Phish-Defence: Phishing Detection Using Deep Recurrent Neural Networks

Aman Rangapur, Tarun Kanakam and Dhanvanthini P

arXiv:2110.13424v4 [cs.CR] 6 Sep 2022

Abstract—In the growing world of the internet, the number of ways to obtain crucial data such as passwords and login credentials, as well as sensitive personal information has expanded. Page impersonation, often known as phishing, is one method of obtaining such valuable information. Phishing is one of the most straightforward forms of cyber-attack for hackers, as well as one of the simplest for victims to fall for. It can also provide hackers with everything they need to access their targets' personal and corporate accounts. Such websites do not provide a service but instead gather personal information from users. In this paper, we achieved state-of-the-art accuracy in detecting malicious URLs using recurrent neural networks. Unlike previous studies, which looked at online content, URLs, and traffic numbers, we merely look at the text in the URL, which makes it quicker and catches zero-day assaults. The network has been optimized so that it may be utilized on tiny devices like Mobiles, Raspberry Pi without sacrificing the inference time.

I. INTRODUCTION

The rapid growth and adoption of new technologies, such as smart devices and 5G connectivity, has resulted in the rapid growth of internet-based services. Some of the services are essential for day-to-day activities. The increasing number of people who utilize web-based services demonstrates their popularity over the years. The goal is to increase the availability and accessibility of web-based services used frequently so that users do not face any inconvenience [1], [2]. Nonetheless, the unlimited access and usage of web-based services or technology on the internet offers potential for cyber-attacks because of non-uniform cyberspace regulations or processes [3].

There has been incredible growth in the number of illegal activities with increased malicious and destructive content in the networks over the past several decades. According to statistics, financial transactions, online gaming services, and social media are among the most popular web-based services with a large user base. These services are highly prone to cyber assaults. The way to protect the user from cyberattacks is called Cybersecurity [4]. Cybersecurity is responsible for safeguarding, prohibiting, and recovering all the assets that use the internet from cyber assaults.

Aman Rangapur is with the School of Computer Science and Engineering, VIT-AP University, Andhra Pradesh, India, 522237. Email: amanrangapur@gmail.com

Tarun Kanakam is with the School of Computer Science and Engineering, VIT-AP University, Andhra Pradesh, India, 522237, Email: kanakamtarunkumartk@gmail.com

Ajith Jubilson is with the School of Computer Science and Engineering, VIT-AP University, Andhra Pradesh, India, 522237, Email: ajith.jubilson@vitap.ac.in

Phishing is a social-engineering-based cyber-attack that aims to steal personal data such as credit card information, login credentials, and passwords. Most of the time, the hacker registers a counterfeit domain address, meticulously creates a website that resembles the organization's actual website, then sends a mass email to thousands, instructing recipients to click on a link on the fake website. These assaults primarily result in the loss of crucial data, financial loss, intellectual property loss, and deterioration of trust and national security [5], [6]. Several cybersecurity professionals and researchers have proposed and built a variety of anti-phishing techniques for identifying phishing websites [7]. Elmer Lastdanger proposed in his study that "Phishing is a scalable act of deception whereby impersonation is used to obtain information from a target" [8].

Machine learning is a field within computer science that differs from traditional computing. In traditional computing, algorithms follow the rules and get executed, whereas Machine Learning algorithms allow computers to train on data and use statistical analysis to output the value. In addition, neural networks and genetic algorithms make the code very efficient. Neural networks are a means of machine learning, in which a computer learns to perform some task by analyzing training data; it's a process that mimics how the human brain operates. A neural network generates the best possible result without redesigning the output criteria.

The motivation of this research is to obtain state-of-the-art accuracy using Recurrent Neural Networks, which is efficient in detecting malicious websites. Furthermore, our model has been optimized and can be utilized on tiny devices like Raspberry Pi without sacrificing the inference time.

II. BACKGROUND STUDY

The constant development of web services and e-commerce platforms has encouraged many phishers and criminals to develop new ways to exploit and deceive novice users into sending their financial information [9], [10].

Phishing is by far the most common cybercrime as of 2022, with the FBI's Internet Crime Complaint Centre reporting more than twice as many phishing cases than any other kind of cybercrime. In a web phishing attack using email, the phisher deceives web users by developing a fake website to steal the financial and personal information of the users. Web phishing attacks can be achieved in many ways [10], [11]. Initially, the phisher makes a phishing website, which looks like the original website in its appearance. Eventually, many emails containing hyperlinks to the phishing website
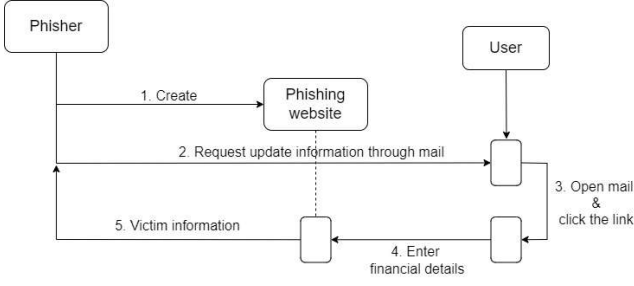
Fig. 1. Life cycle of a web phishing attack.

are continually sent to users by the phisher, which requests validating or updating credentials and financial information to deceive the victims [12]. Finally, the target victims are redirected to the phishing website when the hyperlinks are clicked. When the victim enters the information through the phishing website, the phisher can fully control the victim's financial information. As a result, many financial and identity thefts can be executed after successful web phishing attacks [10], [11], [13]. Online social networks, emails, messengers, blogs, forums, mobile applications, and voice over IP (VoIP) are different communication platforms. Phishing attacks may target these communication systems. Figure 1 shows the life cycle of a web phishing attack.

