International Journal of Gender, Science and Technology
..ISSN: 2040-0748
UGC Care Group I Journal
Vol-13 Issue-02 Aug 2024

Phishing Email Detection Model Using Deep Learning

[1]ChittipothulaMounika,(Chmounika439@gmail.com) [1]M.tech Scholar
[2]Dr.L.Jagadeesh Naik,(jagadeeshnaik.l@hmgi.ac.in) Associate Professor

Holy Mary Institute of Technology and Science, Bogaram, Medchal Malkajgiri dist, Telangana, 501301.

Abstract

One frequent cyber threat that can result in the theft of private information and financial harm is phishing emails. Malicious emails often take the guise of reliable companies or individuals in an attempt to trick recipients into providing private information or money. As technology advances and attackers gain more expertise, it becomes more challenging to recognise and stop email phishing. This paper explores the application of deep learning techniques, including recurrent neural networks (RNNs), bidirectional encoder representations from transformers (BERT), long short-term memory (LSTM) networks, and convolutional neural networks (CNNs), for email phishing attack detection. From a dataset of legitimate and fraudulent emails, relevant properties were extracted using natural language processing (NLP) techniques. The suggested deep learning model was developed and assessed using the dataset. The findings demonstrated that it can identify email phishing with a high degree of accuracy when compared to other cutting-edge research; an accuracy of 99.61% was attained when combining BERT with LSTM. The results demonstrate how deep learning may be applied to improve email phishing detection and protect against this pervasive threat.

Keyword: L stm, Cnn, Dee learning, Rnn.

Introduction

Web-based activities made available through cyberspace have risen as a result of IT's growing popularity and use [1]. Adewole et al. (2019) [2] state that these activities include everything from simple activities like social media and e-health apps to critical services like financial transactions and education. Research indicates that the most popular web-based activities with large user populations are social networking, online gaming services, and financial transactions [3]. Numerous consumers ofthese web-based solutions demonstrate how widely accepted IT has become in recent years. The goal is to increase the availability and accessibility of the web-based solutions that are used on a regular basis. However, the open availability and accessibility of these web-based solutions makes them vulnerable to cyberattacks because there are no universal security measures in cyber space [4].Phishing is a crime that affects everyone worldwide, including governments and organisations. A common cyberattack that can have detrimental effects on an organization's reputation as well as financial loss and identity theft is email phishing. Over the past ten years, the number of people falling victim to phishingscams has increased dramatically, with millions of victims falling victim each month. Organisations are faced with a more challenging task as a result of this growth in trying to defend themselves against this expanding threat. These days, it's harder than ever to identify and stop email phishing. Phishers are always changing their techniques to avoid being discovered by security programmes and law enforcement. If organisations want to be safe from this threat, they need to be able to identify and stop these attacks on a large scale. According to Vrbančič et al. (2018) [5], a phishing assault is a type of widespread fraud in which a phoney website mimics a legitimate one in an attempt to obtain personal information from users who are not cautious. Phishing involves the creation of a clone website that mimics a genuine website, making it challenging for consumers to recognise [6].Phishing is a well-known topic these days, and successful attacks can have severe consequences . Internet users and legitimate web resource owners areat danger due to phishing assaults [7]. The current surge in phishing assaults has caused a loss of trust inlegitimate users, making them feel less safe even withrobust antivirus software.

Artificial neural networks (ANNs) are used in deep learning, a type of machine learning, to analyse and categorise data. Neural networks are useful for classifying text and images because they can handle and learn from massive volumes of data [8]. The goal of this research is to determine which deep learning approaches are most effective for email

phishing detection. The study creates methods for precisely identifying and reporting phishing emails. Our research directly analyses the efficacy of various strategies on a sizable real-world email dataset, in contrast to earlier studies that assessed these techniques separately. This provides unique insights into the relative advantages and disadvantages of different models for phishing detection. Based on a thorough empirical review, our study determines the best deep learning architectures and provides guidance in the selection of appropriate methodologies for developing real-world phishing detection systems. This new direct comparison analysis on a solid dataset closes a crucial gap in the selection of deep learning techniques for email security. The following is the research's contribution: creating deep learning methods that reliably classify emails as authentic or phishing by analysing attributes including the sender's information, content, and subject line.

