BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **22**, No 1
Sofia • 2022 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2022-0004

Enhanced Analysis Approach to Detect Phishing Attacks During COVID-19 Crisis

Mousa Tayseer Jafar[1], *Mohammad Al-Fawa'reh*[2], *Malek Barhoush*[2], *Mohammad H. Alshira'H*[3]

[1]*Philadelphia University, Amman, Jordan*

[2]*Yarmouk University, Irbid, Jordan*

[3]*Al al-Bayt University, Mafraq, Jordan*

E-mails: mjafar@philadelphia.edu.jo fawareh@yu.edu.jo malek@yu.edu.jo alshirah@aabu.edu.jo

Abstract: *Public health responses to the COVID-19 pandemic since March 2020 have led to lockdowns and social distancing in most countries around the world, with a shift from the traditional work environment to virtual one. Employees have been encouraged to work from home where possible to slow down the viral infection. The massive increase in the volume of professional activities executed online has posed a new context for cybercrime, with the increase in the number of emails and phishing websites. Phishing attacks have been broadened and extended through years of pandemics COVID-19. This paper presents a novel approach for detecting phishing Uniform Resource Locators* (*URLs*) *applying the Gated Recurrent Unit* (*GRU*), *a fast and highly accurate phishing classifier system. Comparative analysis of the GRU classification system indicates better accuracy* (*98.30%*) *than other classifier systems.*

Keywords: *Cybersecurity, COVID-19, phishing attack, cybercrime.*

1. Introduction

Global Internet usage increases each year, but 2020 was exceptional due to the COVID-19 pandemic. Public health responses closed conventional workplaces and schools, and confined most people to their homes in most countries during worldwide lockdowns, causing a massive increase in the use of digital technology and the Internet. This was accompanied by a commensurate increase in the volume of online threats, including phishing URLs, one of the biggest online dangers. Phishing is the process of impersonating a trusted party through an e-mail, phone, or text message, through which the scammer can obtain sensitive information, which in turn leads to the scammer obtaining broader powers [1].

There were approximately 4.9 billion Internet users in 2020 [2], calling for huge efforts and advanced technologies to help detect phishing websites and control this active threat. Internet proliferation has changed the ways in which people connect and interact with each other, how they perform business, and how they deliver services. However, there are limited laws regulating people's online behavior, especially in contexts of international user interactions, with a lack of effective control or authority even where legislation exists. The Internet is full of dangers and questionable behaviours [3].

Phishing websites comprise an increasing problem online. Phishing is a type of scam in which a phishing attempt is made to mislead the victim into providing confidential information such as passwords and bank account numbers by posing as an official and authorized institution (e.g., a bank or government agency). Phishing uses fake websites to deceive victims into giving up their sensitive information, which comprises a breach of privacy, and which can be used to enable financial fraud, identity theft, and other malicious criminal activities. This growing threat results in billions of dollars in losses each year [4]. Thus, there is an increasing need for more effective efforts to safeguard users from such websites [5].

One of the popular approaches to detect unsafe websites is the blacklisting and whitelisting approaches. Blacklisting is commonly used by many internet services to warn users of potentially dangerous sites, or to prevent access to them altogether, based on reports of suspicious or criminal activities. However, blacklisting approach is reactive, and it is unable to detect new suspicious websites which are not previously identified as blacklisted [6]. Whitelisting approach seeks to proactively list websites that are legal to display, such as domain names of what can penetrate on a system. Nowadays, advances in artificial intelligence and machine learning have fostered focus on their applications to resolve many cybersecurity problems. Many researchers have examined machine learning techniques to detect unsafe website addresses and categorize unsafe website addresses [7].

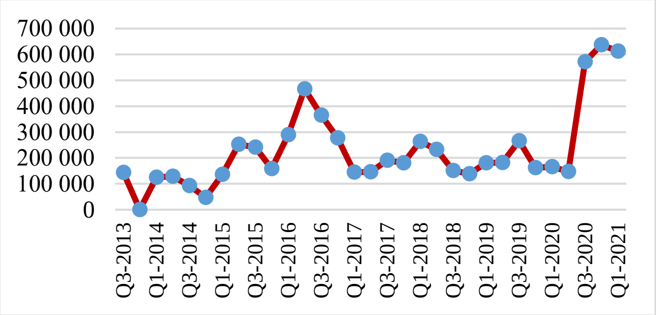

Fig. 1. Number of phishing URLs, Q3 2013 to Q1 2021

The main challenge facing the detection process is the massive number of websites in the world. There are billions of websites on the internet, managing such a huge number of websites is a challenge. Moreover, collecting website features requires time and effort, and unsafe websites tend to be fleeting, as it is easy for criminals to create and remove their websites as required to evade detection [8]. Given the reactive and slow responsiveness of whitelisting and blacklisting approaches, as discussed above, it is vital to categorize new websites by finding effective methods to accumulate and document unsafe websites as quickly and effectively as possible.

Experiments have proven the success of machine learning and decision-making in many areas, and the success of machine learning depends largely on accurate classification of the problem. Since the discovery of phishing requires an accurate classification of websites, this paper proposes a robust classification to distinguish malicious URLs and benign URLs. Existing models are not very accurate in detecting phishing. The main contribution of this paper is utilizing a lexical analysis techniques with GRU as a threat intelligence method to detect phishing URLs.

The main goal of this paper is to build a model to classify Web URLs into the appropriate Web category. The study aims to meet the following objectives:

- To provide a comprehensive groundbreaking study of the concepts surrounding malicious URLs detection systems.
- To perform background research on the concepts of anomaly detection in large network environments.
- To propose a robust detection mechanism to overcome the security issues in the current design of malicious URLs Detection Systems, in order to improve programmatic efficacy and performance.
- Examining and evaluating the strength of the proposed models and their ability to classify and detect novel attacks by measuring the value of the accuracy on a real dataset (ISC2016).

