Machine Learning Techniques for Italian Phishing Detection

Leonardo Ranaldi[1,3], Michele Petito[2], Marco Gerardi[1], Francesca Fallucchi[1], and Fabio Massimo Zanzotto[3]

[1] Department of Innovation and Information Engineering,
Guglielmo Marconi University, Roma, Italy
(l.ranaldi,f.fallucchi)@unimarconi.it
m.gerardi@studenti.unimarconi.it
[2] Agency for Digital Italy (AgID), Roma, Italy
petito@agid.gov.it
[3] Department of Enterprise Engineering,
University of Rome Tor Vergata, Roma, Italy
fabio.massimo.zanzotto@uniroma2.it

Abstract

In recent years, several methods have been developed to combat phishing. These include approaches based on blacklisting/whitelisting, visual similarity, search engines, and, more recently, Machine Learning (ML). Despite scientific research showing how ML-based phishing detection systems are effective and require frequent updates by analysts and the problem of the duration of phishing pages, blacklists do not allow the detection and blocking of so-called zero-day attacks. So, while blacklists are growing in size, phishing attacks are not decreasing.

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. The dataset of phishing URLs and domains related to brands of Italian public and private organizations, mainly banks, financial institutions, and postal services, was provided by the CERT of the Agency for Digital Italy (CERT AgID). The obtained results are promising and show the high performance obtained by models based on pre-trained encoders [29] and models that encode character and word levels in Convolutional Neural Networks and Recurrent Neural Networks with light-weight training for malicious software detection.

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 user commonly perceives the phenomenon of phishing 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. Blacklists (or blocklists) are usually used inside the perimeter security devices (router/firewall/ ids/IPS) to counter this attack. 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 fewer than 24 hours [31].

The Microsoft report [14] highlights how Phishing-as-a-service (PHaaS) can launch attacks on a large scale and with great simplicity. 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. 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. To simplify and speed up the attacker's work as much as possible, the platform made available as many as 100 graphic templates from various brands ready for use. These 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 [32, 2, 41] shows how Machine Learning-based anti-phishing systems are more effective than blacklists. Machine Learning (ML) techniques allow you to build a model starting from data automatically. Thanks to ML and its subset relating to Neural Networks, it is possible to solve any problem in the real world (voice and visual recognition, autonomous driving, smart city, chatbot, etc.), especially when we have a large amount of data available. Many security systems already use ML to prevent and neutralize some cyberattacks and fight cybercrime. ML can be helpful in identifying a phishing attack only based on the analysis of the string contained in the URL.

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 slightly modifies 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 a large amount of data available (legitimate and phishing URLs), ML-based systems allow us to derive a model that can tell us in a short time if a URL is malicious or not.

In this research, some of the best-known ML models were tested on a dataset containing URLs of phishing pages written in Italian. The results showed that the linguistic factor could 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, and Section 3 describes the Italian dataset used in the experiments described in Section 4.1. Finally, Section 5 illustrates a possible implementation of the phishing detector within a browser extension (also mobile). In addition to alerting the user to a possible phishing URL, this extension 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

In the literature, the cyber attack called phishing is treated differently. One of the approaches to counter phishing is a simple blacklist containing known phishing websites, and those blacklists are typically deployed as plug-ins in browsers to check each URL entry in the blacklist. Then it prevents the user whenever he attempts a connection to one of these malicious websites, which are included on the blacklist. However, this approach is not super-fast, and the critical update process is sometimes slow. For this reason, over the years, at the same pace as the spread of Machine Learning (ML), several models for phishing identification based on automatic heuristics have become widespread. ML-based heuristics to detect phishing URLs has a well-defined pipeline. In Figure 1, we can see a dataset, a URL processing phase, the division of processed URLs into train and test, and a learning and fine-tuning phase. URL processing can be of three kinds: based on the construction of a set of fundamental features that analyze the network traffic and the URL, which we will call Features-Based Models (Section 2.1); superficial features that analyze the construction and structure of the URL using complex representations and particular encodings, which we will call Vector-Based Models, (Section 2.2); encodings provided by Transformer-based models and pre-trained on large corpora, much used in Natural

