iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VII **Month of publication:** July 2024

DOI: https://doi.org/10.22214/ijraset.2024.63655

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VII July 2024- Available at www.ijraset.com

Deep Learning Solutions for Phishing by URL Detection

M. Robin Raj Paul[1], P. Sushanth[2], Dr. K Santhi Sree[3]
[1, 2]*Post Graduate Student, M. Tech(CNIS), Department of Information Technology, Jawaharlal Nehru Technological University, Hyderabad, India*
[3]*Professor, Department of Information Technology, Jawaharlal Nehru Technological University, Hyderabad, India*

Abstract: ***In this digital age, phishing attacks are something that are quite prevalent and are on the rise. This paper explores the various avenues for detecting such kind of attacks which will pave way to mitigating such kinds of attacks in the future. We primarily focused on proving that deep learning methods are much more efficient than traditional machine learning models; for this purpose we are evaluating the performance of a traditional machine learning model namely Naive Bayes and two deep learning models which are Convolutional Neural Networks(CNN) and Recurrent Neural Networks(RNN). The process starts with normalizing the input features and then the categorical data is transformed after which the dataset containing the URLs are loaded and are preprocessed. The performance of the models was evaluated against metrics like Accuracy, Precision, Recall and F1-Score.The end results proved that CNN was able to achieve the optimal performance and was capable of outperforming the other two models. Therefore this paper is of the view that such CNN or Neural Network empowered Models are the only way to mitigate these types of attacks and will also act as a catalyst in developing systems or models that are immune to such kinds of attacks.***

Keywords: Phishing Detection, Convolutional Neural Networks(CNN), Recurrent Neural Networks(RNN), Naive Bayes, Neural Networks.

I. INTRODUCTION

Phishing Attacks are a very well known cyberthreat that has become increasingly prevalent in this cyber age , it works by using misleading URLs to deceive users such that they provide their own private information. To curb these kinds of attacks traditional methods like blacklisting and other heuristic based approaches help but are not fully efficient thereby mandating the evolution of novel frameworks or methods to tackle such kinds of attacks. These attackers who divulge in such kinds of attacks often use interesting and clever ways of making the URLs seem legitimate thereby making the job of those responsible to safeguard systems much more difficult, these attacks can take advantage of the educated and digitally aware people, needless to say that it's a bane for those who are not digitally literate. The major problem associated with such kinds of attacks is that we need models that can constantly update , learn and detect on their own thereby throwing such kinds of problems right in the ballpark of deep learning. This Paper also deals with three such models of which one namely Naive Bayes is a Machine Learning Algorithm whereas the other two that is CNN and RNN are Deep Learning Approaches/Techniques. In this paper we, with the help of open access resources and dataset we have created efficient models and trained them well to detect such kinds of malicious URLs, we have also provided various training loss curves and confusion matrices and also compared the performance of the three models in terms of their Accuracy, Precision, Recall and F1-Score.

II. RELATED WORK

Detection of Phishing attack is one such endeavour that several researchers have been at since more than a decade. The Researchers have utilized several machine learning, deep learning models and even tried to create hybrid versions of models that would perform well, therefore most of the related or existing work will also revolve around the aforementioned domains. However two commonly followed approaches are as follows:

A. Customary AI based Approaches

Rule based approaches can be used to detect phishing attacks, these are direct, efficient and use logical reasoning for their detection purposes, however these require regular updates and can be easily bypassed. To ensure perfect phishing detection Jain and Gupta[8],Moghimi and Varjani[11] and Satheesh Kumar [12] have looked into several rule based techniques.

1) *Whitelist-based Methods and Data Mining for Associative Classification:* In these methods such phishing URLs are arranged categorically using associative classification by utilizing data patterns, these can unravel intricate relationships in the data and therefore are preferred when compared to rule based approaches. It is still a hassle to keep an updated whitelist in a dynamic web environment and the processing power required to use the associative classification can be very high, this analysis along with pros and cons was established by Azeez , Abdelhamid [10] and Jain and Gupta[8]
2) *Visual Similarity-Based Approaches:* These methodologies depend on the visual aspect of a webpage to detect phishing attacks or phishing URLs. This basically works by comparing each and every aspect of the legitimate and illegitimate website manually by keeping them one beside the other. The only disadvantage is that it might not function well with websites or pages that change dynamically, because they will require either of two things i.e. either complex obfuscation or a whole lot of processing capability. This concept and research on this was done primarily by Jain and Gupta[13], Medvet [14] and Zhou [16]