This paper is organized as follows. Section 2 presents a background life cycle of phishing attacks. Section 3 explains related studies. Section 4 includes the proposed methodology to detect phishing attacks. Section 5 presents performance evaluation. Section 6 discusses the GRU results. Section 7 concludes the paper and identifies directions for future work.

2. Background life cycle of phishing

Phishing is a technique of cyber-crime that is used to steal sensitive data from individuals like usernames, passwords, personal data, bank account details, critical login credentials, or credit card information.

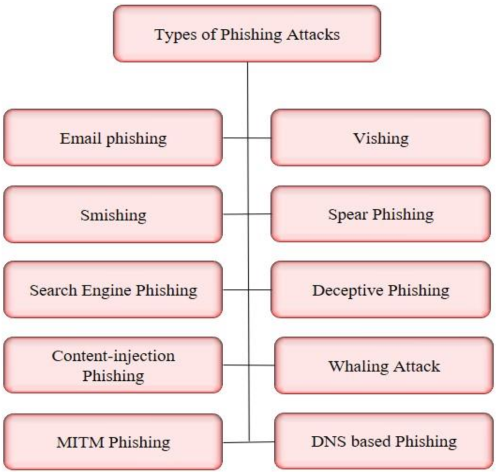

Fig. 2. Types of phishing

Cybercriminals use several methods to obtain such data, including SMS, e-mails, sending attached files, responding to social media connection requests, sending malicious links, using new Wi-Fi hotspots, telephone calls, advertisements, enabling macros in MS Word documents, or any electronic communication types. Fig. 2 illustrates the most important approaches for phishing attacks.

- **Email phishing.** The most extensively known form of phishing, which began as a mass threat during the 1990s. This attack is a cyber-crime that steals classified information via email by sending emails to any email address containing links to malicious websites that are infected with malware. The phishing email raises the challenge of distinguishing between genuine and phishing emails. Some emails are difficult to spot as phishing attacks, especially when they are carefully crafted in terms of language, grammar, and spelling.
- **Vishing.** Vishing (voice phishing) is a cyber-crime technique that uses voice calls via phones to gain personal confidential information from individuals and tempts them to declare sensitive information. Social engineering tactics are applied to support this type of phishing, in order to convince victims to send personal information and access to bank accounts. Generally, the speaker will pose as an employee of the government (e.g., tax department or police) or a bank.
- **Smishing.** Smishing is a more recent type of phishing attack that targets smartphone users by sending malicious links via text messages or SMS to gather sensitive information, such as financial and social insurance data. Smishing attacks take some common forms, such as COVID-19, financial services, gift, invoice or order confirmation, and customer support smishing.
- **Spear phishing.** Spear phishing is a type of phishing attack that plans to target specific victims or groups inside a company. In general, phishing techniques target large samples of random individuals, but spear phishing concentrates on electronic communications to target specific victims or organizations by social engineering.
- **Search engine phishing.** This is a comparatively new type of phishing attack, in which cyber criminals design fraudulent websites to gather individuals' personal data, bank account numbers, direct payments, passwords, social security information, and other data. These fraudulent websites offer inexpensive products and unbelievable business deals to lure online shoppers.
- **Deceptive phishing.** This is the most popular type of phishing attack that people will encounter online. Attackers use famous brands like Amazon and PayPal to pose as legitimate companies to steal personal data or login credentials, then they blackmail the victims to do as the hacker wants.
- **DNS-based phishing.** Domain Name Server (DNS) phishing attack is an attack in which an adjusted DNS table is used to redirect traffic to a fraudulent website. The attacker injects a fake URL into the DNS table, and when the victim requests the real URL, it will automatically redirect the victim to the malicious URL that was injected by the attacker. The hacker can then steal sensitive information, like passwords and account numbers. Fig. 3 illustrates how the URLs inject in the DNS server.

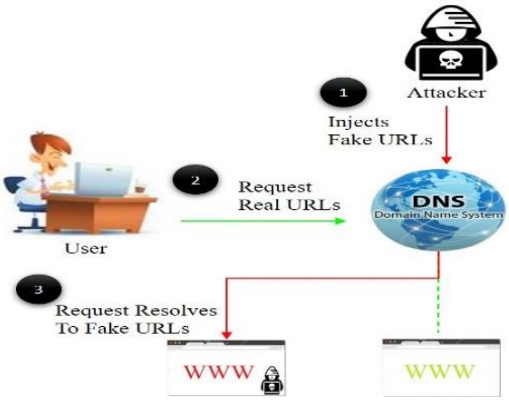

Fig. 3. DNS-based phishing

- **Whaling Phishing.** Whaling attacks are aimed by cybercriminals at the highest levels of target organizations, such as company board members, Chief Financial Officers (CFOs), Chief Executive Officers (CEOs), or anyone who has important authority inside a company. Through social engineering, cybercriminals achieve their goal by encouraging CFOs and CEOs to execute an alternative action to increase their business and profit, masquerading as legitimate emails. The primary goal for cybercriminals is to receive high-value wire transfers from the victims.
- **MITM Phishing.** Man-In-The-Middle (MITM) are a popular type of cyberattack that give attackers permission to manipulate and access unauthorized information or communication between two targets. The attacker uses HTML templates that look like sign-in pages for accounts they wish to target, like Facebook, Google, other ubiquitous sites, and bank accounts, which enable them to obtain user date for subsequent impersonation. The attacker sends malicious emails containing these templates, asking victims to update or share confidential information.

3. Related work

Sahingoz et al. [6] apply Natural Language Processing (NLP) to extract the features from the URLs, and seven machine learning algorithms classifiers have been used to detect phishing URLs. The best result has been given by Random Forest (RF), which achieved 97.98% accuracy. The dataset used to detect phishing has been from 73,575 URLs, consisting of 37,175 phishing and 36,400 benign URLs.

Zouina and Outtaj [9] apply Support Vector Machine (SVM) algorithm to analyze and detect phishing URLs using six features: length of URL, number of hyphens, numeric characters, IP address, number of dots, and distance from target website. The model increases the speed of detection and achieves 95.80% accuracy. They use a small dataset containing 2000 URLs, including 1,000 phishing and 1,000 legitimate URLs.