Phishing attempts against software-as-a-service (SaaS), webmail users were the most common in 2021. During the COVID-19 pandemic, the number of phishing attacks increased significantly. Approximately 70% of healthcare facilities with less than 500 employees are targeted, indicating that these smaller facilities have probably weaker security systems due to low-security budgets. It's also been observed that 75% of phishing websites use SSL (Secure Sockets Layer) security, which means that a website that uses the SSL protocol is not always legitimate. The countermeasures in tackling phishing attacks are categorized as follows.

A. Traditional methods

1) Blacklisting and whitelisting: Blacklisting and whitelisting are phishing URL lists in which the specified URL is compared to a preset phishing URL. The disadvantage of this methodology is that the blacklist cannot cover all phishing sites because a newly created phishing site takes a long time to get updated in the list. This delay in adding the suspect site to the list might be enough for the attackers to achieve their goals [1], [9]. Whitelisting methods are contrary to blacklisting methods. It determines if a URL is legitimate by comparing it to the "whitelist" Database. The "whitelist" Database mainly contains a list of well-known legitimate URLs [10].

2) Visual Similarity: By analyzing the appearance of sites, Visual Similarity techniques may distinguish phishing sites from genuine ones. Attackers embed images, Flash, ActiveX, and Java Applet instead of HTML text to avoid phishing detection. As enrollment items appear on phishing site webpages, visual similarity-based detection algorithms may immediately distinguish them. It classifies URLs based on visual similarity strategies using HTML Document Object Model tree, CSS similarity, visual perception, visual characteristics, pixel-based, and hybrid approaches [14].

B. Non-Traditional methods

1) Machine Learning and Deep Learning: To create phishing detection models, traditional machine learning methods (also known as shallow learning) such as Support Vector Machines (SVM), Decision Trees (DT), and Random Forests (RF) can be used.

The Hybrid Ensemble Feature Selection (HEFS) presented by Chiew K. L. et al. [15] is a feature selection structure for a machine learning-based phishing detection system. In the first step of HEFS, an original Cumulative Distribution Function gradient (CDF-g) calculation is used to generate critical feature sub-sets, which are then used as an input to a data perturbation collection to generate optional feature subsets. The subsequent stage infers pattern features from the optional feature subsets by utilizing a function disturbance set. Compared to SVM, Naive Bayes, C4.5, JRip, and PART classifiers, HEFS performs better when coordinated with the Random Forest classifier.

Deep Learning, which comes under machine learning, focuses on building deep networks with various layer types such as convolutional, pooling, dropout, and fully connected layers. Deep learning-based phishing detection models are built using algorithms like Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Autoencoders.

Wei W. et al. [16] suggested a deep neural network with convolutional layers for identifying phishing sites based solely on the URL address content. Despite previous research that looked at URLs, traffic statistics, and online content, they only looked at the URL text. The network they demonstrated had been appropriately updated, so it can now be used even on mobile phones without affecting its performance. As a result, the approach is faster and can detect zero-day threats.

2) Heuristic based methods: In Heuristic-based methods, minor characteristics of a website are extracted to determine if the URL is malicious or legitimate. In contrast to blacklisting approaches, heuristic-based solutions may detect new phishing websites on a regular basis [17]. Intelligent machine learning classifiers are trained using certain phishing and authentic websites as training data and then utilized to detect newly developed phishing websites accurately [11].

Rao R. S. et al. [18] developed a heuristic technique with TWSVM (twin support vector machine) classifier to detect malicious enrolled phishing websites. Moreover, websites are facilitated on arrangement servers. Their research has focused on detecting phishing sites hosted on arrangement domains by inspecting the sign-in page and the website's main page. The hyperlink and URL – based characteristics are used to identify maliciously enlisted phishing websites.

3) Hybrid Techniques: A combination of different approaches is used to create a better model in terms of accuracy and precision. A DTOF-ANN ("Decision Tree and Optimal Features based Artificial Neural Network"), which is a neural-network phishing identification model based on a decision

tree and ideal feature selection, was proposed by Zhu E. et al. [19] to tackle over-fitting. The initial step was to improve the traditional K-medoids clustering technique by selecting a consistent number of introduction centers to eliminate copy dots in popular datasets. In Addition to DTOF-ANN, optimal feature selection based on innovative features, such as decision trees and neighborhood search techniques, is meant to eliminate the ineffective and undesirable parts. Finally, the optimal architecture of the neural network classifier is constructed by modifying boundaries and training the neural network classifier with the specified ideal features.

III. RELATED WORK

Researchers have studied phishing attacks repeatedly, but most of them were not entirely perfect. Despite getting solutions and accuracy, it required complicated calculations, which made it difficult to use and took a good amount of inference time to detect.

Using a variety of characteristics, da Silva et al. [20] developed a phishing prediction model. The suggested model assesses static characteristics like Keywords and patterns in the phishing URL. A qualitative examination of these characteristics that do not define elements such as relationships and similarities between qualities was undertaken in addition to the quantitative data in the research.