B. *Deep Learning Based Approaches*

1) *Neural Networks and Networks with Long Short Term Memory[LSTM]:*Sequential data such as the characters in a URL which is sequential data can be taken care of nicely by RNNs and LSTMs. These learn patterns over sequencies and therefore are a best fit for analysis of URL Structure and Semantics. These are extremely good at gathering temporal connections in an incremental fashion and contextual information thereby putting it above all other techniques in identifying/detecting phishing attempts. Researchers such as Huang [18], Sahingoz[19] and Singh [20] have proved how well these models fare compared to others.
2) *Attention-Based Models:* Attention based models or attention based Neural Networks have the capacity to dynamically balance the relative value of different URL Components and therefore improves the model's performance and accuracy. It can give priority to important characteristics which go a long way in detecting phishing attacks. Research by Sahingoz [19], Huang [18] and Singh [20] have displayed the pros and cons of working with attention based mechanisms and how good they are in detecting phishing attempts.
3) *Convolution Neural Networks[CNN]:*CNNs have high component extraction capabilities, these originally used for image recognition but can be modified for phishing location. Intricate patterns can be easily identified which makes it easier to deduce the characteristics that can help differentiate the legitimate and illegitimate URLs, they can also identify progressive instances, the utilization of CNNs for this task has been successfully demonstrated by research of Huang [18], Sahingoz [19] and Singh [20].
4) *Hybrid Approaches:* Several researches have tried to create hybrid versions of the models by combining both RNN and CNN thereby accumulating benefits that both of them have to offer; Such a hybrid version can detect sequential and spatial samples inside URLs , therefore it can give high accuracy and performance; the research done by Huang [18], Sahingoz [19] and Singh [20].

All of this research has been quite instrumental in improving systems that detect phishing URLs until now, but since the attacks are becoming much more diverse and sophisticated;
depending on static, rule-based or outwardly focused strategies will no longer be helpful therefore profound learning models are a dignified answer for coping with such type of attacks. Therefore in order to determine the most effective models for phishing detection this paper distinguishes the results of the RNNs, CNNS and the conventional Naive Bayes classifiers. This will help in making the Internet safe for all users.

III.PROPOSED WORK

This Project identifies a way to detect phishing URLs to curb the danger presented by phishing attacks. This paper provides a way to improve and achieve the highest accuracy in detecting such kind of attacks using Convolutional Neural Networks and Recurrent Neural Networks; CNNs can identify hierarchical characteristics from the data and can provide a detailed and open view of all the URL characteristics whereas RNN is used to process sequential data. The way in which this works involves the following steps namely planning, preparing and assessing both the RNN and CNN models in order to distinguish phishing URLs from the dataset of marked URLs. RNN is used to learn the temporal dependencies that have the potential of identifying signs that are indicative of phishing. CNN on the other hand is built in such a fashion that it treats the URLs like a one-dimensional model so it is capable of distinguishing spatial features and other characteristics of URLs, therefore the amalgamation of both these models will provide a complete view of detecting such kind of phishing attacks and or URLs ; After which comparing and contrasting the performance of these models with the most preferred traditional machine learning model namely Naive Bayes classifier which is very popular for text based classification and will provide a real time example of how these can be implemented in the real world.

A. Dataset

The dataset used for this project is the PHI-2018 Phishing URL Dataset[21].This dataset holds a vast collection of URLs that have numerous features and other properties that can be exploited in order to train and evaluate the models in such a manner that they are extremely capable of detecting any kind of phishing attacks. This dataset has a total of 2,35,794 entries and also has 56 features to choose from. It is on the basis of these features that we decide the legitimate nature of the phishing URLs . This dataset was taken from an open source UCI Mchine Learning repository and it is open for academic and research utilization.

The dataset includes the following features:

1) FILENAME: The filename that has the URL in it.
2) URL:The complete address.
3) URLLength: The URL's total length.
4) Domain: The URL's domain name.
5) DomainLength: The domain name's length.
6) IsDomainIP: Returns a value of 0 if the domain is an IP address and 1 otherwise.
7) TLD: The URL's top-level domain.
8) URLSimilarityIndex: A metric that indicates how similar a URL is to other well-known, valid URLs.
9) CharContinuationRate : The rate at which characters continue without a space is known as the CharContinuationRate.
10) TLDLegitimateProb: The likelihood that the TLD is authorised.
11) NoOfSubDomain: The URL's total number of subdomains.
12) HasObfuscation: 0 means the URL is not obfuscated, while 1 means it is.
13) IsHTTPS: denotes whether HTTPS is used by the URL (0 for no, 1 for yes)
14) NoOfImage: The quantity of pictures on the page.
15) NoOfCSS: The amount of CSS files that the website has links to.
16) NoOfJS: The amount of JavaScript files that are connected on the page.
17) label: Denotes a phishing URL (1 being phishing, 0 being legitimate).

