LSTM-PSO: NLP-based model for detecting phishing attacks

Abdulrahman Ayad Alshdadi
Department of Information Systems and Technology,
College of Computer Science and Engineering,
University of Jeddah, Jeddah, Saudi Arabia
alshdadi@uj.edu.sa

Abstract

Detecting phishing attacks involves recognizing and stopping attempts to trick users into revealing information, like passwords, credit card details or personal data without authorization. While most recent related work focus on detecting phishing attacks by analyzing, URLs, email header and content and web pages based on their content, regardless of entering text sequentially into Deep Learning (DL) algorithms. This approach causes the intrinsic richness of the relationship between words and part of speech to be lost. This study main contribution is to detect phishing attacks by introducing an integrated model that emphasizes on analyzing the text content of suspicious web pages a model that detects not on URL addresses. The approach of the proposed model is based on using Natural Language Processing (NLP) for processing web-page content, Particle swarm optimization algorithm (PSO) for optimizing feature extraction process and Deep Learning (DL) algorithms for classifying web page content into phishing or legitimate. NLP techniques are used to preprocess web-page content and word2vector embeddings for Word Representation to extract and select best features into DL algorithm. Two different approaches Long Short-Term Memory (LSTM) are assessed: traditional LSTM and enhanced LSTM-PSO. The results show promising outcomes by the proposed model in detecting phishing attacks as both LSTM and LSTM-PSO achieved an accuracy of 97% and 98.3% respectively.

1 Introduction

Social engineering is a type of cyber-attack where the attacker manipulates or exploits people's behavior to deceive and scam them (Gupta and Singhal, 2017). Social engineering attacks can be carried though phishing attack which is a cyber-attack by hackers who pretend to be an entity or organization to trick people into sharing information like usernames, passwords, credit card numbers or personal details (Ali and Malebary, 2020). The attackers commonly employ methods such as sending emails, text messages, setting up deceiving websites or using social engineering strategies to deceive and exploit their targeted victims. By using strategies like urgency, authority, familiarity or offering rewards the attacker influences the victims to respond and interact with their offering. This could involve persuading them to click on links disclose data or download malicious software. Once successful the attacker on obtaining users credentials or banking information they use it in activities such as identity theft or financial scams (Alam et al., 2020b).

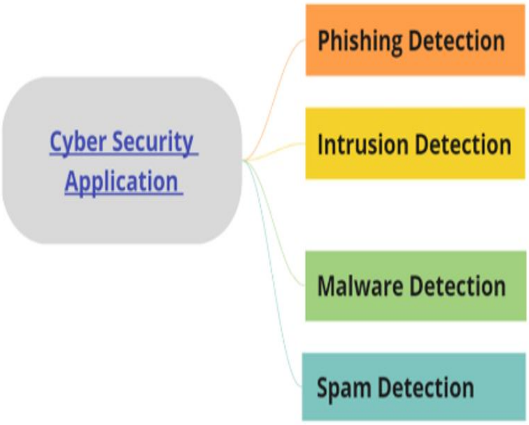

Figure 1: Deep learning application in cyber security

Protecting against phishing attacks requires being cautious confirming sender authenticity avoiding links or attachments updating passwords regularly and following cybersecurity practices (Radha Damodaram and Valarmathi, 2011). The main challenge in detecting phishing attacks lies in the need to effectively spot and prevent these attempts which can endanger individuals, businesses and communities. One of the technical strategies that are widely used nowadays in cyber security application domain as shown in Figure1 is phishing attack detection systems (A, 2022). Nowadays with the fast and vast spread of phishing attacks in cyber space, it is important to stay ahead of evolving phishing strategies as cyber attackers are continuously finding ways to avoid detection methods. In recent years, many studies have investigated different approaches to detect phishing attacks. These studies have made contributions to the field By incorporating model detection methods that analyze text, images, URLs and user behavior patterns to enhance the accuracy of identifying phishing threats (Nordin et al., 2021). In addition, Improving the generalizability and transferability of detection models across datasets and scenarios is crucial for deployment in real world settings.

In recent related work, Deep learning (DL) algorithms like recurrent neural networks (RNNs) or convolutional neural networks (CNNs) are widely employed and was demonstrated a high accuracy rate in detecting different phishing attacks scenarios (Abdelali et al., 2021). In recent years, Phishing attack considered one of the widespread security attacks which target high volume data systems such as emails, social media platforms by sending phishing text content to target victims (Anupam and Kar, 2021) . Analyzing human language by machine require the use of NLP technique to represent human language and it is investigated in recent research works (Abdessaied et al., 2022) . However, in recent DL work reviewed, most attempted analyzed text of web pages without considering words sequence in the text which lack the extract of meaning and semantic of input text. Therefore, there is a need to experiment different models and approaches to detect phishing attacks to address research gap and strengthen phishing detectin strategies and fortifying the security position for both individuals and organizations against phishing attacks. Hence, this work is aimed to detect whether the web page is phishing or ham and the main contributions are as follows:

- **Extraction of language features**: The focus lies on extracting language features from the HTML code of websites using NLP techniques to enable DL algorithm to detect phishing attempts though webpage text content features.

