electronics

MDPI

Systematic Review

A Systematic Review of Deep Learning Techniques for Phishing Email Detection

Phyo Htet Kyaw *, Jairo Gutierrez * and Akbar Ghobakhlou

Department of Computer Science and Software Engineering, Auckland University of Technology, Auckland 1010, New Zealand; akbar.ghobakhlou@aut.ac.nz

* Correspondence: phyohtet.kyaw@autuni.ac.nz (P.H.K.); jairo.gutierrez@aut.ac.nz (J.G.)

Abstract: The landscape of phishing email threats is continually evolving nowadays, making it challenging to combat effectively with traditional methods even with carrier-grade spam filters. Traditional detection mechanisms such as blacklisting, whitelisting, signature-based, and rule-based techniques could not effectively prevent phishing, spear-phishing, and zero-day attacks, as cybercriminals are using sophisticated techniques and trusted email service providers. Consequently, many researchers have recently concentrated on leveraging machine learning (ML) and deep learning (DL) approaches to enhance phishing email detection capabilities with better accuracy. To gain insights into the development of deep learning algorithms in the current research on phishing prevention, this study conducts a systematic literature review (SLR) following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines. By synthesizing the 33 selected papers using the SLR approach, this study presents a taxonomy of DL-based phishing detection methods, analyzing their effectiveness, limitations, and future research directions to address current challenges. The study reveals that the adaptability of detection models to new behaviors of phishing emails is the major improvement area. This study aims to add details about deep learning used for security to the body of knowledge, and it discusses future research in phishing detection systems.

Keywords: deep learning; phishing detection; phishing email; machine learning; PRISMA

Citation: Kyaw, P.H.; Gutierrez, J.; Ghobakhlou, A. A Systematic Review of Deep Learning Techniques for Phishing Email Detection. *Electronics* **2024**, *13*, 3823. https://doi.org/10.3390/electronics13193823

Academic Editor: Wajeb Gharibi

Received: 28 August 2024
Revised: 25 September 2024
Accepted: 26 September 2024
Published: 27 September 2024

1. Introduction

Social engineering tactics exploit unsuspecting individuals by tricking them into thinking they are interacting with a trustworthy, legitimate entity, often with misleading communication messages. Email phishing is a type of social engineering tactic intended to trick individuals or organizations into stealing electronic identities, credentials, and data as well as distributing malicious software. In 2023, almost five million phishing attacks were recorded by the APWG [1]. The rise of phishing incidents caused substantial financial and reputational damage to individuals and organizations. From October 2013 to December 2022, business email compromise (BEC) caused USD 51 billion in losses, and phishing was the most committed cybercrime among the top five cybercrime types [2]. According to Check Point Research [3] and Verizon [4] reports, 86% of all malware was delivered via email in 2022, and among them, 35% was ransomware. Although blacklists, whitelists, and signature-based approaches are available for phishing detection, their shortcomings have led to the creation of more sophisticated solutions because attackers are now using specially crafted lure messages that seem legitimate, malicious code-embedded documents, URLs, and trusted email service providers to bypass traditional spam filtering systems. Moreover, cybercriminals are now using machine learning (ML) techniques to generate more sophisticated email messages to escape from legacy detection systems [5].

In recent years, many researchers paid attention to deep learning (DL) and ML-based phishing detection methods to address the shortcomings of traditional methods and to advance phishing detection accuracy. To recognize patterns and detect data anomalies,

deep learning models were developed to enhance the effectiveness of phishing [6]. Deep learning, a subset of ML, includes models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs). These models are highly flexible in design and can effectively learn data patterns, features, and email structures, enabling them to achieve promising results in the identification of phishing emails [7]. In conventional ML techniques, feature extraction and selection processes need to be performed manually with human experts, whereas deep learning methods can combine these two processes and classify phishing attacks effectively [8].

To build good deep neural network classifiers, the availability of large datasets is very important [9]. The limited availability of big datasets is the major challenge in phishing detection and classification research [10]. The main reason could be privacy concerns because many private organizations are reluctant to share their data. It is crucial to choose a good quality dataset while training DL models as irrelevant features, and poor-quality data input can impact models' classification efficiency [11]. Various researchers proposed DL-based phishing detection methods which were trained by different public and private datasets. However, phishing continues to pose a significant risk to both the cyber community and various institutions. Hence, a systematic literature review is needed to understand how deep learning algorithms were developed, what kinds of datasets were used to train the model, what the gaps are, and what the future direction is.

Many researchers have conducted extensive reviews on various types of phishing attacks and machine learning-based techniques for phishing website detection in recent years [12–17]. Aassal et al. [11] presented a taxonomy of features and compared the features used for training in previous works. Al-Yozbaky and Alanezi [18] reviewed content-based phishing detection techniques utilizing NLP and machine learning. Salloum et al. [19] conducted a systematic literature review on NLP and machine learning-based phishing email detection, following PRISMA guidelines. Quang et al. [20] applied the PRISMA guidelines to review phishing detection systems using machine learning, categorizing the literature by research area, research type, and research contribution. However, their work did not specifically focus on the use of deep learning techniques for phishing email detection. To the best of our knowledge, there are a few articles that have systematically reviewed DL-based phishing email detection using the PRISMA guidelines. This study aims to pinpoint gaps and potential enhancements in the current research on phishing email prevention, with a particular focus on DL approaches, through a comprehensive review of relevant articles and meta-analyses. The contribution of this study is

1. A systematic literature review using the PRISMA approach with transparency and no bias.
2. Conduct an in-depth qualitative analysis of 33 selected papers to categorize and present various deep learning approaches for detecting phishing emails.
3. Discover the strengths and limitations of previous research and suggest potential areas for future investigation.

The rest of the paper is structured as follows: Section 2 outlines the proposed SLR methodology, detailing the selection process for the 33 papers reviewed. Section 3 presents the taxonomy of DL-based phishing email detection systems and a literature review of the findings. Section 4 explores the limitations of current research and possible improvement directions. Section 5 will encompass the conclusion and outline potential avenues for future research.

2. Methodology

Systematic reviews strive to provide an unbiased assessment of a research topic with reliable, thorough, and transparent methodologies. This study reviewed existing literature on DL-based phishing email detection techniques by adopting SLR [21] and PRISMA [22] guidelines. Figure 1 illustrates the adopted PRISMA flow diagram for selecting relevant studies. The review protocol, objectives, research questions, search strategies, selection criteria, and syntheses of results will be presented in the following sections.

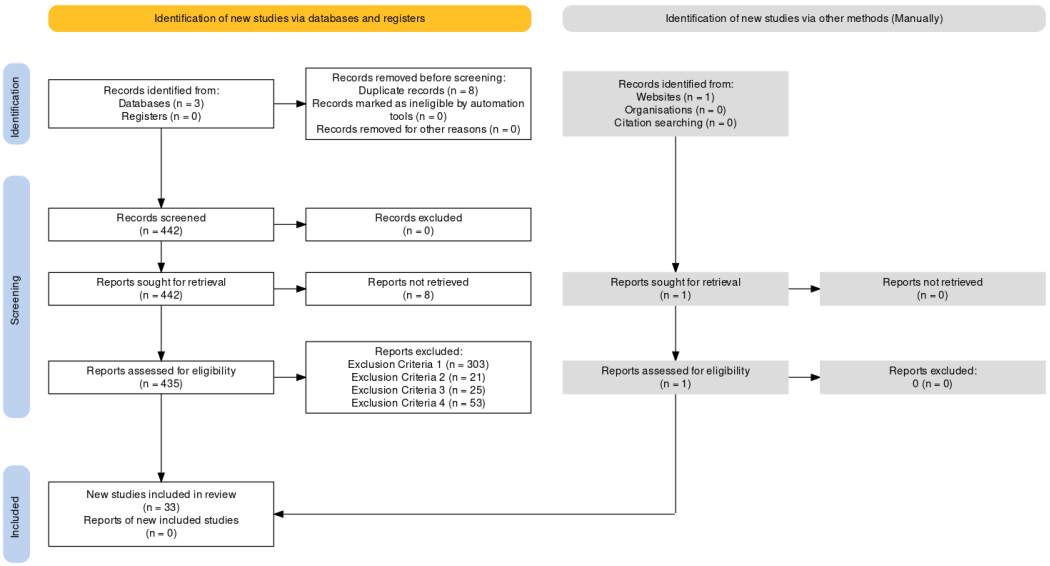


Figure 1. Selection procedure and results depicted in the PRISMA flow diagram.

