Journal of Intelligent System and Applied Data Science (JISADS)

Journal homepage : https://www.jisads.com

ISSN (2974-9840) Online

A REVIEW ON DETECTING PHISHING EMAILS USING CNN AND BI-LSTM TECHNIQUES

Nicholas Muriuki Muriithi[1], Dr.Ephantus Mwangi[1], Dr.Kennedy Malanga[1]*

[1]*School of Pure and Applied Sciences, Kirinyaga University, Kerugoya, 138-10290, Kenya*
nickmuri123@gmail.com, emwangi@kyu.ac.ke, kmalanga@kyu.ac.ke

ABSTRACT

Phishing schemes have become more sophisticated, with the attackers posing as reputable businesses and altering the URLs to acquire the attention of consumers. These tactics such as URL shortening, obfuscation, and targeting multimedia exploit more complicated mechanisms as the detection used in the process. Existing detection methods often work poorly in multilingual content and are mostly based on characters, omitting important word- and context-based cues required to effectively distinguish among formats and languages. The fact that traditional machine learning models depend on human ability to extract features hinders their performance by reducing their adaptation and real-time capacity. The research reviews and assesses current phishing detection methods and provides recommandations for future research aimed at identifying optimal detection models. The proposed solution is to deploy countermeasures to deal with the time-sensitive characteristic of phishing attacks by enhancing real-time detection on fake URLs, especially in email and instant messaging systems. The study shows that the Convolutional Neural Network (CNN) became the most effective algorithm with a score of 15% in the assessment, the next model was Support Vector Machine (SVM) with 13%, and the Long Short-Term Memory (LSTM) network with 10%. The bottom of the ranking went to Natural Language Processing (NLP), Logistic Regression, and the CNN variant with the input of text and images, all with 2%. The review was done from 35 articles from google scholar and 27 articles were selected to analyze the result. The study reviewed high-quality, peer-reviewed papers accessed through Google Scholar, encompassing publications from Web of Science–indexed journals. The CNN and Bi-LSTM hybrid model is the most effective of the models that were examined, offering the best detection performance and making it a great option for real-world phishing prevention systems. In the six models examined the overall frequency score was 44% which gave an average accuracy of 7.32. Standard deviation was found to be ±5.6, which means that there is a significant difference in the models in terms of detection performance. Such dispersion demonstrates the inequity in performance with a small set of models working towards the overall performance and others performing well below average. It is important to note that CNN-BiLSTM hybrid model showed the highest score in detection, which was obvious in comparison with the other methods. Such high performance proves the robustness and reliability of the hybrid architecture as it is a good candidate to be used in the real world phishing detection and prevention systems.

Keywords: Algorithm, Bi-directional Long Short term memory (Bi-LSTM), convolutional Neural Networks (CNN),Neural Networks, Phishing, Universal resource locator (URL);

1. INTRODUCTION

Phishing attacks are one of the most pernicious ways that cybercriminals take advantage of people among the many hazards that both individuals and companies must contend with. Phishing usually entails deceiving someone into disclosing private information, frequently by using malicious URLs that impersonate trustworthy websites. Traditional detection systems are unable to keep up with the rapid evolution of phishing techniques, which calls for the creation of more advanced strategies [1].

In order to overcome these issues, scholars have investigated an extensive variety of machine learning

(ML) and deep learning (DL) systems to identify phishing. The classical ML models, which include Support Vector Machines, Decision Trees, Naive Bayes, K-Nearest Neighbors, and even the Random Forests, have demonstrated good results under controlled conditions and require a lot of hand-crafted features and do not respond well to novel and unknown modes of attack. More recently, deep learning models such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), attention-based architectures and transformer models have been shown to achieve higher performance due to automatic acquisition of complex patterns on large-scale phishing data.

The author [2] investigated the effectiveness of detecting phishing URLs in emails which utilized the use of hybrid techniques which involved the Convolutional neural networks (CNN) with Bidirectional long Term short memory (BiLSTM) which is well suited for evaluating sequential inputs which consists of text based URLs because of its benefit in both forward and reverse orientations. The use of hybrid technique allows the model to learn contextual linkages and semantic patterns across URLs of different lengths, which are frequently used in phishing efforts. Bi-LSTMs can be trained to discover hidden patterns typically used in phishing emails by capturing both structural and sequential properties, which improves the model's capacity to distinguish between authentic and malicious links [2]. The combination of CNN's spatial feature extraction strength with Bi-LSTM's sequential modeling capability yields a powerful solution to phishing detection.

The Convolutional Neural Networks (CNNs) are extremely good at identifying patterns within character sequences on webpages, making them especially valuable for detecting possible dangers in URLs [2]. CNNs excel at capturing URLs' fundamental structure and attributes, resulting in improved feature extraction and threat detection accuracy[3].Combining convolutional neural networks with Bi-directional Long Short-Term Memory(Bi-LSTM) networks, trained to process sequential information in forward and backward directions, allows capturing the structural and contextual relation-ships and enables an efficient detection of the anomalies in URLs in emails [2]. This hybrid approach offers better separation between legal and phishing URLs in that it learns about the sequence of characters as well as the context of a URL and thus increasing its ability to identify phishing attacks more effectively [4].

Bi-LSTMs, a more advanced type of recurrent neural network, are very good at processing data sequences in both forward and backward orientations [2]. This capability makes them suitable for investigating the environmental dependencies and temporal patterns that shape URL behaviors across time. Bi-LSTMs provide a more in-depth comprehension of sequential data by collecting information from both past and future contexts, which is critical for detecting subtle trends in phishing efforts. The model will attempt to combine these two effective concepts in order to detect rogue Email URLs and learn from different phishing tactics [5].

The key contribution of the paper in question is a synthesis of the latest phishing detection studies, including the identification of the key trends, gaps in the methodology, and the problems that remain unsolved in the field. The synthesis of the results of different research works presented in the review offers information about the usefulness of the current methods and explains the direction of the further research in the context of the creation of more effective, explainable, and adaptive phishing detection systems that would help to combat the emergent cyber threats.

2. LITERATURE REVIEW

2.1 Introduction

The issue of phishing detection has gained much research coverage since the cyberattacks are becoming more advanced and dynamic and are now directed at emails, URLs, as well as online resources. Researchers have over the years experimented with an extensive variety of machine learning, deep learning and hybrid techniques to differentiate between phishing scams and legitimate messages. Such techniques differ with regard to a feature representation, learning, scalability, interpretability, and applicability in real time. In this section, the systematic review of available phishing detection methods, such as conventional machine learning algorithms, ensemble models, neural networks, natural language processing, and deep learning architecture, like CNN, LSTM, BiLSTM, or transformer-based models, are provided. Their strengths, limitations, datasets and metrics of performance will be reviewed critically with the intent of establishing research trends, gaps, and opportunities to develop more solid and versatile phishing detection systems.

2.2 Support Vector Machine

Since SVM may employ kernels to translate the features into higher-dimensional space where the data are separable by hyperplane, it is particularly helpful when the data does not lie on a hyperplane. Numerous research using SVM models have demonstrated the accuracy of

these models in categorizing intricate phishing datasets with overlapping patterns. Nevertheless, it becomes very slow when dealing with large data or selecting the appropriate kernel functions, which reduces scalability to true real-time detections [6].The model used phishTank or public dataset with an accuracy of 95.6% which showed strong classification performance which required careful feature tuning. It worked best with TF-IDF textual features and demonstrated robustness in binary classification tasks [6].

In order to assess distinct features extracted from the dataset, the SVM algorithm creates a hyperplane that generates multiple classifications. Any number of vector dimensions can be used with SVM. The method would be a line in two dimensions. It would be considered a hyperplane in three dimensions [7].

The author [7], identified the spam when features size is small with' a good generalization irrespective of where the size is. The researcher used Spam Assassin and phishTank dataset with an accuracy of 93% which demonstrated high classification accuracy and outperformed KNN and Naïve Bayes. However, it was found to be computationally intensive and less interpretable than decision tresses. Feature selection played a crucial role in optimizing performance.