A neural network model based on decision trees and optimal feature selection was suggested by Zhu et al. [19]. Using the classic K-medoids clustering process, they created an incremental selection strategy to eliminate duplicate points from public datasets. An optimum feature selection method was created to exclude the negative and useless traits. Tan et al. [21] developed graph theory-based anti-phishing techniques. The suggested technique's initial stage is deleting hyperlinks from the web page in question and bringing in relevant local web pages.

For phishing detection applications, Zabihimayvan and Doran used Fuzzy Rough Set (FRS) theory to pick significant characteristics from the dataset [12]. Rough Set (RS) theory is extended into the Fuzzy Rough Set (FRS) theory. Websites A and B are both phishing websites. Their features A and B have the same value, so RS is a method to find a decision boundary by calculating the equality of each data point based on certain features and the same classes, such as websites A and B being both phishing websites and their features A and B having the same value. RS is appropriate for the original dataset used in this paper, in which the features are used as a discrete value, i.e., they are a collection of 1, 0, -1 elements. However, after the dataset executes the nominalization process, the value of the feature is transferred to a continuous number from 0 to 1, and the FRS strategy is applied.

In 2021, El-Rashidy proposed a new method for selecting characteristics for an online phishing detection model [22]. There are two parts to the feature selection process. The first phase determines the impact of each feature's absence by training the random forest model on a new dataset with one feature removed. A feature queue graded from high to low accuracy is created after removing each element in the loop.

The model is then trained and tested by starting with one feature, adding a new feature from the ranking feature list each time to create a dataset, calculating the accuracy of each time, and ultimately finding the feature subset with the most excellent accuracy. This strategy is used to find the most useful feature subset. However, because each new dataset must go through the algorithm training and testing procedure, there is a lot of computational complexity and time involved.

Yang et al. [23] proposed a new technique for phishing detection based on an inverted matrix online sequential over-learning machine that characterizes a website using three characteristics. To reduce matrix inversion, they employed the Sherman Morrison Woodbury equation. They introduced the online queue extreme learning machine to modify the training model.

Subasi and Kremic [24] also developed a system for detecting phishing websites. They classified websites as legitimate or phishing using a variety of machine learning algorithms. They presented two types of learners (Adaptive Boosting (AdaBoost) and Multiboost) to help with anti-phishing and attack detection.

For phishing and Botnet attack detection, De La Torre Parra et al. [25] suggested a cloud-based deep learning architecture. The model consists of two basic security mechanisms Distributed Convolutional Neural Network (DCNN) model that works together to detect phishing and application layer Distributed Denial of Service (DDoS) attacks and a cloud-based temporary long short-term memory (LSTM) network model hosted on the backend to detect botnet attacks and feed CNN to detect phishing attacks.

The primary focus of Yi et al. [26] was developing a deep learning framework for detecting phishing websites. The researchers created two types of web phishing features original features and interactive features. These characteristics are employed in a Deep Belief Networks-based detection model (DBN). The DBN-based detection model produced promising results during experiments utilizing natural IP streams.

Wei et al. [27] suggested a lightweight deep learning technique for detecting fraudulent URLs, allowing for a real-time and energy-efficient phishing detection system. They demonstrated that the suggested technology could identify phishing in real time using an energy-efficient integrated single board computer using website URLs. Table I depicts other existing machine learning models for phishing detection.

In this paper, we have developed and assessed web phishing detection models using Recurrent Neural Networks such as LSTM and GRU to achieve maximum accuracy and precision without compromising inference time for detecting malicious websites on small devices.

IV. METHODOLOGY

Long Short-Term Memory [35], [36] was introduced to address the long-term dependency problem. LSTM has a hidden state in which an LSTM block would output and an internal cell that maintains the information across a temporal context. The cell stores the long-term information, and the LSTM can erase, write and read information from that cell

TABLE I
EXISTING MACHINE LEARNING MODELS FOR PHISHING DETECTION

Model or Algorithm	Type	Dataset	Challenges	Limitations	Accuracy
Random Forest (RF_1) [28]	Single	ISCXURL-2016	Achieved high accuracy and low response time without relying on third-party services and using limited features extracted from a URL.	Neither used multiple different datasets to train the model, nor compared the results or evaluated the robustness of the model.	99.57%
Adaboost [29]	Single	Websites (PhishTank, MillerSmiles, Google Search), size of the dataset not mentioned, each instance has 30 features.	The proposed model used Weka 3.6, Python, and MATLAB.	Didn't compare the results and evaluate the robustness of the model.	98.30%
Random Forest (RF_2) [30]	Single	Websites (PhishTank, OpenPhish, Alexa, online payment gateway) 5223 instances, 2500 phishing URLs, 2723 legitimate URLs. Has 20 features.	20 features are manually extracted, some features need to be obtained by calling a third-party service, and some features need to parse the website's HTML source code.	Did not use multiple different datasets to train the model, compare the results, or to evaluate the robustness of the model. The experimental dataset is small.	99.50%
PSL[1] + PART [31]	Hybrid	Websites (PhishTank, Relbank) 30,500 original instances, 20,500 phishing URLs, 10,000 legitimate URLs. Has 18 features.	Extracted 3000 comprehensive features and applied different set of parameters to ML models to compare the experimental results.	The legitimate URLs in the dataset are all related to banks, and some features are limited to e-banking websites.	99.30%
Recurrent Neural Network (RNN) + Convolutional Neural Network (CNN) [32]	Deep learning	Websites (PhishTank, Alexa, etc), 490,408 instances, 245,385 phishing URLs, 245,023 legitimate URLs.	Dataset is large-scale. The first one to use deep learning model to detect malicious URLs.	The maximum length of the URL is 255 characters. The phishing website URLs did not have relevant semantics.	95.79%
Random forest + Neural network + bagging [33].	Hybrid	UCI	No previous research focuses on using a feedforward NN ensemble learning.	Insufficient data sources and lack of a feature extraction process.	97.40%
Auto encoder + NIOSELM [34]	Hybrid	Websites (PhishTank, Alexa, DMOZ), 60,000 legitimate URLs, 5000 phishing URLs. Has 56 features.	The dataset is imbalanced.	The detection accuracy may not be the best compared with the existing methods.	94.60%