2.1. Objectives

- To assess the empirical evidence regarding the efficacy of DL algorithms in detecting phishing emails.
- To summarize the advantages and drawbacks of current implementations of DL algorithms in detecting phishing emails.
- To identify the discrepancies and potential improvements in current phishing email detection research.

2.2. Research Questions

In accordance with objectives, the subsequent research questions were devised for this SLR.

1. How deep learning algorithms are applied and what are the most effective techniques to detect and defend against sophisticated social engineering attacks via email, such as phishing and spear phishing?
2. How does the integration of deep learning algorithms in cybersecurity applications influence the precision and effectiveness of detecting phishing email threats in contrast to conventional ML approaches?
3. What kind of datasets, optimization, and evaluation methods/matrix are used to train and measure the outcome?

2.3. Search Strategy

In this study, the search keywords were defined and combined with Boolean operators to search relevant articles in IEEE Xplore, ScienceDirect (Elsevier), and ACM digital library databases for broader coverage without biases. These databases were chosen because they are widely used in technology, engineering, and computer science. Initially, the Scopus database was included in the search. However, 128 duplicate records were identified when using the same search terms. Consequently, Scopus was excluded to avoid redundancy. The search term is (“All Metadata”:“deep learning”) AND (“All Metadata”:“phishing email”) OR (“All Metadata”:“email phishing”) OR (“All Metadata”:“email spoofing”) OR (“All Metadata”:“scam email”).

2.4. Criteria for Study Selection

The inclusion criteria were defined to focus on the research questions and to gather the most relevant articles published from 2019 to 2024, specifically targeting DL techniques for phishing email detection. In contrast, the exclusion criteria were set to eliminate irrelevant

papers that used rule-based, signature-based, traditional ML models, and the survey papers. Additionally, duplicated publications and non-research papers were excluded to save time and effort of the screening process.

2.4.1. Inclusion Criteria

1. Published within 5 years. However, some literature which influenced the development of DL for phishing detection will be included in the study.
2. Published using the English language.
3. Relevance to topic and/or answer the research questions.

2.4.2. Exclusion Criteria

1. The studies which are not related to the research questions.
2. Traditional machine learning techniques or signature-based solutions for phishing email detection.
3. The survey, or review, or meta-analysis papers.
4. Not a research article, books, chapters, editorials, summaries of workshops, duplicated publication on the same study. For duplicated publications on the same topic, the latest publication will be selected.

2.5. Quality Instrument

After collecting the papers according to the inclusion and exclusion criteria, a weighting or scoring technique practiced by Kitchenham et al. [21] was applied to each quality assessment question to select the most relevant articles for this SLR study. The scores were given "1" if the paper answered "Yes", "0.5" if the paper answered "Partly", and "0" for "No or Unknown" for each quality assessment question. The papers that scored more than or equal to 50% were selected. The four quality assessment questions (QA) are

QA1. Did the authors mention what kind of dataset was used in the experiment?

QA2. Did the authors employ any cross-validation technique or evaluation metrics to evaluate the efficacy of the proposed solution?

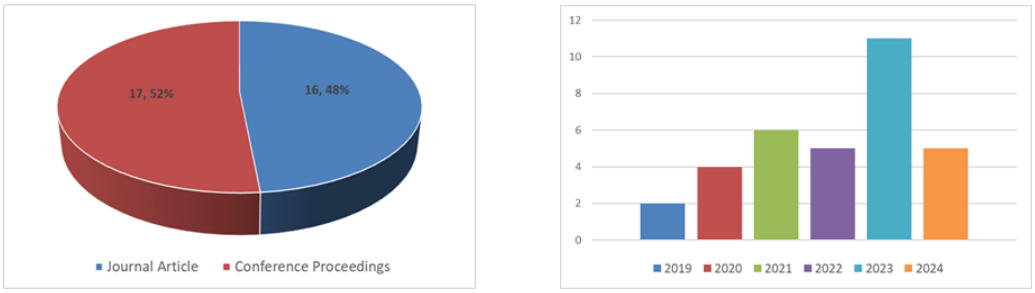
QA3. Did the outcomes undergo comparison with prior studies or other models?

QA4. Is the study/work peer-reviewed?

2.6. Paper Selection and Syntheses

This study applied PRISMA guidelines for the paper selection process that consists of various stages including automatic search, duplicate exclusion, screening of titles and abstracts, selection of full text, etc. There is a total of 443 records (IEEE: 58, ACM: 115, ScienceDirect: 269, manual 1) found after searching in three selected databases. All the search terms, results, authors, source, title, published year, number of citations, journal impact factor, QA scores, and summaries were recorded in Microsoft Excel spreadsheets. Among those records, 8 duplicates were found, 7 in IEEE, and 1 in ACM, respectively. A total of 53 articles, 1 from IEEE, 26 from ACM, and 26 from ScienceDirect, were excluded using exclusion criteria 4 (summaries of workshop, editorials, book chapters, and duplicated publications on the same study by the same authors) before the screening. After that, 384 publications were subsequently analyzed by reviewing their abstracts, introductions, and conclusions, considering the inclusion criteria. The details of search results and selection can be seen in Figure 1. After selecting the papers, the subsequent stage involves evaluating the quality of the chosen papers by scoring with 4 QA questions. Most of the selected 33 papers received a score of 4, except for 3 papers which received 2.5, 3, and 3.5. Hence, all 33 papers were selected for review.

The selected articles were categorized as journal and conference proceedings. Among them, 16 papers are journal articles, and 17 papers are conference proceedings. The distribution of journal and conference proceedings, and years of selected papers are illustrated in Figure 2. The data showed that 11 papers in 2023, 6 papers in 2021, 5 papers in 2022 and 2024, 4 papers in 2020, and 2 papers in 2019 were published.



(**a**) (**b**)

Figure 2. Distribution of publications over the years. (**a**) Proportion of articles from journals and conferences. (**b**) Distribution of publications over the years.

3. Literature Review and Discussion

The selected phishing email detection methods were examined and categorized based on their classification of email segments. Most of the researchers extracted and selected the features from the email body text, 18 out of 33, while other researchers used header, subject, body, URL, attachment, structure, and attachment files features to train the DL model. The taxonomy of their techniques can be seen in Table 1. Regarding the evaluation methods, researchers used different techniques to assess their proposed models' effectiveness such as accuracy, precision, recall, F1 score, false positive rate (FPR), receiver operating characteristic (ROC) curve, and area under the curve (AUC). Hence, it is not possible to compare the effectiveness of each research work, since they used different datasets and different evaluation methods. However, this study will present their accuracy, as that is the most common measure among all researchers, along with the F1 score to assess the balanced performance of the model. Table 2 presents the results, main contributions and limitations of phishing detection systems in the selected articles.

3.1. Classification by Email Body

Chataut et al. [23] proposed phishing email detection with LLM models by analyzing email content. They compare the performance between different LLM models, including a customized version of ChatGPT by using a dataset from Kaggle. An accuracy of 97.46% was achieved by custom ChatGPT, showing that LLM models can be utilized to enhance phishing detection. However, they did not present in detail how the GPT model was customized. Bagui et al. [24] experimented with phishing email detection using different ML algorithms (NB, SVM, and DT) and DL models (LSTM, and CNN) by feeding email body text with and without phrasing. They used a private dataset that contained 18,366 emails and tokenized the body text with n-gram features without removing stop-words. The sigmoid activation function was used for LSTM and word embedding DL, while Relu activation was used for CNN in their experiments, and the word embedding DL model outperformed other models with 98.89% accuracy. DL models performed better than ML models with word phrasing.

Giri et al. [25] conducted a comparison between the integration of two word embedding methods (GloVe and BERT) and deep learning models (CNN and FCN). They used a combination of five public datasets, Ling-Spam, Enron-Spam, Enron, SpamAssassin, and Nazario, in their work. Their model employed the sigmoid function as a classifier, and the GloVe-CNN model performed better with 98% accuracy. However the length of input data was different: GolVe-CNN was trained with a maximum length of 19,000 words, and BERT-FCN was trained with 512 words due to the limitation of the pre-trained BERT model. Zannat et al. [26] also compared DL models (BERT, CNN, and Bi-LSTM) with traditional ML models (NB, KNN, DT, SVM, RF, and AdaBoost) for phishing email detection by training with a private dataset collected from different social media research groups for Bangla emails. They used one-hot encoding for feature extraction, GloVe for word embedding,

CountVectorizer for vectorization, and Adam optimizer for hyperparameter tuning. The Bi-LSTM model achieved the best result with 97% accuracy.