Anand et al. [10] detect URL phishing by using text with the Generative Adversarial Network (GAN). They use an imbalanced dataset (80 for training and 20 for testing).

Adversarial Auto-Encoder (AAE) has been used to analyze and investigate phishing URLs, concentrating on the attacker's goal, knowledge, and influence [11]. Six machine learning classifiers have ben used to build the model: k-Nearest Neighbor (kNN), Decision Tree (DT), GB, RF, SVM-l, and SVM-G. Their model consists of two rows: the top row ordinary auto-encoder reconstructs the data from the latent code; and the discriminatory network predicts whether the samples emerge from the hidden code of the auto-encoder. The model has been applied to four datasets with a total size of 31,000 URLs, comprising 16,076 phishing and 14,924 legitimate. It has achieved average accuracy of 95.47%.

Xiao et al. [12] combine Multi-Head Self-Attention (MHSA) and Convolutional Neural Network (CNN) to detect URL phishing. The features have been extracted by CNN after the URL string has been used as a feed inside the CNN algorithm, in addition to the weights of learned features calculated by using MHSA, to achieve a good rate accuracy of 99.84%.

Kamran, Sengupta and Tavakkoli [13] propose a new architecture from Conditional Generative Adversarial Network (C-GAN) to detect malicious and benign URLs. Their architecture involves a generator and discriminator. The generator contains the auxiliary classification to encode and decode benign or malicious URLs. The discriminator consists of one encoding to classify URLs as benign or malicious. LSGAN has been used to calculate the time for loss function and training network. The dataset used in this model contains more than 500,000 malicious and benign URLs, but they implemented their architecture on 50,000 samples of URLs from a large dataset to achieve an accuracy rate of 95.52%.

Yerima and Alzaylaee [14] combine two convolutional neural networks (CNN1+CNN2) to generate a model to detect phishing URLs. They have implemented their model on a dataset containing 4,898 benign and 6,157 phishing URLs. Several classifiers have been used like SVM, J48, Bayes Net, RF, Naïve Bayes, and Random Tree. CNN1+CNN2 have achieved a high accuracy rate of 96.6%.

Similarly, Yi et al. [15], present a Deep Belief Network (DBN) to detect phishing URLs, concentrating on original features and interaction features. The features have been extracted from ISP traffic flow for forty minutes and one day, achieving accuracy of 89.6%.

Abutair, Belghith and AlAhmadi [16] develop a new model based on a Phishing Detection System integrated with the Case-Based Reasoning (CBR-PDS) to classify illegitimate or malicious URLs. CBR-PDS depends on 21 URL features, and has been implemented on two small datasets containing 500 URLs and 750 URLs, selected based on their characteristics, with a balancing number of malicious and benign URLs. The CBR-PDS has achieved a 96.26% accuracy rate.

Adebowale et al. [17] use Adaptive Neuro-Fuzzy Inference System (ANFIS), kNN, and SVM to detect phishing URLs. ANFIS extracts features based on images, text, and frames to analyze and investigate URLs and the total features set with approximately 35 dimensional features. It has been chosen by the information gain technique and chi-square statistics. ANFIS has obtained 98.30% accuracy.

Babagoli, Aghababa and Solouk [18] integrate a nonlinear regression with a meta-heuristic base (using SVM and harmony search) to introduce a phishing website detection system using 30 features to classify URLs as benign or phishing, using wrapper and decision tree algorithm. Their system has been evaluated on a dataset with 11,055 benign and phishing URLs. The experimental analysis has revealed 92.80% accuracy.

Ferreira et al. [19] utilize ANN-MultiLayer Perception (MLP) algorithm to detect phishing URLs. The ANN-MLP model focuses on URL characteristics by extracting 30 features used to feed the model. The dataset contains 11,055 phishing and legitimate URLs. The ANN-MLP has achieved 98.23% accuracy.

Korkmaz, Sahingoz and Diri [20] use 8 machine learning models using three different datasets. They conclude LR, SVM and NB have low accuracy rate, however in terms of training time NB, DT, LR and ANN models have given better results. They conclude that RF and ANN could be used as phishing detection system due to they have achieved high accuracy rate with less training time.

Alam et al. [21] integrate RF and DT with REF, Relief-F, IG and GR algorithm and principal component analysis to build a detect system for phishing attacks where they use Kaggle dataset with 32 features. The PCA has reduced the irrelevant features in the dataset. Experiments have found the RF achieved an accuracy of 97%.

Kumar et al. [22] have created a realistic dataset of URLs where they solve common problems in other datasets such as data imbalance, biased training, variance and overfitting. Then they extract lexical structure of the dataset. Finally, they use common classifiers such as Logistic Regression, Naive Bayes, RF, Decision Tree and k-Nearest Neighbor. The experiments have indicated all models had almost the same, but the NB have had the highest AUC value. NB has achieved the highest accuracy of 98% with 1, 0.95, 0.97 precision, recall and F1-score respectively.

Do et al. [23] have conducted an empirical trial using Deep Neural Network (DNN), CNN, Long Short-Term Memory (LSTM), and GRU to detect webpage phishing, where they focus on parameter tuning to increase the accuracy rate of these deep learning models. The results obtained from the experiments show that LSTM has achieved the best measures across all models. The DNN has achieved 96.56% accuracy rate while the other models have achieved 97.20, 96.70 CNN, LSTM, GRU respectively. Author suggests future research directions related to deep learning in the phishing detection domain.

Yang et al. [24] use character embedding methods to convert any URLs into fixed-size matrices to extract different features at different levels. They integrate RF, CNN and winner-take-all approach to predict if the URL phishing or not. To test the model being proposed they use their own dataset, the model hasachieved 99.26% accuracy.