- **Word embeddings process**: the process of representing human text to machine as input feature was word2vector embeddings which is a sequential method that describe relationships and semantic of word in spatial distance and vectors.

- **Enhancing feature selection through PSO**: to pinpoint the distinguishing features for precise detection of phishing attacks to increase accuracy and decrease modeling time of LSTM.

- **LSTM modeling** : to classify web page content into phishing or ham, and capture time related patterns and contextual details thereby boosting detection accuracy.

The proposed multi-steps model of this research incorporating NLP, PSO and LSTM to analyze text content, refines feature selection and understands time-based relationships. Moreover, these methods improve the efficiency of the detection process enabling more identification of phishing attacks. In this research work, PSO algorithm is used to enhance feature selection to pick out the language cues thereby enhancing accuracy by recognizing key patterns in the text data. Also, PSO accelerates the convergence of the LSTM model during training cutting down on training time and facilitating deployment of the phishing detection system. The following structure of this article are Section 2 reviews the related work, Section 3 explains proposed re methodology in detail, section 4 highlights the experiment setups and discusses results Section 5 presents conclusions

2 Related works

In this section, a review of related work to this research study main aim of detecting phishing attack is discussed focusing on recent work used NLP and DL in detecting webpage and email suspected text content. In a study conducted by Noor Faisal Abedin et al.(Abedin et al., 2020), the authors discussed the ability of machine learning techniques that can predict if websites are phishing or not. These techniques use features based on URLs that aim to detect websites from fake ones by examining websites' URL. One of the algorithms used is the random forest classifier. This algorithm showed high accuracy results: a precision of 97% a recall of 99% and F1 score of 97% during training. This shows that the model is good at sorting websites into phishing or legitimate categories. One notable advantage of this model is its speed and efficiency. It only needs to analyze the URL to make predictions. It doesn't require resources or features for analysis.

Mohammad Nazmul Alam et al. (Alam et al., 2020a) focus on identifying phishing attempts using machine learning techniques. Random forest (RF) and decision tree (DT) are used. The authors utilized a dataset of phishing attempts probably sourced from platforms, like Kaggle for the machine learning analysis. The model they proposed utilized feature selection methods such as principal component analysis (PCA) to examine the characteristics of the dataset. The authors explained that feature selection is important for pinpointing the attributes that help in effectively detecting phishing attacks. They assessed the model's performance; it achieved an accuracy rate of 97% with the random forest algorithm. Muhammad Waqas Shaukat et al.(Shaukat et al., 2023) used a dataset containing 20,000 website URLs to create a phishing detection model. The phishing detection model utilized a classification method involving machine learning techniques, like SVM, XGBoost, random forest, multilayer perceptron, linear regression, decision tree, naïve Bayes and SVC. Through performance evaluation the model demonstrated phishing detection. XGBoost displayed the performance with accuracy and precision rates of 94% during training and 91% during testing. The multilayer perceptron algorithm also showed performance, with a testing accuracy of 91%. Forest and decision tree algorithms achieved accuracy rates of 91% and 90% respectively. In terms of text based classification, logistic regression and SVM algorithms were employed with accuracy rates of 87% and 88% respectively.

Malak Aljabri et al.(Aljabri and Mirza, 2022) discusses how intelligent techniques like Machine Learning (ML) and Deep Learning (DL) are used to detect phishing websites. Two different datasets were analyzed by the authors, who selected the related features for their study. These features included content-based URL lexical based and domain-based characteristics. The findings highlight how feature selection impacted model performance significantly. The Random Forest (RF) algorithm outperform in accuracy among all models tested on both datasets. This indicates that RF perform more accurate in classifying phishing websites based on specific features. Ishita Saha et al. (Saha et al., 2020)focus on identifying websites through the introduction of a data framework using deep learning techniques. Traditional methods like blacklists, whitelists and antivirus programs have been employed to detect phishing attempts. The researchers suggest utilizing a perceptron (MLP) a type of feed forward network for predicting fraudulent websites. The dataset used in their research was sourced from Kaggle. Comprises information from ten thousand websites. The proposed model achieved an accuracy rate of 95% during training and 93% during testing. The researchers in (Benavides-Astudillo et al., 2023) proposed a method, for spotting phishing attacks by focusing on the text content of web pages instead of just relying on URLs. They used of Natural Language Processing (NLP) techniques and Deep Learning (DL) algorithms to analyse pishing attack of webpages. Ther proposed approach involves an analysis of using NLP and Word Embedding techniques followed by incorporating this data into a DL algorithm. Four different DL algorithms are assessed; Long Short Term Memory (LSTM) Bidirectional LSTM (BiLSTM) Gated Recurrent Unit (GRU) and Bidirectional GRU (BiGRU). The outperforming algorithm among assesses models was BiGRU with an accuracy rate of 97.39%.