Enhancing the effectiveness and velocity of deep learning algorithms for email phishing detection in order to facilitate real-time email analysis as it arrives. experimenting using deep learning in conjunction with methods like feature selection, transfer learning, and graph theory to improve email phishing detection accuracy.comparing the efficacy of various deep learning architectures and methods, such as bidirectional encoder representations from transformers (BERT), recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and convolutional neural networks (CNNs), for email phishing detection.

Literature survey

A study on deep learning methods for identifying spam emails in English-language text messages was carried out by the authors of [19]. The authors put forth a model that classified spam emails using deep learning and characteristics taken from the emails' text. A dataset of spam and non-spam emails was used for training and testing the supervised learning model, which was constructed for this purpose. The study's findings demonstrated that the suggested model could identify spam emails with a high degree of accuracy. The authors also talked about the future directions and possible uses of deep learning methods for spam email identification. Overall, Ref. [19] demonstrated how deep learning may be used to identify spam emails, hence enhancing email system security. Singh and colleagues (2020) [20] carried out research on the use of deep learning methods to the identification of phishing attempts using URLs. Based on information taken from the URL, the authors created a model that used deep learning to identify URLs as either phishing or non-phishing. The system outperforms a prior model that reached 97.98% accuracy, with a maximum accuracy of 98.00%. CNNs may extract pertinent features straight from the URLs, which eliminates the need for manual feature engineering, which is one advantage of the method. This is a big improvement over earlier methods, which can be labour- and time-intensive. All things considered, the method seems to be a viable way to identify and stop phishing attempts.

A methodology for identifying phishing websites through deep learning approaches was presented by Saha et al. (2020) [21]. They analysed a dataset of 10,000 web pages that they had gathered from Kaggle using a multilayer perceptron, also known as a feed-forward neural network. Ten attributes are included in the dataset, including the website's age, the URL, and the existence of specific words or symbols. By dividing the dataset into training and test sets and transforming categoricalattributes to numerical values, the authors preprocessed the data. Subsequently, they used the training data to train the multilayer perceptron and the test data to assess its performance. On the trainingset of data, the model's accuracy was 95%, while on the test set, it was 93%. The authors came to the conclusion that deep learning techniques can identify phishing attacks successfully and recommended moreresearch. They also mentioned that by adding more features or utilising more sophisticated deep learning methods, their system may be made even better.

McGinley and Monroy (2021) [22], obtaining an accuracy rate of 98%, verified the efficacy of CNN models in detecting phishing emails using content analysis. The suggested model receives an email body text embedding as input and outputs a probability indicating the likelihood that the email is malicious. In order to detect different types of wireless attacks, Fetooh et al. (2021) [23] created a real-time attack detection model for wireless networks. This model analyses multiple static and dynamic factors while performing a frame-type analysis. The analysis showed that the model's accuracy on average was 94.40%. A deep learning model that uses the BERT and Distil BERT pre-trained transformer models to identify phishing was proposed by Go go I and Ahmed (2022) [24]. With an accuracy rate of 99%, the suggested detection model effectively tackled the

challenges associated with phishing detection, such as the incapacity of traditional feature extraction techniques to distinguish phishing emails. Doshi et al. (2023) [25] presented a deep learning detection algorithm that uses email body and content features to distinguish between spam and phishing emails. With precise classification, the suggested approach successfully tackles the issue of data imbalance in spam and email phishing classification. The model classifies data examples into the proper classes using a dual-layer architecture, with a learnt or pre-trained model in each layer. CNN, RNN, and ANN models were used in the suggested model. A 99.51%accuracy rate was attained.

Recently, Benavides-Astudillo et al. (2023) [26] presented a phishing attack detection model that uses natural language processing and deep learning to detect phishing assaults on websites. The Phishload dataset was used in the development of the detection algorithm. The text content of the web pages is examined in order to extract features. Following model training, 98% validation accuracy was attained. Aldakheel et al. (2023) [27] described a detection model for phishing website identification intheir work. They used a CNN to successfullydistinguish between phishing and legitimate websites.With the use of the PhishTank dataset—a well- known dataset for identifying phishing websites based solely on URL features—the efficacy of the suggested detection model was assessed. Achieved accuracy rate: 98.77 percent.

Methodology

The methods to put the suggested model into practice are outlined in this part. These include gathering, getting ready, and using a dataset to train and test deep learning models that identify phishing emails. The overall structure for the study is shown in Figure 4, which covers feature extraction, dataset acquisition, data preparation, and training and testing of different deep learning techniques. The research technique is explained in the ensuing subsections.