based on whatever a context defines. In step t, there is a hidden state h_t and a cell state C_t. The selection of the information erased or written, or read is controlled by three corresponding gates each for erasing, writing, and reading respectively. These gate values are dynamic and are learned and computed based on the input at a particular time step and the hidden state from the previous time step.

Figure. 2 is a single neural network with inputs X_{t-1}, X_t, X_{t+1} at $t-1$, t and $t+1$ that has a $tanh$ activation function. The input at a particular time step h_t, as well as h_{t-1} (output of the previous RNN block), which is provided as input, and $tanh$ is applied that gives us h_t. It is given as output and input for the next RNN block at the next time step. LSTM has a similar structure, and it is a chain of repeating modules where the same block applies at every step. The structure of

each block contains four different layers that interact with each other and are not recurrent layers.

The sigmoid function is used in these three blocks because the output should lie between 0 & 1. The cell state (C_t) can remove and add information to the cell state, which is regulated by gates. The following points show the working of LSTM.

- Forget gate layer uses a sigmoid function to decide the information to be thrown away from the cell state.

$$f_t = \sigma(U_f h_{t-1} + x_t W_f + b_f) \qquad (1)$$

- Input gate layer(sigmoid function) decides the values to be updated, and the $tanh$ function creates the new vector value, which is added to the state.

TABLE II
VARIABLES OF LSTM UNIT

Variables of LSTM Unit	
$x_t \in \mathbb{R}^d$	input vector to LSTM Unit
$f_t \in (0,1)^h$	forget gate's activation vector
$i_t \in (0,1)^h$	input/update forget gate's activation vector
$o_t \in (0,1)^h$	output forget gate's activation vector
$h_t \in (-1,1)^h$	hidden state vector also known as output vector
$\tilde{c}_t \in (-1,1)^h$	cell input activation vector
$c_t \mathbb{R}^h$	cell state vector
$W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times d} \ and \ b \in \mathbb{R}^h$	weight matrices and bias vector parameters

$$i_t = \sigma(x_t W_i + h_{t-1} U_i + b_i) \tag{2}$$

$$\tilde{C}_t = tanh(x_t W_C + h_{t-1} U_c + b_c) \tag{3}$$

- Multiply the old cell with the cell which contains information that needs to be dropped, and sum it with the dot product of the result from the previous step.

$$C_t = \sigma(f_t \times C_{t-1} + i_t \times \tilde{C}_t) \tag{4}$$

- A filtered version of output is formed in the cell state; run the sigmoid layer to decide the part of the cell states for output. Apply $tanh$ activator to the cell state and multiply it with the output of the sigmoid gate, where the value lies between 0-1.

$$o_t = \sigma(x_t W_o + h_{t-1} U_o + b_o) \tag{5}$$

$$h_t = o_t \times tanh(C_t) \tag{6}$$

Gated Recurrent Unit [37] was proposed as an alternative to LSTM, which is more straightforward; it combines forget and input gates into a single update gate as well as merges the cell state and hidden state. Figure 4 shows GRU architecture.

The update gate controls what parts of the hidden state are updated and preserved.

$$z_t = \sigma(x_t W^z + h_{t-1} U^z + b_z) \tag{7}$$

Reset gate controls which parts of the previous hidden state are used to compute the fresh subject. New hidden state(reset gate) content selects useful parts of the previous hidden state. It uses the current input to compute new hidden content.

$$\tilde{h}_t = tan(r_t \times h_{t-1} U + x_t W + b) \tag{8}$$

The hidden state (update gate) controls what is kept from the previous hidden state and what is updated to new hidden state content.

$$h_t = (1 - z_t) \times \tilde{h}_t + z_t \times h_t - 1 \tag{9}$$

If reset gate is set to all 1's and update gate to all 0's then in reset gate h_t will be $tanh(W * [h_{t-1}], X_t)$ and in update
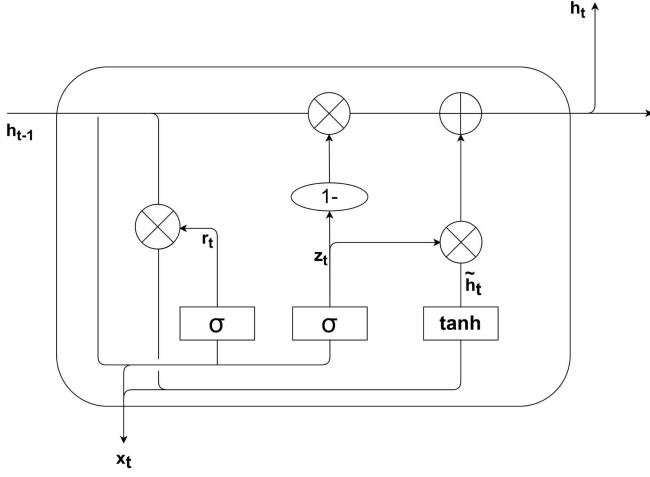

Fig. 2. Vanilla RNN architecture.