Adwan Yasin and Abdelmunem Abuhasan et al.(Yasin and Abuhasan, 2016) introduce the idea of assigning weights to phishing terms to assess how significant they are in each email. They improve the processing stage by including methods like text stemming and using WordNets vocabulary to enrich the model with word variations. The model follows knowledge discovery processes. Applies five known classification algorithms for email categorization. The outcomes reveal an improvement in classification accuracy. Specifically, the Random Forest algorithm achieves a 99.1% accuracy rate while J48 achieves 98.4%. In their research work (Buber et al., 2018) introduces a system, for detecting phishing that uses machine learning algorithms and visual similarity analysis with natural language processing methods. The system underwent testing and the results from experiments indicated that the Random Forest algorithm achieved a success rate of 97.2%. BenavidesAstudillo et al . (BenavidesAstudillo et al., 2024) discusses a research project that centers on creating a user tool named NDLP Phishing designed as an add on for the Google Chrome web browser. This tool leverages learning (DL) and natural language processing (NLP) methods to identify phishing attempts. The research involves

Article	DL	NLP	Optimization algorithm	Web page text
[1]	No	No	No	yes
[2]	No	No	Yes (PCA)	No
[4]	No	No	No	No
[6]	Yes	No	No	No
[7]	Yes	No	No	No
[8]	Yes	Yes	No	No
[9]	No	Yes	No	No
[11]	No	Yes	No	Yes
[12]	Yes	Yes	No	Yes
Proposed Model	Yes	Yes	Yes	Yes

Table 1: A comparison of related work contribution to the proposed model

choosing and tuning hyperparameters for a BiGRU detection model based on DL and NLP. According to the study findings the model demonstrated an accuracy of 98.55% after implementing the optimized hyperparameters.

A summary of related work contributions comparison to our research proposed model is shown in Table 1.

Although the related studies (Basile et al., 2022), (Abdelghaffar et al., 2022) of Table 1 implement DL and NLP techniques, the focus their research were on analyzing the content of the URL and webpage without applying optimization algorithm to enhance feature extraction and enhance time modeling. Only the article of (A et al., 2021) used PCA optimizer and it is neither it is using NLP nor analyzing webpage text.

3 Proposed model methodology

In this research work, the steps of the proposed model methodology to detect phishing attacks is shown in figure2. Firstly, the data gathered contains both legitimate and non-legitimate HTML webpages content. Secondly, the data is preprocessed b using NLP techniques such as tokenizing, parts of speech, lemmatizing and removing stop words. Word2Vec is then utilized for word embeddings to represent words as vectors that capture connections and semantic relationships. Afterwards, the relevant features are identified through feature selection using Particle Swarm Optimization (PSO) algorithms. Finally, LSTM is employed to understand patterns and correlations among these features to classify and detect phishing and ham webpages' content. Performance evaluation metrics, like accuracy, precision, recall and F1 score are used to assess how well the model detects phishing

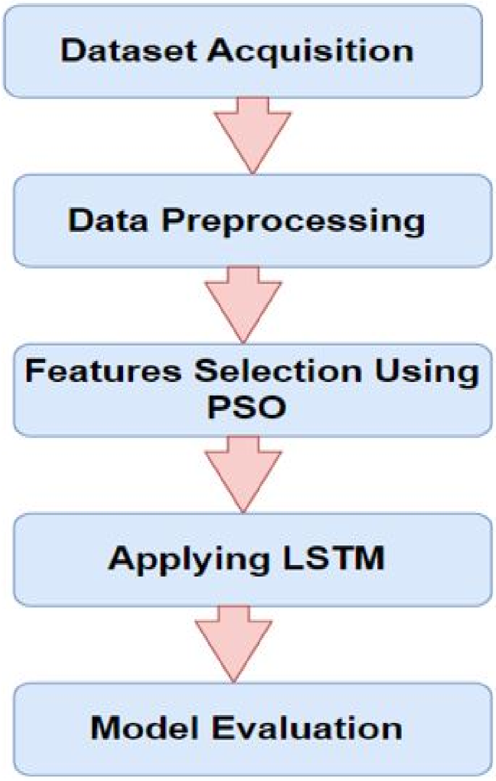

Figure 2: Phishing attack detection model Methodology

.