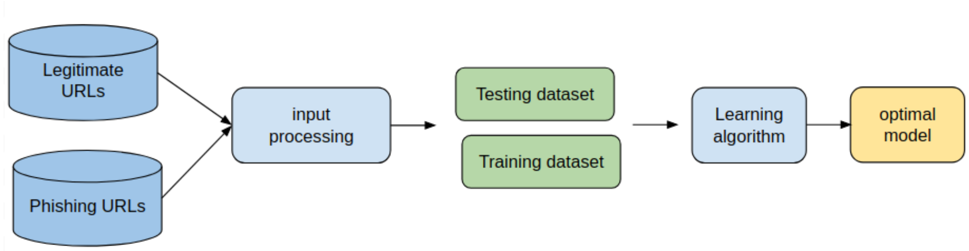

Figure 1: Example of a pipeline of a model used to recognize phishing URLs.

Language Processing tasks, which we will call Holistic Transformers (Section 2.3).

2.1 Features-Based Models

One of the first works to propose a lightweight URL phishing detection system using ML and similarity index was Zouina et al. [41]. They presented a phishing URL detection system based on the SVM network and tests performed on a data sample consisting of 2000 records: 1000 phishing URLs, taken from PhishTank [17] and 1000 legitimate URLs. These URLs were downloaded half from the Alexa top 500 and a half by querying the Google search engine through specific queries that returned domains containing a series of interest keys such as `*.bank.*`, `*.commerce.*`, `*.trade.*`.

The features that were used in [41] were: the size of the URL; the number of dashes; the number of points (for example, this URL contains 3 points `sub-domain2.subdomain3.mcomerce.com`); the number of numeric characters; the value of a Boolean variable indicating the presence of the IP in the URL; the similarity index (measures the similarity of two data, it is equal to 100% if the two words are the same).

Unfortunately, these systems cannot be used on mobile devices due to the required processing power and high battery usage, despite the high accuracy results.

2.2 Vector-Based Models

Sometime later, with the advent of Artificial Neural Networks (ANN), the algorithms based on the computation of features declined because they were not always fast and efficient. In the Phishing URL Detection sector, as in others, techniques for encoding data (Section 2.2.1) begin to become widespread. At the same time, many works based on ANN models are also developed, mainly: Recurrent Neural Networks(RNN) [30, 8, 25] and Convolutional Neural Networks (CNN) [35, 9, 38].

2.2.1 Encoding

The construction of the features resulted in computationally expensive. In fact, with the ANN's diffusion, various solutions have been diffused to represent the input. URL is transformed into one-shot character-level encoding [40]. One-hot encoding is the simplest way to transform a token, or a character, into a vector. To this end, the character is associated with an integer index i of a binary vector of size N (the vocabulary size), containing all zeros except the $i-th$ element equal to 1. One-hot word-level coding is usually used in ANN, but this requires a language-dependent dictionary of words. Therefore, this encoding is not ideal for URLs

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

Figure 2: 70-character dictionary for one-hot encoding.

because they often consist of words written in multiple languages or strings of non-contiguous words. One-hot character-level encoded URLs perform an additional URL preprocessing step to overcome language dependence. 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. In this example, 70 unique characters, consisting of 26 letters, 10 digits, and 33 other characters, in addition to the "newline" character (see Figure 2).

2.2.2 Recurrent Neural Networks

In Recurrent Neural Network (RNN) based systems, URLs are parsed directly, using the particular encoding (Section 2.2.1), rather than using features extracted from URLs. 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 [30]. In addition to RNNs, the most modern Long Short Term Memory (LSTM), a particular type of RNN network, can also be used, which allows for overcoming the training problems of a classic recurring network [25]. Moreover, thanks to a series of visualizers [8], it is possible to identify the portions of the coding that affected the final prediction, making the model explainable.

2.2.3 Convolutional Neural Networks

In Convolutional Neural Network (CNN) based systems, mainly used in image and video recognition applications, in recommender systems, the power of Convolutional layers [32] and self-attention mechanisms are exploited to find regularities in URL encodings [9, 38]. They seem to be able to outperform Recurrent networks even in contexts where the training data is unbalanced, thanks to a self-generating mechanism usable in CNNs [35]. As in RNNs, also in the case of CNNs, the input is no longer processed following a long phase of feature processing but is the result of encoding at a character level, as explained in Section 2.2.1. These types of input adapt very well to the architectures of CNNs.