Fig. 3. LSTM architecture.

Fig. 4. GRU architecture.

Fig. 5. Comparison of Accuracy at different epochs.

Fig. 6. Comparison of Accuracy at different epochs.

gate, h_t becomes, h_{t-1} respectively. Input and forget gates of LSTM are coupled by an update gate in GRU; the reset gate in GRU is applied directly to the previous hidden state.

TABLE III
VARIABLES OF GRU UNIT

Variables of GRU Unit	
x_t	input vector
h_t	output vector
$\tilde{h}_t$	candidate activation vector
z_t	update gate vector
r_t	reset gate vector
$W, U \ and \ b$	parameter matrices and vector

V. PROPOSED MODEL

A. Preparation of Dataset

For effective detection of malicious URLs, the dataset should contain recent URLs which are malicious, for recognizing fresh features to train the model. Attackers will change the production of phishing links by anti-phishing regulations and procedures that have been released. Anti-phishing models and algorithms must also be improved based on new phishing data. Furthermore, the training dataset's quantity and validity significantly impact the performance of machine learning-based solutions. The performance of deep learning models increases with the variety of content in the training dataset. Hence, it is advised that phishing URLs and legitimate URLs should be extracted from data repositories. The dataset used in this paper consists of numerous legitimate and malicious URLs which are taken from PhishTank, OpenPhish, and Common Crawl. It consists of 46839 instances, and we merely looked into the text in the URL and extracted features to train the model. The dataset was split into 75% and 25% for training and testing, respectively.
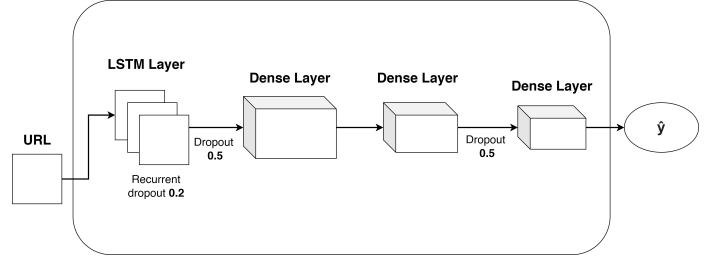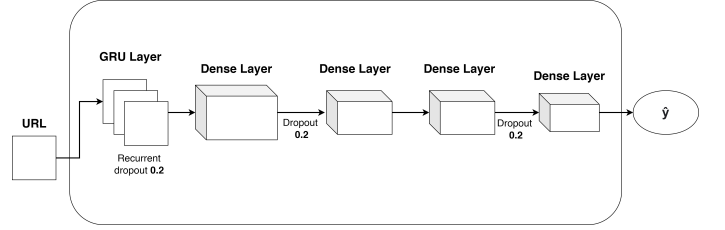
B. Network Architecture and Training Parameters

All the features captured from the URL are fed to the LSTM layer with an orthogonal recurrent initializer. The embedded matrix is used as weights, further combined with dense layers with a sigmoid activation function. A dropout of 0.5 is added between the LSTM layer and dense layer. Similarly, the features are captured from the URL and fed to the GRU layer with an orthogonal recurrent initializer and sigmoid recurrent activation, combined with dense layers with a softmax activation function. A dropout of 0.2 is added between the dense layers. Figure 5 and 6 depict the network architecture of Phish-Defence of LSTM and GRU, respectively. The trained model is converted to a tflite model for producing faster inference time on small edge devices like Raspberry Pi.

The model is trained for 40 epochs with a batch size of 500 using the $tanh$ activation function for the LSTM model and the sigmoid activation function for the GRU model. Binary cross entropy loss with Adam optimizer is attached to the dense layer for classifying legitimate and malicious URLs in both LSTM and GRU models. Binary cross-entropy loss is defined as

$$loss = -\frac{1}{N} \sum_{i=1}^{N} y_i \bullet log\left(p\left(y_i\right)\right) + \left(1 - y_i\right) \bullet log\left(1 - p\left(y_i\right)\right)$$

(10)

VI. RESULTS

As described in Section V-B the network architectures were trained with a batch size of 500 with 70 training steps for 40 epochs, and the training and testing loss started oscillating. The Adam optimization method was utilized with an initial learning rate of 1e-3. This learning rate was decreased by a learning rate scheduler, which checks for lack of reduction in training loss for five epochs and decreases the learning rate by a factor of 0.1. The reduction was stopped when the

learning rate was reduced to an absolute minimum of 1e-5. Early stopping was set at 6 epochs, which ends the training process when the training loss oscillates for more than 6 epochs, yielding a maximum validation accuracy of 90.50% for the LSTM model.