3.1 Data acquisition

In this study, the authors collect data from the Phishload dataset (. and Chandra, 2022) which is a collection of HTML code from both phishing and non-phishing web pages. The dataset was originally in a SQL format. Was converted to CSV format to be used in Python. The dataset contains three tables. Our analysis primarily focused on the "websites" table. From this table the author extracted two columns: 1. "htmlContent" column; This column includes the HTML code and text content of the web pages. 2. "isPhish" column; This column indicates whether a website is identified as phishing or non-phishing. Initially the dataset had 10,488 rows. After eliminating rows with missing data fields, 10,373 rows remain in total. Among these 9,198 rows were categorized as phishing websites while 1,176 rows were labeled as phishing websites. It is important to note that there is an imbalance in the dataset due to the difference, in the number of phishing and phishing instances. To address the imbalance data and ensure the reliability of the experiment, the author utilized the K cross validation method with shuffle = true and K = 5. This method includes splitting the dataset into five segments with 80% of the data allocated for training and 20%, for testing in each segment. By adopting this strategy, it enables an assessment and evaluation of performance when dealing with an imbalanced dataset ref(A et al., 2021).

3.2 Data preprocessing

The process of word analysis involves stages; To start regular expressions are utilized to eliminate elements such s URLs, mentions, HTML tags, digits and miscellaneous characters. Next the split function breaks down the text into segments using a designated separator. In this research work, the author preprocess text data using NLP techniques by applying following steps in (A, 2022)

- **Common words elimination** : little significance stop words like "a," "an " "the " "is," are filtered out from the list of words. The elimination of stop words is a practice in NLP tasks to improve efficiency, accuracy and interpretability. By Using a Predefined Stop Word Lists, such as NLTK which include predefined lists of stop words(. et al., 2022).

- **Tokenization**: Tokenization is defined as breaking down text into units known as tokens is a process in natural language processing. Tokenization plays a role in detecting phishing by extracting features with tokens acting as the basis for recognizing signs of phishing activities. In this research scenario, the method texts to sequences was utilized to convert atext input into a sequence of numbers. This technique is commonly employed in natural language processing (NLP) libraries such as TensorFlow or Keras to change a text collection into a format that can be analyzed by machine learning algorithms. Each distinct word in the text receives a number and the resulting sequence represents the text based on these numbers (Rubino et al., 2022).

- **Lemmatization**: simplifies words to their base forms with the assistance of the WordNetLemmatizer class. These procedures serve to refine the text by discarding components and converting words into their forms to reduce exclusive words in the corpus and improve precision and effectiveness(A, 2022).

- **POS tagging** : The POS tagging is heuristics method that is utilized for tagging Parts of Speech (POS). In this research work, POS tagging involves assigning parts of speech (like nouns, verbs, adjectives, adverbs, etc.) to words in a provided text or sentence. Through the assignment of part of speech tags to words it facilitates a profound analysis and understanding of the texts meaning and context offering insights that help in identifying word combinations that could signal phishing content. In the field of phishing detection, POS tagging is valuable for pinpointing errors like incorrect verb forms or inconsistent noun verb agreement(. et al., 2022).

- **Word2Vector** : The feature extraction process plays a role in building performing models in DL. It focuses on reducing the number of features to concentrate on the ones for efficient training. Word embedding is a technique in NLP that aligns with the hypothesis suggesting that words with similar meanings often appear in similar linguistic contexts. Word embedding represents words as valued numeric vectors within a vector space aiming to capture features based on neighboring words. Numeric representations of words allow for operations and comparisons between words. In this study, Continuous Bag of Words (CBOW) is applied to predict a target word in webpage content from its context. Three layers are used in CBOW implementation. First layer is Input layer which relates to the context. Second is the hidden layer which pertain to the prediction of each word feed from the input layer into weighting matrix. The third layer is the output layer which is projected by the weighting matrix. Finally, the model compare between its output and the word itself to correct the representation using error gradient technique of back propagation (Abdul-Mageed et al., 2021). Efficient Estimation of Word Representations in Vector Space.