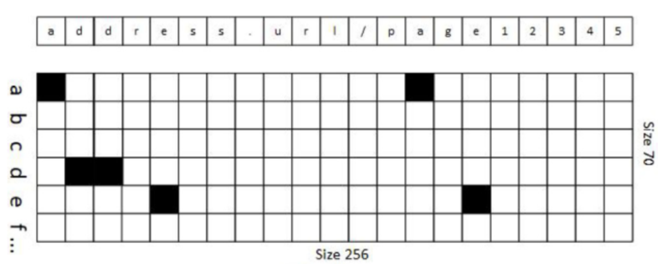

Figure 3: One-shot encoding of the URL `address.url/12345`

2.3 Holistic Transformers

Transformers-based architectures are achieving state-of-the-art results in many NLP tasks. At the core of these models is precisely a neural architecture called Transformer [29]. Transformer offsets can be used in various configurations like [5, 4, 37]. Maneriker et al. [13] proposed URLTran, which uses Transformers to significantly improve phishing performance over a wide range of meager false-positive rates compared to other deep learning-based methods. The sustainable performance seems to result from extensive pre-training and accurate fine-tuning. In addition, the raw input processing pipeline seems to work very well, even in the case of URLs, as in this task. This last apparatus of text pre-processing was not analyzed superficially in Features-based models and was completely ignored in one-hot encodings.

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. Thanks to OSINT activities and spontaneous reports from public and private organizations or ordinary citizens, the activity is carried out. CERT's security experts analyze malicious campaigns and, subsequently, census and share indicators of compromise through the MISP threat intelligence platform [15] or through a simple feed to accredited Public Administrations.

Figure 4: Example of a Misp event describing a phishing campaign against the CREDEM bank.

As shown in Figure 4, in addition to the landing page URL, other helpful information for classification is associated with the phishing campaign, such as the theme (banking, delivery, payment, etc.), the TLP [27], 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

[1]Italian public agency that deals with the 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

composed of 3714 examples, was divided into 70% for the training set and 30% for the testing set.

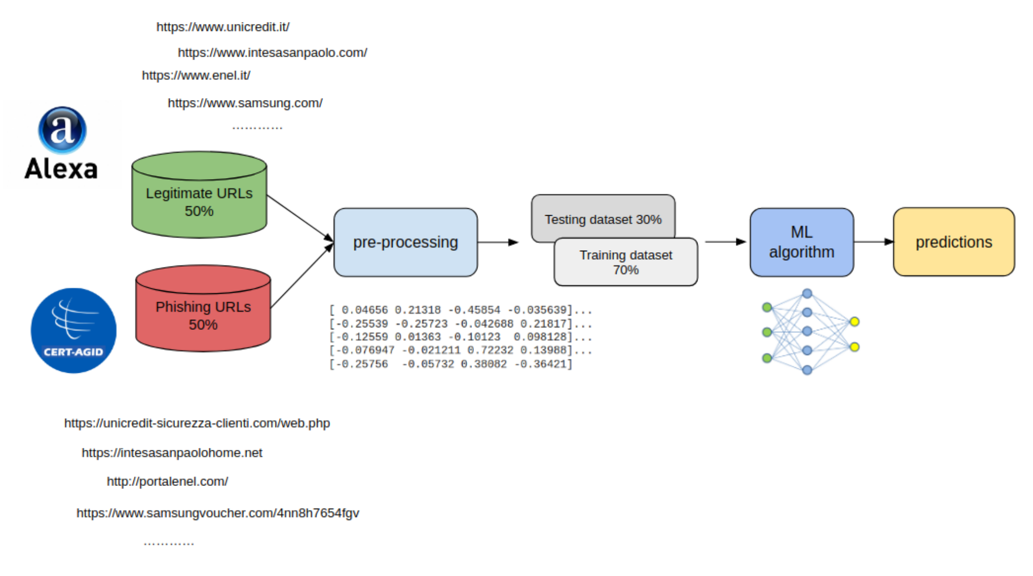

Figure 5: Framework of the proposed Machine Learning phase.