TABLE IV
RESULTS OF PHISH-DEFENSE

| Model	Precision	Inference time	F-Score	Recall
Random Forest (RF_1) [28]	98.30	2.30	99.01	99.65
Adaboost [29]	99.00	2.80	98.80	98.60
Random Forest (RF_2) [30]	99.40	1.95	99.40	-
PSL ^{1} + PART [31]	98.70	3.10	99.00	99.30
Recurrent Neural Network (RNN) + Convolutional Neural Network (CNN) [32]	97.33	2.03	95.52	93.78
Random forest + Neural network + bagging [33]	96.00	1.90	97.00	98.10
Auto encoder + NIOSELM [34]	97.80	2.20	95.52	98.30
PD-LSTM	91.02	0.60	96.40	97.54
PD-GRU	99.08	0.53	99.24	98.97
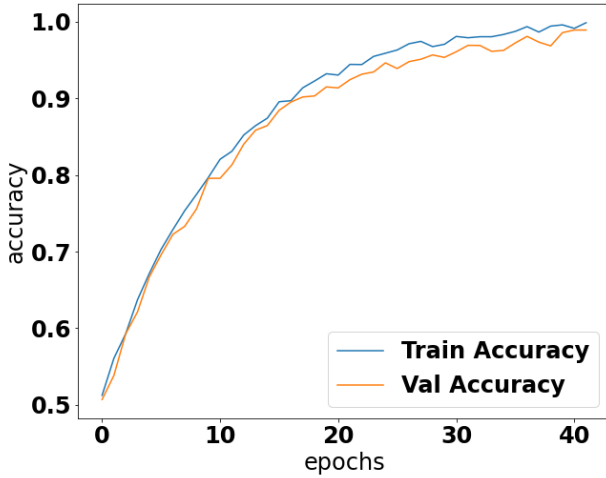
Fig. 7. Accuracy vs Epochs for GRU model.

Later, we built and trained a fresh RNN model using GRU and achieved a state-of-the-art accuracy of 98.51%. Figure 7 and 8 shows the accuracy vs epochs and loss vs epochs graph for Gated Recurrent Unit model. This accuracy was achieved using the softmax activation function with a 0.25 dropout. After performing a significance test, the resulting trained model was found to perform with an inference time of 0.8 seconds on Raspberry Pi-4. Figure 9 and 10 illustrate the accuracy and inference time in the form of a line graph. Table IV shows the results of Phish-Defense compared to other machine learning algorithms.
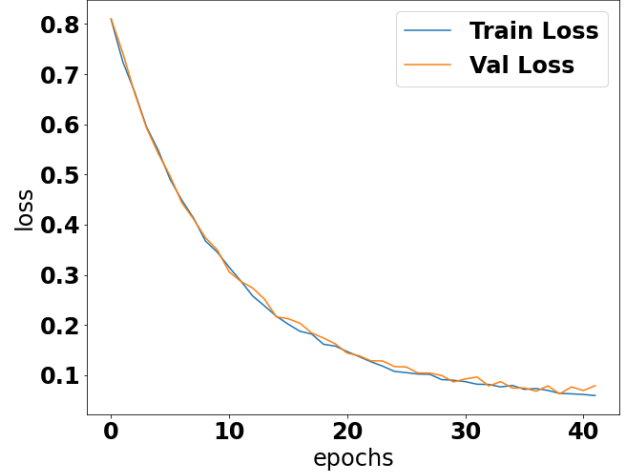

Fig. 8. Loss vs Epochs for GRU model.

VII. CONCLUSION

This study recommended a web phishing detection approach using deep recurrent neural networks to predict phishing websites. In phishing site prediction, the most influential features and optimal weights of website features were tokenized, and these features were used to train the RNN to make better predictions of phishing websites. The experimental results demonstrated that the proposed phishing websites prediction approaches based on RNN increased the classification performance using fewer features. Additionally, the proposed method approaches produced the competitive classification accuracy of phishing websites among all other classifiers with all the feature selection methods used in this study with the lowest inference time. Thus, the proposed approach based on RNNs can also be used on smaller devices like Raspberry-Pi and Arduino to successfully predict phishing websites to contribute and provide more confidence for online commerce and business customers.