4 Features selection using PSO

In this study, the author apply particle swarm optimization (PSO) algorithm which is an optimization method inspired by how natures collective behavior works. PSO shows promise in enhancing detection systems. PSO support the process of choosing the most relevant features that distinguish phishing attacks from legitimate content efficiently (Agarwal et al., 2022). Steps of using PSO for feature extraction is shown in Figure 3. The Steps for Using PSO for Feature Selection are explained as follows:

- **Initialization:**

  1. Swarm initialization: Create a swarm of particles where each particle represents a potential solution. In the context of feature selection, each particle's position can be a binary vector where each bit represents the inclusion (1) or exclusion (0) of a feature.

  2. Velocity initialization: Initialize the velocity of each particle randomly.

- **Fitness evaluation**: Fitness Function: Define a fitness function to evaluate the quality of each particle's position. This could be the accuracy of a machine learning model trained on the selected features or a combination of accuracy and the number of selected features to ensure model simplicity

- **Update velocity:** Update the velocity of each particle based on its personal best position (pbest) and the global best position (gbest). The velocuty update rule can be defiend as:

- **Update position:** update the position of each particle using its updated velocity: Apply a sigmoid function to ensure the position values remain within the [0, 1] range, and then convert them to binary values for feature selection.

- **Iteration:** repeat the steps of fitness evaluation, velocity update, and position update until a stopping criterion is met (e.g., a maximum number of iterations or a satisfactory fitness level).

- **Result:** the global best position (gbest) at the end of the iterations represents the optimal set of features selected by the PSO algorithm.

Therefore, in this research work, the PSO plays an important role in finding the right parameter values for LSTM model in order to detect phishing content as it enhances feature selection, fine tune model parameters. Through exploring parameter settings PSO guides the optimization process towards parameter configurations leading to better detection accuracy and time modelling.

The algorithm for the PSO-based feature selection algorithm is provided in Algorithm 1 used for feature selection in our phishing detection model. The process begins with initializing a swarm of particles, each representing a potential solution in the form of a binary vector that indicates the inclusion or exclusion of features. The velocity and position of each particle are iteratively updated based on both their own best-known position (*pbest*) and the best-known position of the entire swarm (*gbest*). The particles' positions are then converted into binary values to determine the selected features. The fitness of each particle is evaluated using a predefined fitness function, typically based on the accuracy of a machine learning model trained with the selected features. The algorithm continues to iterate until a stopping criterion is met, such as a maximum number of iterations or a satisfactory fitness level. Finally, the algorithm outputs the global best position, which represents the optimal set of selected features. By using PSO, the author aimed to enhance the feature selection process, improving the accuracy and efficiency of the phishing detection model.

5 Applying LSTM

In this study, the author assesses the performance of the LSTM model by two features feedings: • Word2vector direct features feeding to LSTM and refered to as "LSTM model" • Word2vector and PSO enhanced features feeding to LSTM referred to as "LSTM-PSO model". The LSTM is proven high accurate results in examining patterns in text data that unfold over time (Abdel-Salam, 2022). It is used in this research due to the capability of LSTM layers in analyzing the input sequence and

Algorithm 1 Feature Selection Using PSO

```
Initialize the swarm with $N$ particles, each representing a potential solution (binary vector of feature inclusion/exclusion).
Initialize velocity vectors for each particle randomly.
Evaluate the fitness of each particle based on a predefined fitness function (e.g., accuracy of a machine learning model using the selected features).
Initialize the personal best position (pbest) of each particle to its current position.
Initialize the global best position (gbest) to the position of the best fitness particle in the swarm.
while stopping criterion not met do:
    for each particle $i$ in the swarm do:
        Update particle's velocity:
            $v_i = \omega v_i + c_1 r_1 (pbest_i - x_i) + c_2 r_2 (gbest - x_i)$
        Update particle's position:
            $x_i = x_i + v_i$
        Apply a sigmoid function to ensure position values remain within the [0, 1] range:
            $x_i = \frac{1}{1 + e^{-x_i}}$
        Convert positions to binary values for feature selection:
            $x_i = \begin{cases} 1 & \text{if } x_i > 0.5 \\ 0 & \text{otherwise} \end{cases}$
        Evaluate the fitness of the updated position.
        if current fitness better than pbest then:
            Update pbest to current position.
        end if
        if current fitness better than gbest then:
            Update gbest to current position.
        end if
    end for
end while
Output the global best position (gbest) as the optimal set of selected features.
```

grasp the connections between words. In this research work, to train the LSTM model, we utilized our proposed preprocessed dataset features with splitting of dataset to 80% training and 20% validation. During this phase the model gains an understanding of patterns and characteristics within data that differentiate between phishing attempts and false content. Following training the performance of the LSTM model is assessed using a test dataset. Common evaluation metrics, for detecting phishing may encompass accuracy, precision, recall and F1 score. Upon completion of training and evaluation processes the LSTM model can predict the likelihood of phishing in text data that it has not encountered before. The model takes in input text runs it through its LSTM layers and generates a prediction (phishing or legitimate) based on its patterns (A, 2022).