Dangwal and Moldovan in [25] combine two datasets one of them contains 30 and the other 48 features, they identify 18 common features. In addition they use feature selection methods to identify the best 13. The conducted experiments prove that using RF algorithm with these 13 features has achieved better performance compared with RF with 30 features of the same dataset. The best performance achieved among all experiments has been 93.7% accuracy rate.

4. Proposed methodology

This paper introduces a robust technique to analyze and detect phishing URLs by using GRU. The approach proposed in this paper analyzes the website URLs through extracting features using lexical method, and then trains the GRU classifier on a dataset that has malicious URLs and benign URLs.

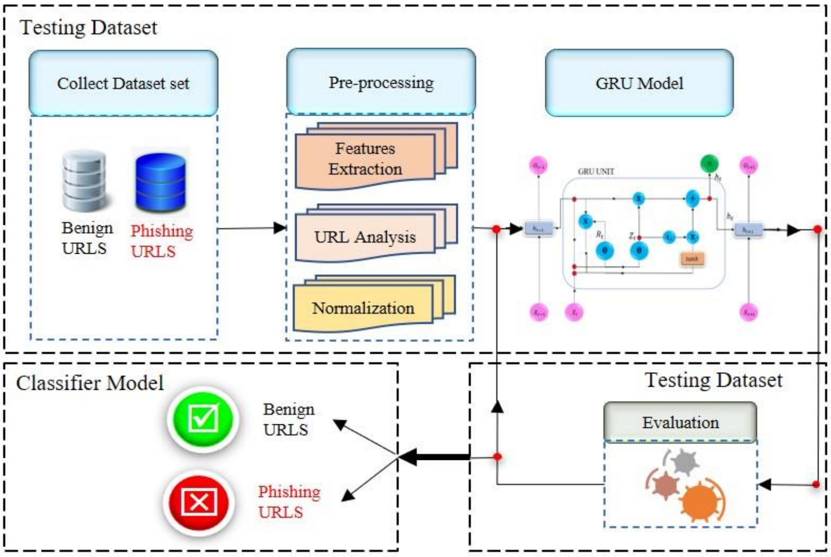

Fig. 4. Proposed methodology

The experiments in this work implement GRU, a powerful machine learning model, employed to detect suspicious websites based on features extracted from the URLs. The proposed approach phases are illustrated in Fig. 4.

The experiments have been performed as follows to accomplish the research objective:

- ✓ Build GRU classifier.
- ✓ Employ GRU machine learning model to detect phishing URLs.
- ✓ Evaluate GRU classifier performance.

The following phases are included in this research:

- Data collection.
- Feature extraction.
- Eliminating duplicate URLs from the dataset.
- Handling missing values and outfitters using median values.
- Handling imbalanced data using over-sampling methods.
- Data normalization.
- Feature selection.
- Build URL classifier using GRU.
- Training URL classifier.
- Testing URL classifier.

4.1. Data collection

Finding a suitable dataset to perform experiments is a difficult challenge facing cybersecurity researchers. Many potentially useful datasets are highly confidential and restricted due to privacy issues, which makes it more difficult to find suitable datasets for analysis in cybersecurity studies. In addition, many datasets are anonymous, and do not reflect existing cybersecurity objectives. This phase explains the collection process to enable the model to achieve its goals. The process of collecting data is very important for the analysis and investigation phases. These data are considered as evidence is used in cybercrime. Otherwise, in machine learning, it is considered as a feed for training and testing phases in the selected models. The dataset has been used in this paper adapted from a well-known dataset called ISC2016, the dataset comprises to 35,300 benign URLs and 10,000 phishing ones. The dataset is imbalanced; hence, we balanced it by choosing an equal number of benign and phishing URLs.

4.2. Data pre-processing

The phase of data pre-processing is considered the most important phase, due to data being converted to a form suitable to be fed into the GRU model, or any selected machine learning models. It is important to include only important data with the important features and preparation of data sets for classification tasks.

4.3. Removing duplicate data

This phase deals with inappropriate and missing data values, such as {NAN, Infinity}. During this phase, duplicates will be eliminated from the dataset while training the GRU model.

4.4. Feature extraction

Feature extraction is the most significant part of finding the perfect features from the raw data to solve research problems. The extractor algorithm analyzes website URLs to choose the best feature.

- URL structure:

URL (Uniform Resource Locator) is an important and unique concept of the Web that locates a resource on the Internet. The URL structure comes from multiple parts that help determine how and where to retrieve a resource on the web. It includes the protocol, domain name, file name, path, and parameters to the webserver [26]. Fig. 5 explains the URL structure.

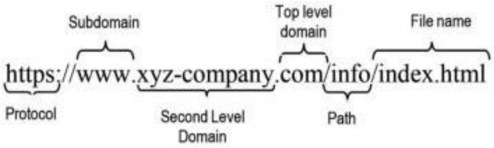

Fig. 5. URL structure

Characteristic points of a URL that can be extracted during feature extraction include the length of the URL, the number of arguments, the average domain token length, the total number of dots in the file, the number of hyphens in the domain, and more. Table 1 shows a list of useful features that were extracted to feed the GRU classifier. Finally, benign URLs were labeled with one, and phishing URLs were labeled with zero.