ACKNOWLEDGMENTS

REFERENCES

[1] R. Mohammad, F. Thabtah, and T. Mccluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, vol. 25, pp. 443–458, 08 2013.
[2] G. Vrbančič, I. Fister, and V. Podgorelec, "Swarm intelligence approaches for parameter setting of deep learning neural network: Case study on phishing websites classification," in *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, WIMS '18, (New York, NY, USA), Association for Computing Machinery, 2018.
[3] R. Verma and A. Das, "What's in a url: Fast feature extraction and malicious url detection," in *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*, IWSPA '17, (New York, NY, USA), p. 55–63, Association for Computing Machinery, 2017.
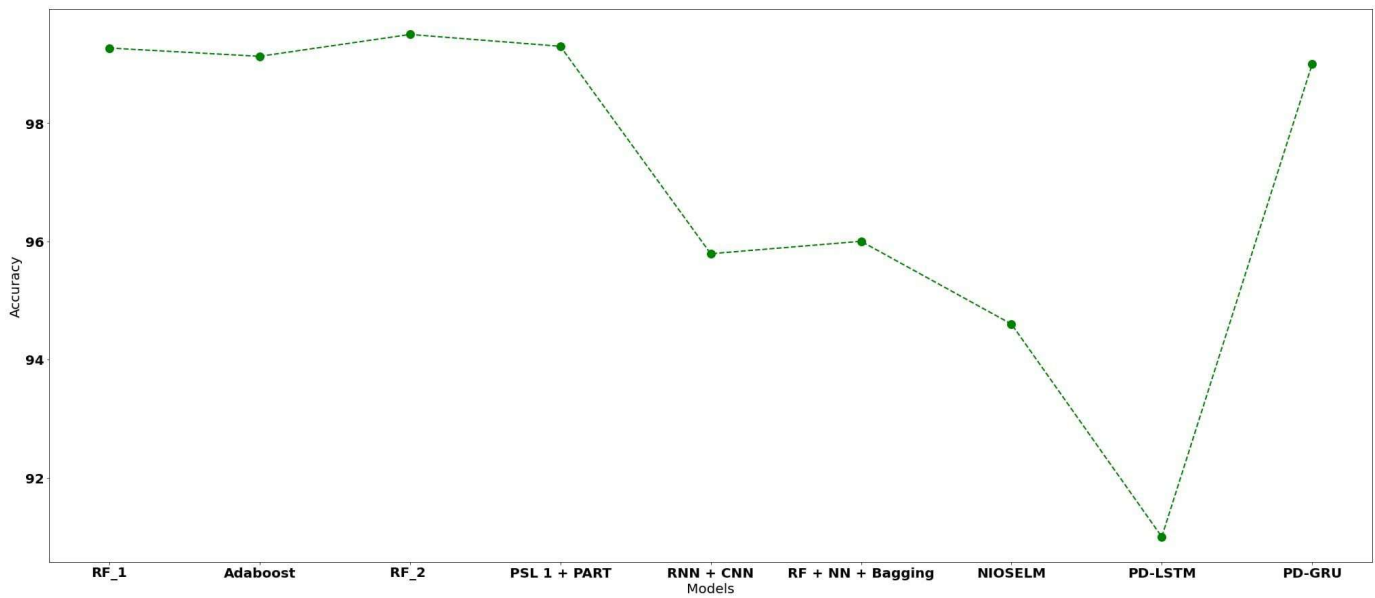
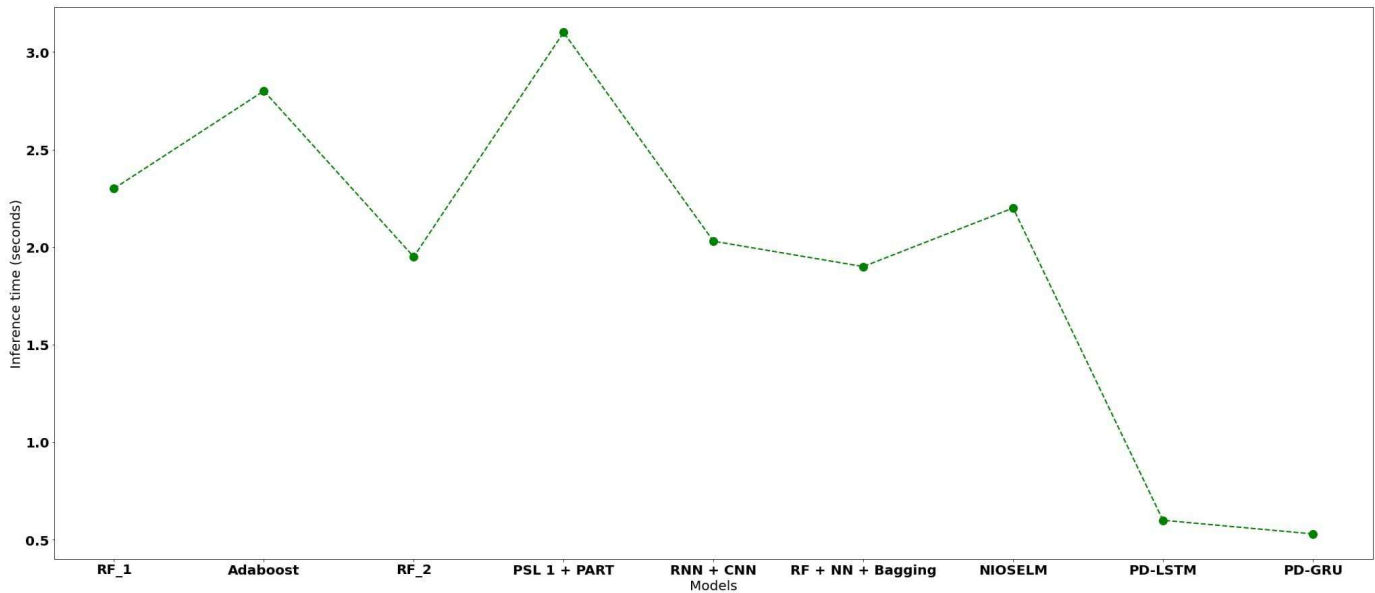Fig. 9. Comparison of accuracy with existing models.

Fig. 10. Comparison of inference time with existing models.

[4] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of ai-enabled phishing attacks detection techniques," *Telecommunication Systems*, 01 2021.

[5] A. Aleroud and L. Zhou, "Phishing environments, techniques, and countermeasures: A survey," *Computers & Security*, vol. 68, 04 2017.

[6] Z. Ramzan and C. Wueest, "Phishing attacks: Analyzing trends in 2006," in *CEAS 2007 - The Fourth Conference on Email and Anti-Spam, 2-3 August 2007, Mountain View, California, USA*, 2007.

[7] A. Ubing, S. Kamilia, A. Abdullah, N. Zaman, and M. Supramaniam, "Phishing website detection: An improved accuracy through feature selection and ensemble learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, 01 2019.