A technique for identifying spam in online social networks is presented by [8] Combining spam messages from one social network to another is the main emphasis of their job. They collected 10938 ham and 1836 spam tweets from Twitter for processing. In addition, they used 9275 ham posts and 1328 spam posts. In TSD, 23.4% of tweets contained different terms, while 75.6% of tweets featured URL URLs for spam tweets. Of the 10941 ham tweets, 36.1% had just words and 62.9% featured both words and URL links. The remaining 67.2% of FSD spam postings are made up entirely of text, while 31.8% of messages include various web links [8]. Web links make up 95.1% of the 9275 ham posts, while words make up the remaining 4.9%. They made use of the top twenty feature terms from the spam data on Twitter and Facebook. They separate the training dataset and the testing dataset from the TSD and FSD. The author [9] reported a fast and accurate phishing detection method that integrated Naïve Bays (NB) and Support Vector Machine (SVM) using URL and webpage content data. NB was used for web page detection. However, if the websites were not well-discovered and continue to be questionable, SVM was utilized to reclassify them. The training set consisting of 100 authentic and 100 phishing websites, while the remaining 600 phishing websites serving as testing data. Phish Tank was used to build the dataset. According to experimental findings, the recommended approach achieved a high detection accuracy and a short detection time.

The author [10] worked with the hybrid algorithms which comprised of SVM, KNN and logistic Regression algorithms achieving the accuracy of 98.0% using Alexa and PhishTank dataset but the model was able to support a subset of instances of 3502 legitimate out of 35390 and 3655 phishing out of 36175 which reduced detection reliability in large scale environment compromising the security.

2.3 Decision Tree (DT)

Decision tree is a commonly used ML algorithm that can be applied for regression and classification. A recursive partitioning algorithm is applied to test the availability of attributes or features considering specific purity indices [11].The Gini Index and Entropy are the most commonly used indices, with the former applied to measure the probability that a randomly chosen feature will be misclassified indexes, where the former is applied to measure the probability of a randomly chosen feature that is incorrectly classified [11].The degree of uncertainty proportional to the information gain is called entropy amount that is proportional to the information gain is referred to as Entropy [11]. By means of these indexes, the required position of the entities, whether an internal node or a root, can be determined features [11].

The work of [7] with the binomial classification of spam and ham emails, DT has been applied in the tier three level. The model could identify spam in real time. For this feature, DT offers valuable insights since it has a straightforward computational process, which is necessary for effective real-time computing needs. The algorithm has been frequently used for easier explanations and visualizations. The author [7], on his research was mostly applicable in detecting of patterns of repetitive keywords in spam based on the structure carbon copy (Cc) or Blind Carbon Copy (Bcc), domains and header. The researcher used UCI based or custom phishing email dataset where the model efficiently relied heavily on feature selection which is less interpretable than other models with an accuracy of 96% accuracy detection.

The author [12] had created a smart model of phishing sites which was identified by forest technique, a combination of forests of the decision trees. It was evaluated using ROC curve, accuracy and f-measure [12] Models based on the k-NN, SVM, ANN, Rotation Forest, C4.5, CART, and NB algorithms, which can be applied as single classifiers in ensemble approaches were compared

to the constructed approach. As anticipated, the random forest model gave the best model with an accuracy of 97.35, f-measure of 0.974 and AUC value of 0.996. The avoided study has had some limitations in that it compared the random forest to individual classifiers e.g., KNN, SVM, and ANN, which among the models used generate ineffective models when compared with the simple random forest model [12]. Due to these difficulties, a new approach to classify phishing websites was proposed by the researcher [12], the Phishing Websites Classification Using Association Classification (PWCAC) that uses an association rule to perform a genuine or phishing classification of a web site.

In the work of [13] proposed GADT, a unique hybrid machine learning technique that combines genetic algorithms with decision trees, for the detection of spam emails. It is believable that the performance of decision trees for text classification can be enhanced with genetic algorithms in a precise and efficient manner. The best value for a parameter called the confidence factor, which regulates the decision tree's pruning, is found using a genetic algorithm [13].A significant issue with any text classification application, such as spam detection, is the abundance of features that reduce the classifiers' accuracy.

Decision tree is suitable for simple, structured phishing detection tasks which is limited in manual feature engineering and it has poor adaptability thus resulting to a traditional method which cannot solve most of the attacks that are happening currently on emails.

In the work of [8] defined that decision tree algorithm used optimal phishing website detection with the main goal of improving classification of phishing website as legitimate or phished website .The authors conducted the study using the publicly available dataset from UCI machine learning repository which comprised of 4698 phishing websites and 6157 legitimate websites .The study obtained 98.80% accuracy with a feature selection strategy

2.4 K-Nearest Neighbor (KNN)

The K-Nearest Neighbors (KNN) algorithm is a non-parametric estimation algorithm that is in-stance-based, implying that it can effectively work when the input version is noisy. The KNN also classifies the new data points according to their closeness to the already identified labeled samples in feature space and therefore it is very intuitive as well as flexible in classification as well as regression. This property enables the algorithm to give discrete classes results as well as continuous regressions of the results based on the needs of the applications. Nevertheless, even with these benefits, the method is not a main algorithm in large-scale studies, because of a number of inherent limitations, the most obvious being its extreme sensitivity to outliers in the data set, and its high cost of computation in high-dimensional data [7].

These limitations are of great importance in the context of phishing detection. Real-world email corpus or URL repository phishing datasets are usually noisy, unbalanced, and have mislabeled samples. The user-reported phishing samples can produce noisy data, in which inconsistencies in labeling or failures to extract a feature will create uncertainty in the dataset. In spite of this, KNN has been observed to have some degree of resilience to random noise in that the local decision boundaries that are created by it are driven by the density of the neighborhood, and not by global assumptions. This local decision making model makes KNN to be resistant to small changes in data distribution especially when a sufficient value of k (number of nearest neighbors used) is taken.

The author [14] found that KNN on top of deep learning-generated feature representations greatly enhanced the accuracy of phishing detection and particularly in the difficult “edge cases a traditional model may fail. In particular, they were able to show that feature-based KNN on top of a hybrid Convolutional Neural Network-Bidirectional Long Short-Term Memory (CNN-BiLSTM) architecture yielded more context-based classifications. CNN layers were effective in capturing spatial and local features in email or URL patterns, whereas BiLSTM layers learned sequential dependencies, and thus, the model was able to better represent both temporal and linguistic dependencies. The issuance of deep feature embeddings combined with the similarity principle of KNN, which relies on a distance, enabled making fine-grained classification choices, especially when phishing features were not explicit or clear.

In the work of [14] claimed, the hybrid method was experimented on a combination of UCI datasets and synthetic phishing data, the detection accuracy was around 87%. Nevertheless, they also pointed out that the system did not handle the outliers and unequal distribution of data too well- problems that the KNN algorithm is known to have. Since KNN involves direct use of the training data to classify (using all the samples in memory), this can cause distortion in the neighborhood structure in the presence of outliers. This leads to wrong distance measurements and misclassification especially where the dominant majority class prevails. This disparity is a continuous problem in phishing email detection because there are very many legitimate emails and very few phishing email messages, which tend to bias a prediction

in the majority group.

2.4.1 Strengths and Suitability of KNN to Phishing Detection

The main advantage that KNN possesses is its simplicity and interpretability. Contrary to the complex deep learning models, which need a lot of training and hyper-parameter optimization, KNN is a lazy learner, i.e. it does not construct a model in the training phase, but waits to compute when making a classification [14]. This makes it possible to adapt to new data quickly and update easily in cases when new samples of phishing are available. The algorithm is non-parametric in nature, i.e. it does not assume anything about the underlying data distribution and thus is especially effective in detecting phishing, and data might not be normally or linearly distributed.

Moreover, KNN inherently learns local features in the data, which is useful in phishing tasks where small lexical, syntactic and style differences are useful in distinguishing between legitimate and malicious emails. As an example, URL length, the frequency of special characters, domain entropy, and the existence of misleading tokens can be widely different in legitimate domains with respect to phishing detection based on URLs. The similarity-based metrics (e.g., Euclidean, cosine distance) available in KNN allow the latter to cluster such samples well based on local relationships between features than overall trends.

KNN can also be effectively used in high-dimensional feature spaces of textual phishing problems with feature engineering or dimensionality reduction methods like the Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE). In addition, ensemble methods, which consist of integrating KNN with other machine learning algorithms, e.g. Random Forests, Support Vector Machines (SVMs), or Gradient Boosting can be employed to increase robustness and classification stability. These hybrid systems can take advantage of local generalization capabilities of KNN but use the global predictive abilities of ensemble models.