Table 1. Types of features used and their DL models.

Type of Features	Authors	DL Models
Email body	R. Chataut et al. [23]	LLM
	Bagui et al. [24]	LSTM, CNN, and word embedding DL
	Giri et al. [25]	GloVe+CNN, and BERT+FCN
	McGinley et al. [27]	CNN
	Zannat et al. [26]	CNN, Bangla-BERT, Bi-LSTM
	Ramprasath et al. [28]	RNN with LSTM cells
	Valecha et al. [29]	Bi-LSTM
	Paradkar [30]	LSTM, Bi-SLTM, CNN
	Paliath et al. [31]	NN
	Divakarla et al. [32]	LSTM, Bi-LSTM, CNN+RNN
	Gholampour et al. [33]	ALBERT, ROBERTA, BERT, DEBERTA, DEBERT, SQ, and YOSO
	Bountakas et al. [34]	BERT
	Qachfar et al. [35]	BERT
	Sachan et al. [36]	CNN+BiLSTM+GRU
	Alhogail et al. [37]	GCN
	Nicholas et al. [38]	CNN
	AbdulNabi et al. [39]	BERT
	Hina et al. [40]	LSTM-GRU
Header, subject, and email body	D. He et al. [41]	LSTM, Bi-SLTM
	Aassal et al. [11]	Multiple models from AutoSklearn and TPOT
	Alotaibi et al. [42]	CNN
	Fang et al. [43]	RCNN (with Bi-LSTM)
	Kaddoura et al. [44]	FFNN (MLP), BERT
	Salloum et al. [45]	MLP
	Jáñez-Martino et al. [46]	BERT
	Doshi et al. [47]	ANN, CNN, RNN
	Krishnamoorthy et al. [48]	DNN+BiLSTM
	Borra et al. [49]	DLCNN
Header, subject, email body, and URL	T. Saka et al. [50]	BERT
	Magdy et al. [51]	ANN
	Bountakas et al. [52]	DT+KNN+MLP
Header, subject, email body, URL, and attachment	T. Muralidharan et al. [10]	BERT, CNN
Email structure, body, and URL	J. Lee et al. [7]	CNN-LSTM, BERT

Table 2. Results, main contribution, and weakness/limitation.

Authors	DL Models	Results	Main Contribution	Weakness/Limitation
Chataut et al. [23]	LLM	Accuracy = 97.46%, F1 = 0.9668	Demonstrate potential of LLMs for phishing identification.	Cannot detect malicious link or attachment inside email body. Not a real-time processing. Require high computational resources. Dataset is small and model may overfit.
Bagui et al. [24]	LSTM, CNN, and Word Embedding DL	Word embedding DL Accuracy = **98.89%**, F1 = NA DT, NB, SVM, CNN, LSTM Accuracy = < 97.50%, F1 = NA	Showed the context of the email is important in detecting phishing email.	Cannot detect malicious link or attachment inside email body.
Giri et al. [25]	GloVe+CNN, and BERT+FCN	GloVe+CNN Accuracy = **98%**, F1 = **0.9749** BERT+FCN Accuracy = 96%, F1 = 0.9576	Compare the combination of word embedding techniques and DL architectures.	Cannot detect malicious link or attachment inside email body. BERT model has limitation of maximum 512 tokens (words length).
McGinley et al. [27]	CNN	Accuracy = 98.139%, F1 = 0.9819	Ablation study for best performing setting in CNN architecture for phishing email text classification.	Cannot detect malicious link or attachment inside email body. Dataset is small and model may overfitted.
Zannat et al. [26]	CNN, Bangla-BERT, Bi-LSTM	Bi-LSTM Accuracy = **97%**, F1 = **0.8889** CNN Accuracy = 96.8%, F1 = 0.8745 Bangla-BERT Accuracy = 96.4%, F1 = 0.8635 NB, KNN, DT, SVM, AdaBoost, RF Accuracy = < 93.6%, F1 = NA	New labeled dataset for Bangla email.	Cannot detect malicious link or attachment inside email body. No dropout layer to prevent overfitting.
Ramprasath et al. [28]	RNN with LSTM cells	RNN Accuracy = **99.1%**, F1 = **0.958** SVM Accuracy = 98.2%, F1 = 0.932 CkNN Accuracy = 98.1%, F1 = 0.928	NA	Cannot detect malicious link or attachment inside email body. No dropout layer to prevent overfitting.
Valecha et al. [29]	Bi-LSTM	Accuracy = 95.97%, F1 = 0.9569	Phishing detection based on gain and loss persuasion cues of text context.	Cannot detect malicious link or attachment inside email body. Manual coding of persuasion cues labels and manual hyperparameter tuning.
Paradkar [30]	LSTM, Bi-SLTM, CNN	CNN Accuracy = **98.05%**, F1 = **0.9826** LSTM Accuracy = 97.32%, F1 = 0.9786 Bi-LSTM Accuracy = 98.04%, F1= 0.9825 NB, LR, SVM, DT Accuracy = < 73.23%, F1 = NA	NA	Cannot detect malicious link or attachment inside email body.

Table 2. *Cont.*

Authors	DL Models	Results	Main Contribution	Weakness/Limitation
Paliath et al. [31]	NN	NN Accuracy = **99.44%**, F1 = **0.9915** SVM, NB, RS, RF, RT Accuracy = < 99.21%, F1 = < 0.9878	NA	Cannot detect malicious link or attachment inside email body. Dataset is small and model may overfitted. Not scalable and may have limitation in real-world application.
Divakarla et al. [32]	LSTM, Bi-LSTM, CNN+RNN	LSTM Accuracy = **98.8%**, F1 = **0.987** Bi-LSTM Accuracy = 95.4%, F1= 0.95 CNN+RNN Accuracy = 97.9%, F1 = 0.956	NA	Cannot detect malicious link or attachment inside email body. No dropout layer to prevent overfitting.
Gholampour et al. [33]	ALBERT, ROBERTA, BERT, DEBERTA, DEBERT, SQ, and YOSO	BERT and its variants Accuracy = 98%∼99%, F1 = 0.92∼0.97	Developed new adversarial ham/phish dataset. Proposed ensemble method with KNN as shield model to assign correct label before feeding to DL models.	Cannot detect malicious link or attachment inside email body. Require high computational resources. Maximum tokens of BERT an ALBERT is 512. Dataset is small and model may overfitted.
Bountakas et al. [34]	BERT	Balance dataset: Word2Vec+RF Accuracy = **98.95%**, F1 = **0.9897** Other combinations Accuracy = < 97%, F1 = 0.9744 Imbalanced dataset: Word2Vec+LR Accuracy = **98.62%**, F1= **0.9241** Other combinations Accuracy = < 98.42%, F1 = 0.8996	Compare the combination of NLP techniques and ML models.	Cannot detect malicious link or attachment inside email body. BERT model has limitation of maximum 512 tokens.
Qachfar et al. [35]	BERT	BERT F1 = **0.991 to 0.998** RF, DT, SVM, SGD, KNN, GNB, LR, LSTM, CNN F1 = 0.72 to 0.99	Propose method to reduce the impact of imbalanced data by adding synthetic training data.	Cannot detect malicious link or attachment inside email body. BERT model has limitation of maximum 512 tokens.
Sachan et al. [36]	CNN+BiLSTM+GRU	CNN+BiLSTM+GRU Accuracy = **97.32%**, F1 = **0.9545** NB, RF, KNN, SVM Accuracy = <92.4%, F1 = <0.915	Show stacking DL models performed better than ML and single DL model.	Cannot detect malicious link or attachment inside email body. The three block stacking model is complex and may need high computing resources.
Alhogail et al. [37]	GCN	Accuracy = 98.2%, F1 = 0.9855	Propose NLP+GCN.	Cannot detect malicious link or attachment inside email body. No dropout layer to prevent overfitting.

