Machine Learning Techniques for Italian Phishing Detection

Leonardo Ranaldi[1], Michele Petito[2], Marco Gerardi[3], and Francesca Fallucchi[4]

[1] Gugliemlo Marconi University, Roma, Italy
l.ranaldi@unimarconi.it
[2] Agency for Digital Italy (AgID), Roma, Italy
petito@agid.gov.it
[3] Guglielmo Marconi University, Roma, Italy
m.gerardi@studenti.unimarconi.it
[4] Guglielmo Marconi University, Roma, Italy
f.fallucchi@unimarconi.it

Abstract

In recent years, several methods have been developed to combat phishing. These include approaches based on blacklisting / whitelisting, visual similarity, heuristics, search engines and more recently on machine learning (ML). In this article we will focus mainly on the latter method and on the possible advantages of ML to the more classic and widespread use of blacklists which is becoming less and less effective due to the shorter duration of phishing sites.

In addition to the problem of the duration of phishing pages, blacklists do not allow the detection and blocking of so-called zero-day attacks (phishing pages not yet detected by any security system) and require frequent updates by analysts. So while blacklists are growing in size, phishing attacks aren't decreasing.

On the other hand, scientific research shows how ML-based phishing detection systems are more effective. In the past these methods could not be used due to the large amount of data to be processed, but thanks to the progressive increase in computing power and hardware dedicated to deep learning, today it is possible to use machine learning techniques to detect phishing with a very high accuracy.

In this research, the CERT of the Agency for Digital Italy (CERT AgID) provided a dataset of URLs and phishing domains linked to brands of Italian public and private organizations, mainly banks, financial institutions and postal services. The best machine learning models available at the state of the art have been applied to this dataset. The results of the laboratory experiments on three families of models (features based, vector based and embedding) were surprisingly positive, as even without any parameterization, it was possible to obtain an accuracy greater than 99%, as in the case of the character and word level CNN. The same experiments were subsequently compared with a generic dataset not specifically linked to Italy and the results obtained in terms of accuracy were worse. This demonstrates how the use of a dataset of domains and URLs linked to a specific country could improve the accuracy of ML-based phishing URL detection systems.

1 Introduction

Phishing attacks against the most well-known Italian brands are detected every day by the Cert of the Agency for Digital Italy (Cert-AgID) and described in its weekly summaries [3]. The phenomenon of phishing is commonly perceived by the user as a less dangerous event than malware: in reality, most malware attacks occur as a result of a phishing attack, usually emails with office attachments containing malicious macros. To counter this type of attacks, black lists (or block lists) are usually used inside the perimeter security devices (router / firewall / ids /

ips). The downside to these lists is that they are generated through the collection and analysis of various external sources and require continuous updating by security analysts. Furthermore, these lists lose value very quickly as 80% of phishing pages have an average duration of less than 24 hours [20].

Attacks are therefore increasingly rapid and this is to be attributed to the greater simplicity with which attackers can now set up a phishing campaign. The Microsoft report [10] highlights how Phishing-as-a-service (PHaaS) is able to launch attacks on a large scale and with great simplicity. According to Redmond analysts, the online service BulletProofLink (also known as Anthrax) was able to create a phishing campaign by generating 300,000 unique subdomains. The platform, to simplify and speed up the attacker's work as much as possible, made available as many as 100 graphic templates from various brands ready for use. These types of platforms greatly facilitate criminal activity, even if they do not have specific technical skills and feed what is more generically defined as Cybercrime-as-a-Service.

Some research [21, 2, 26] shows how ML-based anti-phishing systems are more effective than blacklists. In general, machine learning techniques allow you to automatically build a model starting from data. Thanks to machine learning and in particular to its subset relating to ”neural networks” and deep learning (neural networks of greater complexity), it is basically possible to solve any problem in the real world (voice and visual recognition, autonomous driving, smart city, chat bot , etc.) especially when we have a large amount of data available. Many security systems already use machine learning to prevent and neutralize some cyber attacks and fight cybercrime. ML can be useful in identifying a phishing attack only based on the analysis of the string contained in the URL.

As is known, attackers use domain or subdomain names to create phishing pages so that the URL looks legitimate. Domain registration uses the so-called typosquatting technique which consists in slightly modifying the structure of the brand name making it visually similar. One of the ML techniques allows you to ”see” URLs as images and thanks to the large amount of data available (legitimate and phishing URLs), ML-based systems allow us to derive a model that can tell us in a very short time if a URL is malicious or not.

In this research, some of the best known machine learning models were tested on a dataset containing URLs of phishing pages written in Italian. The results showed that the linguistic factor can positively affect the accuracy of the results.

The rest of the paper is organized as follows: section 2 introduces the best known ML models used for phishing detection (SVM, RNN and CNN), section 3 describes the Italian dataset used in the experiments described in section 4. Finally, section 5 illustrates a possible implementation of the phishing detector within a browser extension (also mobile). This extension, in addition to alerting the user to a possible phishing URL, could allow the same user, thanks also to a blockchain-based incentive system, to report a malicious URL as phishing or non-malicious (false positive).

2 Related Works

2.1 Support Vector Machine based systems

Support-vector machines (SVM) are supervised learning models associated with learning algorithms for regression and classification. They operate through the kernel trick, increase the size of the dataset and then find the hyperplane with greater separation. When projected into the original space of the dataset, the hyperplane is no longer a place of linear points. The advantage of SVMs is that they are mathematically simple to model but less ”customizable”

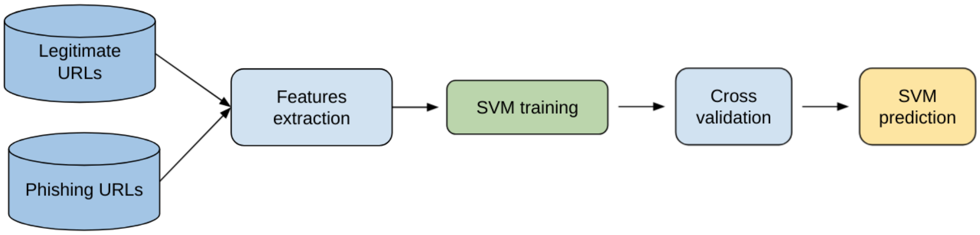


Figure 1: Example of an SVM network used for recognizing phishing URLs.

than NN techniques (which rely on gradient descent).

Research [26] presents an SVM network-based URL phishing detection system. Based on tests performed on a sample of data consisting of 2000 records, the system has an accuracy of 95.80%. For the input dataset, 1000 phishing URLs, taken from PhishTank [13] and 1000 legitimate URLs, were used. These URLs were downloaded half from the Alexa top 500 and half by querying the Google search engine through specific queries that returned domains containing a series of interest keys such as `*.bank.*`, `*.commerce.*`, `*.trade.*`.

For recognition, six characteristics were analyzed within the URL:

- the size of the URL;
- the number of dashes (legitimate sites rarely use ”-” characters);
- the number of points (for example this url contains 4 points: `sub-domain2.subdomain3.sub-domain4.mcomerce.com`);
- the number of numeric characters (usually numeric characters are not used on legitimate sites);
- the value of a Boolean variable indicating the presence of the IP in the URL (equal to 1 if there is an IP in the URL, 0 otherwise);
- the similarity index (measures the similarity of two data, it is equal to 100% if the two words are the same).

The calculation of the characteristics is done for each pair of phishing URLs and the corresponding legitimate URL.

The advantage of this solution is the good accuracy, but these systems cannot be used on mobile devices due to the required processing power and the consequent high battery usage.

2.2 Systems based on neural networks: RNN e CNN

There are different types of neural networks in the field of deep learning, and new methods or modifications to existing models are published and discussed every day. In the Phishing URL Detection sector, publications mostly belong to the following three classes of artificial neural networks: Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN).

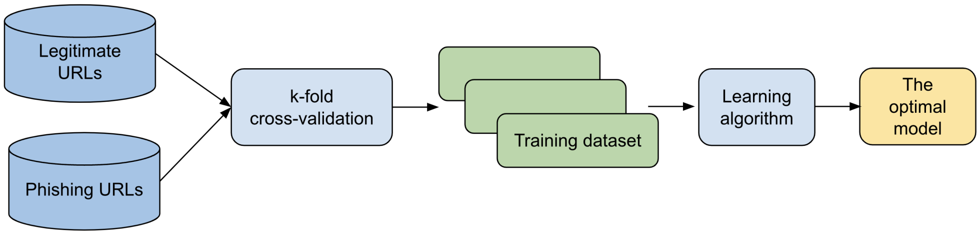


Figure 2: Example of a CNN network used for recognizing phishing URLs.

2.2.1 Recurrent neural network (RNN)

In recurrent neural network (RNN) based systems, URLs are parsed directly, rather than using features extracted from URLs. In fact, RNNs allow analyzing temporal phenomena and, in the case of anti-phishing systems, these networks are used to sequentially analyze the characters of the URL. In addition to RNNs, the most modern Long Short Term Memory (LSTM), a particular type of RNN network, can also be used, which allow to overcome the training problems of a classic recurring network.

2.2.2 Convolutional Neural Networks (CNNs / ConvNets)

Networks belonging to the convolutional family (CNN) are mainly used in applications for image and video recognition, in recommendation systems, in natural language processing (NLP) and in bioinformatics. A widely used CNN was LeNet 5 in the year 1998. This was applied by several banks to recognize handwritten numbers on checks, scanned into 32×32 pixel images.

The Figure 2 illustrates the general scheme of the system [1] capable of detecting phishing URLs thanks to a CNN network. The model basically consists of two large input datasets:

1. 50% consisting of a set of 10,604 phishing URLs obtained from PhishTank [13].

2. 50% consisting of a set of legitimate URLs obtained thanks to the Common Crawl Foundation. [5].

To leave as much space as possible for encoded characters, the protocol prefixes (http:// and https://) have been removed from the URLs. In this way, 7-8 characters were saved that were not used to identify phishing. The dataset was divided by 80% in the training set and 20% by the testing set. Subsequently, 5-fold cross validation was applied, a statistical technique used for training.

Some data regarding the dataset used:

- Total dataset set: 21,208 URLS with 10,604 phishing URLs and the other half legitimate.

- Number of characters in the CNN input vector: 256. The average length of the URLs was 186 characters and only 9% of the addresses were longer than 256 characters. So to optimize the network and define an appropriate number of convolutional layers it was decided to cut the URLs to the maximum length of 256 characters.

- Dictionary size for one-shot encoding: 70.

- Longest URL: 1149 characters.

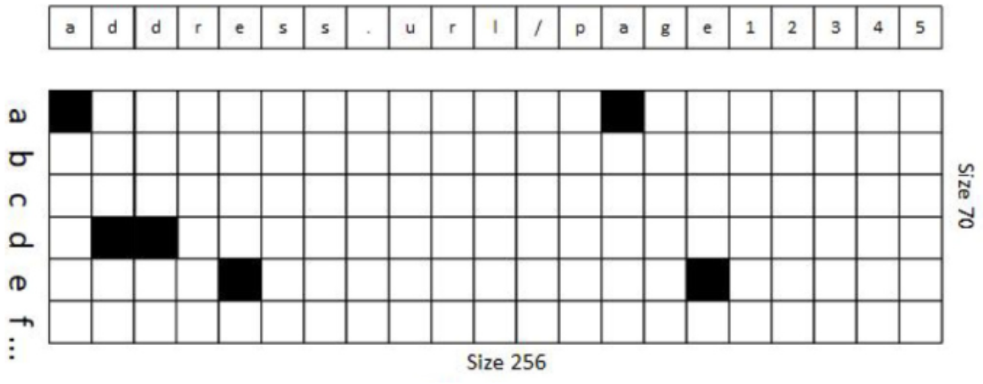


Figure 3: One-shot encoding of the URL `address.url/12345`

abcdefghijklmnopqrstuvwxyz0123456789
-,;.!?:'''/\|_@#$%^&*~`+-=<>()[]{}

Figure 4: 70-character dictionary for one-hot encoding.

- Shortest URL: 186 characters.

After an initial ”training” phase, learning is performed through a selected learning algorithm. The result is a very good model that almost instantly allows for binary output (legitimate or malicious URL).

Regarding the URL encoding, one-shot character-level encoding [25] is used. One hot encoding is the simplest way to transform a token (a character in this case) into a vector. To do this, the character is associated with an integer index i of a binary vector of size N (the size of the vocabulary), containing all zeros except the i-th element which is equal to 1. One-hot word-level coding is usually used in neural networks, but this requires a language-dependent dictionary of words. This encoding is therefore not ideal for URLs, because they often consist of words written in multiple languages or strings of non-contiguous words. To overcome the language dependency, one-hot character-level encoded URLs are then supplied to the neural network. For example, the Figure 3 shows the one-shot encoding of the URL address.url/12345. The URL is encoded in its entirety rather than being split into parts. The characters that define the URL should be defined in a dictionary (see Figure 4) of 70 unique characters, consisting of 26 letters, 10 digits and 33 other characters, in addition to the ”new line” character.

The characters of the URL not belonging to the alphabet defined above are removed. The encoded URL is essentially treated by the convolutional network as an input image to the CNN

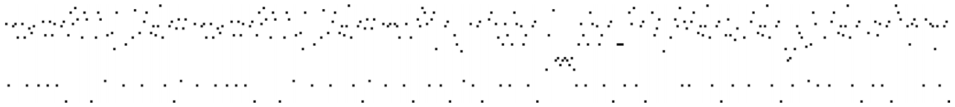
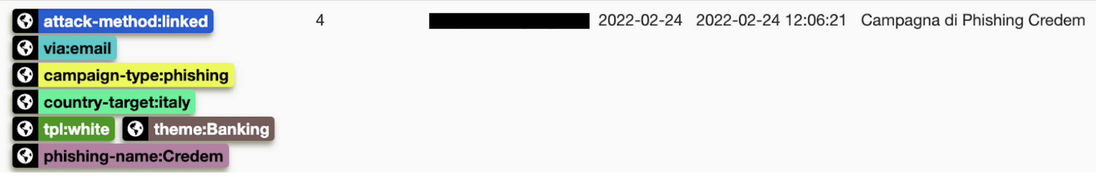


Figure 5: The input image created from a suspected UR by character-level one-hot encoding.



4 [redacted] 2022-02-24 2022-02-24 12:06:21 Campagna di Phishing Credem

Figure 6: Example of a Misp event describing a phishing campaign against the CREDEM bank

network (see Figure 5).

As mentioned above, CNN networks are useful for image classification and are designed not to be sensitive to image distortions. In the search, the CNN network has been modified to detect these distortions, in particular the distortions in URLs.

Experiments of this type demonstrate how convolutional networks allow the analysis of URLs without a dictionary using a convolutional neural network with very high accuracy (99.98% in this particular experiment). The trained system does not require high computational resources, therefore it can be used on mobile devices, especially in the most recent ones, where there are chipsets optimized for calculations related to machine learning.

3 Italian dataset

This research was made possible thanks to the invaluable contribution of the Computer Emergency Response Team (CERT) of the Agency for Digital Italy.[1]. This agency collects daily indicators of compromise related to malware and phishing campaigns in Italy. The activity is carried out thanks to OSINT activities and spontaneous reports from public and private organizations or ordinary citizens. CERT's security experts analyze malicious campaigns and subsequently census and share indicators of compromise through the MISP threat intelligence platform [11] or through a simple feed to accredited Public Administrations.

As shown in Figure 6, in addition to the landing page URL, other information useful for classification is associated with the phishing campaign, such as the theme (banking, delivery, payment, etc.), the TLP [19], the type of campaign, the name of the campaign (i.e. the brand of the impacted company) and the communication channel (e-mail, sms, social media, etc.). For this research, only the IoCs were sufficient, but it is not excluded that in a future possible evolution of the system, such information could be included for a possible automatic classification of the campaign.

For the export of IoCs from MISP, a search filter was set on the following tags: country-target: Italy and campaign-type: phishing. At the end of the export, a dataset containing 1857 ioc was obtained, of which 807 domains, 193 URLs in http and 857 URLs in https.

The dataset was integrated with 1857 benign domains, using a random scraper that browsed Italian domains. To verify the veracity of the extracted domains, a query was made to the Alexa[2] service and low ranking domains were removed. The dataset constructed is composed of 524 domains, 210 URLs in http and 1123 URLs in https. The final dataset, balanced and composed of 3714 examples, was divided into 70% for the training set, 10% for the validation set, and 20% for the testing set. The final dataset, balanced and made up of 3714 examples,

[1] Italian public agency that deals with technological innovation of the Public Administration under the direction and control powers of the President of the Council of Ministers or of the minister delegated by him.

[2] https://www.alexa.com/topsites

was divided into 70% for the training set, 10% for the validation set and 20% for the testing set.

4 Empirical comparison of Machine Learning Models

This section explains the general process of phishing ULR detection. The central idea of this work is to create phishing URL detection framework using a Machine Learning (ML) approach. To achive this goal we conducted an anlisys on the collected corpora using approaches: feature-based (Section 4.3), lexical-based (Section 4.4), holistic Transformers (Section 4.5).

4.1 Framework

The framework is largely structured into three phases.

1. Firstly, phishing URLs are collected. In this work, sources provided by the Computer Emergency Response Team (CERT) of the Agenzia per l'Italia Digitale and a set of randomly extracted URLs were used, as described in Section 3.

2. When an HTTP request is made by the client (browser), the phishing detector system checks the URL and decides whether it can be downloaded.

3. Finally, the framework, on a daily schedule, monitors the presence of new URLs in PhishTank [13] and trains the model with a larger dataset. Training takes about few minutes while checking the URL takes two milliseconds.

4.2 Methods & Models

The cornerstone of the proposed framework is a learning-based model. The code of the models in the following sections is opensource and available at the following repository[3]. In this section we propose several learning-based approaches, specifically fine-tuned for the ULR detection task.

4.3 Features-Based Models

The URL is the first thing to analyze of a website to decide whether it is phishing or not. As we mentioned in Section 2, URLs of phishing domains have some distinctive points such as length, presence of affixes or suffixes, or presence of some non-ASCII characters. The features that have been considered in this work are: URL length, URL depth (how many sub-paths x1/x2/x3/x4), Presence http/https, Presence of characters outside ASCII format, Presence of suffixes followed by ”-”, Presence of affixes preceded by ”-”, IP retrievable.

Extracted features were classified using linear Support Vector Machines (SVM), Decision Tree, XGBoost, Random Forest, Shallow MultiLayer Perceptrons MLP with three hidden layers of 100 neurons each, Autoencoder with 10 dense layers and 10 training epochs.

[3] https://github.com/LeonardRanaldi/ItalianPhishingDetection

4.4 Lexical-Based Models

While ML algorithms perform very well on categorical values, deep learning (DL) algorithms express the best potential when used with input encodings, as introduced in Section**??**. To investigate the role of ULR embeddings, we processed the input using one-hot encoding at the character-level, word-level, and symbolic-structural levels. Character-level and word-level embeddings were constructed on a 200-dimensional vector space using one-hot representations while symbolic-structural-level embeddings were constructed using the KERMIT [24] framework that produced syntactic encodings of dimension 4000 ready to be processed by a neural network. Then, we used seven different classifiers based on three Neural Networks architetures: a Convolutional Neural Network (CNN), a Recurrent Neural Network (RNN) and a FeedForward Neural Network (FFNN). The proposed classifiers are structured as follows:

- **Input layer:** at character level, at word level with size 200, while in the presence of both, the inputs were concatenated after the respective CNN and RNN layers;
- **CNN layer:** a convolutional layer is applied with a 200 dimension filter, then a max-pooling operation is applied on the feature map in order to calculate the feature vector and apply a softmax classifier to predict the outputs.
- **RNN layer:** a gated recurrent unit with size 200 is applied followed by three dense layers with size 4000,512 and 256 and softmax classifier to predict the outputs.
- **FFNN:** is composed of two layers of 4000 and 2000 respectively, finally the output layer of size 2. Between each layer a ReLu activation function and a dropout of 0.1 is used to avoid overfitting on the training data.

During training, the dropout probability regularization technique (p) 0.2 is used with deep networks in which network units are turned off randomly to avoid overfitting. All models were optimized using Adam [7] and the MSE loss function. After a fine-tuning phase, we decided to set the training epochs to 10 after a trade-off analysis between the required training time, computational resources, and the achieved accuracy.

4.5 Holistic Transformers

Transformers-based architectures are achieving state-of-the-art results in many NLP tasks. In this work, six Transfomers-based encoders were tested. Encoders for all the proposed models were implemented using Huggingface's Transfomers library [22]. The output of each encoder was decoded by a FeedForward Neural Network (FFNN) to a single output layer has the as input size 768 and output the number of classes, this had the function of classifier. For the training phase, optimizer Adam [7] and cross entropy loss function was used for 10 epochs as in [9].

4.6 Experiments

This section describes the general set-up of our experiments and the specific configurations adopted.

The performances of the models described in section 4.2 were tested on the dataset described in section 3, provided by the Computer Emergency Response Team (CERT) and integrated with a dataset of random Italian legit domains of the same size. Each experiment was repeated 5 times initializing the models with neural networks with 5 different seeds, this to make sure that the models based on neural networks do not produce anomalous results

Category	Model	Accuracy
Holistic Transformers	BERT$_{base}$ [6]	89.96
	Electra [4]	**90.37**
	XLNet [23]	88.89
	Ernie [18]	83.23
	BERT$_{multi}$ [14]	85.86
	RoBERTA [8]	85.95
	DistilBERT [17]	86.8
	BERT$_{italian-version}$ [15]	87.3
Vector-Based Models	CNN(Character embedd.)	77.96
	CNN(Word embedd.)	76.53
	CNN(Character+Word embedd.)	93.85
	RNN(Character embedd.)	76.88
	RNN(Word embedd.)	75.93
	RNN(Character embedd.+word embedd.)	**95.36**
	KERMIT(syntax Encoding)	92.78
Features-Based Models	SVM	68.3
	MultiLayer Perceptrons	69.2
	Random Forest	68.7
	Autoencoder	68.6
	XGBoost	68.5
	Decision Tree	67.9

Table 1: Comparison of different performances for on Italian dataset for phishing detection.

4.7 Discussion

We explored the performance of the models described in Section 4.2 on the dataset described in Section 3. The results reported in Table 1, show that natural language processing (NLP) algorithms combined with Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) that consider character and word embeddings perform better in terms of accuracy. Although, the supremacy of accuracy is held by CNN and RNN, the FeedForward Neural Network (FFNN) with syntactic input (KERMIT) also achieved sustainable results, ranking third. If the architectures mentioned so far are very complex, the models based on Transformers encoders, manage to obtain extraordinary results by simply using a universal encoder and a shallow classification layer. These models based on Transformers, although they seem to overfit some corpora and fail on others [16]. In this task they get very good results because it seems that the pre-processing pipeline composed by tokenizer, padder and other parts works very well as the total encoder. From the point of view of training require little effort, which is not true for the effort of the model itself, very heavy computational. Weak note, for the algorithms based on features, in fact it seems they have not been able to obtain very good performances. The reasons for the disappointing performances can be attributable to the number of features used not very significant. From the point of view of the computational elaboration and of the memory consumption the best results are those of the features-based algorithms (see Table 2), but at the same time they require a lot of time for the construction of the features. The fastest but also the heaviest are the algorithms based on Transformers, because they take in input pre-trained models and make a minimal part of fine-tuning to adapt few parameters.

Model	**Memory**	**Avg Time (Learning & Predictions)**
Holistic Transformers	∼1.5/2.5 Gb	5 min
CNN & RNN	∼750Mb/1 Gb	2 min
KERMIT	∼1.0Gb	10 min
Features-based	∼100Mb	15 min

Table 2: Report average learning times and memory consumption of proposed models.

5 Future works: domain monitoring, browser integration and user incentives

5.1 Domain monitoring

According to a report by the Anti-Phishing Working Group (APWG) [1] in 2016, out of 255,065 phishing attacks detected worldwide, 100,051 were domains of compromised sites and 95,424 were domain names registered by phishers, thus more than 50% of phishing was due to compromised sites. Over the years, the phenomenon of phishing has increased: as stated in the fourth quarterly report APWG of 2021, the number of attacks has risen to 316,747, but according to the report of PhishLabs [12] in February 2022 the percentage of compromised sites would have dropped to 36, 2%, while the remaining percentage is attributable to dissemination methods that exploit tunneling services, URL shortener, free hosting and above all paid and free domain registrations. These last two methods of diffusion, although today they represent only 21.4% of phishing attacks, could be effectively countered with a service for monitoring the TLD lists of the new registered domains applied to machine learning. Detecting new possible phishing domains in advance could be very helpful. According to data from APWG [1], new phishing domains sometimes take weeks or months to be used in attacks. This period of time could therefore be used to monitor the eventual publication of content and the execution of the take-down request process of the malicious site.

5.2 Browser integration and user incentives

Current last generation browsers (e.g., Mozilla Firefox and Google Chrome just to name two of the most famous) provide the possibility to create software packages to be inserted as plugins inside the browser. Taking advantage of this potentiality, it would be useful to implement an automatic system of detecting deceptive sites (for example, phishing and drive-by downloads) inside the browser's extension. The extension will have to foresee a system to block connections to the potentially risky site, notifying to the user the typology detected, who has signaled it as malicious, and all the information (like date and time of domain registration and kind of SSL certificate) that can help the surfer to choose whether to continue or interrupt the connection.

In addition, the plug-in must provide three other basic features:

The first one is related to the reporting of a false positive site, that is indicated as malicious but that in reality is not (based on the experience of the user who is browsing), in this case, the user can send a notice indicating the alleged error.

The second one refers to the possibility of reporting a malicious site based on the user's browsing experience if the plugin has identified it as safe.

The third and last feature, but not the least, should include a whitelisting system through which it is not possible to mark some sites as malicious (because one or more users could intentionally blacklist specific sites). Moreover, always in the whitelist, specific sites recognized as trusted will be indicated.

The browser extension should also integrate a cryptocurrency wallet to engage and incentivize users to contribute in exchange for a reward. The remuneration could be either in the form of a proprietary token that allows the user to receive benefits such as discounts, invitations to courses, promotions, and so on or with a token that can be spent on a decentralized exchange or exchange platform such as Automated Market Making (AMM) or similar. In addition, upon reaching some predefined thresholds (for example 10, 50, 100 reports) could be issued to the user a badge as a reward for the work done. This prize could be a product in NFT (Non-Fungible Token)[4] format, therefore unique and resalable. The ranking and distribution of prizes must be transparent and publicly available through a website developed with web3 technology. The ranking could list the top 100 signallers representing the Hall of Fame that will always be visible, uncensored, and certified as it is written on the blockchain. The signallers will also be able to access their personal profile through a special web page3.

Currently, there are many blockchains that could be used for this purpose. Among the most famous we can definitely cite the Ethereum ecosystem that unfortunately still suffers from very high fees to justify its use for this use case. Alternatively, we could use blockchain that exploits a different consensus system and therefore have greatly reduced costs, in some cases almost zero as for example the Polygon[5] network that is technologically based on the Ethereum[6] network (defined layer 1) as it has its own consensus system and a ”parallel” blockchain (defined layer 2). On layer 1 there are however alternative solutions to Ethereum very valid to be taken into consideration such as Basic Attention Token (BAT)[7], a solution already adopted by the browser Brave to reward users who use its technology and authorize the vision of advertisements. Another solution could be represented by the use of the XDC token of the XinFin foundation[8] based in Singapore, which aims to lower transaction costs by improving transparency and traceability of information. Last but not least, the Italian or European Blockchain Service Infrastructure (IBSI or EBSI) could be used, even if they are still in an experimental phase and we will have to wait a few more months before they are up and running.

6 Conclusion

In this article, we have explored the possibility of applying different machine learning techniques in order to evaluate their performance for detecting phishing URLs. To perform the experiments, a dataset provided by the Agid CERT was used: compared to other datasets, this one stands out for the URLs and phishing domains associated only with Italian campaigns. The use of a dataset targeted to the specific territory of a nation, has made it possible to obtain greater accuracy in the detection phase of new URLs.

This approach was able to detect, through the analysis of the URL with the trained model, almost all of the attacks with average detection times of around two milliseconds. In particular, algorithms that made use of convolutional neural networks proved superior. This result makes ML-based phishing detector systems a valid integration / alternative to support the classic black / white list, which can be implemented on all devices, including mobile ones.

[4] https://ethereum.org/en/nft/

[5] https://polygon.technology/

[6] https://ethereum.org

[7] https://basicattentiontoken.org/

[8] https://xinfin.org/

The detection system could also be integrated into today's browsers, through the implementation of a software extension capable of detecting in real time any phishing URLs, even of the ”zero-day” type. Such an extension could also integrate a cryptocurrency wallet to incentivize users to provide their contribution in exchange for a reward.

References

[1] Anti-Phishing Working Group (APWG). Global phishing survey: Trends and domain name use in 2016. https://docs.apwg.org/reports/APWG_Global_Phishing_Report_2015-2016.pdf, last viewed March 2022, 2017.

[2] Alejandro Correa Bahnsen, Eduardo Contreras Bohorquez, Sergio Villegas, Javier Vargas, and Fabio A. González. Classifying phishing urls using recurrent neural networks. In *2017 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–8, 2017.

[3] CERT-AgID. Weekly summaries of the cert-agid. https://cert-agid.gov.it/tag/riepilogo/, last viewed February 2022, 2020-2022.

[4] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

[5] Common Crawl Foundation. Common crawl datasets. https://commoncrawl.org/, last viewed February 2022, 2022.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[9] Pranav Maneriker, Jack Stokes, Edir Lazo, Diana Carutasu, Farid Tajaddodianfar, and Arun Gururajan. Urltran: Improving phishing url detection using transformers, 06 2021.

[10] Microsoft Security Blog. Catching the big fish: Analyzing a large-scale phishing-as-a-service operation. https://www.microsoft.com/security/blog/2021/09/21/catching-the-big-fish-analyzing-a-large-scale-phishing-as-a-service-operation/, last viewed February 2022, 2021.

[11] Misp Community. Open source threat intelligence sharing platform (misp. https://www.misp-project.org/, last viewed February 2022, 2022.

[12] PhishLabs. Quarterly threat trends intelligence. https://info.phishlabs.com/quarterly-threat-trends-and-intelligence-february-2022, last viewed March 2022, 2022.

[13] Phishtank. Phishtank database. https://phishtank.org/developer_info.php, last viewed February 2022, 2022.

[14] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert?, 2019.

[15] Marco Polignano, Pierpaolo Basile, Marco Degemmis, Giovanni Semeraro, and Valerio Basile. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *CLiC-it*, 2019.

[16] Leonardo Ranaldi, Aria Nourbakhsh, Arianna Patrizi, Elena Sofia Ruzzetti, Dario Onorati, Francesca Fallucchi, and Fabio Massimo Zanzotto. The dark side of the language: Pre-trained transformers in the darknet, 2022.

[17] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[18] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *ArXiv*, abs/2107.02137, 2021.

[19] The Forum of Incident Response and Security Teams (FIRST). Traffic light protocol (tlp) — version 1.0. https://www.first.org/tlp/, last viewed March 2022, 2016.

[20] Webroot. 84% of phishing sites exist for less than 24 hours. https://www.webroot.com/in/en/about/press-room/releases/quarterly-threat-update-about-phishing, last viewed February 2022, 2016.

[21] Wei Wei, Qiao Ke, Jakub Nowak, Marcin Korytkowski, Rafał Scherer, and Marcin Woźniak. Accurate and fast url phishing detector: A convolutional neural network approach. *Computer Networks*, 178:107275, 2020.

[22] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.0, 2019.

[23] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.

[24] Fabio Massimo Zanzotto, Andrea Santilli, Leonardo Ranaldi, Dario Onorati, Pierfrancesco Tommasino, and Francesca Fallucchi. KERMIT: Complementing transformer architectures with encoders of explicit syntactic interpretations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 256–267, Online, November 2020. Association for Computational Linguistics.

[25] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA, 2015. MIT Press.

[26] Mouad Zouina and Benaceur Outtaj. A novel lightweight URL phishing detection system using SVM and similarity index. *Human-centric Computing and Information Sciences*, 7(1):17, 2017.