2.5 Random Forest

According to author [15], using a small number of machine learning algorithms without knowledge of the hyper-parameter configuration or comparison with any previous results, the random forest classifier was able to achieve an accuracy of 97% on the data gathered from 11504 URLs on Kaggle.

The author [15] employed the UCI machine learning repository of 11,055 URLs, which included 6157 phishing URLs and 4698 legitimate cases. They were able to reach an accuracy rate of 97.35%, 97.43%, and 97.24%, respectively, after three tests. Due to UCI's open nature and lack of normalized features, which exclude the original URLs, the study did not employ several datasets to assess the model. KNN, Decision tree and SVM are simpler and more useful in controlled scenarios which rely heavily on manual features extraction, lack adaptability which struggles in real time phishing detection .This can only be overcome by CNN-BiLSTM which its able to interact with structural and semantic features for emails and real time detection which makes it more effective for modern phishing threats detections [16].

2.6 Neural Networks

The principle of neural networks proposed by [6] is that neural networks are built around the interrelation of linked artificial neurons that are organized into layers that process the input data with weightings and activation functions. The different layers in the network store increasingly abstract representation of the data, with the simple lexical features of the data being captured by the lower layers of the network, to the complex contextual and semantic features that are captured in the deeper layers of the network. Applied to the phishing detection, the mechanism allows the network to learn specific nuances in the textual content, URLs, metadata or even the stylistic approach of the fraudulent emails, which can also be the signs of the phishing attack. Such networks can be trained by showing them large amounts of labeled data, i.e. phishing and legitimate web pages or email that the network adapts its own internal settings to reduce the rate of error in making predictions. With repeated repetitions, the model learns a strong perception of trends in the email subject lines or email header or hyperlinks or HTML on a consistent pattern that are often linked to phishing and as such, it becomes very accurate when distinguishing between safe and malicious emails.

Specifically, the author [6] drew attention to the idea that neural networks can be trained to understand phishing emails on the basis of concealed associations between textual indicators, embedded links, sender metadata, as well as structural features/elements that the human analyst would not detect at a glance. This is enabled by the fact that deep learning architectures are an effective learning model in terms of feature representation learning- an operation upon which a network learns the best possible set of features that optimize the network in terms of classification. Traditional systems have the feature of features developed by hand by security analysts or data scientists through domain knowledge, like length of URL,

and number of special characters or the age of the domain. But in the case of neural networks these discriminating features do not require this kind of manual intervention because the model automatically learns them. This is a significant paradigm shift of cybersecurity analytics. The research by [15] took this research direction and adopted the neural networks as a deep learning to detect phishing URLs. Their paper used the GitHub data and had a very high accuracy of 96.60 percent- a sign of the enhanced generalization capability of the neural models. In contrast to the conventional Whois command that finds the domain registries and is frequently both slow and incomplete, the deep neural network may identify the connection between the URL segments, subdomains, and lexical formations at a considerably quicker rate, with the predictive accuracy being quite high. Not only is this accuracy computationally efficient, but it is also vital to real-time phishing prevention systems running in browsers and email clients, the time it takes a user a few milliseconds can decide whether they are a victim of an attack or not.

One of the most innovative works in this area was done by [17], who introduced their own self-organizing neural network that was specifically created to detect phishing websites. Their design showed that neural networks do not work in fixed neural hardware structures but can re-organize themselves on the fly to enhance learning results. The researchers proved the scalability and flexibility of their model by applying 17 attributes based on 600 legitimate and 800 phishing websites which were obtained through the Phish-Tank and Miller Smiles archives. Most of these features were founded on external indicators of services like domain validity, the state of an SSL certificate and content based features. It was found that the self-structuring neural network, besides being highly accurate, also had a high level of generalization, i.e. it was able to classify phishing sites with high accuracy that it had never encountered before or that belonged to other domains not in the training set. Such generalizability is a peculiarity of deep learning systems because phishing attacks are often based on novelty, even minor modifications to the URL, content phrasing, or visual representation will suffice to mislead rule-based or classic ML classifiers. Neural networks on the other hand memorize the underlying representational structure which is persistent even when the surface-level information changes and hence they are immune to adversarial effects.

2.7 Fuzzy decision tree and Naïve Bayes

The author [8] offered an alternative method for spam detection which comprised the combinations of the two algorithms. To identify trends in spam behavior, they employed the baking voting algorithm because the real world lacks observable traits. The level of cross-linking used to describe or explain characteristics is neutral and logical. To distinguish between ham and spam emails, decision trees employed fuzzy Mamdani rules. Next, they apply the Naïve Bayes classifier to the dataset. Finally, votes are divided into smaller portions and the baking procedure is applied. This method provided them with an optimum weight that can be applied to the per-centages that are collected in order to attain a higher degree of accuracy. 650 (65%) of the 1000 emails in the sample utilized in this study were ham, and 339 (34%) were spam [8].

A supervised machine learning-based email categorization method for Internet of Things systems was presented by author [8].They employed a Multiview approach that emphasized gathering more de- tailed data for categorization. Internal and external feature sets were combined to form a double view dataset. The suggested method was tested on two datasets with an actual network environment and may be applied to both labeled and unlabeled data. The study's findings suggest that the Multiview model outperforms simple email classification in terms of accuracy. Ultimately, the Multiview model was contrasted with other models that already exist identified by author [8].

The Neuro-Fuzzy Scheme, which combines fuzzy logic and neural networks, was used in this work in place of a stand-alone fuzzy system. This integration makes it possible to use both numerical and language features. This scheme's primary contribution was the extraction of 278 features from five inputs (Legitimate site regulations, User-behavior profile, PhishTank, User-specific sites, and Pop-Ups from emails) that weren't employed in tandem on a single system platform. Although neural networks are good at handling raw data, fuzzy logic uses linguistic and numerical features to have a high degree of reasoning [9].

According to the researcher [9] the use of neuro-Fuzzy scheme was chosen because of its capacity to generate linguistic rules from a fuzzy perspective and learn data from a neural network point of view. Using 2-Fold cross-validation, the experiment evaluated 278 characteristics, yielding an accuracy of 98.5%.

2.8 Natural Language Processing

The author [11] Reviewed 100 research articles published over the period between 1906 and 1921 in accordance with predetermined criteria and consisting of 100 research articles. Features of the phishing email, the datasets and

resources utilized in phishing emails, assessment measures, and natural language processing (NLP)-machine learning (ML) algorithms and optimization strategies are now the core areas of research study in phishing email detection. In the work of [8] stated that a critical systematic literature review of natural language processing procedures based on detecting phishing emails does not exist. As the researcher has shown, it is needed to carry out further re-search to implement the deep learning method, such as CNN-based models and LSTM, in the investigation of phishing emails detection.

2.9 Long short term memory and Artificial Neural Networks

The author [18] provided substantial information on the dynamic nature of the relationship between conventional machine learning tools and new deep learning algorithms in the sphere of phishing detection, especially when the systems are subject to the use of URL-based to make predictions. Their study particularly tested the Long Short-Term Memory (LSTM) network as an abstract component of an overall model that aims at identifying phishing websites. The study was fueled by the growing sophistication of phishing attacks, over which malicious individuals are continuously changing the names of websites and domain hierarchies to avoid the traditional rule-based or fixed machine learning models. Adebowale et al. aimed to establish how sophisticated neural networks like LSTM can win over less sophisticated algorithms such as Random Forests (RF) that use handcrafted features extensively. The authors in their research compared an RNN-based model (with an LSTM core) with a Random Forest classifier, whereby a shared set of 14 lexical and statistical URL features were used. These attributes were carefully chosen to observe the underlying trends that can distinguish a legitimate site and a phishing one, thus providing a moderate measure between the traditional and deep learning paradigms.

Parameters used in their study in lexical and statistical aspects were; length of URL, number of subdomains, special characters in the URL e.g., @, -, and underscore, ratio between digits and letters in domain name, use of HTTPS, domain age, and entropy, among others. All these features are famous signs of the phishing motive. As an example, phishing URLs can be characterized by a tendency to be longer in length, with an abnormal number of subdomains, and with the use of deceptive brand names to trick internet users into thinking it is a trustworthy one. Similarly, characteristic elements, like the use of HTTPS and age of domain, are vital since phishing sites are typically temporary and can be uncertified with regard to the use of the Secure Socket layer. Through the analysis of these 14 features, Adebowale et al. hoped that they would be able to create a complete report of the structural and lexical composition of URLs. In this way, they could easily contrast the performance of manual feature engineering (as in Random Forests) with the automatic feature extraction features of deep learning models such as LSTM.