Table 2. *Cont.*

Authors	DL Models	Results	Main Contribution	Weakness/Limitation
Nicholas et al. [38]	CNN	Accuracy = 98.75%, F1 = NA	Use Sand Cat Swam Optimization (SCSO) to tune the weight in CNN.	Cannot detect malicious link or attachment inside email body. SCSO is computationally more expensive than other optimization techniques.
AbdulNabi et al. [39]	BERT	BERT Accuracy = **97.30%**, F1 = **0.9696** BiLSTM Accuracy = 96.43%, F1 = 0.96 KNN and NB Accuracy = <94%, F1 = <0.94	NA	Cannot detect malicious link or attachment inside email body. Maximum input sequence length is 300.
Hina et al. [40]	LSTM-GRU	LSTM+GRU Accuracy = **95%**, F1 = **0.95** LR, SVM, SGD, NB, RF Accuracy = <92%, F1 = <0.90	Show stacking DL models performed better than ML in multiclassification.	Cannot detect malicious link or attachment inside email body.
He et al. [41]	LSTM, Bi-SLTM	LSTM-XGB Accuracy = **98.35%**, F1 = **0.9824** L-SVM, L-GNB, L-DTC Accuracy = <97%, F1 = <0.96	Double-layer detection mechanism for both phishing and insider threats.	Cannot detect the image links embedded in phishing emails. Dataset is small and model may be overfitted.
Aassal et al. [11]	Multiple models from AutoSklearn and TPOT	With header: LR, SVM, Auto-Sklearn Accuracy = **99.95%**, F1 = **0.9995** DL: Accuracy = 99.85%, F1 = 0.9985 Without header: Auto-Sklearn Accuracy = **99.09%**, F1 = **0.9909** DL Accuracy = 97.86%, F1 = 0.9789	Proposed new phishing research benchmarking framework (PhishBench).	Cannot detect malicious link or attachment inside email body.
Alotaibi et al. [42]	CNN	Accuracy = 99.42%, F1 = 0.9917	NA	Cannot detect malicious link or attachment inside email body. No dropout layer to prevent overfitting.
Fang et al. [43]	RCNN (with Bi-LSTM)	RCNN (with Bi-LSTM) Accuracy = **99.84%**, F1 = **0.9933** LSTM Accuracy = 97.38%, F1 = 0.8783 CNN Accuracy = 96.58%, F1 = 0.849	Embedding both character level and word level.	Cannot detect malicious link or attachment inside email body. Maximum input sequence length is 300.
Kaddoura et al. [44]	FFNN (MLP), BERT	FFNN Accuracy = NA, F1 = 0.9922	NA	Cannot detect malicious link or attachment inside email body. Purely feedforward. Limitation to capture local patterns and context analysis.
Salloum et al. [45]	MLP	MLP Accuracy = **94.63%**, F1 = **0.9478** KNN, DT, LR, SVM, RF, NB, XGBoost Accuracy = <93.7%, F1 = 0.9376	New Arabic-English parallel corpus.	Cannot detect malicious link or attachment inside email body. Purely feedforward. Limitation to capture local patterns and context analysis.

Table 2. *Cont.*

Authors	DL Models	Results	Main Contribution	Weakness/Limitation
Jáñez-Martino et al. [46]	BERT	TF-IDF+LR Accuracy = **94.6%**, F1 = **0.953** BERT+LR Accuracy = 94.2%, F1 = 0.939	Propose SPEMC-15K-E and SPEMC-15K-S datasets and multiclassification, used OCR to scan text in the picture of HTML email body.	Cannot detect malicious link or attachment inside email body. Limitation to capture local patterns and context analysis.
Doshi et al. [47]	ANN, CNN, RNN	Dual layer CNN Accuracy = **99.40%**, F1 = **0.992** Dual layer RNN Accuracy = 99.10%, F1 = 0.995 Dual layer ANN Accuracy = 99.51%, F1 = 0.989 Traditional ML models Accuracy = <98.5%, F1 = 0.978	Dual layer approach to overcome class imbalance.	Cannot detect malicious link or attachment inside email body. Splitting different class label data to train with different model seperaely may lead to overfitting for majority class.
Krishnamoorthy et al. [48]	DNN+BiLSTM	DNN-BiLSTM Accuracy = **98.69%**, F1 = **0.9869** LR, RF, RNN, CNN, LSTM Accuracy = <96.39%, F1 = NA	AES encryption in preliminary stage.	Cannot detect malicious link or attachment inside email body.
Borra et al. [49]	DLCNN	DLCNN Accuracy = **98.43%**, F1 = **0.9707** LR, SVM, NB, AdaBoost Accuracy = <89%, F1 = 0.8066	Multiclassification with the combination of PCA for feature extraction, PSO for feature selection, and DL-CNN for classification.	Cannot detect malicious link or attachment inside email body. PSO+DLCNN may computationally expensive. No dropout layer to prevent overfitting.
Saka et al. [50]	BERT	BERT+DBSCAN Accuracy = **99.2%**, F1 = NA BERT+Agglomerative Accuracy = 98.7%, F1 = NA BERT+K-Mean Accuracy = 98.0%, F1 = NA	Compare the combination of BERT and unsupervised clustering algorithms.	Cannot detect malicious attachment inside email body. Manual labeling. Need feature extraction and selection processes. BERT model has limitation of maximum 512 tokens.
Magdy et al. [51]	ANN	Accuracy = 99.94, F1 = 0.9935	Present multiclassification with fast training time (max 78.6 milliseconds).	Cannot detect malicious attachment inside email body. Purely feedforward. Limitation to capture local patterns and context analysis.
Bountakas et al. [52]	DT+KNN+MLP	KNN-DT+ArgMax Accuracy = **99.43%**, F1 = **0.9942** KNN-DT+MLP Accuracy = 99.07%, F1 = 0.9907 LR, GNB, KNN, DT, RF, MLP Accuracy = <98.6%, F1 = <0.9856	Present ensemble learning to train hybrid features with fast training time (31 milliseconds).	Cannot detect malicious attachment inside email body. Limited adaptability to unseen features.
Muralidharan et al. [10]	BERT, CNN	Accuracy = 99.2%, F1 = 0.941	Ensemble learning to analyze all email segments including attachment.	Inference time to process and may need high computing resources.
Lee et al. [7]	CNN-LSTM, BERT	RF-BERT+RF-CNN+LSTM AUPRC = **0.9997**, F1 = NA RF-Word2Vec+LSTM-CNN+LSTM AUPRC = **0.9851**, F1 = NA	Propose moduler architecture to analyze all components of email except attachment.	Cannot detect malicious attachment inside email body. BERT model has limitation of maximum 512 tokens (words length). Require high computational resources.

McGinley and Monroy [27] experimented with the CNN model for phishing email detection with various settings, and an ablation study, by analyzing email body text. They employed character-level word embeddings, which were subsequently input into multiple layers of a CNN model for feature selection and classification. A combination of Nazario, Enron, and Enron-spam datasets was used to train the model, and the sigmoid function was utilized at the output layer. The system attained an accuracy of 98.139%, but they did not compare with other models or previous work. Ramprasath et al. [28] proposed RNN with an LSTM cells model and compared the performance with traditional ML models (SVM and CkNN). They used a public dataset from Kaggle to train the models, and RNN achieved 99.1% accuracy. However, they did not present the detail setting of the model.

The anti-phishing mechanism by analyzing persuasion cues in the email body text was proposed by Valecha et al. [29]. The combination of Millersmile and Enron corpus datasets was used in their experiments. They manually labeled the pilot dataset, 1029 emails, and used NLTK for tokenization. After that, Bi-LSTM was used to train and generate persuasion labels for the remaining dataset. Lexical features were extracted with the Word2Vec technique and used traditional ML models (NB, LR, RF, and SVM) as the classifiers. LR, RF, and SVM classifiers achieved more than 95% accuracy in the experiments. They used manual hyperparameter tuning by randomly searching with 10-fold cross-validation which needs manual effort and is time consuming.

Paradkar [30] experimented and compared conventional ML techniques (RF, LR, NB, SVM, and DT) with DL models (LSTM, Bi-LSTM, and CNN) by training with the Enron corpus. They used Keras, a Python-based open-source deep learning framework, for tokenization and labeling. According to their results, the CNN model performed best with 98.05% accuracy. All DL models predicted better than ML models. The combination of NLP and GCN models for identifying phishing was presented by Alhogail and Alsabih [37] by classifying the email body text. Term Frequency-Inverse Document Frequency (TF-IDF) was employed to compute the edge values, and the model was trained using Radev's dataset. The proposed technique can detect phishing emails with 98.2% accuracy.