6 Experiment setup

Ib this study, the author used Python 3.5.2 on Jupyter Notebook 6.0.2 is to code NLP, PSO and LSTM algorithms. Additionally the libararies used are, as Keras, NLTK, NumPy, pandas, requests, scikit learn and TensorFlow. These libraries offer features and utilities for tasks, like DL, NLP, data handling and model development and assessment. By using these tools, the author successfully carried out research experiment.

7 Evaluation metrics

Ib this study, the author used Python 3.5.2 on Jupyter Notebook 6.0.2 is to code NLP, PSO and LSTM algorithms. Additionally the libararies used are, as Keras, NLTK, NumPy, pandas, requests, scikit learn and TensorFlow. These libraries offer features and utilities for tasks, like DL, NLP, data handling and model development and assessment. By using these tools, the author successfully carried out research experiment.

- **True positive (TP)** : Represents the number of correctly classified positive data items.
- **True negative (TN)** : Represents the number of classified data items.
- **False positive (FP))** : Indicates the number of classified data items.
- **False negative (FN)** : Indicates the number of classified data items.

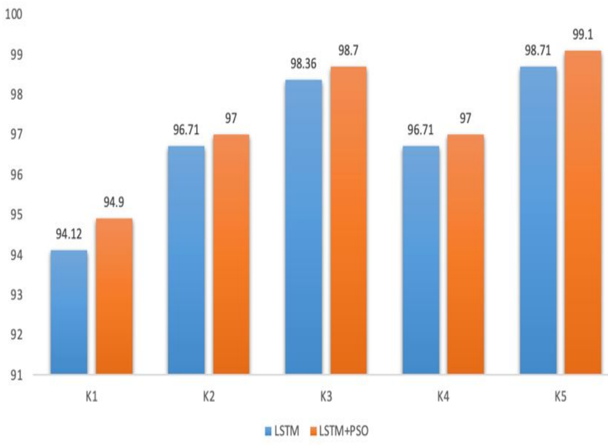

Figure 3: Cross Validation accuracy

Both models as shown in Figure 4 performe well in identifying phishing attacks with accuracies exceeding 94% across all K folds. The LSTM+PSO model consistently outperfoem the LSTM model in terms of accuracy inicaging that incorporating PSO for feature selection boosts the model's effectiveness. The accuracies of both models exhibit variations among segments indicating their ability to generalize well to diverse data subsets. Moreover, the LSTM+PSO model consistently achieves accuracies reaching a peak accuracy of 99.1% in K5. Overall, the findings of k cross-valdiaton metric highlight that integrating PSO for feature selection enhances the phishing detection capabilities of the LSTM+PSO model compared to the LSTM model. This improvement is evidenced by accuracies, across various segments underscoring its efficacy in detecting phishing attacks.

With regards to LSTM model accuracy values as shown in Figure 5 During the training process the accuracy of the model steadily increases with each epoch. Starting at 92% in epoch 0 it progresses to 97% by epoch 17.5 showing that the model is learning and getting better at classifying the training data. Similarly, the validation accuracy also improves as epochs increase. Beginning at 94% in epoch 0 it reaches 98.5% by epoch 17.5 indicating that the model is adapting well to data and enhancing its performance over time. When comparing training and validation accuracies it is noticeable that validation accuracy consistently surpasses training accuracy. This suggests that the model is not overly fixated on the training data and can generalize effectively. The slight disparity between both accuracies implies that there is no overfitting issue. The optimal performance point is observed at epoch 17.5 where a validation accuracy of 98.5% is achieved. This indicates that the model excels in generalizing to data at this stage. However, factors like resources, training duration and potential overfitting should be considered when determining the epoch for model training.