Table 1. Sample of features

Description of features					
F1	“domain_token_count”	F27	“dld_domain”	F53	“Querylength”
F2	“dld_filename”	F28	“ArgUrlRatio”	F54	“dld_getArg”
F3	“argDomanRatio”	F29	“Query_DigitCount”	F55	“argPathRatio”
F4	“charcompace”	F30	“ldl_filename”	F56	“avgdomaintokenlen”
F5	“subDirLen”	F31	“domainlength”	F57	“ldl_url”
F6	“ISIp AddressInDomainName”	F32	“ldl_path”	F58	“ldl_getArg”
F7	“Filename_LetterCount”	F33	“tld"avgpathtokenlen”	F59	“urlLen”
F8	“longdomaintokenlen”	F34	“dld_url”	F60	“this.fileExtLen”
F9	“NumberRate_Extension”	F35	“pathLength”	F61	“domainUrlRatio”
F10	“Entropy_ Afterpath”	F36	“pathurlRatio”	F62	“executable”
F11	“path_token_count”	F37	“pathDomain Ratio”	F63	“host_DigitCount”
F12	“fileNameLen”	F38	“NumberofDotsinURL”	F64	“LongestPathTokenLength”
F13	“charcompvowels”	F39	“File_name_Digi tCount”	F65	“SymbolCount_Domain”
F14	“ldl_domain”	F40	“CharacterContinuityRate”	F66	“Entropy_Filename”
F15	“dld_path”	F41	“Arguments_LongestWord Length”	F67	“Domain_ LongestWordLength”
F16	“ArgLen”	F42	“NumberRate_URL”	F68	“Entropy_Extension”
F17	“isPortEighty”	F43	“SymbolCount_Afterpath”	F69	“SymbolCoun t_ Directoryname”
F18	“Extension_DigitCount”	F44	“SymbolCount_FileName”	F70	“SymbolCount_URL”
F19	“subDirectory_LongestWordLength”	F45	“Entropy_Domain”	F71	“URL_Letter_Count”
F20	“delimeter_path”	F46	“SymbolCount_Extension”	F72	“Path_LongestWordLength”
F21	“Entropy_ DirectoryName”	F47	“Host _letter_count”	F73	“LongestVariableValue”
F22	“Entropy_URL”	F48	“NumberRate_FileName”	F74	“NumberRate_Directory Name”
F23	“Extension_ LetterCount”	F49	“delimeter_Count”	F75	“NumberRate_Domain”
F24	“NumberRate_AfterPath”	F50	“URLQueries_variable”	F76	“Directory_LetterCount”
F25	“URL_sensitiveWord”	F51	“spcharUrl”	F77	“URL_DigitCount”
F26	“delimeter_ Domain”	F52	“Query_LetterCount”	F78	“Directory_DigitCount”

4.5. Handling missing data

This phase starts after converting URLs to features and labeling them as phishing and benign. Usually, the raw data contains at least one missing value or duplicate value, all of which need to be removed. Afterward, the missing and inappropriate values are replaced by median values.

4.6. Data normalization

The process of minimizing redundancy from a relationship in a dataset is called normalization or scaling. Each dataset has redundancy in relationships that cause deletion or insertion. The normalization phase eliminates or reduces redundancy in the database.

4.7. Build GRU classifier

The GRU is a powerful and similar to LSTM network and considers a new version of Standard Recurrent Neural Network (RNN). GRUs can effectively contain longterm dependencies in sequential data, while addressing the “short-term memory” issue plaguing vanilla RNNs. GRUs use internal gating mechanisms to control and regulate the movement of information between cells in the neural network [27]. The gates play an important role in increasing the accuracy of learning for the GRU cell by storing or erasing information. Fig. 6 illustrates the architecture of GRU.

The update gate (z_t) appears through joining input gates with the forget gate [28]. the primary function of the update gate is to help determine the amount of previous information in memory and maintain the amount of new information to be controlled and held. This makes the model more robust, whereby it copies all previous information and eliminates the risk. The next equation is used to calculate the update gate:

(1) $$z_t = \sigma(w_z.[h_{t-1}, x_t]).$$

The second gate is the reset gate (r_t), which is responsible for the architecture of this model to decide how much of the previous information must access to forget [29]. The reset gate is derived from the current input and previous hidden state (previous memory). Mathematically, the next equation is used to calculate the reset gate:

(2) $$r_t = \sigma(w_z.[h_{t-1}, x_t]).$$

Tanh is a hyperbolic tangent function. The output range for Tanh is (–1, 1). The important function for Tanh is to keep the values between –1 and 1, which helps the model to control the output of the network, thus helping activation of the network [30]. The next equations are used for this function:

(3) $$h_t = \tanh(r_z * [h_{t-1}, x_t]),$$

(4) $$h_t = (1 - z_t) * h_{t-1} + z_t * h_t.$$

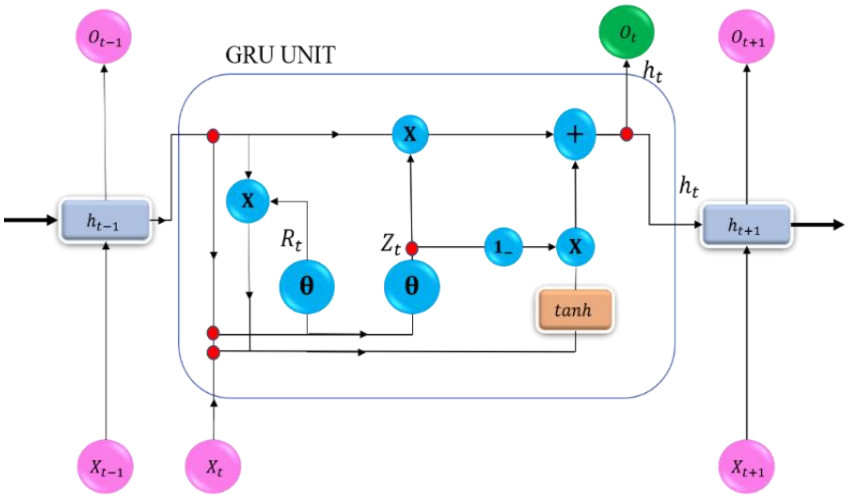

Fig. 6. GRU architecture

4.8. Training and testing sets

The dataset has been divided into a training set (80%) and testing set (20%), the latter of which is used for model performance evaluation, after training the model using the training set.

5. Performance evaluation

To measure the functionality of any model, the outputs must be evaluated. Essentially, the evaluation is a core part and substantial in measuring the performance of the selected model, and building an effective model. The test dataset should contain the correct labels for all data instances. These labels are used to compare between the predicted labels for performance evaluation after classification. There are various evaluation metrics; the GRU model is evaluated using accuracy, recall, precision, sensitivity, and F1 score. A GRU classifier predicts all data instances of a test dataset as either positive or negative, which can fall into one of the following four categories.