Two new features, the ratio of uppercase and lowercase letters (capRatio) and the number of phishing words, were proposed by Paliath et al. [31] and tested with out-of-the-box algorithms available in the WEKA tool. They used a combination of Nazario and SpamAssassin datasets. The Information Gain method was used for feature selection. Experimental results showed that the NN model outperformed ML algorithms with 99.44% accuracy, but the authors did not explain the detail settings of the NN such as whether it was an RNN or CNN, how many layers, height, width, etc. Divakarla and Chandrasekaran [32] compared different ML and DL algorithms for phishing email detection and phishing URL detection. They combined PhishTank and Kaggle datasets to train the models. The experimental findings indicated that CNN and RNN are two DL techniques that enhance the ability to identify phishing. However, the authors did not mention the detail settings of DL architectures.

Gholampour and Verma [33] researched phishing email detection robustness on the adversarial attacks with transformer-based deep learning models (ALBERT, ROBERTA, BERT, DEBERTA, DEBERT, SQ, and YOSO) for NLP-based classification and KNN. They used IWSPA 2.0, and adversarial phishing attack emails generated by the GPT-2 model with a TextAttack framework, to train the models. All BERT-based models can detect phishing emails with 98% accuracy, and they proposed an ensemble KNN-based black box adversarial attack detection technique. Sachan et al. [36] proposed an ensemble learning model (CNN-Bi-LSTM-GRU) to classify spam email. They used SpaCy for tokenization, Word2Vec for feature extraction, and a combination of multiple datasets (Ernon, Phished corpora, and Hate Speech & Offensive). Their stacking DL models performed better than traditional ML models and achieved 97.32% accuracy.

Qachfar et al. [35] presented phishing email detection with a three-phase approach with the BERT model. First, they produced synthetic emails with the LeakGAN adversarial neural network, and then generated synthetic emails were labeled by Positive-Unlabeled

learning with SVM. Finally, the generated data were combined with SpamAssassin, Enron, Wikileaks, and Nazario datasets and fed to a pre-trained BERT model. Their proposed model achieved a 0.99 F1 score. AbdulNabi and Yaseen [39] also proposed spam email detection with a pre-trained BERT model and compared the performance with Bi-LSTM. They used Scikit-learn, an open-source Python library, for data processing and tokenization. The combination of UCI ML and SpamFilter datasets was used to train the BERT, and AdamW optimizer was used for hyperparameter tuning. Their proposed model outperformed Bi-LSTM with 97.30% accuracy.

Bountakas et al. [34] compared different combinations of NLP techniques (TF-IDF, Word2Vec, and BRET) with traditional ML classifiers (RF, DT, LR, GBT, and NB) for phishing email detection. NLP methods were employed for feature extraction, while the features were selected with the Chi-square technique. The combination of the Nazario phishing corpus and the Enron dataset was used in their experiments. Their results showed that BERT did not perform well for phishing email feature extraction when compared to Word2Vec. Hina et al. [40] proposed a multiclass email classification method (SeFACED) to identify ham, phish, harassment, and suspicious with LSTM-based GRU by analyzing the email body text. They used NLTK for data cleansing, SpaCy for tokenization, TF-IDF for feature extraction, Adam optimizer, and the combination of Enron, CLAIR, and Hate Speech & Offensive datasets. Their proposed model achieved 95% accuracy for multiclassification.

Nicholas and Nirmalrani [38] presented spam email detection with a combination of the CNN model and bio-inspired Sand Cat Swarm Optimization algorithm, which optimized the weight parameters during each epoch. A bag of words (BoW) was used for feature extraction and selection after data processing. They used the UCI ML dataset in their experiments and achieved 98.75% accuracy. However, they did not compare the proposed technique with other models or studies.

3.2. Classification by Header, Subject, and Email Body

Aassal et al. [11] presented a benchmarking framework (PhishBench) for the phishing detection of URLs, websites, and emails (with headers and without headers). It has five modules, an input module, a feature extraction and ranking module, a classification module, and an evaluation module. The feature selection module has four algorithms, Information Gain, Gini Index, Chi-Square Metric, and Recursive Feature Elimination, and TF-IDF was used for word embedding. Multiple classifiers from AutoSklearn, Tree-Based Pipeline Optimization Tool (TPOT), and Library for Scalable Online Learning (LIBSOL) were used in their experiment. They used a combination of various public datasets (Wikileaks, Enron, SpamAssassin, and Nazario) to train the models. With headers and body features, LR, SVM, and AutoSklearn achieved 99.95% accuracy, whereas DL obtained 99.85% accuracy. Without header features, AutoSklearn can detected with 99.09% accuracy, while DL attained 97.86% accuracy.

The double-layer detection method to detect both phishing emails and insider threat, which consisted of seven tuples of behavioral sequences, was designed by He et al. [41]. They used LSTM for feature extraction of the subject, Bi-LSTM for email body text, and XGBoost with a custom loss function for classification in the proposed design. The Enron dataset and phishing email dataset from monkey.org were used to train the model, and the proposed technique achieved 98.38% accuracy. Alotaibi et al. [42] presented phishing email detection with CNN. They trained the model using a combination of the Phishing Corpus and SpamAssassin datasets. Their model can detect phishing emails with 99.42% accuracy.

Fang et al. [43] proposed an RCNN framework incorporating words and character vectors and an attention mechanism for identifying phishing emails. The combination of IWSPA-AP, Wikileaks, Enron, SpamAssassin, and Nazario datasets was used in their experiment. They divided the email body and header into both character and word levels. Word2Vec was used for word embedding, and the tanh activation function was used in each layer of the RCNN. Instead of RNN, Bi-LSTM was used in their RCNN and attained 99.84% accuracy. Kaddoura et al. [44] researched phishing email detection with the FFNN

model and compared it with BERT. They used the Enron dataset and TF-IDF for word embedding before feeding to FFNN. The proposed model achieved a 0.9922 F1 score.

A new English–Arabic parallel corpus was created by Salloum et al. [45] by a combination of the IWSPA-AP dataset and its Arabic translation. They tested that dataset with KNN, DT, LR, SVM, RF, NB, XGBoost, and Multilayer Perceptron classifiers. NLTK for data processing and TF-IDF for feature selection were used in their research. Among the tested classifiers, the MLP model achieved the highest accuracy, 94.63% for the English emails and 96.82% for the Arabic emails. Jáñez-Martino et al. [46] evaluated and proposed phishing email multiclass classification with a combination of four NLP techniques (TF-IDF, BoW, Word2Vec, and BERT) and four traditional ML classifiers (SVM, NB, RF, LR). The email subjects were extracted from headers, and HTML email bodies were transformed into grayscale images, and then texts were extracted with OCR. They created private datasets, SPMEC-15K-E and SPEMC-15K-S, for their experiments. The accuracy for BERT-LR achieved 94.2% while TF-IDF-LR attained 94.6% for the English dataset.

Doshi et al. [47] proposed a two-layer architecture, one layer for the majority class and another for the minority class, for phishing and spam email detection with DL (ANN, CNN, and RNN) and traditional ML models. The combination of Nazario and SpamAssassin datasets was used to train the models. The data were defined as majority class if they had more high-frequency data instances. They used NLTK for data processing and TF-IDF for tokenization. The highest accuracy was achieved by a dual-layer ANN with 99.51%. Krishnamoorthy et al. [48] presented the DNN-BiLSTM model for phishing email detection. They used NLTK for data cleaning, SpaCy for tokenization, TF-IDF for feature extraction, Information Gain for feature selection, and Enron datasets. Their proposed model detected phishing emails with 98.69% accuracy.

The phishing email detection with the CNN model was presented by Borra et al. [49]. The combination of UCI ML, CSDMC, and SpamAssassin datasets was used to train the proposed model. They used Principal Component Analysis (PCA) for feature extraction, Particle Swarm Optimization (PSO) for feature selection, and Stochastic Gradient Descent (SGD) for parameter optimization. Their proposed model attained 98.43% accuracy.