Figure 1. Methodology for phishing email detection.

The Python Tensor Flow and Keras packages were used to create the deep learning models. In particular, high-performance numerical calculations and data pipelines were made possible with Tensor Flow. The deep learning models were constructed and trained using Keras on top of Tensor Flow using high-level APIs including functional and sequential APIs. The dense, LSTM, bidirectional, and dropout layers are the main Keras layers that are used. Additionally, text processing wasdone using Keras preprocessing tools including pad sequences and the tokenizer. For machine learning activities like data splitting and model evaluation, Scikit-learn was utilised. Important NLP libraries are the Tensor Flow Transformers library for BERT implementation and the NLTK library for textprocessing. For data handling, Pandas and NumPy were utilised, and for visualisation, Mat plot lib and Sea born were employed. This set of resources offeredthe ideal setting for quickly creating and testing deep learning models for phishing detection. This study's research methodology applied deep learning algorithms to identify phishing emails over a series ofphases. Among these actions were:

Data gathering: obtaining a dataset of benign and phishing emails in order to train and test themodel. Data preparation: Cleaning and tokenizing the text, among other preprocessing operations, to make the data ready for input into the model.

- Writing code to extract pertinent email features that can be used to train the model is known as feature extraction.
- Choosing the best deep learning algorithms to categorise phishing emails requires studying a number of different algorithms.
- Data processing: Utilising Python code, the data are processed and then saved to a CSV file. dividing the data into datasets for testing andtraining.
- Training the model involves applying the chosen algorithms to the training dataset. For subsequent use, the trained model was stored.
- Model evaluation involves classifying emails in the test dataset using the trained model, then utilising a variety of indicators to assess the model's performance.
- Comparison: Evaluating the effectiveness of various deep learning algorithms to determine which model is most effective in detecting phishing emails.
- This study applies deep learning techniques to open-source datasets of benign and phishing emails using a quantitative methodology.
- Deep learning-based algorithms for phishing email detection were trained, tested, and analysed using these datasets.

Dataset Selection

Any research on machine learning or deep learning must start with the selection of a dataset because the model's performance can be greatly impacted by the variety and quality of the data. A few things to think about when choosing a dataset for phishing email detection are as follows: Size: To give the model enough examples to train from, it's critical to use a sizable dataset. A sufficiently sampled dataset can aid in the model's improved generalisation to new, untested data. Diversity: A range of emails reflecting both phishing and benign communications should be included in thesample. This can aid in the model's learning of more reliable and precise patterns for phishing email detection.

Quality: There should be few mistakes or discrepancies in the data, making it of excellent quality. This may contribute to the model's increased accuracy.

Relevance: The information must be pertinent to the current task. The dataset should contain a representative sample of benign and phishing emails, similar to what the model will see in the real world, for the purpose of phishing email detection. This study trained and tested deep learning methods for phishing email classification using datasets that were made available to the public. It is crucial to choose a relevant and high-quality dataset because it might have a big impact on the model's performance and accuracy. The UCI Machine Learning Repository, Kaggle, and Google Dataset Search are a few tools for locating publically accessible datasets. This study made use of publically accessible datasets. These datasets were retrieved in text and CSV file formats, and different deep learning techniques were utilised for training, testing, and preliminary analysis. To determine the most effective model for spotting phishing emails, the performance was assessed and contrasted.

Phishing Email Dataset

A group of emails that have been specially chosen or produced to be utilised for phishing assault study and analysis is known as an email phishing dataset. These datasets usually contain a collection of phishing emails that have been gathered from different sources, including Spam Assassin and the UCI machine learning repository. They could be tagged or arranged in a certain way, like by the kind of phishing attack or the sector or company that is the target. Email phishing datasets can be utilised forstudy on the traits and techniques of phishing assaults, as well as for the training and assessment of deep learning-based phishing detection systems.

Benign Email Dataset

The email messages and metadata from the Enron Corporation, an American energy, commodities, and services business at the heart of one of the worst corporate scandals in history, are collected in the Enron email dataset. Over 500 customers' emails totaling over half a million are included in the dataset, which was assembled as part of the inquiry into the company's financial collapse. Other data included in the dataset include financial information, news items, and message routing details. Research on corporate governance, organisational communication, deep learning, machine learning, and natural language processing has made extensive useof the Enron email collection. It is a rich source of information for researchers due to the quantity of dataand the range of topics addressed. It has been utilised

to train deep learning and machine learning models that classify emails into various categories, such phishing or spam. It is important to note that private material and personally identifiable information (PII) have been removed from the Enron email collection through preprocessing and cleaning. In general, academics can benefit from the Enron email collection, particularly those who are interested in corporate governance, machine learning, and natural language processing.