FILE NAME	URL	Domain	TLD	label
521848.txt	https://www.southbankmosaics.com	www.southbankmosaics.com	com	1
31372.txt	https://www.uni-mainz.de	www.uni-mainz.de	De	1
597387.txt	https://www.voicefmradio.co.uk	www.voicefmradio.co.uk	Uk	1
554095.txt	https://www.sfnmjournal.com	www.sfnmjournal.com	Com	1
151578.txt	https://www.rewildingargentina.org	www.rewildingargentina.org	Org	1

Table -1: Dataset Example

B. Architecture

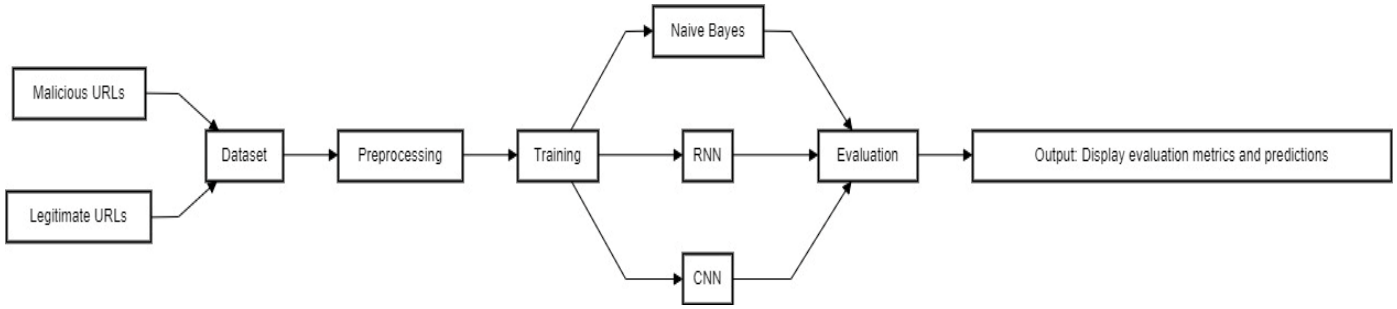


Figure -1: Architecture

This particular architecture typically depicts a comprehensive methodology for detecting phishing URLs by employing three models namely Convolutional Neural Networks(CNN], Recurrent Neural Network(RNN) and Naive Bayes. The process starts with the dataset that contain both the authentic and fraud URLs, which then goes through preprocessing in order to normalize and standardise the information after which the data that has been preprocessed is used to train the models, to be specific CNN,RNN and Naive Bayes. The Final Performance of each model is then used to determine their overall performance and the comparison results are portrayed in order to determine how well they can perform in real time situations or the real world