4 Empirical comparison of Machine Learning Models

This section explains the general process of phishing URL detection. The central idea of this work is to create a phishing URL detection framework using a Machine Learning (ML) approach. To achieve this goal we conducted an analysis on the collected corpora using approaches: Feature-based (Section 4.2), Lexical-based (Section 4.3), Holistic Transformers (Section 4.4).

4.1 Methods & Models

The cornerstone of the proposed framework is a learning-based model, Figure 5. The code of the models in the following sections is open source and available at the following repository[3]. This section proposes several learning-based approaches, fine-tuned explicitly for the URL detection task.

4.2 Features-Based Models

The URL is the first thing to analyze 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 or presence of affixes and suffixes. The features that have been considered in this work are: URL length, URL depth (how many sub-paths x1/x2/x3/x4), presence of HTTP/HTTPS, presence of characters outside ASCII format, presence of suffixes/affixes followed/preceded by "-," IP retrievable.

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

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.

4.3 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 Section2.2. To investigate the role of URL 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 [39, 21] framework that produced syntactic encodings of dimension 4000 ready to be processed by a Neural Network (NN). Then, we used seven different classifiers based on three NNs architectures: a Convolutional Neural Network (CNN), a Recurrent Neural Network (RNN), and a Feed-Forward 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 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 a softmax classifier to predict the outputs.
- **FFNN:** is composed of two layers of 4000 and 2000 respectively, and finally, the output layer of size 2. Between each layer, a ReLu activation function and a dropout of 0.1 are used to avoid overfitting the training data.

During training, the dropout probability regularization technique (p) 0.2 is used with NNs in which network units are turned off randomly to avoid overfitting. All models were optimized using Adam [11] 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.

4.4 Holistic Transformers

Transformers-based architectures are achieving state-of-the-art results in many NLP tasks. In this work, eight Transformers-based encoders were tested. The proposed encoders differ by corpus and parameters set in the pre-training phase. Encoders for all the proposed models were implemented using Huggingface's Transformers library [33]. The output of each encoder was decoded by a FeedForward Neural Network (FFNN) to a single output layer has the input size 768 and output the number of classes; this had the function of a classifier. For the training phase, optimizer Adam [11] and cross-entropy loss function was used for 10 epochs as in [13].

4.5 Experiments

This section describes the general set-up of our experiments and the specific configurations adopted. The performances of the models described in Section 4.1 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 NN with 5 different seeds to ensure that the NN-based models did not produce anomalous results. The metric that best describes performance is accuracy since the classification is binary and the dataset is balanced.

4.6 Discussion

We explored the performance of the models described in Section 4.1 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 CNN and RNN hold the supremacy of accuracy, the Feed-Forward 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 are based on Transformers; although they seem to overfit some corpora and fail on others [23], one of the possible reasons could be the learning mode [22]. In this task, they get excellent results because it seems that the pre-processing pipeline composed of tokenizer, padder, and other parts works very well as the total encoder. From the point of view, training requires little effort, which is not true for the effort of the model itself, which is very heavy computational. Weak note, for the feature-based algorithms, has not been able to achieve excellent performance with the proposed features. We plan to expand the features in future developments to study how much these may affect the boosted performance. From the point of view of computational processing and memory consumption, the best results are those of feature-based algorithms (see Table 2), but at the same time, they take a long time for feature construction. The fastest and 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 a few parameters.

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 [16] 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. Although today they represent only 21.4% of phishing attacks, these last two methods of diffusion could be effectively countered with a service for monitoring the TLD lists of the newly 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 could therefore be

Category	Model	Accuracy
Holistic Transformers	$BERT_{base}$ [5]	89.96
	Electra [4]	**90.37**
	XLNet [37]	88.89
	Ernie [26]	83.23
	$BERT_{multi}$ [18]	85.86
	RoBERTA [12]	85.95
	DistilBERT [24]	86.8
	$BERT_{italian-version}$ [19]	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 of models proposed on the Italian dataset for phishing detection.

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

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

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 plug-ins 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 the user of 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 is related to reporting a false positive site, which is erroneously indicated as malicious. In this case, the user can report the alleged error. The second one refers to the possibility of reporting a malicious site based on the user's browsing experience if the plug-in 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. 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 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 (10 or 100 reports) could be issued to the user a badge as a reward for the work. This prize could be a product in NFT (Non-Fungible Token)[7] 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 profile through a special web page3.

Currently, many blockchains could be used for this purpose. Excluding the famous Etherium [34] which has high fees, we could use blockchain that exploits a different consensus system and therefore has greatly reduced costs, in some cases almost zero for example the Polygon network [20] that is based on the Ethereum network (defined layer 1) as it has its consensus system and a "parallel" blockchain (defined layer 2). On layer 1, there are alternative solutions to Ethereum very valid to be taken into consideration such as Basic Attention Token (BAT)[28], which is already adopted by the browser Brave to reward users who use it and authorize the vision of advertisements.

Another solution could be represented by the use of the XDC token of the XinFin Foundation [36] based in Singapore, which aims to lower transaction costs by improving transparency and information traceability. Last but not least, the Italian or European Blockchain Service Infrastructure (IBSI [10] or EBSI [6]) 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 to evaluate their performance in detecting phishing URLs. A dataset provided by CERT Agid was used to run the experiments: compared to other datasets, this one stands out for the phishing URLs and 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 phishing URLs with very high accuracy values, in particular with algorithms based on Convolutional Neural Networks. 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.

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] 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.

[6] EBSI. Experience the future with the european blockchain services infrastructure (ebsi)., 2022.

[7] Ethereum. Non-fungible tokens (nft)., 2022.

[8] Tao Feng and Chuan Yue. Visualizing and interpreting rnn models in url-based phishing detection. In *Proceedings of the 25th ACM Symposium on Access Control Models and Technologies*, SACMAT '20, page 13–24, New York, NY, USA, 2020. Association for Computing Machinery.

[9] Yongjie Huang, Qiping Yang, Jinghui Qin, and Wushao Wen. Phishing url detection via cnn and attention-based hierarchical rnn. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 112–119, 2019.

[10] IBSI. Ibsi italian blockchain service infrastructure., 2022.

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

[12] 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.

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

[14] 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.

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

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

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

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

[19] 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.

[20] Polygon. Bringing the world to ethereum., 2022.

[21] Leonardo Ranaldi, Francesca Fallucchi, Andrea Santilli, and Fabio Massimo Zanzotto. Kermitviz: Visualizing neural network activations on syntactic trees. In Emmanouel Garoufallou, María-Antonia Ovalle-Perandones, and Andreas Vlachidis, editors, *Metadata and Semantic Research*, pages 139–147, Cham, 2022. Springer International Publishing.

[22] Leonardo Ranaldi, Francesca Fallucchi, and Fabio Massimo Zanzotto. Dis-cover ai minds to preserve human knowledge. *Future Internet*, 14(1), 2022.

[23] 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.

[24] 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.

[25] Yang Su. Research on website phishing detection based on lstm rnn. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 1, pages 284–288, 2020.

[26] 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.

[27] 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.

[28] Basic Attention Token. Bat – making crypto and defi accessible and useable for everyone., 2022.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[30] Weiping Wang, Feng Zhang, Xi Luo, Shigeng Zhang, and Angel M. Del Rey. Pdrcnn: Precise phishing detection with recurrent convolutional neural networks. *Sec. and Commun. Netw.*, 2019, jan 2019.

[31] 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.

[32] 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.

[33] 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.

[34] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger.

[35] Xi Xiao, Wentao Xiao, Dianyan Zhang, Bin Zhang, Guangwu Hu, Qing Li, and Shutao Xia. Phishing websites detection via cnn and multi-head self-attention on imbalanced datasets. *Computers Security*, 108:102372, 2021.

[36] xinfin. Enterprise ready hybrid blockchain for global trade and finance., 2022.

[37] 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.

[38] Suleiman Y. Yerima and Mohammed K. Alzaylaee. High accuracy phishing detection based on convolutional neural networks, 2020.

[39] 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.

[40] 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.

[41] 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.