3.3. Classification by Header, Subject, Email Body, and URL

Saka et al. [50] researched phishing email classification by a combination of BERT and three ML classifiers (K-Means, DBSCAN, and Agglomerative). The consolidated Nazario and Enron datasets were used in their experiments. They used BERT, Chi-square, Manual, bag-of-topics, and Tranco list for features extraction of the body, header, subject line, and URL, respectively. BERT-DBSCAN performed the best with a 99.2% accuracy in their experimentation.

Magdy et al. [51] proposed an email classification model with ANN to identify ham, spam, or phishing. They utilized three benchmark datasets from the UCI ML archive (SpamBase, CSDMC2010, and SpamAssassin+Nazario). The features were extracted from the header, address, text, and URL from the email body. After that, features were selected with low-variance, PCA, and Chi-squared methods. The proposed technique achieved 99.94% accuracy.

Bountakas and Xenakis [52] presented an ensemble learning approach for identifying phishing emails. They compared two methods: method 1 applied MLP as a Tier-2 classifier, and method 2 used the ArgMax fusion algorithm as a Tier-2 classifier. Both methods used the DT-KNN stacking model as a Tier-1 base learner. In their research, Word2Vec was employed for feature extraction, the Mutual Information filter method was utilized for feature selection, and the model was trained using a combination of the Enron and SpamAssassin datasets. Although MLP achieved the high accuracy detection with the hybrid features, Voting Ensemble learning (DT-KNN+ArgMax) performed best with 99.43% accuracy.

3.4. Classification by Header, Subject, Email Body, URL, and Attachment

Muralidharan and Nissim [10] proposed a deep ensemble learning method for malicious email detection. Their system examined all parts of the email with three pipelines: header, body, and attachment. For the header segment, ASCII character image plots were classified by custom CNN, and header strings were classified by pre-trained BERT. In the next segment, body texts were classified with BERT. The attachments were transformed into images with a byteplot and classified with a custom CNN model. All of the probability values made by each base learner of three pipelines were used as input to the XGBoost Meta-classifier for final prediction. They used a collection of emails from the VirusTotal dataset to train the model and achieved a 0.968 F1 score.

3.5. Classification by Email Structure, Body, and URL

In another study, a multi-modular architecture with three learning modules for structure analysis, text analysis, and URL analysis for phishing email detection, named D-Fence, was proposed by Lee et al. [7]. The structure analysis module analyzed values of the headers and HTML content structure such as the number of hyperlinks, unique domain names, linked URLs, and Document Object Model and then classified them with the RF model. In the text analysis module, they used pre-trained BERT for word embedding and a RF classifier. The URL analysis module used the CNN-LSTM model, CNN extracted local features and LSTM learned the prolonged correlations from the encoded characters. The predictions of each module were fed to the XGBoost meta-classifier for final prediction. They used a private dataset EES 2020 that was collected from multiple enterprises. Their proposed model claimed 0.99 AUPRC.

4. Limitations

The limitation of the proposed phishing detection systems in the selected articles will be discussed in this section.

4.1. Dataset

This study found that the availability of big datasets is one of the limitations in phishing email detection. The details of different datasets used in selected articles can be found in Table A1 in the Appendix A. The efficacy of DL models heavily depends on the size, quality, and variety of the data utilized during training. If the input data are small, homogeneous, and contain irrelevant features, the model may not generalize new phishing patterns [11]. As a result, the model may become overfitted, leading to inaccurate prediction results in production environments. While there is no specific research indicating the minimum dataset size required to train DL models for phishing detection, a general guideline for traditional classification models suggests at least 3000 training samples [53]. Furthermore, deep learning models, which consist of multiple layers, typically require larger datasets than simpler models to achieve effective generalization.

Some researchers utilized private datasets obtained from various private corporations to address this problem. On the other hand, many researchers may not have such an opportunity due to privacy concerns. The generation of synthetic phishing samples using generative AI systems could be an option, as demonstrated by the work of Qachfar et al. [35], to overcome this challenge. Data augmentation is a widely adopted technique that has been shown to effectively enhance the generalization performance of models [54]. This approach can enhance the diversity of the training dataset, allowing models to learn from a broader range of scenarios and potential attack patterns. The synthetic data may effectively supplement existing datasets. However, it is time consuming, and the similarity between synthetic and genuine phishing emails is uncertain. Therefore, it is important to select recent, adequate and diverse datasets for training the model when building DL-based phishing detection systems. Additionally, future studies should experiment with various contexts from diverse data sources.

4.2. Future Engineering

This study discovered that some researchers selected the features manually, while others applied various methods such as PCA, PSO, Chi-square, mutual information, information gain, etc. The manual feature selection process requires expert knowledge, manual effort, and is prone to human errors. Moreover, while supervised feature selection methods can perform very well during model training, their performance may degrade when applied to unseen data in real-world applications. These techniques also lack the ability to adapt to the evolving nature of attack patterns. Thus, future research should prioritize the use of automatic feature selection techniques.

4.3. Flexible and Robust System

In selected articles, there are different approaches to analyze email segments as presented in the previous section. Most of the studies have limitations to detecting phishing emails based on their headers, structure, and attachments, which diminishes their effectiveness in identifying more sophisticated phishing attacks. In many cases, phishing emails closely mimic legitimate ones in their content. Therefore, analyzing only the semantic features of the email body may not be sufficient. On one hand, there could be ethical and privacy concerns with extracting features from headers such as sender and receiver addresses. However, these challenges can be addressed by using encryption techniques at preliminary stages or transforming the data into another form as proposed by Krishnamoorthy et al. [48] and Muralidharan and Nissim [10]. Moreover, the DL model should be developed with cross-domain system testing. For example, it should be trained with one dataset and tested with a different one from another source to verify the robustness and effectiveness of classification on different types of attack patterns. To ensure that the model is robust and generalizes well, K-fold cross-validation should apply during model building. It is also recommended to test the model using recent, real-world, and sophisticated phishing emails to ensure better performance evaluation and generalization of the results on newer phishing emails. In addition, the researchers should develop a flexible model capable of analyzing any segment of an email to detect phishing. Finally, future research should explore deep reinforcement learning and deep online learning to adapt to evolving changes in attack patterns.

4.4. Sophisticated Phishing Techniques

In several research papers, various methods have been presented for effectively detecting phishing emails. These methods include the use of NLP, hybrid models (combined with genetic algorithms), transfer learning, stacking, and ensemble learning to analyze the email context, headers, and URLs. Among these, Jáñez-Martino et al. [46] proposed a system capable of detecting advanced phishing techniques such as text written in images and salting by utilizing optical character recognition (OCR) during the data processing stage. Furthermore, Muralidharan and Nissim [10] converted attachments into images using byteplot to classify malicious attachments with CNN. However, there are significant research gaps in preventing more sophisticated attacks, such as URL obfuscation, which uses URL shortening and QR codes to conceal malicious links in email and zero-day attacks.

5. Conclusions

In conclusion, this systematic literature review (SLR) examined the advancement of DL-based techniques, the types of datasets used, and the effectiveness of DL in comparison to traditional ML methods; identified current limitations and gaps; and suggested possible future improvements in the field of phishing prevention research. Numerous studies in the selected articles have demonstrated that various DL-based techniques surpass traditional machine learning models with promising results in phishing detection. In contrast, there are three articles that indicate traditional ML (KNN, DT, LR, SVM, and RF) performed better than DL (BERT and MLP), yet they have weaknesses. These traditional models may face difficulties in generalizing to unseen and complex patterns in data. Although traditional

models are computationally inexpensive and simple to deploy, they lack the ability to learn and adapt to evolving attack patterns. As a result, they require frequent retraining to keep up with changes in data. Additionally, these models are not easily scalable, limiting their effectiveness in handling large datasets or rapidly changing environments.

Some studies demonstrated that the ensemble learning and modular architecture approach appears well suited to detecting sophisticated phishing attacks by effectively analyzing all components of an email. This approach leverages multiple models, each specializing in different parts of the email, such as the header, body, and attachments. Moreover, this design allows for flexibility, enabling the system to handle complex phishing tactics that might evade detection by a single model. On the other hand, it is crucial to consider the trade-off between detecting sophisticated phishing email and system complexity, resource requirements, and deployment efficiency in production environments. The ensemble and modular DL models may need higher computational power, longer processing times, higher costs, and slower deployments. Finding the right balance between detection capability and system efficiency is essential to create a robust yet practical phishing detection system for real-world applications.