C. Methodology

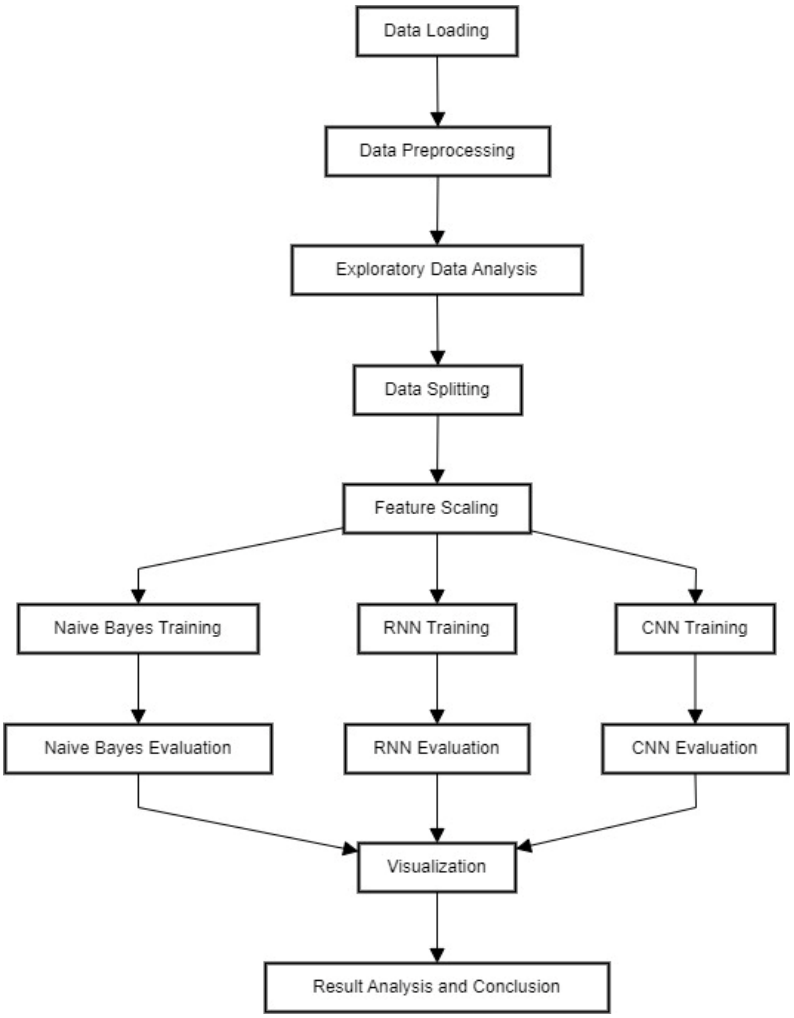


Figure -2: Flowchart

The first step is data loading and preprocessing , since the dataset is in a csv file, so the first step is to load the file containing the features using Pandas then the next step is to implement error handling functions for potential parsing issues. Then the most important step is to convert non-numeric columns to numeric format using python's and scikit-learn's 'LabelEncoder' function also such operations are done to ensure that the data is in a numerical format for the machine learning models.

The next step is Exploratory Data Analysis, this can be implemented with matplotlib, visualizing the dataset using a label distribution graph which shows legitimate vs phishing URLs, this helps in understanding class balance and dataset distribution. The next task is Data Splitting which can be implemented using Scikit-learn's 'train_test_split' to divide the dataset in a 80:20 ratio for training and testing respectively then evaluating the validity of the trained models; Also the data should be Normalized/Standardized using Scikit-learn's 'StandardScaler' also we have to ensure that features are lying on the same range in order to enhance the accuracy which is crucial for deep learning models like RNN and CNN.

The next important step is Model Training, we have first trained three distinct models such as Naive Bayes(Gaussian Naive Bayes), RNN and CNN. The Naive Bayes can be implemented using 'GaussianNB' classifier from Scikit-learn, the RNN and CNN can be implemented using PyTorch('nn.RNN'-RNN and 'nn.Conv1d'-CNN).Then the models have to be optimized using Adam Optimizer i.e. 'optim.Adam' and the cross-entropy loss has to be minimized using 'nn.CrossEntropyLoss'.Then the Models have to be trained for several iterations or epochs using loops like 'for epoch in range(num_epochs)'.

The final and most important step is Model Evaluation and Analysis followed by Visualisation the evaluation metrics are computed using Scikit-learn: Accuracy, Precision, Recall and F1 score. Then the Confusion Matrices should be generated to evaluate the model's performance in it's ability to differentiate between Authentic and Fraud URLs . Then Curves such as Training loss curves are plotted. Finally the performance of all the three models are displayed.

IV. EXPERIMENTAL ANALYSIS AND RESULTS

The majority of the model training and evaluation are heavily dependent on the number of legitimate and phishing URLs in the dataset, getting a thorough understanding of the URL distribution is very important in order to successfully categorize the classes, therefore such kind of a visualization provides a clear understanding of the dataset's class balance.

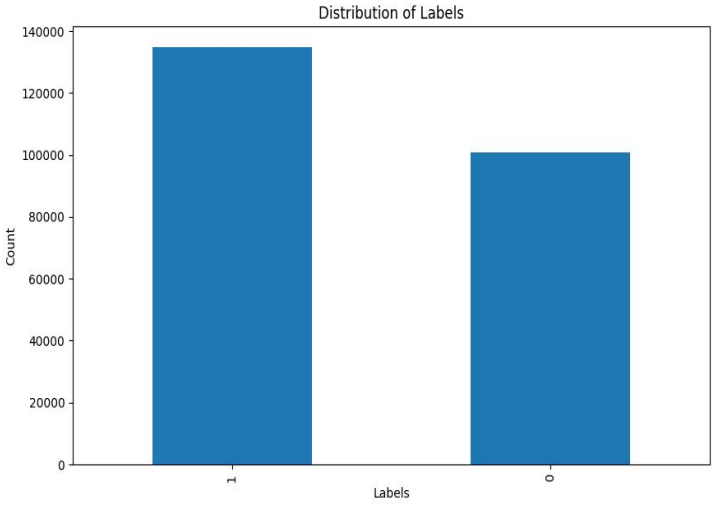


Figure -3: Dataset Label distribution

The training losses of RNN and CNN are as shown below, the RNN Training loss goes upto a peak of 170 before saturating around 150. The CNN starts from about 30 and goes towards 0. Therefore this is a clear indication that that CNN Model is learning at a very positive and greater rate when in comparison to the RNN Model.