- True Negative (TN): the number of instances that are classified and detected as false. TN is defined as the ratio of negatives instances that are categorized correctly.
- True Positive (TP): the number of instances that are classified as true and detected as true. TP rate is the percentage of positive instances that are accurately categorized.
- False Positive (FP): The number of instances that are wrongly detected as positive. FP rate is the percentage of negatives cases that are incorrectly classified as positive.
- False Negative (FN): the number of positive cases that are predicted as negative. FN rate is the percentage of positives cases that are incorrectly classified as negative.

According to the above categories: accuracy, recall, precision, sensitivity, F1 score can be calculated as follows.

- Accuracy: the percentage of correct predictions for the test data. It is the most common metric used to judge a model [31]. It is calculated using the equation

$$\text{Accuracy } = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}. \tag{5}$$

- Recall: a percentage of positive instances from the total actual positive instances for the GRU model predicted as positives (TP) [32]. It is calculated using the equation

$$\text{Recall} = \frac{\text{TP}}{\text{TP+ FN}}. \tag{6}$$

- Precision: the percentage of positive instances from the total predicted positive instances [31]. This explains how accurate the selected model is. It is calculated using the equation

$$\text{Precision } = \frac{\text{TP}}{\text{TP+ FP}}. \tag{7}$$

- F1 Score: indicates the equilibrium between recall and precision results, with the contributions of both results [32]. Furthermore, it makes efficiency than accuracy, as an F1 score is not considered in any TN cases. It is calculated using the equation

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}. \tag{8}$$

- Sensitivity: calculated as the number of correct positive predictions divided by the total number of positives [32]. It is also called RECall (REC) or True Positive Rate (TPR). It is calculated using the equation

(9) $$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+ FN}}.$$

The models have been performed on Intel laptop Intel® core™ I7-6500 CPU @ 2.50 GHz 2.60 Hz and 16 GB RAM, Windows 10 professional 64-bits operating system.Table 2 depicts the testing environment. The architecture was implemented using Keras V2.2.4 and TensorFlow.

Table 2. System specifications

Processor	Intel® core™ I7-6500 CPU @ 2.50 GHz 2.60 Hz
RAM	16 GB
System type	64-bit operating system, ×64-based processor

Table 3 demonstrates the average of the relevance measures from different runs of the GRU model: 50, 100, 150, and 200 epochs. The experiments for the work presented in this paper have been performed on a balanced dataset containing both phishing and benign URLs.

Table 3. GRU model experimental results

Detection model	Number of ran epoch times			
	50	100	150	200
Training time	81.2233	170.551	246.427	338.612
Testing time	0.36472	0.48935	0.41911	0.42110
Accuracy	98.00%	98.10%	98.10%	98.30%
Sensitivity	97.88%	97.51%	98.01%	98.08%
Precision	98.10%	98.70%	98.30%	98.60%
Recall	97.80%	97.40%	97.90%	98.00%
F1-Score	97.90%	98.10%	98.10%	98.30%

6. Results and discussion

Accuracy, precision, sensitivity, recall, F1-score, sensitivity, and ACU are presented in Table 2 for the GRU classifier system. The results show the GRU classifier system accuracy rates when run with the following number of epochs: 50 (98.00%), 100 (98.10%), 150 (98.10%), and 200 (98.30%). The results in Table 4 compare GRU classifier system performance with that of various other detection approaches, and Fig. 7 compares GRU accuracy. It is noteworthy the main limitation of this paper using a medium size of dataset while the state of art of deep learning in large and complex dataset.

The computational complexity of deep learning models is represented by Space complexity (Memory usage), time complexity (Number of serial steps) and the floating-point operations per second [39]. The computation complexity for any simple single-layer RNN is linear with the length of the input sequence [40]. Table 5 shows the RNN complexity as a function sequence length where *T* is the length of the input sequence and **Big *O*** Notation represent the complexity of an algorithm [41].

Table 4. The experimental results for the GRU model compared with different models

Reference	Detection technique	Accuracy	Sensitivity	Precision	F1-score	AUC	Dataset
Xiao et al.[12]	CNN	92.51%	91.90	93.03	92.46	0.9251	88984
Adebowale et al. [17]	ANFIS	98.30%	x	98.31	98.28	x	13000
Chatterjee and Namin [33]	Reinforcement Learning	90.10%	88.00	0.867	87.30	x	73,575
Kamran, Sengupta, and Tavakkoli [13]	C-GAN	95.52%	96.00	95.08	95.54	0.9552	50000
Babagoli, Aghababa and Solouk [18]	HS& SVM	92.80%	x	95.40	96.30	x	11050
Anand et al. [10]	Text-GAN	91.35%	92.10	90.73	91.41	0.9135	300000
Abutair, Belghith and AlAhmadi [16]	CBR-PDS	96.26%	x	x	96.25	x	1250
Zhang et al. [34]	ELM	97.50%	x	97.96	97.48	x	6905
Yerima and Alzaylaee [14]	SVM	86.38%	88.22	85.08	86.62	0.8638	11050
Yi et al. [15]	DBN	89.60%	x	x	x	x	2018734
Sahingoz et al. [6]	RF	97.98%	99.00	97.00	98.00	x	73575
Ferreira et al. [19]	ANN-MLP	98.23%	x	x	x	x	3000
El-Alfy [35]	PNNs	96.79%	x	95.48	96.67	x	11050
Shirazi et al. [11]	RF	86.88%	85.08	88.25	86.64	0.8688	31000
Zouina and Outtaj [9]	SVM & Gaussian kernel	95.80%	x	x	x	x	2000
Montazer and Arab Yarmohammadi [36]	Fuzzy rough	88.00%	x	87.90	87.95	x	-
Yadollahi et al. [37]	XCS	98.39%	98.41	98.39	98.29	x	8000
Alshira’h and Al-Fawa’reh [38]	RF	98.00%	x	99.00	98.00	x	45300
Yang et al. [24]	CNN + RF	99.26%	99.29	99.19	99.23	x	131067
Kumar et al.[22]	Naive-Bayes	98.00%	95.00	99.00	97.00	98.70	117000
Korkmaz, Sahingoz and Diri [20]	RF	94.59%	x	x	x	x	126077
Alam et al. [21]	RF-PCA	97.00%	x	96.89	90.84	x	2211
Do et al. [23]	DNN	97.29%	97.53	97.53	97.53	99.40	11055
The proposal model	GRU	98.30%	98.08	98.60	98.30	98.30	45300