Data Preparation and Preprocessing

The data must be cleaned and any extraneous words or characters must be eliminated in order to prepare the dataset for feature extraction and deep learning. Natural language processing (NLP) methods in Python were used to sanitise the datasets containing phishing and benign emails . Before being fed into the deep learning models, the implemented data underwent preprocessing processes to prepare the email text data. Using regular expressions, the NLTK library, and custom functions, the text data for the CNN, RNN, and LSTM models were converted to lowercase and devoid of special characters, numerals, and stop words. After that, the remaining words were reduced to their base form by stemming them with the Snowball stemmer. Usingthe Bert Tokenizer from the Transformers library, thetext data were tokenized for the BERT model bygenerating input IDs and attention masks and truncating sentences to 512 tokens. By reducing noise in the data, these preprocessing strategies improved the efficiency with which the deep learning models learned from the cleaned text. By emphasising the crucial textual elements and patterns for spotting phishing attempts, preprocessing the email data was crucial to increasing model accuracy.

Data Organisation and Labelling:

A Python script was used to identify and extract information from datasets of benign and phishing emails. The script produced a CSV file with all the features that were found, indicated by a 1 if they were present or a 0 if they weren't, along with the email's total character count. Additionally, theemail was categorised as either non-phishing (markedwith a 0) or phishing (marked with a 1). To train and test deep learning models for email phishing detection, this CSV file was utilised.

Splitting data

The next stage was to divide the data into training and test datasets after the features were extracted, the data were labelled, and they were ready to be saved in a CSV file.

The deep learning model was trained on the training dataset, and its performance was assessed on the test dataset There are several methods for dividing data into test and training sets. One method is a fixed split, in which a predetermined portion of the data are assigned to the test set and the remaining portion to the training set. For instance, the data might be split 70/30, meaning that 30% would be used for testing and the remaining 70% for training. We used a 70/30 splitting strategy in our suggested detection model. Utilising stratified sampling, which divides the data in a way that maintains the relative proportions of various classes or categories inside the dataset, is an additional strategy. Assume, for instance, that the dataset has an equal proportion of phishing and non- phishing emails. Stratified sampling can thus guarantee that there is an equal distribution of phishing and non-phishing emails in the training and test sets. To guarantee the model can generalise to new data and to precisely assess its performance, the data must be carefully divided into training and test sets.

Deep Learning Models: Training and Testing

Several deep learning models, including long short-term memory (LSTM) networks, recurrent neural networks (RNNs), bidirectional encoder representations from transformers (BERT), and convolutional neural networks (CNNs), were used in this study. The many deep learning models that were used in the study will be covered in the sections that follow. Using Keras and Scikit-learn, the suggested model performed standardised feature extraction and model training procedures. Using a maximum vocabulary size of 10,090 words, the email text was tokenized using the Keras Tokenizer into numerical representations for the CNN, RNN, and LSTM classifiers. For batch training, the sequences were padded to equal lengths. Using the pre-trained BERT tokenizer from Hugging Face, features were extracted for BERT, producing Word Piece tokens and attention masks up to 512 tokens in length. To avoid overfitting, the suggested model was optimised during training by utilising dropout and early stopping strategies. To be used again in the classification of phishing emails, the trained models were serialised. Overall, quick testing and assessmentof the deep learning architectures on the real-world

email dataset was made possible by utilising reliable feature extraction and training methods in Python.

CNNs, or Convolutional Neural Networks

CNNs are deep learning models that are frequently used for tasks like natural language processing, picture and video recognition, and other applications. CNNs are built with the ability to automatically and adaptively identify spatial hierarchies of characteristics from incoming data. An input layer, several hidden layers, and an output layer are the standard components of a CNN. The input layer is where text or images are received as input data. Convolutional layers, which are hidden layers, extract features from the input data by applying mathematical operations to it. The output, such as a probability distribution or a classification label, is generated by the final output layer. An activation function for the binary classification, like the Sigmoid function, is present in the output layer. A CNN can be optimised by a variety of techniques, including regularisation, early stopping, hyper parameter adjustment, and data preprocessing.