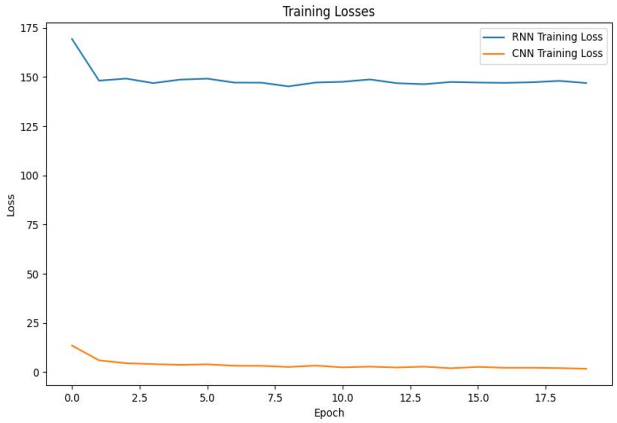


Figure -4: Training Loss Curves

For Naive Bayes the result indicated that there were 20,114 True Negatives with 26,824 True Positives the model performs nicely however there are 211 misleading Negatives and 10 Up-Sides as well which depicts Great Execution but Some misclassification.

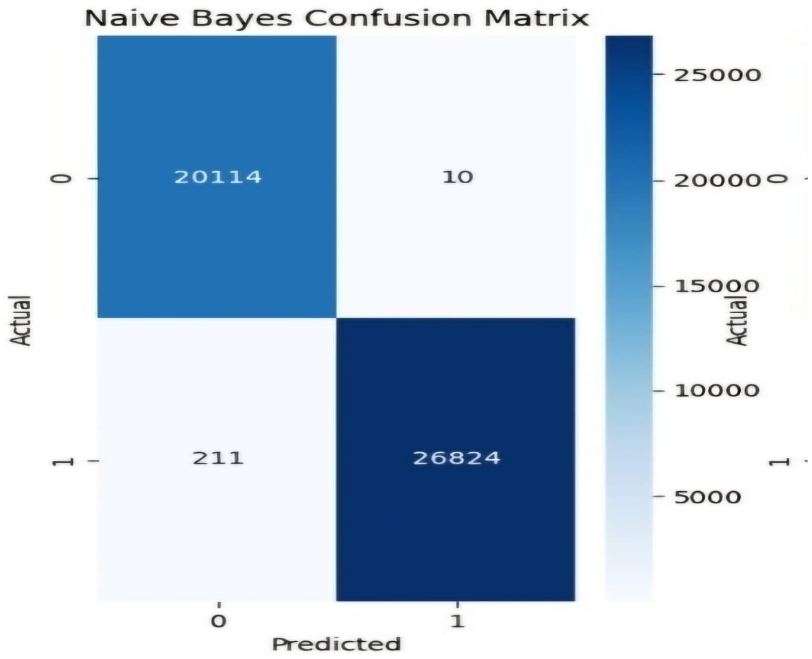


Figure-5: Confusion Matrix for Naive Bayes

For RNN the result shows that 20,123 True Negatives and 27,035 True Positives have been obtained, it is near to perfect since it has only one False Positive and Zero False Negatives