Table 5. RNN complexity

Detection model	Memory	Compute	Serial steps
Inference	$O(1)$	$O(T)$	$O(T)$
Training BPTT	$O(T)$	$O(T)$	$O(T)$
Training BPTT $h(x, y^*)$	$O(1)$	$O(T)$	$O(1)$

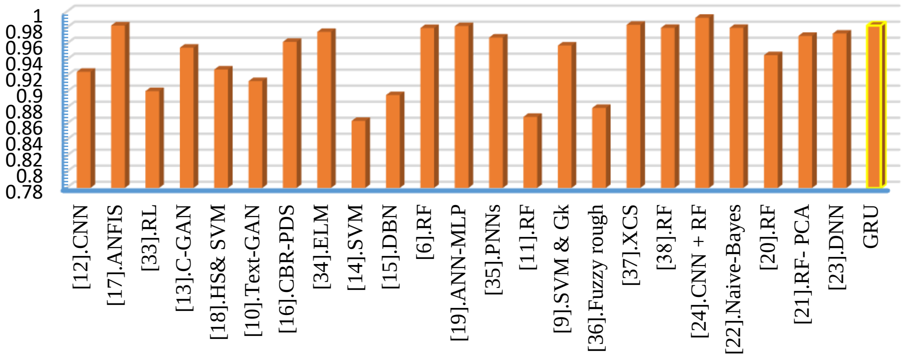

Fig. 7. The experimental results for the GRU model compared with different models

7. Conclusion

During the COVID-19 period, massive efforts have been made to combat the enormous increase in the number of malicious websites on the internet. This paper has proposed a new phishing classifier system by applying a GRU. This model concentrates on using the gate, which increases the speed of the model to detect phishing URLs. This approach can work advantageously and quickly in the cybersecurity domain. The GRU classifier system presents excellent results compared with other models, including accuracy of 98.30%, and it can support efforts to meet the international challenge of detecting potential phishing URLs. The performance of any model either machine or deep learning depends on the data collection and the pre-processing phases, consequently this paper utilizes a systematic approach and optimized model, that is the main advantage of our model over other models.

References

1. Phishing|General Phishing Information and Prevention Tips (Accessed 18 February 2022). **https://www.phishing.org/**
2. Internet-Statistics (Online). **https://www.broadbandsearch.net/blog/internet-statistics**
3. Whitman, M. E., H. J. Mattord. Principles of Information Security. Cengage Learning, 2011.
4. Trautman, L. J., M. Hussein, E. U. Opara, M. J. Molesky, S. Rahman. Posted: No Phishing. – In: Emory Corp. Gov. Account. Rev., 2020.
5. Alqurashi, R. K., M. A. AlZain, B. Soh, M. Masud, J. Al-Amri. Cyber Attacks and Impacts: A Case Study in Saudi Arabia. – Int. J., Vol. **9**, 2020, No 1.
6. Sahingoz, O. K., E. Buber, O. Demir, B. Diri. Machine Learning Based Phishing Detection from URLs. – Expert Syst. Appl., Vol. **117**, 2019, pp. 345-357.
7. Buczak, A. L., E. Guven. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. – IEEE Commun. Surv. Tutorials, Vol. **18**, 2015, No 2, pp. 1153-1176.
8. Luo, Z. A Study of Accuracy and Reliability of CBIR-Based Phishing Filter. Purdue University, 2013.
9. Zouina, M., B. Outtaj. A Novel Lightweight URL Phishing Detection System Using SVM and Similarity Index. – Human-Centric Comput. Inf. Sci., Vol. **7**, 2017, No 1, pp. 1-13.
10. Anand, A., K. Gorde, J. R. A. Moniz, N. Park, T. Chakraborty, B.-T. Chu. Phishing URL Detection with Oversampling Based on Text Generative Adversarial Networks. – In: Proc. of 2018 IEEE International Conference on Big Data (Big Data'18), 2018, pp. 1168-1177.
11. Shirazi, H., S. R. Muramudalige, I. Ray, A. P. Jayasumana. Improved Phishing Detection Algorithms Using Adversarial Autoencoder Synthesized Data. – In: Proc. of 2020 IEEE 45th Conference on Local Computer Networks (LCN'20), 2020, pp. 24-32.
12. Xiao, X., D. Zhang, G. Hu, Y. Jiang, S. Xia. CNN-MHSA: A Convolutional Neural Network and Multi-Head Self-Attention Combined Approach for Detecting Phishing Websites. – Neural Networks, Vol. **125**, 2020, pp. 303-312.
13. Kamran, S. A., S. Sengupta, A. Tavakkoli. Semi-Supervised Conditional GAN for Simultaneous Generation and Detection of Phishing URLs: A Game Theoretic Perspective. arXiv Prepr. arXiv2108.01852, 2021.
14. Yerima, S. Y., M. K. Alzaylaee. High Accuracy Phishing Detection Based on Convolutional Neural Networks. – In: Proc. of 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS'20), 2020, pp. 1-6.
15. Yi, P., Y. Guan, F. Zou, Y. Yao, W. Wang, T. Zhu. Web Phishing Detection Using a Deep Learning Framework. – Wirel. Commun. Mob. Comput., Vol. **2018**, 2018.

16. Abutair, H., A. Belghith, S. AlAhmadi. CBR-PDS: A Case-Based Reasoning Phishing Detection System. – J. Ambient Intell. Humaniz. Comput., Vol. **10**, 2019, No 7, pp. 2593-2606.

