel, E.A.; Zakariah, M.; Gashgari, G.A.; Almarshad, F.A.; Alzahrani, A.I. A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators. *Sensors* **2023**, *23*, 4403. [CrossRef]
28. He, D.; Lv, X.; Xu, X.; Yu, S.; Li, D.; Chan, S.; Guizani, M. An effective double-layer detection system against social engineering attacks. *IEEE Netw.* **2022**, *36*, 92–98. [CrossRef]
29. Bagui, S.; Nandi, D.; Bagui, S.; White, R.J. Classifying phishing email using machine learning and deep learning. In Proceedings of the 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, 3–4 June 2019.
30. Tang, L.; Mahmoud, Q.H. A deep learning-based framework for phishing website detection. *IEEE Access Pract. Innov. Open Solut.* **2022**, *10*, 1509–1521. [CrossRef]
31. Murthy MY, B.; Mastanbi, S.; Sujitha, B.; Babu, K.R. Evaluating deep learning algorithms for natural language processing. In *Algorithms for Intelligent Systems*; Springer Nature Singapore: Singapore, 2023; pp. 709–720.

32. Koushik, J. Understanding Convolutional Neural Networks. *arXiv* **2016**, arXiv:1605.09081.
33. Salehinejad, H.; Sankar, S.; Barfett, J.; Colak, E.; Valaee, S. Recent advances in recurrent neural networks. *arXiv* **2018**, arXiv:1801.01078.
34. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef] [PubMed]
35. Ozdil, U.; Arslan, B.; Tasar, D.E.; Polat, G.; Ozan, S. Ad text classification with bidirectional encoder representations. In Proceedings of the 2021 6th International Conference on Computer Science and Engineering (UBMK), Ankara, Turkey, 15–17 September 2021.
36. Goularas, D.; Kamis, S. Evaluation of deep learning techniques in sentiment analysis from twitter data. In Proceedings of the 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), Istanbul, Turkey, 26–28 August 2019.