[8] E. Lastdrager, "Achieving a consensual definition of phishing based on a systematic review of the literature," *Crime Science*, vol. 3, p. 9, 09 2014.

[9] M. He, S.-J. Horng, P. Fan, K. Khan, R.-S. Run, J.-L. Lai, R.-J. Chen, and A. Sutanto, "An efficient phishing webpage detector," *Expert Syst. Appl.*, vol. 38, pp. 12018–12027, 09 2011.

[10] R. Mohammad, F. Thabtah, and T. Mccluskey, "Tutorial and critical analysis of phishing websites methods," *Computer Science Review*, vol. 17, 05 2015.

[11] W. Ali, "Phishing website detection based on supervised machine learning with wrapper features selection," *International Journal of Advanced Computer Science and Applications*, vol. 8, 01 2017.

[12] M. Zabihimayvan and D. Doran, "Fuzzy rough set feature selection to enhance phishing attack detection," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–6, 2019.

[13] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," *Expert Systems with Applications*, vol. 41, p. 5948–5959, 10 2014.

[14] A. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Security and Communication Networks*, vol. 2017, pp. 1–20, 01 2017.

[15] K. L. Chiew, C. L. Tan, K. Wong, K. Yong, and W. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Information Sciences*, vol. 484, pp. 153–166,

05 2019.

[16] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast url phishing detector: A convolutional neural network approach," *Computer Networks*, vol. 178, p. 107275, 04 2020.

[17] R. Mohammad, F. Thabtah, and T. Mccluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, vol. 25, pp. 443–458, 08 2013.

[18] R. Rao, A. Pais, and P. Anand, "A heuristic technique to detect phishing websites using twsvm classifier," *Neural Computing and Applications*, vol. 33, pp. 1–20, 06 2021.

[19] E. Zhu, Y. Ju, Z. Chen, F. Liu, and X. Fang, "Dtof-ann: An artificial neural network phishing detection model based on decision tree and optimal features," *Applied Soft Computing*, vol. 95, p. 106505, 06 2020.

[20] E. Feitosa, C. Silva, and V. Garcia, "Heuristic-based strategy for phishing prediction: A survey of url-based approach," *Computers & Security*, 09 2019.

[21] C. L. Tan, K. L. Chiew, K. Yong, S. Sze, J. Abdullah, and Y. Sebastian, "A graph-theoretic approach for the detection of phishing webpages," *Computers & Security*, vol. 95, p. 101793, 05 2020.

[22] M. El-Rashidy, "A smart model for web phishing detection based on new proposed feature selection technique," *Menoufia Journal of Electronic Engineering Research*, vol. 30, pp. 97–104, 01 2021.

[23] L. Yang, J. Zhang, X. Wang, Z. Li, Z. Li, and Y. He, "An improved elm-based and data preprocessing integrated approach for phishing detection considering comprehensive features," *Expert Systems with Applications*, vol. 165, p. 113863, 08 2020.

[24] A. Subasi and E. Kremic, "Comparison of adaboost with multiboosting for phishing website detection," *Procedia Computer Science*, vol. 168, pp. 272–278, 01 2020.

[25] G. Parra, P. Rad, K.-K. R. Choo, and N. Beebe, "Detecting internet of things attacks using distributed deep learning," *Journal of Network and Computer Applications*, vol. 163, p. 102662, 04 2020.

[26] P. yi, Y. Guan, F. Zou, Y. Yao, W. Wang, and T. Zhu, "Web phishing detection using a deep learning framework," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–9, 09 2018.

[27] B. Wei, R. Hamad, L. Yang, X. He, H. Wang, B. Gao, and W. L. Woo, "A deep-learning-driven light-weight phishing detection sensor," *Sensors*, vol. 19, p. 4258, 09 2019.

[28] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing urls detection using lexical based machine learning in a real-time environment," *Computer Communications*, vol. 175, pp. 47–57, 2021.

[29] A. Odeh, I. Keshta, and E. Abdelfattah, "Phiboost- a novel phishing detection model using adaptive boosting approach," *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 07, pp. 65–74, 03 2021.

[30] E. Gandotra and D. Gupta, "Improving spoofed website detection using machine learning," *Cybernetics and Systems*, vol. 52, no. 2, pp. 169–190, 2021.

[31] P. Barraclough, G. Fehringer, and J. Woodward, "Intelligent cyber-phishing detection for online," *Computers & Security*, vol. 104, p. 102123, 2021.

[32] W. Wang, F. Zhang, X. Luo, and S. Zhang, "Pdrcnn: Precise phishing detection with recurrent convolutional neural networks," *Security and Communication Networks*, vol. 2019, pp. 1–15, 10 2019.

[33] A. Zamir, H. Khan, T. Iqbal, N. Yousaf, F. Aslam, A. Anjum, and M. Hamdani, "Phishing web site detection using diverse machine learning algorithms," *The Electronic Library*, vol. ahead-of-print, 01 2020.

[34] L. Yang, J. Zhang, X. Wang, Z. Li, Z. Li, and Y. He, "An improved elm-based and data preprocessing integrated approach for phishing detection considering comprehensive features," *Expert Syst. Appl.*, vol. 165, p. 113863, 2021.

[35] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 338–342, 01 2014.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[37] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (gru) neural networks," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1597–1600, 2017.