Random Forest (RF) models were used as the machine learning baseline in their experiments. Random Forests is an ensemble learning algorithm, which builds many decision trees throughout the training process and returns the mode of the classes (in classification tasks). They are characterized by their strength and capacity to accommodate non-linear feature relationships and are a popular option when it comes to the phishing detection using manually engineered features. The 14 handcrafted features used in the RF model by Adebowale et al. were the input features, which are interpreted easily and readily computed. Nevertheless, RF is not without its drawbacks, in larger scale or feature constrained systems, RF has an inherent weakness in that it cannot dynamically learn new representations based on new data without first re-engineering its features. RF models are constantly brought up to date, or their feature set re-defined, as phishing strategies may develop (with obfuscation, or homograph attacks, or URL shortening). This weakness highlights the main rationale of studying LSTM-based architectures in detecting phishing, which can learn to represent data autonomously with time.

The author [9] used the Long Short-Term Memory (LSTM) model where the input was the URL strings in the form of sequences and the model used the contextual dependencies among characters and tokens. There is also a difference between LSTMs and Random Forests in that the latter relies on features that are fixed and static, whereas the former uses sequences of characters per URL and learns temporal relationships between them. This methodology enables the model to retain pat-terns including repeated brand-name insertions, misleading word combinations or manipulation at domain levels that change with time. Indicatively, phishing websites usually replicate trusted brand names (e.g., www.paypal.verify-login.com) -a.characteristic LSTM model was able to learn through positional dependencies and character distributions that are out of place when compared to real URLs. The gating process in LSTM, which consists of an input, forget and output gates, assists in memorizing essential sequence data and forgetting unimportant data; which avoids vanishing gradient problems inherent to standard re-current neural networks (RNNs). This renders

LSTMs very appropriate in sequence based phishing detection whereby subtle contextual information is used to tell whether a URL is malicious or not.

2.10 Convolutional Neural Networks and Bi-directional short term memory

Email phishing has also undergone a long overdue improvement in the recent past with the discovery of deep learning, that is, Deep Learning networks that have adopted the use of both Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) network. CNNs have a high performance in identifying local details in text, including n-grams and patterns that are too common in phishing e-mails (i.e., click here or verify account).

CNN is most effective in the analysis of multiple levels of emails header and body whose levels also encompass the character and words levels and this renders it more effective in detecting email phishing [19].

In the work of [20]CNN have been used in learning text embedding from individual characters which has contributed to an efficient in phishing email detection because it's capable of identifying sub-word structure, unusual punctuations or deliberate misspelling that attackers may use to evade traditional filters. The researcher contribution was vital in or more robust in obfuscation techniques which mostly includes inserting random symbols ,mixing uppercase and lowercase letters, or using visually similar characters from other alphabets which is more common with current phishing attacks on emails (Maneriker et al. 1920).

In the work of an author [21] transformer model was best suited in semantic comprehension and multi-context integration which requires a high cost and encounters token length constraints thus in phishing detection ,CNN can outperform transformers in highly obfuscated ,character manipulated dataset but transformer win most on datasets where nuanced semantic understanding is crucial. The re-searcher argued that its best to use CNN since it utilizes a low cost computation and in short text it's more effective to analyze which results to fewer token restrictions following their extraction, these features are fed into BiLSTM networks, which improve semantic understanding by capturing the words' contextual links and sequential dependencies in both forward and backward directions. The author [22] showed how effective Text-CNN is in detecting phishing indicators from email content, exhibiting a high degree of accuracy in identifying patterns of fraudulent language.

In a similar vein, the author [23] emphasized the potential of sequence-based models such as LSTM for phishing attack detection utilizing text and email information. The author [24] expanded on this by putting out a hybrid CNN-BiLSTM model that combined the advantages of both architectures, beating standalone models in terms of accuracy and resilience across a number of phishing datasets. This model enhanced the detection of sophisticated phishing techniques by utilizing CNN for initial feature extraction and BiLSTM for sequence modeling. The researcher used phishTank, Spam Assassin dataset to detection phishing attacks in emails that showed strong performance on both textual and URL based phishing detection of emails threats with 97.8% accuracy.

The author [25] used CNN in spam detection because of its strong feature extraction where it combined tweet text with meta led to a high accuracy of 99.31 %,precision level of 99.45% and F1-score of 99.68% but during the detection it declined its performance when textual data is used.

In the work of [26] enhanced the CNN with Word2vec embedding sina weibo dataset which enabled the model to achieve 91.35% accuracy but after running for sometimes it posed a model complexity challenge. The author [27] combined CNN text and image data which enabled the model to achieve 98.11% accuracy which proved that the model is adaptable across use of different input types.

According to [28] used a hybrid of CNN with BI-LSTM and word2vec which obtained an ac-curacy of 94.56% which depicted that the model was able to detect phishing detection in emails.

The hypothesis by [29] about how the phishing email detection model could improve phishing email detection is that they could collect the features present in the body of the email through text analysis and machine learning and deep learning to improve phishing email finding. Supervised learning model the model was developed on a GCN (convolutional network). The publicly used dataset on fraud where both fraud and genuine emails were available was used to train and test the algorithm. The quality and the format of the dataset were appropriate to apply in supervised learning techniques, and the collected data were balanced. The results of the testing showed that, the accuracy of the proposed model in identifying phishing email messages was 98 percent and the false-positive rate 0.015 percent. The research question as stated by [30] involved the pro-posed effective deep learning model adapted to the processing and classification of documents at document level. The researcher prosed the CNN-

BiLSTM model of Document level sentiment analysis using the work the Doc2vec word embedding whereby the model was made to test against the CNN model, LSTM model, BiLSTM or the CNN-LSTM model and experimented to show that the model (CNN-BiLSTM) was a better sentiment analysis model than the other models capable of classifying the French press articles to 90.66 percent accuracy.

2.11 Bi-LSTM with self-Attention and transformers

The bidirectional LSTM models with self-attention processes have been highly efficient in con-text information retrieval, such as text, in spam identification. The author [31] used self-attention Bi-LSTM and ALBERT to work with Twitter and Weibo datasets, they succeeded in achieving 91 percent of the accuracy rate and 90 percent of the F1-score. [28] demonstrated a very good performance with F1-score of 95.2 per cent and accuracy of 95 using Bi-LSTM combining CNN and word2Vec algorithms.

3. Metrics for email URLs Phishing detection Techniques

The technical solution to the problem of phishing attack in cybersecurity is the high-tech creation of an email URL phishing detection model, which refers to a hybrid approach of using Convolutional Neural Networks (CNN) and Bi-directional Long Short-Term Memory (Bi-LSTM) algorithm. CNNs in this model will deal with the retrieval of useful information as URLs in emails, capability to understand when suspicious characters, domains, and structuring of the URL re-source, which is frequently linked with phishing are present. No wonder, CNN models were also effective when it comes to detecting phishing emails based on the contents of the emails, and in this instance, it resulted in 98 percent accuracy [29]. The proposed model generates a probability that expresses the achievement of the likelihood that the email is malicious in reference to an input of a specified text that has been embedded on the email body. CNNs perform well with local n-gram retrieval and capture patterns, therefore they are significant to the detection of obfuscated phishing words and URLs [32]. More to the point, Bi-LSTM networks are more capable of presenting the sense of information flow and long-term relationships, which allows them to be more contextual when interpreting emails [33].

In order to detect threats, a number of studies coupled CNN and LSTM using Alexa and phish tank datasets. These studies achieved a 98.61% success rate with the available genuine and phished URLs [34].Working with 73,575 URLs from GitHub, [35] found 99.67% accuracy using CNN. However, because CNN was created for desktop browsers, it faced the difficulty of rapid updates to trust worthy domain lists.

The author [36] applied CNN to a huge dataset of 212,540 URLS, they achieved an accuracy of 88.90%. However, they encountered several difficulties because they did not employ hybrid algorithms and used fewer characteristics.