Finally, this study revealed a significant gap in defenses against sophisticated attacks, such as URL obfuscation. In future work, expanding shortened URLs and QR decoding at the initial data processing stages will be considered to address this gap. Furthermore, we plan to investigate deep online learning techniques through rigorous testing and validation with varied datasets across domains to build a resilient and adaptable system.

Author Contributions: Conceptualization, P.H.K.; methodology, P.H.K.; formal analysis, P.H.K.; investigation, P.H.K.; resources, J.G.; writing—original draft preparation, P.H.K.; writing—review and editing, J.G. and A.G.; visualization, P.H.K.; supervision, J.G. and A.G.; project administration, J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: This study conducted a systematic literature review, and no new data were created.

Conflicts of Interest: The authors have no competing interests to declare that are directly or indirectly related to the content of this article. The authors declare no competing interests.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
FFNN	Feedforward Neural Network
GCN	Graph Convolutional Network
GRU	Gated Recurrent Unit
LLM	Large Language Model
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
NN	Neural Network
RCNN	Recurrent Convolutional Neural Networks
RNN	Recurrent Neural Network

Appendix A

Table A1. The list of datasets used in selected articles.

Datasets	Size	Authors
Phishing email detection dataset	Total 828 (ham 504, phishing 324)	Chataut et al. [23]
Private dataset	Total 18,366 (ham 14,950, phish 3416)	Bagui et al. [24]
Ling-Spam, Enron-Spam, Enron, SpamAssassin, and Nazario	Total 22,965 (ham 15,502, phish 7463)	Giri et al. [25]
Enron-Spam, Enron, and Nazario	Total 3804 (ham 1870, phish 1934)	McGinley and Monroy [27]
Private dataset (EES2020)	Total 291,702 (ham 224,137, phish 67,565)	Lee et al. [7]
Private dataset	Total 5572 (ham 4572, phish 1000)	Zannat et al. [26]
Enron and monkey.org	Total 4000 (ham 2000, phish 2000)	He et al. [41]
Kaggle	Total 5572	Ramprasath et al. [28]
Wikileaks, Enron, SpamAssasin, and Nazario	Total 22,000 (ham 10,500, phish 10,500)	Aassal et al. [11]
PhishingCorpus and SpamAssassin	Total 6428 (ham 4150, phish 2278)	Alotaibi et al. [42]
IWSPA-AP, Wikileaks, Enron, SpamAssassin, and Nazario	Total 8780 (ham 7781, phish 999)	Fang et al. [43]
Millersmile and Enron corpus	Total 38,084 (ham 19,661, phish 18,423)	Valecha et al. [29]
Enron corpus	Total 20,000 (ham 11,664, phish 8336)	Paradkar [30]
Nazario and SpamAssasin	Total 1256 (ham 842, phish 414)	Paliath et al. [31]
Kaggle and PhishTank	Total 5171 (ham 3672, phish 1499)	Divakarla and Chandrasekaran [32]
Enron	Total 32,638 (ham 16,094, phish 16,544)	Kaddoura et al. [44]
IWSPA 2.0 and generated dataset	Total 7286 (ham 5692, phish 1594)	Gholampour and Verma [33]
Nazario and Enron	Total 15,407 (ham 14,000, phish 1407)	Bountakas et al. [34]
Nazario and Enron	Total 4472 (ham 2193, phish 2279)	Saka et al. [50]
SpamAssasin, Enron, Wikileaks, and Nazario	Total 21,000 (ham 10,500, phish 10,500)	Qachfar et al. [35]
IWSPA-AP and Arbaic-translated	Total 84,033 (ham 47,692, phish 36,341)	Salloum et al. [45]
Ernon corpora, Phished emails corpora, and Hate Speech & Offensive	Total 42,153 (ham 12,498, harassment 19,190, suspicious 5323, phish 5142)	Sachan et al. [36]
Radev	Total 8579 (ham 4894, phish 3685)	Alhogail and Alsabih [37]
SPEMC-15K-E, and SPEMC-15K-S	Spam 15,000 each	Jáñez-Martino et al. [46]
Nazario and SpamAssasin	Total 5554 (ham 2664, phish 4204, spam 1350)	Doshi et al. [47]
UCI ML	Total 23,386 (ham 14,011, phish 4864, spam 4511)	Magdy et al. [51]
UCI ML	Total 9120 (ham 1200, phish 7920)	Nicholas and Nirmalrani [38]
Enron, SpamAssassin, and Nazario	Total 35,511 (ham 32,051, phish 3460)	Bountakas and Xenakis [52]
VirusTotal	Total 32,676 (ham 20,037, phish 9996)	Muralidharan and Nissim [10]
Enron	Total 33,727 (ham 16,563, phish 17,188)	Krishnamoorthy et al. [48]
UCI ML, CSDMC, and SpamAssassin	NA	Borra et al. [49]
UCI ML and SpamFilter	Total 5000 (ham 3000, spam 2000)	AbdulNabi and Yaseen [39]
Enron, CLAIR, and Hate Speech & Offensive	Total 32,427 (ham 9001, harassment 9138, suspicious 5287, phish 9001)	Hina et al. [40]

References

1. Anti-Phishing Working Group (APWG). Phishing Activity Trends Report: 4th Quarter 2023. 2023. Available online: https://www.apwg.org/trendsreports/ (accessed on 27 February 2024).
2. Federal Bureau of Investigation (FBI). 2022 Internet Crime Report. 2022. Available online: https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf (accessed on 27 February 2024).
3. Check Point Research. 2023 Cyber Security Report. 2023. Available online: https://resources.checkpoint.com/report/2023-check-point-cyber-security-report (accessed on 23 February 2024).
4. Verizon. Data Breach Investigations Report 2022. 2022. Available online: https://www.phishingbox.com/downloads/Verizon-Data-Breach-Investigations-Report-DBIR-2022.pdf (accessed on 23 February 2024).
5. Yamin, M.M.; Ullah, M.; Ullah, H.; Katt, B. Weaponized AI for cyber attacks. *J. Inf. Secur. Appl.* **2021**, *57*, 102722. [CrossRef]
6. Kocher, G.; Kumar, G. Machine learning and deep learning methods for intrusion detection systems: Recent developments and challenges. *Soft Comput.* **2021**, *25*, 9731–9763. [CrossRef]