RNN

Text, audio, and time series data can all be processed sequentially using RNNs, which are deep learning models. RNNs have the ability to update their hidden states in response to both the input at that moment and earlier hidden states. Because of this, RNNs can maintain track of dependencies and context across time, which makes them ideal for applications like speech recognition, sentiment analysis, and language translation. An input layer, one or more recurrent layers, and an output layer make up an RNN. Based on the input received today and the hidden state from earlier, the recurrent layer changes the hidden state and produces a forecast. A number of techniques, including gradient clipping, have been proposed for RNN optimisation in additionto techniques for CNNs .Memory for Long Short Term(LSTM)

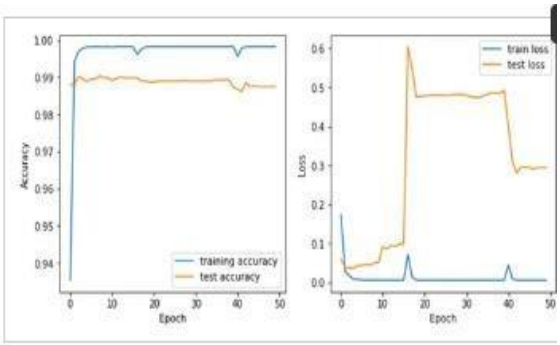
An issue with regular RNNs is vanishing gradients, which is addressed with the LSTM form of RNN. In order to better manage long-term dependencies in sequential data, LSTM makes use of a unique kind of memory cell that has the ability to selectively remember or **forget information from earlier time steps .**

Transformer-Based BidirectionalEncoder Representations (BERT)

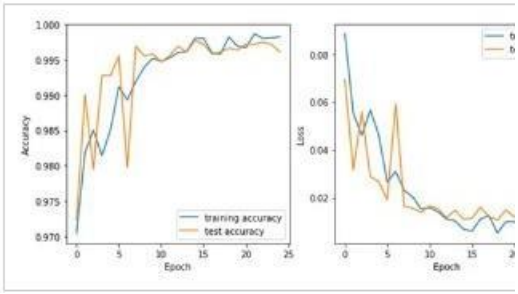
BERT is a deep learning model created to handle problems related to naturallanguage processing, including text classification, translation, and language understanding.

Based on thetransformer architecture, BERT allows the model to focus on different parts of the input data based on the job by utilising self-attention methods. Pre-training is the method used to train BERT, in which a sizable corpus of text data is used to teach the model general language representations. By superimposing a few task-specific layers on top of the pre-trained model, BERT can be refined on a particular job, such text categorization, once it has been pre-trained. The initial step in utilising BERT for text classification is to refine the BERT model that has already been trained using a labelled dataset of text data. After it has been adjusted, the BERT model can be used to categorise previously unknown text input by using the representations it has acquired throughout pre- training and fine-tuning. BERT's capacity to comprehend textual context is one of its benefits, as text categorization jobs require this skill. Additionally, BERT can extract features from text input that may be applied to other models. Use of a high-quality and pertinent dataset, careful selection of the hyper parameters (e.g., number of layers, number of memory cells, learning rate), and regularisation strategies like dropout to avoid overfitting are all necessary for optimising BERT for text classification

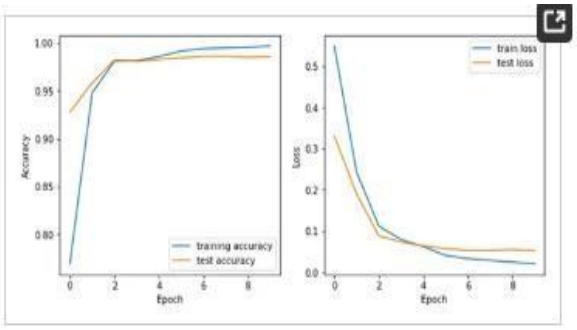
Results:



CNN model training and testing accuracy and loss over 50 epochs of training



ERT with LSTM model training and testingaccuracy and loss over the 25 epochs of training.



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

When compared to the previous state of the art in using deep learning for phishing detection, the suggested research offers insightful new information. Compared to models such as Singh et al.'s CNN method, the proposed model demonstrated a recall improvement of up to 5% and much higher accuracy . This illustrates how a bigger dataset can lead to more reliable training. Furthermore, the vanishing gradients constraint that limited earlier RNN models to 74% accuracy was overcome by the suggested LSTM model, enabling LSTM to capture a longer context and achieve 98.89% accuracy. Furthermore, scenarios in which the model's performance varied across measures were examined, and it was discovered that while LSTM best balanced recall and accuracy, BERT had a precision edge. Important insights on the relative strengths and drawbacks of different strategies were obtained by direct comparison, as opposed to separate evaluations. For example, RNNs were not as good at extracting features from raw emails as CNNs and LSTMs were. This crucial insight will help future studies modify deep learning to improve the identification of phishing attempts.

Address Bar based features	HTML and JavaScript based features	Domain based features	Abnormal based features
Use of IP address	Website Forwarding	DNS Record	Request URL
Long URL to hide the suspicious part	Status Bar Customization	Website traffic	Server form handler (SFH)
Adding prefix or suffix separated by '-' to the domain	HTML links to third-party resources like Google Analytics, Facebook, Cloudflare, etc.	Domain registration length	Links in <Meta>, <Script> and <Link> tags
URL's having '@' Symbol	Using Pop-up window	PageRank	URL of anchor
Existence of 'HTTPS' in the domain part of the URL	IFrame redirection	Google Index	Submitting information to Email
Using URL shortening services	<Body> length in tags	Number of links pointing to page	Abnormal URL
Sub domain and multi sub domains	Disabling Right Click	Statistical-reports based feature	
HTTPS (HTTP with SSL)	Missing Title	Using non-standard port	
Redirecting using '//'	Favicon		

Conclusion

Phishing emails are becoming more frequent. The goal of these scam emails is to trick gullible people into responding in a way that gives hackers access to their personal data and makes them victims. Such assaults have the potential to weaken an organization's cybersecurity defences and provide hackers access to private information. Cyberattacks frequently involve phishing emails, which is why they need to be treated carefully. The purpose of this work was to assess deep learning models utilising authentic and publicly accessible phishing datasets. After being extracted, the emails in raw format were saved as CSV files. The emails were cleaned and processed using Python programming.

References

1. ang-Jaccard, J.; Nepal, S. A survey of emerging threatsin cyber security. *J.Comput. Syst. Sci*. 2014, *80*, 973–993.[GoogleScholar][CrossRef]
2. Adewole, K.S.; Akintola, A.G.; Salihu, S.A.; Faruk, N.;Jimoh,R.G.Hybrid rule-based model for phishing URLs detection. In *Proceedings of the Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*;SpringerInternational Publishing: Basel, Switzerland, 2019; pp.119–135.[GoogleScholar]
3. Elijah,A.V.;Abdullah,A.;JhanJhi,N.Z.;Supramaniam, M.; Abdullateef, B. Ensemble and deep-learningmethodsfortwo-classandmulti-attackanomaly intrusion detection: An empirical study. *Int. J.Adv.Comput.Sci.Appl.IJACSA* **2019**, *10*,520–528.[GoogleScholar][CrossRef]
4. Alsariera, Y.A.; Elijah, A.V.; Balogun, A.O. Phishingwebsitedetection:Forestbypenalizingattributes algorithmanditsenhancedvariations. *Arab.J.Sci.Eng.* **2020**, *45*,10459–10470.[GoogleScholar][CrossRef]
5. Vrbančič,G.;Fister,I.,Jr.;Podgorelec,V.Swarmintelligence approaches for parameter setting of deeplearningneuralnetwork:Casestudyonphishing
6. websitesclassification.InProceedingsofthe8thInternational Conference on Web Intelligence, MiningandSemantics—WIMS'18,NoviSad,Serbia,25–27June2018.[GoogleScholar]
7. Zamir, A.;Khan, H.U.; Iqbal, T.; Yousaf, N.;Aslam,F.; Anjum, A.; Hamdani,M. Phishing website detectionusingdiversemachinelearningalgorithms. *Electron.ibr.*2020,*38*,65–80.[GoogleScholar][CrossRef]
8. AntiPhishingWorkingGroup(APWG).TrendsReport.Availableonline: https://apwg.org (accessedon12

December2022).
9. LeCun, Y.; Bengio, Y.; Hinton, G. Deeplearning. *Nature* 2015, *521*, 436–444. [Google Scholar][CrossRef][PubMed]
10. Castillo,E.;Dhaduvai,S.;Liu,P.;Thakur,K .S.;Dalton,A.;Strzalkows ki,T.Emailthreatdetectionusingdistinctne uralnetworkapproaches.InProceedingsoft heFirstInternationalWorkshoponSocial Threats in Online Conversations: Understandingand Management, Marseille, France, 11–16 May 2020;pp.48–55.[GoogleScholar]
11. Do, N.Q.; Selamat,A.;Kre jcar, O.; Herrera-Viedma,E.;Fujita,H.Deeplearningforphishingd etection:Taxonomy, current challenges and futuredirections. *IEEE Access: Pract. Innov. OpenSolut.* 2022, *10*,36429–36463.[GoogleScholar][CrossRef]
12. Albrecht, K.; Burri, N.; Wattenhofer, R. Spamato - An Extendable Spam Filter System. In Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS), Stanford University, Palo Alto, California, USA, July 2005.
13. Alsaid, A.; Mitchell, C.J. Installing fake root keys in a pc. In Proceedings of EuroPKI, 2005, pp. 227–239.
14. 13. Anti-Phishing Working Group. Phishing activity trends report, Jan. 2005. Available online:
15. [http://www.antiphishing.org/reports/apwg_re port_jan_2006.pdf](http://www.antiphishing.o rg/reports/apwg_report_jan_2006.pdf).
16. 14. Apache Software Foundation. Spamassassin homepage, 2006.Available_online:http://spamassassin.ap ache.org/](http://spamassassin.apache.org/).
17. 15. Apache Software Foundation. Spamassassin public corpus, 2006. Available online: [http://spamassassin.apache.org/publiccorpus/] (http://spamassassin.apache.org/publiccorpus/) .
18. 16. Breiman, L. Random forests. Mach. Learn. 2001, 45, 5–32.
19. 17. Chandrasekaran, M.; Karayanan, K.; Upadhyaya, S. Towards phishing e-mail detection based on their structural properties. In Proceedings of the New York State Cyber Security Conference, 2006.
20. 18. Chou, N.; Ledesma, R.; Teraguchi, Y.; Mitchell, J.C. Client-side defense against web-based identity theft. In Proceedings of NDSS, 2004.
21. 19. Cohen, W. Learning to classify English text with ILP methods. In Advances in Inductive Logic Programming; De Raedt, L., Ed.; IOS Press, 1996; pp. 124–143.
22. 20. Cranor, L.; Egelman, S.; Hong, J.; Zhang, Y. Phinding phish: An evaluation of anti-phishing toolbars. Technical report, Carnegie Mellon University, Nov. 2006.
23. 21. Cristianini, N.; Shawe-Taylor, J. An introduction to support Vector Machines: and other kernel-based learning methods. Cambridge University Press: New York, NY, USA, 2000.
24. 22. FDIC. Putting an end to account-hijacking identity theft, Dec.2004.Available
25. online: [http://www.fdic.gov/consumers/consumer/idtheftstudy /identity_theft.pdf](http://www.fdic.gov/consumers/co nsumer/idtheftstudy/identity_theft.pdf).
26. 23. Fette, I.; Sadeh, N.; Tomasic, A. Learning to detect phishing emails. Technical Report CMU-ISRI-06-112, Institute for Software Research, Carnegie Mellon University, June 2006. Available online: http://reports-archive.adm.cs.cmu.edu/anon/isri2006/abstracts/06-112.html.
27. 24. Gandon, F.L.; Sadeh, N.M. Semantic web technologies to reconcile privacy and context awareness. J.Web Semantics 2004, 1, 241–260.
28. 25. Gilby Productions. Tinyurl, 2006. Available online: http://www.tinyurl.com/](http://www.tinyurl.com/).
29. 26. Graham, P. Better bayesian filtering. In Proceedings of the 2003 Spam Conference, Jan 2003.
30. 27. Leiba, B.; Borenstein, N. A multifaceted approach to spam reduction. In Proceedings of the First Conference on Email and Anti-Spam (CEAS), 2004.

Author's Details:

Dr.L.JagadeeshNaik working as an Associate Professor in the Department of ECE at Holy Mary Institute of Technology and Science Keesara Hyderabad-501301 India. He has 12 Years of teaching experience in various engineering colleges.

He published 13 research articles in National and International Journals. He has attended 20 Faculty development Programs. He has attended 03 International conferences. His research areas are Wireless Sensor Networks and IOT.He has professional membership in IEI.

Ch.Mounika M tech withspecification of Electronics and communication Engineering in HolyMary Institute of Technology and Science ,Bogaram.