17. Adebowale, M. A., K. T. Lwin, E. Sanchez, M. A. Hossain. Intelligent Web-Phishing Detection and Protection Scheme Using Integrated Features of Images, Frames and Text. – Expert Syst. Appl., Vol. **115**, 2019, pp. 300-313.

18. Babagoli, M., M. P. Aghababa, V. Solouk. Heuristic Nonlinear Regression Strategy for Detecting Phishing Websites. – Soft Comput., Vol. **23**, 2019, No 12, pp. 4315-4327.

19. Ferreira, R. P., et al. Artificial Neural Network for Websites Classification with Phishing Characteristics. – Soc. Netw., Vol. **7**, 2018, No 2, p. 97.

20. Korkmaz, M., O. K. Sahingoz, B. Diri. Detection of Phishing Websites by Using Machine Learning-Based URL Analysis. – In: Proc. of 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT'20), 2020, pp. 1-7.

21. Alam, M. N., D. Sarma, F. F. Lima, I. Saha, S. Hossain. Phishing Attacks Detection Using Machine Learning Approach. – In: Proc. of 3rd International Conference on Smart Systems and Inventive Technology (ICSSIT'20), 2020, pp. 1173-1179.

22. Kumar, J., A. Santhanavijayan, B. Janet, B. Rajendran, B. S. Bindhumadhava. Phishing Website Classification and Detection Using Machine Learning. – In: Proc. of 2020 International Conference on Computer Communication and Informatics (ICCCI'20), 2020, pp. 1-6.

23. Do, N. Q., A. Selamat, O. Krejcar, T. Yokoi, H. Fujita. Phishing Webpage Classification via Deep Learning-Based Algorithms: An Empirical Study. – Appl. Sci., Vol. **11**, 2021, No 19, p. 9210.

24. Yang, R., K. Zheng, B. Wu, C. Wu, X. Wang. Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning. – Sensors, Vol. **21**, 2021, No 24, p. 8281.

25. Dangwal, S., A.-N. Moldovan. Feature Selection for Machine Learning-Based Phishing Websites Detection. – In: Proc. of 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA'21), 2021, pp. 1-6.

26. Berners-Lee, T., L. Masinter, M. McCahill. Uniform Resource Locators (URL). 1994.

27. Bibi, I., A. Akhunzada, J. Malik, J. Iqbal, A. Mussaddiq, S. Kim. A Dynamic DL-Driven Architecture to Combat Sophisticated Android Malware. – IEEE Access, Vol. **8**, 2020, pp. 129600-129612.

28. Kula, S., M. Choraś, R. Kozik, P. Ksieniewicz, M. Woźniak. Sentiment Analysis for Fake News Detection by Means of Neural Networks. – In: Proc. of International Conference on Computational Science, 2020, pp. 653-666.

29. Chung, J., C. Gulcehre, K. Cho, Y. Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv Prepr. arXiv1412.3555, 2014.

30. Kalman, B. L., S. C. Kwasny. Why Tanh: Choosing a Sigmoidal Function. – In: Proc. of 1992 IJCNN International Joint Conference on Neural Networks, Vol. **4**, 1992, pp. 578-581.

31. Baldi, P., S. Brunak, Y. Chauvin, C. A. F. Andersen, H. Nielsen. Assessing the Accuracy of Prediction Algorithms for Classification: An Overview. – Bioinformatics, Vol. **16**, 2000, No 5, pp. 412-424.

32. Saito, T., M. Rehmsmeier. The Precision-Recall Plot is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. – PLoS One, Vol. **10**, 2015, No 3, p. e0118432.

33. Chatterjee, M., A.-S. Namin. Detecting Phishing Websites through Deep Reinforcement Learning. – In: Proc. of 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC'19), Vol. **2**, 2019, pp. 227-232.

34. Zhang, W., Q. Jiang, L. Chen, C. Li. Two-Stage ELM for Phishing Web Pages Detection Using Hybrid Features. – World Wide Web, Vol. **20**, 2017, No 4, pp. 797-813.

35. El-Alfy, E.-S. M. Detection of Phishing Websites Based on Probabilistic Neural Networks and K-Medoids Clustering. – Comput. J., Vol. **60**, 2017, No 12, pp. 1745-1759.

36. Montazer, G. A., S. ArabYarmohammadi. Detection of Phishing Attacks in Iranian e-Banking Using a Fuzzy-Rough Hybrid System. – Appl. Soft Comput., Vol. **35**, 2015, pp. 482-492.

37. Yadollahi, M. M., F. Shoeleh, E. Serkani, A. Madani, H. Gharaee. An Adaptive Machine Learning Based Approach for Phishing Detection Using Hybrid Features. – In: Proc. of 2019 5th International Conference on Web Research (ICWR'19), 2019, pp. 281-286.

38. Alshira'h, M., M. Al-Fawa'reh. Detecting Phishing Urls Using machine Learning Lexical Feature-Based Analysis. – Int. J. Adv. Trends Comput. Sci. Eng., Vol. **9**, 2020, No 4, pp. 5828-5837.

39. Al-Ruzouq, R. et al. Sensors, Features, and Machine Learning for Oil Spill Detection and Monitoring: A Review. – Remote Sens., Vol. **12**, 2020, No 20, p. 3338.

40. Deep Learning – What is the Complexity of a Bidirectional Recurrent Neural Network? – Data Science Stack Exchange (Accessed 18 February 2022). **https://datascience.stackexchange.com/questions/82766/what-is-the-complexity-of-a-bidirectional-recurrent-neural-network**

41. Computational Complexity of ML Models|by Paritosh Kumar|Analytics Vidhya|Medium (Accessed 18 February 2022). **https://medium.com/analytics-vidhya/time-complexity-of-ml-models-4ec39fad2770**

Received: 29.12.2021; Second Version: 19.02.2022; Accepted: 22.02.2022 (*fast track*)