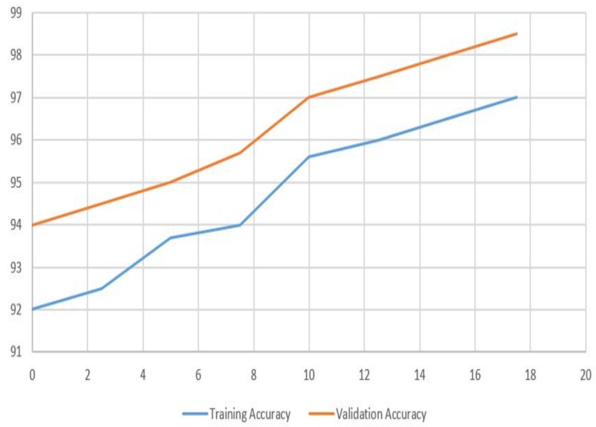

Figure 4: LSTM accuracy

On the other hand The accuracy vlaues of LSTM-PSO is illustrated in Figure 6. of the training set keeps improving as the epochs progress. It begins at 90% at epoch 0. Steadily rises to 97% by epoch 17.5. This shows that the model is learning and getting better at classifying the training data. Similarly, the validation accuracy also displays an trend with increasing epochs. It starts at 92% at epoch 0. Gradually increases to 98.9% by epoch 17.5. This suggests that the model is adapting well to data and enhancing its performance over time.In comparing the training and validation accuracies we notice that the validation accuracy remains consistently higher than the training accuracy. This indicates that the model is not overly focused on fitting to the training data but can generalize effectively. The minimal difference between these two accuracies implies that there is no overfitting issue with the model, which's a positive outcome. When we look at when it achieves its validation accuracy we see that it happens during epoch 17.5 where it reaches an accuracy of 98.9%. This signifies that this stage represents performance, for generalizing to data. However, it is crucial to take into account factors like computing resources, training duration and the risk of overfitting when deciding on the epoch for model training.

Based on the data shown in the figure, the model reaches its peak performance around epoch 17.5 boasting a training accuracy of 98.3% and a solid

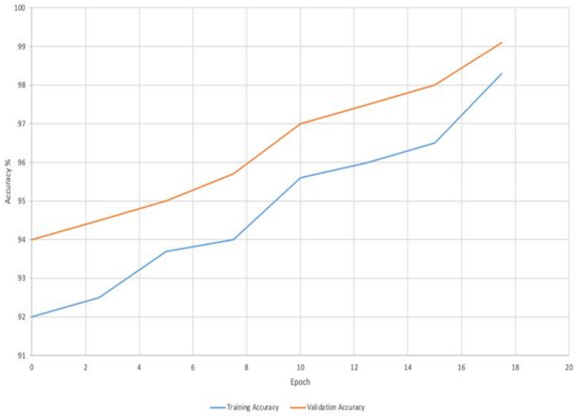

Figure 5: LSTM-PSO accuracy

validation accuracy of 98.9%. This indicates that the model has successfully grasped patterns from the training data and can generalize effectively to examples. Notably the models performance shows enhancement across epochs with advancements even in the initial stages.

8 Conclusion

In conclusion, this research study is research proposed an integrated approach to detect phishing attack webpages text content by utilizing the Keras Embedding Layer with word2vector to capture both the meaning and structure of text found on web pages. In addition to employing word level embedding methods, the model transformed these characteristics into vector representations which were then feeded into deep learning algorithms like LSTM, and the vector representation of word2vector featrures were enhanced and feeded into LSTM-PSO to detect phishing websites. The results of the proposed LSTM-PSO model indicate a higher accuracy rate of 98.3% in comparison to LSTM accuracy rate of 97%. The literature review conducted in this study illustrated a gap in research studies related to the analysis of web page content using natural language processing and deep learning. Most existing studies have focused on mitigating phishing emails or examining URLs rather than analysing the text content of web pages an.

The author aim for future work to test the model using word embedding methods such as FastText and GloVe to investigate how well they perform in processing webpage text content in comparison to word2vector embbidings. Moreover, the author intend to conduct phishing attacks detection using different DL algorithms and different ensemble methods of at least two DL on more data segment such as third-party information, and Web content level.

References

Mamta ., Asif Ekbal, Pushpak Bhattacharyya, Tista Saha, Alka Kumar, and Shikha Srivastava. 2022. HindiMD: A multi-domain corpora for low-resource sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7061–7070, Marseille, France. European Language Resources Association.

Sanju . and Subhash Chandra. 2022. Pāṇinian phonological changes: Computation and development of online access system. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 24–28, Marseille, France. European Language Resources Association.

Anirudh A, Aman RAJ Singh, Anjali Goyal, Lov Kumar, and N L Bhanu Murthy. 2021. Prediction of video game development problems based on postmortems using different word embedding techniques. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 465–473, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Sujan Reddy A. 2022. Automating human evaluation of dialogue systems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 229–234, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Reem Abdel-Salam. 2022. reamtchka at SemEval-2022 task 6: Investigating the effect of different loss functions for sarcasm detection for unbalanced datasets. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 896–906, Seattle, United States. Association for Computational Linguistics.

Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2021. QADI: Arabic dialect identification in the wild. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 1–10, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mohamed A Abdelghaffar, Amr El Mogy, and Nada Ahmed Sharaf. 2022. Adapting large multilingual machine translation models to unseen low resource languages via vocabulary substitution and neuron selection. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 287–297, Orlando, USA. Association for Machine Translation in the Americas.

Adnen Abdessaied, Ekta Sood, and Andreas Bulling. 2022. Video language co-attention with multimodal fast-learning feature fusion for VideoQA. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 143–155, Dublin, Ireland. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Noor Faisal Abedin, Rosemary Bawm, Tawsif Sarwar, Mohammed Saifuddin, Mohammd Azizur Rahman, and Sohrab Hossain. 2020. Phishing attack detection using machine learning classification techniques. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pages 1125–1130. IEEE.

Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. 2022. Entity linking via explicit mention-mention coreference modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4644–4658, Seattle, United States. Association for Computational Linguistics.

Mohammad Nazmul Alam, Dhiman Sarma, Farzana Firoz Lima, Ishita Saha, Sohrab Hossain, et al. 2020a. Phishing attacks detection using machine learning approach. In *2020 third international conference on smart systems and inventive technology (ICSSIT)*, pages 1173–1179. IEEE.

Mohammad Nazmul Alam, Dhiman Sarma, Farzana Firoz Lima, Ishita Saha, Rubaiath-E Ulfath, and Sohrab Hossain. 2020b. Phishing attacks detection using machine learning approach. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 1173–1179.

Waleed Ali and Sharaf Malebary. 2020. Particle swarm optimization-based feature weighting for improving intelligent phishing website detection. *IEEE Access*, 8:116766–116780.

Malak Aljabri and Samiha Mirza. 2022. Phishing attacks detection using machine learning and deep learning models. In *2022 7th International Conference on Data Science and Machine Learning Applications (CDMA)*, pages 175–180. IEEE.

Sagnik Anupam and Arpan Kumar Kar. 2021. Phishing website detection using support vector machines and nature-inspired optimization algorithms. *Telecommunication Systems*, 76(1):17–32.

Valerio Basile, Zornitsa Kozareva, and Sanja Stajner, editors. 2022. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Dublin, Ireland.

Eduardo Benavides-Astudillo, Walter Fuertes, Sandra Sanchez-Gordon, Daniel Nuñez-Agurto, and Germán Rodríguez-Galán. 2023. A phishing-attack-detection model using natural language processing and deep learning. *Applied Sciences*, 13(9):5275.

Eduardo BenavidesAstudillo, Walter Fuertes, Sandra Sanchez-Gordon, and Daniel Nuñez-Agurto. 2024. Ndlp phishing: A fine-tuned application to detect phishing attacks based on natural language processing and deep learning. *International Journal of Interactive Mobile Technologies*, 18(10).

Ebubekir Buber, Banu Diri, and Ozgur Koray Sahingoz. 2018. Nlp based phishing attack detection from urls. In *Intelligent Systems Design and Applications: 17th International Conference on Intelligent Systems Design and Applications (ISDA 2017) held in Delhi, India, December 14-16, 2017*, pages 608–618. Springer.

Surbhi Gupta and Abhishek Singhal. 2017. Phishing url detection by using artificial neural network with pso. In *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, pages 1–6. IEEE.

Noor Syahirah Nordin, Mohd Arfian Ismail, Tole Sutikno, Shahreen Kasim, Rohayanti Hassan, Zalmiyah Zakaria, and Mohd Saberi Mohamad. 2021. A comparative analysis of metaheuristic algorithms in fuzzy modelling for phishing attack detection. *Indones. J. Electr. Eng. Comput. Sci*, 23(2):1146–1158.

MCA Radha Damodaram and ML Valarmathi. 2011. Phishing website detection and optimization using particle swarm optimization technique. *International Journal of Computer Science and Security (IJCSS)*, 5(5):477.

Melanie Rubino, Nicolas Guenon des Mesnards, Uday Shah, Nanjiang Jiang, Weiqi Sun, and Konstantine Arkoudas. 2022. Cross-TOP: Zero-shot cross-schema task-oriented parsing. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 48–60, Hybrid. Association for Computational Linguistics.

Ishita Saha, Dhiman Sarma, Rana Joyti Chakma, Mohammad Nazmul Alam, Asma Sultana, and Sohrab Hossain. 2020. Phishing attacks detection using deep learning approach. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 1180–1185. IEEE.

Muhammad Waqas Shaukat, Rashid Amin, Muhana Magboul Ali Muslam, Asma Hassan Alshehri, and Jiang Xie. 2023. A hybrid approach for alluring ads phishing attack detection using machine learning. *Sensors*, 23(19):8070.

Adwan Yasin and Abdelmunem Abuhasan. 2016. An intelligent classification model for phishing email detection. *arXiv preprint arXiv:1608.02196*.