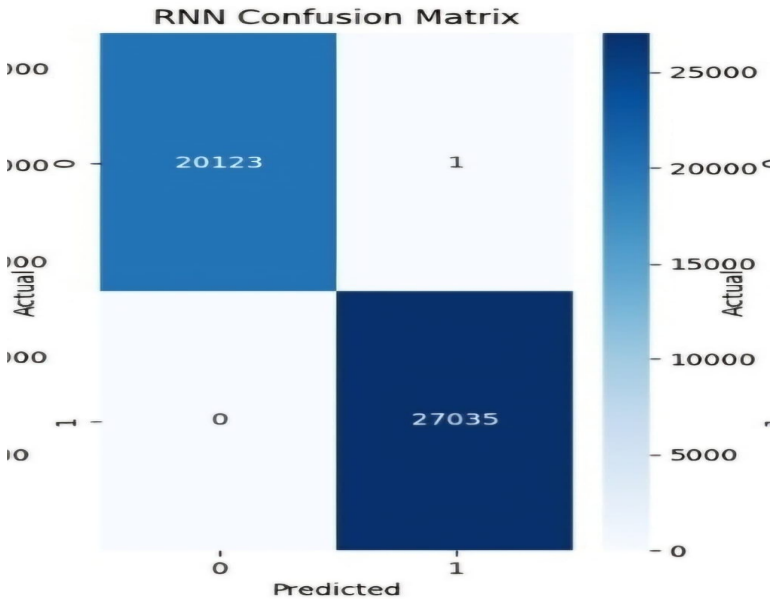


Figure-6:Confusion Matrix for RNN

For CNN, the result shows that it received 20,123 True Negatives and 27,034 True Positives, CNN also performs near to perfection and in similar lines to RNN, however it has One False Positive and One False Negative.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VII July 2024- Available at www.ijraset.com

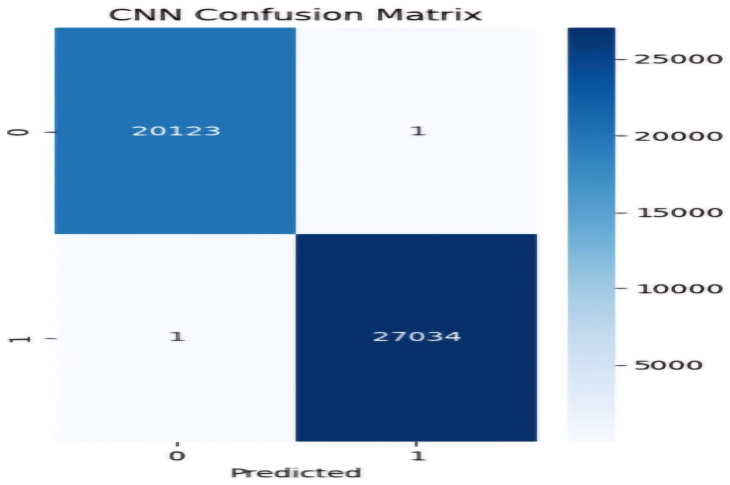


Figure-7:Confusion Matrix for CNN

A. Evaluation Criteria

The performance of the model is evaluated against parameters like Accuracy, Precision, Recall and F-Score. The results are as depicted.

Model	Accuracy	Precision	Recall	F1-Score	Comments
Naive Bayes	0.9953	0.9954	0.9953	0.9953	Good overall performance with slight misclassifications.
RNN	0.9999	0.9999	0.9999	0.9999	Classification that is close to perfect, with only a few misclassifications.
CNN	0.9999	0.9999	0.9999	0.9999	Excellent performance with virtually no classification errors.

Table -2: Results

Therefore after considering the performance of all the models and analysing them it has been found that CNN is the best model that outperforms all other models in various aspects,making it the most optimal solution amongst all.