7. Lee, J.; Tang, F.; Ye, P.; Abbasi, F.; Hay, P.; Divakaran, D.M. D-Fence: A flexible, efficient, and comprehensive phishing email detection system. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 6–10 September 2021; pp. 578–597.
8. Apruzzese, G.; Colajanni, M.; Ferretti, L.; Guido, A.; Marchetti, M. On the effectiveness of machine and deep learning for cyber security. In Proceedings of the 2018 10th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 29 May–1 June 2018; pp. 371–390.
9. Ahmad, R.; Alsmadi, I. Machine learning approaches to IoT security: A systematic literature review. *Internet Things* **2021**, *14*, 100365. [CrossRef]
10. Muralidharan, T.; Nissim, N. Improving malicious email detection through novel designated deep-learning architectures utilizing entire email. *Neural Netw.* **2023**, *157*, 257–279. [CrossRef]
11. El Aassal, A.; Baki, S.; Das, A.; Verma, R.M. An in-depth benchmarking and evaluation of phishing detection research for security needs. *IEEE Access* **2020**, *8*, 22170–22192. [CrossRef]
12. Odeh, A.; Keshta, I.; Abdelfattah, E. Machine learningtechniquesfor detection of website phishing: A review for promises and challenges. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Virtual, 27–30 January 2021; pp. 0813–0818.
13. Zaimi, R.; Hafidi, M.; Lamia, M. Survey paper: Taxonomy of website anti-phishing solutions. In Proceedings of the 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), Paris, France, 14–16 December 2020; pp. 1–8.
14. Zaimi, R.; Hafidi, M.; Lamia, M. A literature survey on anti-phishing in websites. In Proceedings of the 4th International Conference on Networking, Information Systems & Security, Kenitra, Morocco, 1–2 April 2021; pp. 1–7.
15. Tang, L.; Mahmoud, Q.H. A survey of machine learning-based solutions for phishing website detection. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 672–694. [CrossRef]
16. Aung, E.S.; Zan, C.T.; Yamana, H. A survey of URL-based phishing detection. In *DEIM Forum*; 2019; pp. G2–G3. Available online: https://db-event.jpn.org/deim2019/post/papers/201.pdf (accessed on 8 April 2024).
17. Benavides, E.; Fuertes, W.; Sanchez, S.; Sanchez, M. Classification of phishing attack solutions by employing deep learning techniques: A systematic literature review. In *Developments and Advances in Defense and Security: Proceedings of MICRADS 2019*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 51–64.
18. Al-Yozbaky, R.S.; Alanezi, M. A Review of Different Content-Based Phishing Email Detection Methods. In Proceedings of the 2023 9th International Engineering Conference on Sustainable Technology and Development (IEC), Erbil, Iraq, 21–23 February 2023; pp. 20–25.
19. Salloum, S.; Gaber, T.; Vadera, S.; Shaalan, K. A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access* **2022**, *10*, 65703–65727. [CrossRef]
20. Quang, D.N.; Selamat, A.; Krejcar, O. Recent research on phishing detection through machine learning algorithm. In Proceedings of the Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices: 34th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2021, Kuala Lumpur, Malaysia, 26–29 July 2021; Proceedings, Part I 34; Springer: Berlin/Heidelberg, Germany, 2021; pp. 495–508.
21. Kitchenham, B.; Brereton, O.P.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering—A systematic literature review. *Inf. Softw. Technol.* **2009**, *51*, 7–15. [CrossRef]
22. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*.
23. Chataut, R.; Gyawali, P.K.; Usman, Y. Can AI Keep You Safe? A Study of Large Language Models for Phishing Detection. In Proceedings of the 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2024; pp. 0548–0554.
24. 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; pp. 1–2.
25. Giri, S.; Banerjee, S.; Bag, K.; Maiti, D. Comparative Study of Content-Based Phishing Email Detection Using Global Vector (GloVe) and Bidirectional Encoder Representation from Transformer (BERT) Word Embedding Models. In Proceedings of the 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), Trichy, India, 16–18 February 2022; pp. 01–06.
26. Zannat, R.; Mumu, A.A.; Khan, A.R.; Mubashshira, T.; Mahmud, S.R. A Deep Learning-Based Approach for Detecting Bangla Spam Emails. In Proceedings of the 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Tenerife, Spain, 19–21 July 2023; pp. 1–6.
27. McGinley, C.; Monroy, S.A.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; pp. 5609–5613.
28. Ramprasath, J.; Priyanka, S.; Manudev, R.; Gokul, M. Identification and mitigation of phishing email attacks using deep learning. In Proceedings of the 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 12–13 May 2023; pp. 466–470.

29. Valecha, R.; Mandaokar, P.; Rao, H.R. Phishing email detection using persuasion cues. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 747–756. [CrossRef]
30. Paradkar, N.S. Phishing Email's Detection Using Machine Learning and Deep Learning. In Proceedings of the 2023 3rd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS), Ernakulam, India, 18–20 May 2023; pp. 160–162.
31. Paliath, S.; Qbeitah, M.A.; Aldwairi, M. PhishOut: Effective phishing detection using selected features. In Proceedings of the 2020 27th International Conference on Telecommunications (ICT), Bali, Indonesia, 5–7 October 2020; pp. 1–5.
32. Divakarla, U.; Chandrasekaran, K. Predicting Phishing Emails and Websites to Fight Cybersecurity Threats Using Machine Learning Algorithms. In Proceedings of the 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 29–31 December 2023; pp. 1–10.
33. Mehdi Gholampour, P.; Verma, R.M. Adversarial robustness of phishing email detection models. In Proceedings of the 9th ACM International Workshop on Security and Privacy Analytics, Charlotte, NC, USA, 26 April 2023; pp. 67–76.
34. Bountakas, P.; Koutroumpouchos, K.; Xenakis, C. A Comparison of Natural Language Processing and Machine Learning Methods for Phishing Email Detection. In Proceedings of the 16th International Conference on Availability, Reliability and Security, New York, NY, USA, 17–20 August 2021; ARES '21. [CrossRef]
35. Qachfar, F.Z.; Verma, R.M.; Mukherjee, A. Leveraging synthetic data and pu learning for phishing email detection. In Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy, Baltimore, DC, USA, 25–27 April 2022; pp. 29–40.
36. Sachan, S.; Doulani, K.; Adhikari, M. Semantic Analysis and Classification of Emails through Informative Selection of Features and Ensemble AI Model. In Proceedings of the 2023 Fifteenth International Conference on Contemporary Computing, Noida, India, 3–5 August 2023; pp. 181–187.
37. Alhogail, A.; Alsabih, A. Applying machine learning and natural language processing to detect phishing email. *Comput. Secur.* **2021**, *110*, 102414. [CrossRef]
38. Nicholas, N.N.; Nirmalrani, V. An enhanced mechanism for detection of spam emails by deep learning technique with bio-inspired algorithm. *e-Prime-Adv. Electr. Eng. Electron. Energy* **2024**, *8*, 100504. [CrossRef]
39. AbdulNabi, I.; Yaseen, Q. Spam Email Detection Using Deep Learning Techniques. *Procedia Comput. Sci.* **2021**, *184*, 853–858. [CrossRef]
40. Hina, M.; Ali, M.; Javed, A.R.; Ghabban, F.; Khan, L.A.; Jalil, Z. Sefaced: Semantic-based forensic analysis and classification of e-mail data using deep learning. *IEEE Access* **2021**, *9*, 98398–98411. [CrossRef]
41. He, D.; Lv, X.; Xu, X.; Chan, S.; Choo, K.K.R. Double-layer Detection of Internal Threat in Enterprise Systems Based on Deep Learning. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 4741–4751. [CrossRef]
42. Alotaibi, R.; Al-Turaiki, I.; Alakeel, F. Mitigating email phishing attacks using convolutional neural networks. In Proceedings of the 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 19–21 March 2020; pp. 1–6.
43. 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* **2019**, *7*, 56329–56340. [CrossRef]
44. 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), Virtual, 4–6 November 2020; pp. 193–198.
45. Salloum, S.; Gaber, T.; Vadera, S.; Shaalan, K. A New English/Arabic Parallel Corpus for Phishing Emails. *ACM Trans. Asian Low Resour. Lang. Inf. Process.* **2023**, *22*, 1–17. [CrossRef]
46. Jáñez-Martino, F.; Alaiz-Rodríguez, R.; González-Castro, V.; Fidalgo, E.; Alegre, E. Classifying spam emails using agglomerative hierarchical clustering and a topic-based approach. *Appl. Soft Comput.* **2023**, *139*, 110226. [CrossRef]
47. 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]
48. Krishnamoorthy, P.; Sathiyanarayanan, M.; Proença, H.P. A novel and secured email classification and emotion detection using hybrid deep neural network. *Int. J. Cogn. Comput. Eng.* **2024**, *5*, 44–57. [CrossRef]
49. Borra, S.R.; Yukthika, M.; Bhargavi, M.; Samskruthi, M.; Saisri, P.V.; Akhila, Y.; Alekhya, S. OECNet: Optimal feature selection-based email classification network using unsupervised learning with deep CNN model. *e-Prime-Adv. Electr. Eng. Electron. Energy* **2024**, *7*, 100415. [CrossRef]
50. Saka, T.; Vaniea, K.; Kökciyan, N. Context-Based Clustering to Mitigate Phishing Attacks. In Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security, Los Angeles, CA, USA, 11 November 2022; AISec'22, pp. 115–126. [CrossRef]
51. Magdy, S.; Abouelseoud, Y.; Mikhail, M. Efficient spam and phishing emails filtering based on deep learning. *Comput. Netw.* **2022**, *206*, 108826. [CrossRef]
52. Bountakas, P.; Xenakis, C. Helphed: Hybrid ensemble learning phishing email detection. *J. Netw. Comput. Appl.* **2023**, *210*, 103545. [CrossRef]

53. Koshute, P.; Zook, J.; McCulloh, I. Recommending training set sizes for classification. *arXiv* **2021**, arXiv:2102.09382.
54. Ying, X. An overview of overfitting and its solutions. *J. Phys. Conf. Ser.* **2019**, *1168*, 022022. [CrossRef]