In the work of an author [37], When CNN and BiLSTM are combined together in detecting phishing detection in emails will allow the model to capture both fine-grained local patterns and global sequential context since CNN first extract important local features that will assist in detecting anomalies on an email.Bi-LSTM Techniques processes these features extracted in sequence to understand the temporal /contextual relationships.

3. METHODOLOGY

The literature review followed the PRISMA methodology, with the search limited to English-language publications within a defined publication period; grey literature and commercial security tools were excluded to ensure reliance on peer-reviewed academic sources.

4. RESULTS

The following table shows the findings from the 28 selected journals out of 39 journals from the google scholar. The researcher analyzed the various algorithms from the selected journals as distributed below in Table 4.1 Selected Journals

Selected Journals

Author & Title of Journal & Year	Technique / Algorithm Used	Dataset	Strengths	Weaknesses Security Loophole	Accuracy (%)
[6]	Support Vector Machine	PhishTank, Public Dataset	Good generalization, effective with high-dimensional data	Sensitive to noisy data	95.6
[7]	Support Vector Machine	Spam Assassin, PhishTank	Efficient classification	May not scale well with large datasets	93
[8]	Support Vector Machine	Twitter, Facebook	Handles unstructured social data	Social media data may have biases	-
[9]	SVM + Naïve Bayes	PhishTank	Combined strengths improve classification	Naïve Bayes assumes independence	-
[15]	SVM + KNN + Logistic Regression	Alexa, PhishTank	Ensemble increases accuracy and robustness	Computationally expensive	98
[7]	Decision Tree	UCI or custom phishing email dataset	Easy to interpret, fast	Overfitting with complex datasets	96
[12]	Random Forest (DT Ensemble)	UCI phishing dataset	High accuracy, handles overfitting	Slower than single trees	97.35
[12]	PWCAC (DT-based)	UCI phishing dataset	Adaptive decision mechanism	Might lack interpretability	-
[13]	GADT (DT + Genetic Algorithm)	Custom	Optimizes feature selection	Complexity in tuning genetic parameters	-
[14]	K-Nearest Neighbors	UCI + Synthetic	Simple, non-parametric	Slow with large datasets	87
[45]	Random Forest	Kaggle URLs	Effective with URL-based features	May not perform well on email body text	97
[45]	Neural Network	GitHub Dataset	Learns deep representations	Requires large data and time	96.6
[17]	Neural Network	PhishTank, Miller Smiles	Flexible for various data types	Overfitting if not regularized	-
[29]	Feedforward Neural Network	Kaggle URLs	Fast training with simple architecture	Limited memory of sequence data	93
[8]	Fuzzy DT + Naïve Bayes	Spam emails (1000)	Handles uncertainty in decision-making	Less scalable	-
[9]	Neuro-Fuzzy Scheme	Custom	Good in capturing uncertain and fuzzy features	Computational cost	98.5
[11]	NLP (Review)	Multiple	Comprehensive language understanding	Not specific to phishing classification	-
[19]	LSTM	Phishing URL dataset	Captures sequential patterns in URLs	Long training time	-
[10]	ANN/DNN	Custom	Learns complex patterns	Overfitting without enough data	-
[19]	CNN + LSTM	Phishing Websites	Combines spatial and temporal features	Computationally intensive	-
[43]	CNN	Custom	Effective with spatial patterns in text	Ignores sequence if not combined with RNN	95.97
[44]	CNN + LSTM + Attention	Phishing URLs	Focus on important features	Model complexity	98.25
[26]	CNN + Word2Vec	Sina Weibo	Embedding improves feature understanding	May struggle with sarcasm or informal text	91.35
[27]	CNN + Text + Image	Custom	Multimodal detection	Requires image preprocessing	98.11
[28]	CNN + BiLSTM + Word2Vec	Custom	Strong semantic and sequential feature capturing	Computationally expensive	94.56
[24]	CNN + BiLSTM	PhishTank, SpamAssassin	Handles both spatial and temporal features	Model size	97.8
[31]	BiLSTM + Self-Attention	Twitter, Weibo	Selective focus on important parts of sequence	Complex tuning	91
[28]	BiLSTM + CNN + Word2Vec	Custom	Balanced feature extraction	Resource intensive	95
[21], Applying Deep Learning for Detecting Phishing on Emails	CNN, SVM, LSTM, Bi-LSTM, CNN-BiLSTM	Email phishing dataset	Highly flexible in learning patterns, structures, and features. Hybrid CNN-BiLSTM achieved highest performance.	NLP showed poor results due to language limitations; not effective for dynamic email threats.	CNN-BiLSTM: 99.41% SVM: High NLP: 2%
[38], Website Phishing Detection	CNN, LSTM, LSTM-CNN	Web-based phishing dataset	CNN robust in extracting features; LSTM-CNN and LSTM models handle sequential data well.	May struggle with semantic complexity and advanced obfuscation.	CNN: 99.2% LSTM-CNN: 97.6% LSTM: 96.8%
[39], Malicious vs Benign URLs Detection	LSTM, Bi-LSTM	URL datasets (malicious and benign)	Bi-LSTM handles sequence data bidirectional; excellent at detecting contextual patterns in URLs.	Might be resource-intensive and sensitive to input structure.	LSTM: 97% Bi-LSTM: 99%
[42], Email Phishing Detection with Traditional and Transformer-based Models	Logistic Regression, XLNet, BERT	Email dataset	Transformer models (BERT, XLNet) had high accuracy.	Logistic Regression had low performance due to misclassification of grammar errors, leetspeak, and HTML content.	Logistic Regression: 2% BERT: 99.1% XLNet: 98.84%

Online: https://doi.org/ 10.64680/jisads.v3i2.55

Table 4.1 Selected Journals

Table 4.1 above shows the list of the selected journals from google scholar which were used in the analysis.

Techniques Distribution

Table 4.1 Techniques Distribution

Algorithm	Frequencies
CNN	7
SVM	6
LSTM	5
Decision Tree (DT)	4
BiLSTM	4
NN / ANN / DNN	4
Naïve Bayes	3
Word2Vec	3
Random Forest	2
K-Nearest Neighbor (KNN)	2
Self-Attention	2
Fuzzy Decision Tree	2
Logistic Regression	1
Transformer (ALBERT)	1
Text/Image Input	1
NLP	1

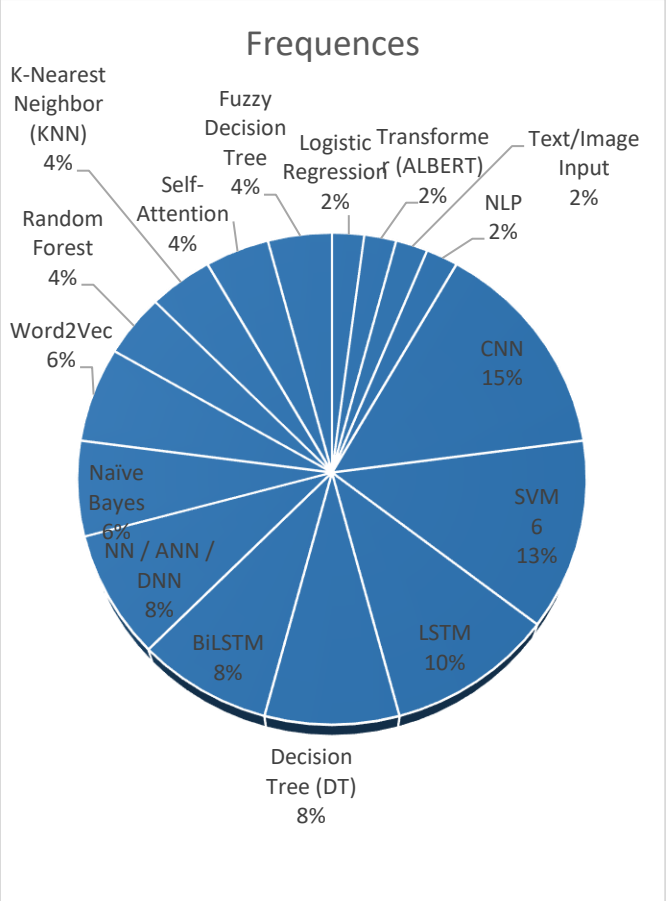

Figure 4.1 Algorithm Used

Table 4.1 above shows the frequency of the various algorithms used by different authors in phishing detection model for emails attacks using supervised learning.

5. DISCUSSION

According to Figure 4.1, the comparative assessment of the various machine learning and deep learning models used in detecting phishing email shows that there is evident variance in the performance, accuracy, and the ability to adapt to the complex data structures. The Convolutional Neural Network (CNN) became the most effective with a score of 15% in the assessment, the next model was Support Vector Machine (SVM) with 13%, and the Long Short-Term Memory (LSTM) network with 10%. The bottom of the ranking went to Natural Language Processing (NLP), Logistic Regression, and the CNN variant with the input of text and images, all with 2%. These performance gaps underscore the role of model structure, data encoding and learning processes towards algorithmic determination of detection capabilities especially in the separation of phishing emails and legitimate email messages. In his study of the issue of applying deep learning to phishing detection on email, the author[27] indicated that the models of deep learning including CNN, LSTM, and Bi-LSTM demonstrated the ability to effectively, specifically, and automatically learn the patterns, textual and structural features of emails, thus giving better detection results than other machine learning algorithms.

The excellence of CNNs in this research can be explained by the fact that CNNs are able to obtain local in the input data. Although CNNs were originally used to classify images, these algorithms have shown abnormal flexibility in text phishing detection, as they have convolutional layers, and these layers can effectively extract n-grams, word embedding's, and local patterns in the email text, subject lines, and URLs. CNNs can detect subtle differences and associations signifying phishing intent, in the form of abnormal lexical syntax, suspicious URL tokens, or even the existence of psychologically manipulative keywords by going through sequences of words or to-kens. Convolution and pooling have allowed CNNs to remove noise, highlight the high impact features and generalize well even when faced with previously unknown phishing email formats. Moreover, CNNs are characterized by shorter training durations than recurrent networks like LSTM, as convolutional filters could be run simultaneously hence CNNs are computationally efficient without losing high precision. This effectiveness might have contributed to the

highest score of CNN in this research at 15%, which reinforces the findings provided by [21], who concludes that CNNs are the most resilient among the deep learning-based classifiers in phishing email detection.

The second and the biggest classifier was the Support Vector Machine (SVM) with 13% and yet again, it showed a good performance and emphasized that it was still relevant as a classical machine learning framework to binary classification with a phishing and an authentic email. SVMs are also effective because they can form the best hyperplanes that maximize the distance between data points that are of different classes. Applied to phishing detection, SVMs do suc-cessfully classify structured representations of features based on email metadata, URLs, and content-based features like word frequency and character distribution. The performance degradation relative to CNN is however a hint that SVMs are good in separating non-linearly or linearly separable datasets by a series of kernel tricks, but they may not be able to handle the hierarchical and sequential nature of natural language data, even with their tricks. Such contextual information is inherently represented in deep learning models such as CNN and LSTM and requires extensive manual feature engineering and preprocessing in SVMs. However, its relatively high score suggests that the SVM can be used to achieve the same level of accuracy as deep learning models with reasonably good feature extraction strategies, e.g., TF-IDF, n-grams, or Word2Vec embeddings, so it can still be used when dealing with smaller datasets or resources-limited environments.

The Long Short-Term Memory (LSTM) model with the score of 10% performed better, but a little worse than CNN and SVM. LSTM networks are a variant of recurrent neural network (RNN) that has the ability to learn long-term dependencies in sequential data. They are designed with memory cell and gating procedures such that they can store contextual data in a long sequence and are therefore best suited to detect phishing activities which are text-based and in which the context of words in a sentence is important. An example is the use of the terms verify your account, update your password or urgent action required, which can be used in phishing, however, in official communication between companies, they might be harmless. LSTMs are able to differentiate between such situations by looking at the context in which such phrases are used. Nevertheless, the strengths come with several weaknesses, such as LSTMs being computationally heavy and subject to overfitting especially where training data is small or unbalanced. They also take longer to train than CNNs because of their sequential processing nature and that could be the reason why CNN was better than LSTM in this analysis. Nevertheless, according to [21], LSTMs and their time-aware counterpart (Bi-LSTM) are also useful to contextual semantics and time-related rela-tions that other models do not account for.

At the bottom of the performance table were Natural Language Processing (NLP)-based traditional models, Logistic Regression, and CNNs that were trained on mixed text/image data, these all scored 2%. The fact that standalone NLP models perform poorly might be due to having hand-crafted linguistic features like token frequencies, part-of-speech tags, and sentiment predictions that are not effective at capturing the multidimensional and dynamic trends of phishing emails. In contrast to deep learning models, traditional NLP methods do not learn representations, but rely on fixed rules or small statistical models that might not be effective at detecting novel phishing techniques. Another base model, Logistic Regression must have performed poorly due to the same reasons. Although it is efficient and interpretable, Logistic Regression is based on the assumption of a linear relationship between features and the target class which, in the case of phishing detection problems with nonlinear and high-dimensional data, is not often true. More-over, Logistic Regression is not very effective with text features in the form of sparse vectors, e.g. those created by TF-IDF or Bag-of-Words, which results in insufficient generalization when faced with a variety of email formats, or misleading linguistic signals.

The poor results of CNN models trained on text and image inputs (2%) are also intriguing results that should be further interpreted. Primarily, the multimodal methods, i.e., an analysis of both textual and visual information (e.g., logos, in-text imagery, or template of a brand) ought to be more successful in distinguishing between authentic and fake phishing sites because they offer more information on authenticity. Nonetheless, in reality, there are challenges that are brought about by this approach. To start with, image data used in phishing emails may differ in quality, resolution and encoding and in the process, CNNs are not always able to retrieve meaningful visual features. Secondly, the image and text representation fusions involve complicated network structures and even-balanced datasets so that either of the modalities does not overpower the learning. In case the text component already contains enough discriminative information, the addition of the image features can only introduce noise instead of enhancing accuracy. Hence, bad performance of CNN on text/image input might be explained by an imbalance in data, low image quality, or the lack of multimodal fusion strategy during training.

All these results support a key finding in the research of phishing emails detection: deep learning algorithms are

significantly better than traditional machine learning algorithms because of their capacity to learn and generalize on large-scale data without any explicit feature engineering. Specifically, CNN still reigns supreme in terms of the performance metrics since it gathers ac-curacy, computational efficiency, and noise resistance. This conclusion is also supported in the work of [21], which clearly states that deep learning architectures (particularly CNN, LSTM, and Bi-LSTM) are not only flexible, but they can also adapt to the constantly changing environment of phishing attacks. With the use of more advanced forms of social engineering, the more traditional models, which rely on common patterns or are based on rules, cannot keep up to the dynamism of deep neural networks, which are trained on new context and structure signals with each new piece of data.

The other dimension that can be addressed is the feature segmentation approach that is used in phishing detection. The CNNs, LSTMs, and other modern models are more effective in the case when the input data is separated into semantically meaningful segments: a subject line of the email, the email body, the email header, the email sender address, and email URLs. All these elements have their unique patterns that can be used to identify phishing attacks. To illustrate, the header can show discrepancies in the sender domain, and the URL can have obfuscated or misspelled brand names that are meant to mislead the users. These segmented inputs are well processed by CNNs with parallel feature extraction pipelines and resulting to a more holistic view of phishing intent. SVM and Logistic Regression models on the other hand, which require features to be engineered manually, might not be effective at capturing these multi-dimensional relationships. Accordingly, the performance ranking obtained illustrates the value of the architectural depth and feature representation in phishing email detection.

Another important factor that determines the applicability of model decisions to real-world cyber security context is their interpretability. Despite their superior accuracy, CNNs and LSTMs are commonly regarded as black-box models since it is not easy to interpret their inner decision-making processes. This makes the security analysts difficult because they can be requested to explain the results of classification, particularly in the corporate or legal environments. On the other hand, simpler models like the Logistic Regression and SVM are more open and under-standable, but less predictive. Consequently, a more efficient phishing detection scheme will be a hybrid one, when deep learning models are used as a first detection tool, and more classical algorithms are used as verification or explain ability tools, after the initial detection. Such combination could trade off precision with readability so that phishing identification systems could be both effective and reliable.

It is also important to mention that percentages of the performance that are reported in Figure 4.1 do not only indicate the raw capability of each algorithm but also the quality and the size of the dataset, the methods of feature extraction applied, and the measures of evaluation adopted. As an example, the performance of CNN can be significantly different between the text representation with or without word embeddings: Word2Vec, GloVe, or BERT. On the same note, SVM accuracy can also be influenced by the decision of the kernel function (linear, radial basis, or polynomial) and hyper parameter optimization. The comparatively close results of CNN and SVM (15% vs. 13%) indicate that both models were optimized successfully and that CNN had a marginal ad-vantage of having hierarchical learning abilities of the features. The low scores of the other models on the other hand might be because of lack of proper data preprocessing, imbalance of features or poor parameter tuning.

Overall, the analysis of Figure 1.3.1 and the literature references allow concluding that Convolutional Neural Networks (CNNs) are still the most appropriate model to use to detect phishing emails because they possess structural benefits in acquiring hierarchical patterns, processing large amounts of text, and being computationally efficient. SVMs are also still competitive when the data size is less or when features like interpretability are of importance. The Long Short-Term Memory (LSTM) networks are more effective in comprehending the contextual associations but can be outperformed by CNN because of the calculation cost. In the meantime, the traditional models such as Logistic Regression and NLP-based models are proving to be less useful in the contemporary phishing detection because of their inability to handle non-linear and dynamic data structures. Lastly, multimodal CNN techniques with text and images demonstrate a promising future but demand more advanced methods of data fusion to realize useful gains.

Generally, the trends shown in Figure 1 and supported by [21] indicate unequivocally that the major shift in phishing detection systems will be to deep learning-based systems with such models as CNN, LSTM, and Bi-LSTM becoming the basis of future innovation in defending against cybercrime. Going forward, since phishing methods are becoming more sophisticated, the ability of these architectures to learn features automatically, be flexible, and scalable will be instrumental in creating sturdier, smarter, and active email security systems that will protect their users against the new digital threats that continue to emerge.

Online: https://doi.org/ 10.64680/jisads.v3i2.55

The research conducted by [38] on phishing websites showed the following accuracy results 99.2% 97.6% and 96.8% concerning CNN, LSTM-CNN and LSTM. These outcomes add resilience to the performance of CNN in the extraction of Internet-based information and thus more attractive to phishing categorization. The relevance of sequence data modelling in phishing detection is emphasized by the result of the LSTM-CNN hybrid and LSTM models.

The author [39] Used malicious and benign URLs datasets and applied LSTM and Bi-LSTM in the research where they achieved an accuracy of 97% and 99.0%, respectively. LSTM is essential to the detection of phishing as it is designed to process sequential data and recognize long-term connections and as such, it is especially applicable to the assessment of character-level or token-level trends as evident in phishing hyperlinking. Bi-LSTM however takes this a notch higher as it uses both forward and backward processing of the data enabling the model to absorb context about both ends of a URL.

In deep learning the combination of CNN and Bi-LSTM contributed an accuracy of 99.41% and Bi-LSTM and SVM with an accuracy of 95% which contributed that with the use of hybrid algorithm of CNN and Bi-LSTM has the highest accuracy [21]. Support Vector Machines (SVMs) are commonly used in phishing assault detection because they excel at handling classification tasks, which is exactly what phishing detection entails. This is due to their ability to discriminate between harmful and trustworthy websites or emails.

Thus, the CNN and Bi-LSTM hybrid model is the most effective of the models that were examined, offering the best detection performance and making it a great option for real-world phishing prevention systems. Building on this, the goal is to blend CNN with Bidirectional Long Short-Term Memory (Bi-LSTM) to create a more effective CNN detection model. With this hybrid approach, the advantages of both architectures are combined: CNN is excellent at extracting local spatial features from email content, such as HTML patterns, embedded URLs, or suspicious phrases, while Bi-LSTM efficiently captures the contextual and sequential relationships in text data by processing input in both forward and backward directions. The algorithm is better able to comprehend phishing indications at the word and sentence levels. Along with methods like dropout, batch normalization, and Hyper parameter tuning to increase training efficiency and model generalization, the hybrid model also intend to incorporate a pre-trained word embedding layer to improve the semantic understanding of input text. The goal of this honest combination of CNN and Bi-LSTM is to produce a scalable and reliable phishing detection system that performs better than conventional models, especially when managing intricate, dishonest phishing efforts that change over time.

In the work of [21] , NLP is limited in its ability to be widely employed in detecting dangers in other areas of emails because it has been largely used to study language translation, mostly in Arabic text thus even the analysis done on the research it had the lowest accuracy of 2%. The study concluded that because NLP is limited, no further research has been conducted with it. This poor performance shows that NLP might not be a good technique to detect phishing threats in emails, particularly when interpreting contextual complex semantic patterns that evolves in real time. A small 2% accuracy highlights the necessity for hybrid techniques that go beyond language features, as [21] found. Future studies should concentrate on combining multimodal feature ex-traction and deep learning with natural language processing.

Logistic regression recorded accuracy results achieving 2 % according to analysis results which was the lowest accuracy of email phishing detection. The lowest accuracy of 58.77 as compared to XLNet and Bert with an accuracy of 98.08 and an F1 score of 0.9831, XLNet 0.9884 and BERT 0.9911 was observed, as this means that, logistic regression was not as effective in detecting anomalous phishing patterns in an email as it encountered numerous patterns of errors characterized by grammatical errors, mixed language content, and leetspeak which were classified as non-phishing as well as HTML code and Phishing URL.

The author [40] Compared BERT and Word2Vec where BERT failed to perform well when applied with feature extraction that were chosen via the Chi-square technique and did not show any such positive results on phishing emails. BERT yielded an accuracy of 98.2 and Word2Vec achieved an accuracy of 98.8 consequently surpassing BERT technique which scored a percentage of 2 % according to the research thus it is not the most appropriate algorithm that can be used in email phishing detection.

The author [41] found that BERT being transformer algorithm offers a strong representations learning for phishing but it faces practical limitations of high resource consumption, dataset imbalance, latency, adversarial sensitivity and interpretability limitations which is complicated to use and incur a lot of cost which can be overcome by using a lighter hybrid model (CNN-BiLSTM) techniques for phishing detection which has shown a great impact on the anal-lysis done.

Online: https://doi.org/ 10.64680/jisads.v3i2.55

6. CONCLUSION

According to the study's findings, the CNN and Bi-LSTM hybrid model was the most successful in detecting phishing emails because it outperformed other methods and had the highest accuracy (99.41%). Because CNN and LSTM-based models could handle sequential input and extract features, they proved to be reliable. SVM did well on classification tests as well. However, be-cause of their shortcomings in managing intricate email patterns, techniques like NLP, logistic regression, and BERT demonstrated low accuracy (around 2%). These results demonstrate the superiority of hybrid deep learning models and the necessity of more sophisticated methods in next studies on phishing detection.

7. FUTURE STUDY

Future studies ought to be done to investigate hybrid and ensemble phishing detection models that combine deep learning and existing machine learning methods systematically because current research shows performance improvement, although no standard comparison has been done. Multimodal methods involving the integration of email text, URLs, email headers, and visual characteristics should be further examined because these are not currently studied thoroughly and not consistently tested on a dataset-wide basis. Also, research that evaluates the model generalizability, scalability and robustness to changing and zero-day phishing attacks based on a variety of and real-world data sets are needed. Lastly, the systematic reviews of the future ought to focus on explain ability, computational efficiency and benchmarking practices to facilitate deployment of phishing detectors in the real world and their reproducibility.

REFERENCES

[1] Basit, A., Zafar, M., Javed, A. R., & Jalil, Z. (2020). A novel ensemble machine learning method to detect phish-ing attack. In 2020 IEEE 22rd International Multitopic Conference (INMIC) (pp. 1-5). IEEE.d

[2] Zhu, E., Ju, Y., Chen, Z., Liu, F., & Fang, X. (2020). DTOF-ANN: An Artificial Neural Network phishing detec-tion model based on Decision Tree and Optimal Features. Applied Soft Computing, 95, and 106505. https://doi.org/10.1016/j.asoc.1919.106505

[3] Aljofey, A., Jiang, Q., Qu, Q., Huang, M., & Niyigena, J.-P. (2020). An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL. Electronics, 9(9), Article 9. https://doi.org/10.3290/electronics9091514

[4] Kim, Y., Lee, S., & Kim, H. (2020). A deep learning-based approach for detecting phishing URLs. Computers & Security, 89, 101674.

[5] Zieni, R., Massari, L., & Calzarossa, M. C. (2023). Phishing or not phishing? A survey on the detection of phish-ing websites. IEEE Access, 11, 9499-9518.

[6] Madhavaram, C., Konkimalla, S., Rajaram, S. K., Gollangi, H. K., & Reddy, M. (2023). AI/ML-Powered Phishing Detection: Building an Impenetrable Email Security System. 10–18.

[7] Raza, M., Jayasinghe, N. D., & Muslam, M. M. A. (2021). A Comprehensive Review on Email Spam Classifica-tion using Machine Learning Algorithms. 2021 International Conference on Information Networking (ICOIN), 316–321.

[8] Ahmed, D. S., Hussein, K. Q., & Allah, H. A. A. A. (2022). Phishing websites detection model based on decision tree algorithm and best feature selection method. Turkish Journal of Computer and Mathematics Educa-tion, 13(1), 100-107.

[9] Zuraiq, A. A., & Alkasassbeh, M. (2019). Review: Phishing Detection Approaches. 2019 2nd International Con-ference on New Trends in Computing Sciences (ICTCS), 1–6.

[10] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing de- tection from URLs. Expert Systems with Applications, 117, 334–345.

[11] Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2022). A systematic literature review on phishing email detec-tion using natural language processing techniques. Ieee Access, 10, 65703-65726.

[12] Alsariera, Y. A., Adeyemo, V. E., Balogun, A. O., & Alazzawi, A. K. (2020). AI Meta-Learners and Extra-Trees Algorithm for the Detection of Phishing Websites. IEEE Access, 8, 141381–141421. IEEE Access. https://doi.org/10.1109/ACCESS.1919.2913599

[13] Taloba, A. I., & Ismail, S. S. I. (2019). An Intelligent Hybrid Technique of Decision Tree and Genetic Algorithm for E-Mail Spam Detection. 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), 99–104.

[14] Slam, M. R., Abawajy, J. H., & Watters, P. A. (2022). A hybrid deep learning and KNN model for phishing detec-tion. Journal of Information Security and Applications.

[15] Singh, R., Kumar, R., & Singla, R. K. (2020). A heuristic-based phishing detection approach using machine learning techniques. International Journal of Information Technology, 12(3), 685–690.

[16] Bu, S. J., & Cho, S. B. (2021). Deep character-level anomaly detection based on a convolutional auto encoder for zero-day phishing URL detection. Electronics, 10(12), 1492.

[17] Zabihimayvan, M., & Doran, D. (2019). Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detec-tion. 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 1–6. https://doi.org/10.1109/FUZZ-IEEE.1918.8858884

[18] Adebowale, M. A., Lwin, K. T., & Hossain, M. A. (2023). Intelligent phishing detection scheme using deep learning algorithms. Journal of Enterprise Information Management, 35(3), 745-766.

[19] Singh, R., Kumar, R., & Singla, R. K. (2020). A heuristic-based phishing detection approach using machine learning techniques. International Journal of Information Technology, 12(3), 685–690.

[20] Maneriker, P., Stokes, J. W., Lazo, E. G., Carutasu, D., Tajaddodianfar, F., & Gururajan, A. (2021, November). Urltran: Improving phishing url detection using transformers. In MILCOM 2021-2021 IEEE Military Commu-nications Conference (MILCOM) (pp. 187-194). IEEE.

[21] Kyaw, P. H., Gutierrez, J., & Ghobakhlou, A. (2024). A systematic review of deep learning techniques for phishing email detection. Electronics, 13(18), 3722.

[22] Zhang, Y., & Lee, D. (2019). Text-CNN for phishing email detection. IEEE Access, 7, 172832–172840.

[23] Bahnsen, A. C., et al. (2018). Detecting phishing emails using supervised learning algorithms. In 2018 IEEE International Conference on Data Science and Advanced Analytics (DSAA).

[24] Wang, D., et al. (2021). A hybrid deep learning model for phishing email detection. Computers, Materials & Continua, 67(3), 3343–3362.

[25] Alom, M., Carminati, B., & Ferrari, E. (2020). Spam detection in Twitter using Convolutional Neural Net-works and metadata fusion. Journal of Information Security and Applications, 54, 102418.

[26] Feng, Q., Zhou, Y., Fan, J., & Wang, W. (2018). Detecting spam URLs in social media via convolutional neu-ral networks and word embeddings. IEEE Access, 6, 13758–13767.

[27] Seth, J., & Biswas, A. (2018). Multimodal deep learning approach for image and text based spam detection. In 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (pp. 1–4). IEEE.

[28] Shahariar, M. S., Ahmed, M., & Nur, M. M. (2019). Spam detection in reviews using deep learning. In 2019 International Conference on Bangla Speech and Language Processing (ICBSLP) (pp. 1–5). IEEE.

[29] Atawneh, S., & Aljehani, H. (2023). Phishing Email Detection Model Using Deep Learning. Electronics,12(19),Article19.https://doi.org/10.3290/electronics12094151

[30] Rhanoui, M., Mikram, M., Yousfi, S., & Barzali, S. (2019). A CNN-BiLSTM model for document-level senti-ment analysis. Machine Learning and Knowledge Extraction, 1(3), 831-845.

[31] Xu, H., Zhou, Y., & Liu, Q. (2021). Semantic spam detection in microblogs using a self-attention BiLSTM model with ALBERT embeddings. Journal of Intelligent & Fuzzy Systems, 40(2), 3345–3357.

[32] Saxe, J., & Berlin, K. (2018). Deep neural network-based malware detection using two- dimensional binary program features. In MALWARE.

[33] Phan, T. Q., Bui, H. D., & Nguyen, T. T. (2020). Phishing URL detection using deep learning techniques. Pro-ceedings of the International Conference on Artificial Intelligence.

[34] Sirigineedi, P., Bhattacharya, P., & Chakraborty, R. S. (2020). PhishAri: A hybrid model for phishing website detection using CNN and LSTM. Procedia Computer Science, 167, 1046–1057.

[35] Wei, F., Zeng, Y., Yang, Z., & Ma, H. (2020). Phishing detection based on deep learning and visual similarity. IEEE Access, 8, 221412–221382.

[36] Korkmaz, S., Arslan, M., & Doğru, İ. A. (2021). Detecting phishing websites using deep learning and convolu-tional neural networks. Neural Computing and Applications, 32, 11217–11227.

[37] Nanda, M., & Goel, S. (2024). URL based phishing attack detection using BiLSTM-gated highway attention block convolutional neural network. Multimedia Tools and Applications, 83(26), 69334-69365.

[38] Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q. E. U., Saleem, K., & Faheem, M. H. (2023). A deep learn-ing-based phishing detection system using CNN, LSTM, and LSTM-CNN. Electronics, 12(1), 222.

[39] Roy, S. S., Awad, A. I., Amare, L. A., Erkihun, M. T., & Anas, M. (2022). Multimodel phishing url detection using lstm, bidirectional lstm, and gru models. Future Internet, 14(11), 330.

[40] Thakur, K., Ali, M. L., Obaidat, M. A., & Kamruzzaman, A. (2023). A Systematic Review on Deep-Learning-Based Phishing Email Detection.

Electronics, 12(20), Article 20. https://doi.org/10.3290/electronics12104343

[41] Jamal, S., Wimmer, H., & Sarker, I. H. (2024). An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach. Security and Privacy, 7(5), e392.

[42] Meléndez, R., Ptaszynski, M., & Masui, F. (2025). Comparative Investigation of Traditional Machine-Learning Models and Transformer Models for Phishing Email Detection. Electronics, 13(23), 4677.

[43] Rathee, T., & Mann, P. S. (2022). Performance evaluation of deep learning models for phishing email detec-tion. Journal of Information and Knowledge Management, 20(04), 2150061.

[44] Peng, Y., Zhang, X., Xu, S., & Li, W. (2019).A malicious URL detection model based on CNN-LSTM with atten-tion mechanism.IEEE Access, 7, 144379–144388.

[45] Safi, S., & Singh, M. (2023).Malicious URL detection using machine learning techniques: A comparative analy-sis.International Journal of Advanced Computer Science and Applications, 14(1), 313–321.