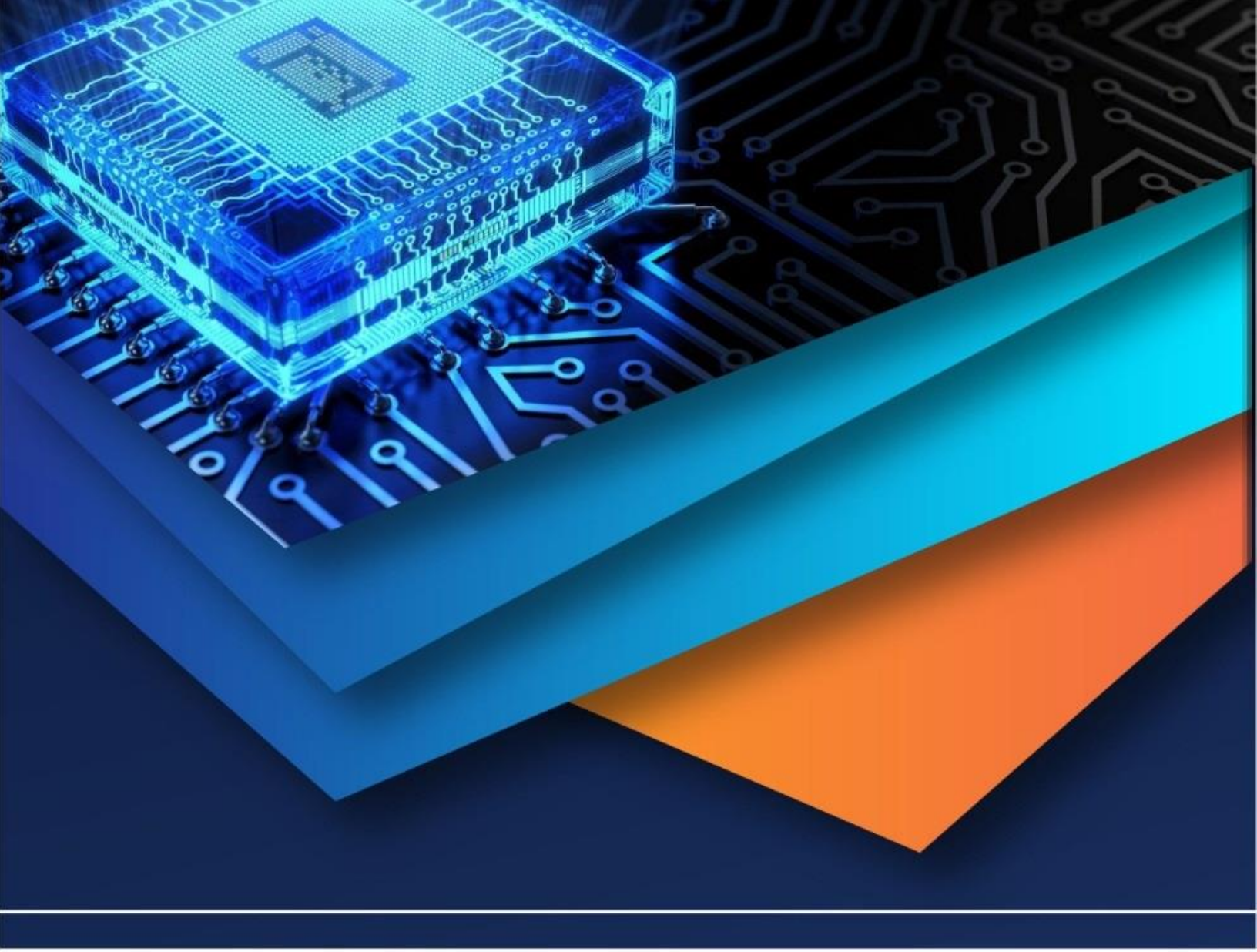
V. CONCLUSIONS

In this study we have executed and analyzed different learning models to distinguish phishing URLs. We have highlighted the strength of deep learning techniques in detecting such phishing attacks or threats by utilizing a improved CNN, RNN and then we have compared it's performance with a machine learning algorithm called Naive Bayes. The Bayes Model acted as the reference and the RNN and improved CNN have given us exceptional design acknowledgement capacities, Through our results we were able to come to the conclusion that Accuracy, Precision, Recall and F1-Score of the improved CNN model were by far the best and finer than the results of those of Naive Bayes. The CNN model includes several convolutional layers, batch normalization and dropout systems showed higher capability in differentiating potential phishing URLs. The results itself are an indication of how important these models can be in securing the internet and it's users.

Further expansion on this work will involve formulating the formed models into ongoing phishing location frameworks thereby upgrading their importance and application in network safety. Having a much more diverse dataset will always help the model to train effectively and classify further such events properly; going much more in depth rather than URL analysis, integrating the Natural Language Processing techniques can also guarantee the detection of phishing content and is capable of offering a comprehensive security detail. Therefore this can act as the basis and can keep cyberthreats and risks at bay.

REFERENCES

[1] Anti-Phishing Working Group. (Sep. 2022). Phishing Attacks Trends Report-Q2 2022. Accessed: Oct. 15, 2022. [Online]. Available: https://apwg.org/trendsreports/
[2] Cloudflare's 2023 Phishing Threats Report. Accessed: Oct. 1,2023.[Online].Available:https://www.cloudflare.com/lp/2023-phishing-report/
[3] M. Volkamer, K. Renaud, B. Reinheimer, and A. Kunz, ''User experiences of TORPEDO: Tooltip-powered phishing email detection,''Comput. Secur., vol. 71, pp. 100–113, Nov. 2017, doi: 10.1016/j.cose.2017.02.004.
[4] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita,''Deep learning for phishing detection: Taxonomy, current challenges and future directions,'' IEEE Access, vol. 10, pp. 36429–36463, 2022, doi: 10.1109/ACCESS.2022.3151903.
[5] T. Mahara, V. L. H. Josephine, R. Srinivasan, P. Prakash, A. D. Algarni, and O. P. Verma, ''Deep vs. shallow: A comparative study of machine learning and deep learning approaches for fake health news detection,'' IEEE Access, vol. 11, pp. 79330–79340, 2023, doi: 10.1109/ACCESS.2023.3298441.
[6] Google Safe Browsing. Accessed: Oct. 1, 2023. [Online]. Available: https://safebrowsing.google.com/
[7] (2019). Office 365 Advanced Threat Protection Safe Links. Accessed:Jul. 10, 2023. [Online]. Available: https://docs.microsoft.com/enus/office365/securitycompliance/atp-safe-links
[8] A. K. Jain and B. B. Gupta, ''A novel approach to protect against phishing attacks at client side using auto-updated white-list,'' EURASIP J. Inf. Secur., vol. 2016, no. 1, pp. 1–11, Dec. 2016, doi: 10.1186/s13635- 016-0034-3.
[9] N. A. Azeez, S. Misra, I. A. Margaret, L. Fernandez-Sanz, and S. M. Abdulhamid, ''Adopting automated whitelist approach for detecting phishing attacks,'' Comput. Secur., vol. 108, Sep. 2021, Art. no. 102328, doi: 10.1016/j.cose.2021.102328.
[10] N. Abdelhamid, A. Ayesh, and F. Thabtah, ''Phishing detection based associative classification data mining,'' Expert Syst. Appl., vol. 41, no. 13, pp. 5948–5959, Oct. 2014, doi: 10.1016/j.eswa.2014.03.019.
[11] M. Moghimi and A. Y. Varjani, ''New rule-based phishing detection method,'' Expert Syst. Appl., vol. 53, pp. 231–242, Jul. 2016, doi: 10.1016/j.eswa.2016.01.028.
[12] M. SatheeshKumar, K. G. Srinivasagan, and G. UnniKrishnan, ''A lightweight and proactive rule-based incremental construction approach to detect phishing scam,'' Inf. Technol. Manage., vol. 23, no. 4, pp. 271–298, Dec. 2022, doi: 10.1007/s10799- 021-00351-7.
[13] A. K. Jain and B. B. Gupta, ''Phishing detection: Analysis of visual similarity based approaches,'' Secur. Commun. Netw., vol. 2017, pp. 1–20, Oct. 2017, doi: 10.1155/2017/5421046.
[14] E. Medvet, E. Kirda, and C. Kruegel, ''Visual-similarity-based phishing detection,'' in Proc. 4th Int. Conf. Secur. Privacy Commun. Netowrk, Sep. 2008, pp. 1–6, doi: 10.1145/1460877.1460905.
[15] W. Liu, X. Deng, G. Huang, and A. Y. Fu, ''An antiphishing strategy based on visual similarity assessment,'' IEEE Internet Comput., vol. 10, no. 2, pp. 58–65, Mar. 2006, doi: 10.1109/MIC.2006.23.
[16] Y. Zhou, Y. Zhang, J. Xiao, Y.Wang, andW. Lin, ''Visual similarity based anti-phishing with the combination of local and global features,'' in Proc. IEEE 13th Int. Conf. Trust, Secur. Privacy Comput. Commun., Sep. 2014, pp. 189–196, doi: 10.1109/TRUSTCOM.2014.28.
[17] G. Varshney, M. Misra, and P. K. Atrey, ''Improving the accuracy of search engine based anti-phishing solutions using lightweight features,'' in Proc. 11th Int. Conf. Internet Technol. Secured Trans. (ICITST), Dec. 2016,pp. 365–370, doi: 10.1109/ICITST.2016.7856731.
[18] Y. Huang, Q. Yang, J. Qin, and W. Wen, ''Phishing URL detection via CNN and attention-based hierarchical RNN,'' in Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng., Aug. 2019, pp. 112–119, doi: 10.1109/Trustcom/BIGDATASE.2019.00024.
[19] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, ''Machine learning based phishing detection from URLs,'' Expert Syst. Appl., vol. 117, pp. 345–357, Mar. 2019, doi: 10.1016/j.eswa.2018.09.029.
[20] S. Singh, M. P. Singh, and R. Pandey, ''Phishing detection from URLs using deep learning approach,'' in Proc. 5th Int. Conf. Comput., Commun. Secur. (ICCCS), Oct. 2020, pp. 1–4, doi: 10.1109/ICCCS49678.2020.9277459.
[21] Datset:Prasad,Arvind and Chandra,Shalini. (2024). PhiUSIIL Phishing URL (Website). UCI Machine Learning Repository. https://doi.org/10.1016/j.cose.2023.103545.
[22] O. K. Sahingoz, E. BUBEr and E. Kugu, "DEPHIDES: Deep Learning Based Phishing Detection System," in IEEE Access, vol. 12, pp. 8052-8070, 2024, doi: 10.1109/ACCESS.2024.3352629.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY