Proceedings of the 2nd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/73/20240376

AccuRapidGuard: Advancing phishing website detection with high performance and efficiency

Qi Sun

School of Software Engineering, Xi'an Jiaotong University, Xi'an, China

972264218@stu.xjtu.edu.cn

Abstract. Cyber attackers continually develop sophisticated techniques to create deceptive phishing websites, resulting in financial losses and the exposure of personal information. As a result, there is a growing demand for phishing detection solutions that are practical for everyday use. However, current deep learning-based models offer good accuracy and generalization but require extensive computational resources, making them impractical due to long training times and slow inference speeds. To overcome the limitation, we propose AccuRapidGuard, a deep learning model that utilizes parallel CNN layers and an RNN layer with attention mechanism. Extensive experiments demonstrate that AccuRapidGuard not only achieves great performance with high accuracy, low false positive rates, and excellent generalization, but also provides outstanding efficiency, significantly reducing training time and inference speed compared to state-of-the-art baseline models. This combination of exceptional performance and superior efficiency positions AccuRapidGuard as a highly valuable and practical solution for phishing detection.

Keywords: Phishing Detection, Cyber Security, Deep Learning

1. Introduction

With the rapid development of the internet, accessing knowledge and information has become significantly easier, greatly enhancing the convenience of our lives. However, this convenience has also led to a concerning surge in cyber attacks and frauds. Cyber attackers, equipped with advanced techniques, can create deceptive phishing websites that closely mimic legitimate businesses. Their goal is to unlawfully obtain users' personal identity data and financial account credentials, which they then propagate through email, links, social media, and other channels. These deceptive activities consistently result in significant financial losses. The latest Phishing Activity Trends Report from the Anti-Phishing Working Group (APWG) reveals a staggering 1,350,037 observed phishing attacks in the fourth quarter of 2022, setting a new record and marking the worst quarter for phishing that APWG has ever witnessed. Furthermore, APWG has noted a steep increase in the total number of phishing attacks, with an annual acceleration of over 150% from 2019 to 2022 [1].

To counter the threat posed by phishing websites, researchers have been diligently exploring various approaches to detect their legitimacy. The current prominent detection methods primarily fall into categories such as list-based, heuristic-based, visual similarity-based, machine learning-based, and deep learning-based techniques [2]. Among these models, particularly those based on machine learning and deep learning, there are some that achieve outstanding performances with high accuracy, low false

DOI: 10.54254/2755-2721/73/20240376

positive rates, and excellent generalization. However, few of them achieve high efficiency as well, with least training time, test time and inference speed, due to the nature of extensive computational resources requirement of deep learning model.

Moreover, from the user's perspective, there is a pressing need for a practical solution that can effectively detect phishing websites in daily use, offering both low latency and high accuracy. Therefore,an efficient solution that automatically and swiftly extracts valuable features from the URL of a suspicious website, enabling accurate detection of phishing websites, appears promising.

In this paper, we propose a deep learning-based solution, named AccuRapidGuard, that exclusively extracts features from the URL of a suspected website, enabling precise legitimacy detection. Specifically, we employ parallel CNN layers with different convolutional kernel sizes to extract crucial character-level local features of varying scales from the URL. This design allows for the extraction of comprehensive local features while significantly reducing the training and inference time. These local features are then concatenated into a feature map. Subsequently, an RNN layer with an attention mechanism is utilized to extract contextual features based on the aforementioned local feature map. By collectively extracting the most significant local features, the input to the RNN layer contains minimal noise, resulting in relatively accurate contextual features. Additionally, the attention mechanism is employed to compute the importance of each attribute feature, thereby highlighting the key distinguishing features that differentiate phishing websites from legitimate ones and further enhancing the model's accuracy. Considering that the Gated Recurrent Unit (GRU) offers comparable efficiency to the Long Short-Term Memory (LSTM) while significantly improving computational speed[3, 4], we select GRU as the backbone of our RNN layer, which not only saves training and inference time but also ensures high detection accuracy from contextual features. Finally, we employ a Multi-Layer Perceptron (MLP) to determine the legitimacy of the website.

The main contributions of our work are summarized as follows:

a) We propose a deep learning-based model named AccuRapidGuard that simultaneously achieves high accuracy and low latency. By leveraging the strengths of both CNN and GRU, we can extract comprehensive features from the suspected URL, enabling the model to make accurate classifications while ensuring a limited runtime.
b) We have built a large-scale dataset with over 500,000 URL samples, consisting of a set of benign URLs crawled from Common Crawl and a set of phishing URLs obtained from PhishTank. The dataset is regularly updated, with dead websites being removed to maintain its relevance.
c) We conducted extensive experiments, comparing our model with four baseline models to demonstrate its superiority. The results showed that our model outperformed the others in terms of both latency and accuracy, with the lowest runtime and the second highest accuracy.

2. Overview of Proposed Model

2.1. Motivation

By analyzing the existing approaches, we conclude that most of current methods suffer from the following drawbacks:

a) Dependency on third-party services: such as blacklists, whitelists, search engines, etc [2]. This dependency can introduce latency, instability, and reliance on external sources, which may not adequately address new and emerging phishing attacks.
b) Dependency on handcrafted features and expert knowledge: Heuristic-based methods and machine learning-based methods may rely on the accuracy of expert knowledge, rules, and manually designed feature extraction techniques, leading to false positives and false negatives in phishing detection[2, 5].
c) Training and inference time: Training and using a model for inference based on deep learning methods with high accuracy and generalization ability can be time-consuming, requiring extensive computational resources. And the need for accessing external services for list-based and heuristic-based methods can introduce extra inference time[6].

DOI: 10.54254/2755-2721/73/20240376

Given these limitations, our objective is to address these challenges by introducing a deep learning-based model that operates independently of third-party services and expert knowledge. This model will have the ability to automatically extract features from raw datasets, thereby eliminating the need for manual feature engineering. Additionally, we aim to mitigate the training and inference time by optimizing the model's architecture, allowing for efficient computations without compromising accuracy or generalization. We firmly believe that such a comprehensive model, free from the aforementioned drawbacks, will prove invaluable for practical use in the realm of phishing detection.

2.2. Problem Definition

To abstract the problem of phishing detection, we formulate it as a binary classification with suspected URLs as inputs. In training dataset $T = \{x_1, x_2, \cdots, x_n\}$, $i = 1,2, \cdots, n$, where x_i represents a suspected URL and $y_i \in \{0,1\}$ stands for its label and n is the number of URLs in dataset T. when $y_i = 1$, it represents a phishing URL. On the other hand, when $y_i = 0$, it signifies a legitimate one. The aim of proposed method is to compose a network f, obtaining all predicted label $\widehat{y_k} = f(x_k)$ for every x_k in T and calculate the loss function $\sum L(y_k, \widehat{y_k})$ for backpropagation to find the best weight parameters.

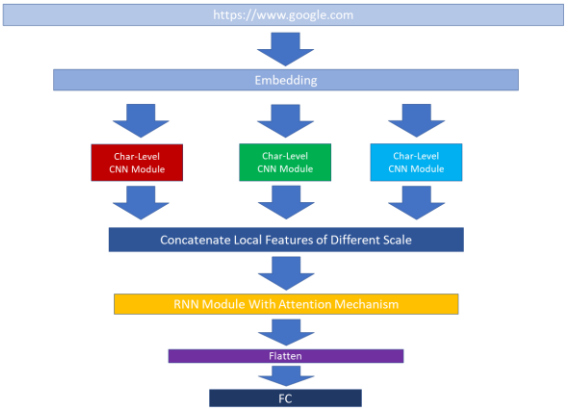


Figure 1. AccuRapidGuard

2.3. Model design

In order to achieve our aim, we proposed a deep learning-based model consisting of CNN, RNN and Attention module named AccuRapidGuard. The layers of the model are well organized, resulting of high performance of accuracy and low training and inference cost. Fig.1. depicts the structure of proposed model.

a) *Embedding layer.* To mitigate the potential larger attack surface caused by unknown characters in URLs, we collect all characters that appear in the training, validation, and testing sets to strive for a comprehensive vocabulary. Only the characters that do not appear in the dataset will be considered as unknown characters and represented as $< unk >$. All characters in the vocabulary are encoded and then embedded into k-dimensional vectors in the embedding layer. The character embedding matrix starts with random initialization and is learned during the end-to-end optimization process. In this way, the representation of each character is stored in the embedding matrix $E \in \mathbb{R}^{M \times K}$, where M equals the size of vocabulary and each row corresponds to the embedding vector of a character. Consequently, a URL is represented as a matrix $X \in \mathbb{R}^{L \times K}$, where L is the padding length of URLs and each row of X represents a character in URL.

b) *CNN layer.* Convolutional Neural Networks (CNNs) are designed to extract local features by utilizing rolling convolutional kernels, and are commonly applied in handling classification problems. According to the research conducted by C.Alexis et al. [7], The convolutions capture n-gram features from tokens (characters), where the length of the n-gram varies to capture both short-term and long-term relationships. In other words, these convolutional layers capture different lengths of n-gram features by using convolutional kernels of varying sizes. The size of

DOI: 10.54254/2755-2721/73/20240376

each kernel determines the length of the n-grams it can capture. By applying multiple kernels of different sizes, the convolutional layers can simultaneously capture n-gram features of different lengths. This provides the ability to encode important relationships and semantic information across different ranges in the text and filter non-important local noise. In proposed work, we employed a parallel CNN layer with three CNN modules with different kernel size show in Fig.2.

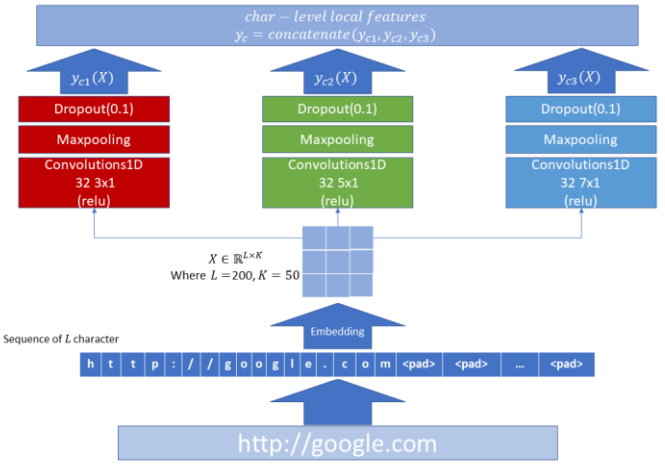


Figure 2. Structure of CNN Layer

The advantages of the design is: (1) It can capture multiple useful local features from raw URLs, which guarantee a comprehensive feature extraction of both short-term and long-term relationships. (2) It can reduce the running time, obtaining a low latency while keeping a high effective feature extraction, which helps it achieve outstanding performance.

c) *RNN layer.* RNN excel in processing sequential data, and effectively address the challenge of capturing the dependencies between preceding and subsequent elements within a sequence. it possesses the ability to retain and utilize information from the past when performing computations on the current input, allowing for effective modeling of sequential relationships. Given that the inputs are URL texts and we aim to extract contextual and structural features, it is appropriate to utilize RNN module in this task.
Based on the experiments conducted by Kyunghyun et al.[4] and Junyoung et al.[3], the Gated Recurrent Unit (GRU) offers comparable efficiency to the Long Short-Term Memory (LSTM) while significantly improving computational speed, we apt to utilize GRU in AccuRapidGuard to shrink latency. As a matter of fact, the contextual and structural feature extraction is based on the output feature map from aforementioned CNN layer which contains the important relationships and semantic information from raw URL and fewer non-important local noise, facilitating the efficiency and accuracy of contextual and structural feature extraction Consequently.

d) *Attention mechanism.* Furthermore, we embedded attention mechanism in AccuRapidGuard. By introducing attention mechanism, the focus on key features in model have been effectively enhanced. The attention mechanism enables AccuRapidGuard to dynamically learn and adopt key features from different phishing URLs. It automatically adjusts weights of features and emphasizes those closely associated to phishing behavior while disregarding features with little impact on classification results. This improvement leads to higher detection rates and accuracy in identifying phishing attempts.

DOI: 10.54254/2755-2721/73/20240376

Table 1. Information of Datasets

Dataset	Label	Train	Validation	Test	All
DS1	Phishing	220991	27624	27624	564434
	Benign	230556	28819	28820	
DS2	Phishing	607425	76162	75775	1559362
	Benign	640064	79774	80162	

3. Experiment

3.1. Datasets

To ensure the generalization and stability of the experimental models, we used two different datasets consisting of real-world URLs.

The first dataset, DS1, was constructed by us. We utilized a web crawler program to scrape the latest 556,305 phishing URLs from the PhishTank website. We then filtered out any URLs that were inaccessible or contained errors, resulting in 276,239 phishing URLs that met the experimental requirements. The benign URLs were obtained from Common Crawl, and after data cleaning and processing, we retained a total of 288,195 benign URLs.

The second dataset, DS2, is a larger publicly available dataset from an experiment led by Tao et al. [8]. It comprises a total of 1,559,362 URLs, including 759,362 phishing URLs and 800,000 benign URLs. Also, the phishing URLs and benign URLs were obtained from PhishTank and Common Crawl respectively, too.

Each dataset was divided into training, validation, and test sets in an 8:1:1 ratio. Detailed information has been summarized in Table I.

3.2. Baselines

There are three state-of-the-art deep learning-based models as baselines in this experiment. The first one is PDRCNN [9] which employs first a bidirectional LSTM networks and then a CNN module. The second is BiGRU [8] which consists of a bidirectional GRU layer. The last one is CNN-Fusion [10], a lightweight convolutional model with one-layer CNN in parallel with varying kernel sizes.

3.3. Evaluation Metrics

We will primarily evaluate the proposed model and baseline models from two aspects: performance and efficiency, which are two distinct aspects when evaluating a model.

a) Performance refers to how well a model accomplishes its intended task or objective based on the model's ability to produce accurate and reliable results. It measures the effectiveness of the model in achieving the desired outcomes. In the context of a classification model, performance metrics such as accuracy, precision, recall, and F1 score are used to assess how accurately the model classifies the data.

b) Efficiency relates to how effectively a model utilizes computational resources, such as time, memory, and processing power, to achieve its objectives. It measures the speed, resource utilization, and scalability of the model. Efficiency is often evaluated based on factors such as training time, testing time, and inference speed.

DOI: 10.54254/2755-2721/73/20240376

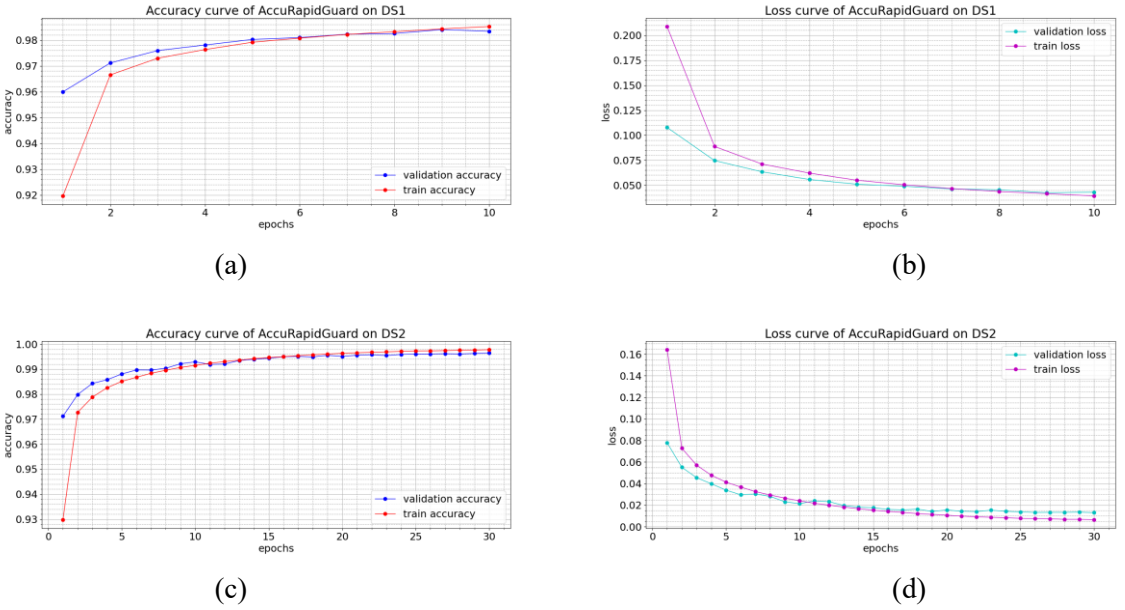


Figure 3. Accuracy and loss curves of AccuRapidGuard

Table 2. Performences on DS1

Model	Test accuracy	Precision	Recall	F1-score	Test loss
AccuRapidGuard	0.983	0.980	**0.987**	0.984	**0.044**
PDRCNN	0.981	0.979	0.983	0.981	0.050
CNN-Fusion	**0.985**	0.984	0.986	**0.985**	0.061
BiGRU	0.982	**0.989**	0.976	0.983	0.046

Table 3. Performances on DS2

Model	Test accuracy	Precision	Recall	F1-score	Test loss
AccuRapidGuard	0.997	0.997	**0.996**	0.997	**0.011**
PDRCNN	**0.998**	**0.999**	0.996	**0.998**	0.016
CNN-Fusion	0.995	0.996	0.994	0.995	0.030
BiGRU	0.997	0.997	0.996	0.997	0.013

To depict the comprehensive performance of these models and assess the efficiency of AccuRapidGuard, we employ the following evaluation metrics using the third-party module scikit-learn: accuracy, precision, recall, F1-score, loss, training time, test time, and inference speed. These metrics provide a holistic view of the model's performance, offering intuitive insights into its classification ability, generalization capability, training effectiveness, and real-world performance.

Accuracy, precision, recall, F1-score, inference speed can be calculated as follows:

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

DOI: 10.54254/2755-2721/73/20240376

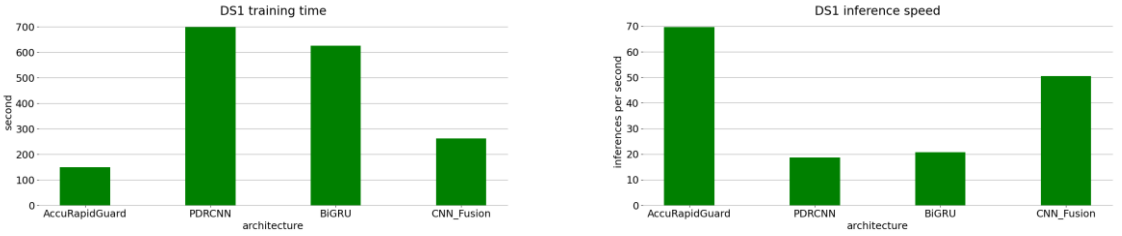
$$F1 = 2 \times \frac{TP \times FP}{TP + FP} \tag{4}$$

$$Inference\ Speed = \frac{M}{T}(IPS) \tag{5}$$

Where TP, TN, FP, FN represent true positives, ture negatives, false positives, false negatives respectively, and M represents the number of URLs in test set while T represent test time.

Table 4. Efficiency on DS1

Model	Training time(s)	Training time radio to AccuRapidGuard	Test time(s)	Inference speed(IPS)	Inference speed radio to AccuRapidGuard
AccuRapidGuard	158.521	1.000	0.810	69.683	1.000
PDRCNN	700.768	4.420	3.029	18.636	0.267
CNN-Fusion	262.082	1.653	1.118	50.481	0.724
BiGRU	626.432	3.952	2.726	20.703	0.297



(a) (b)

Figure 4. Traning time and inference speed on DS1

3.4. Results

In this section, we will provide an explanation of the experimental results obtained from our approach, assessing its performance and efficiency, and comparing it against other baseline models.

a) ***Performances***.

The accuracy and loss learning curves of AccuRapidGuard on datasets DS1 and DS2 are illustrated in Fig.3. Upon observing the figures, it becomes evident that the training and validation losses converge to a stable point, exhibiting minimal generalization gap. This convergence signifies a favorable fit of the models. Detailed evaluation metrics for the performance of AccuRapidGuard and the baseline models are summarized in Table 3 and Table 4.

Analyzing the test results on dataset DS1, it is apparent from Table 3 that AccuRapidGuard delivers exceptional performance in recall and loss, outperforming the baselines and achieving values of 0.987 and 0.044 respectively. Accuracy, F1-score, and precision also showcase commendable results, with an accuracy of 0.983, a F1-score of 0.984 (slightly lower than CNN-Fusion), and a precision of 0.98.

Similarly, examining the test results on dataset DS2, Table 4 reveals that AccuRapidGuard maintains its leading position in recall and loss, attaining high value of 0.996 and low value of 0.011 respectively. Furthermore, AccuRapidGuard secures the second-best performance in accuracy, precision, and F1 score, with all three metrics reaching 0.997, with negligible differences to the highest values(slightly lower than PDRCNN). This indicates that even on a large-scale dataset, AccuRapidGuard accurately predicts the labels of test samples and demonstrates robust generalization to unseen data.

Based on these experimental results, it is conclusive that AccuRapidGuard exhibits strong performance and notable generalization capabilities. It surpasses state-of-the-art techniques in terms of recall and loss, while achieving accuracy, precision, and F1 scores exceeding 0.98, reaching the level of state-of-the-art methods. AccuRapidGuard satisfactorily meets the requirements of phishing detection.

DOI: 10.54254/2755-2721/73/20240376

b) Efficiency.

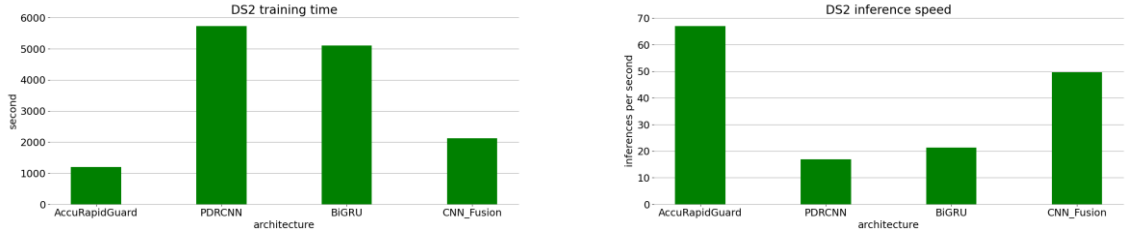
Based on datasets DS1 and DS2, we conducted an analysis and evaluation of the efficiency of AccuRapidGuard, considering factors such as training time, test time, and inference speed. We calculated the ratios of training time and inference speed to AccuRapidGuard in order to obtain a quantitative assessment of the proposed model compared to the baselines. The specific information regarding the efficiency of AccuRapidGuard and the baseline models is summarized in Table 5 and Table 6, respectively, based on DS1 and DS2. Moreover, we have depicted the details of the efficiency evaluation metrics in Figures 5 and 6, providing an intuitive representation of the superior efficiency of AccuRapidGuard compared to the other baseline models.

From Table 5, it can be observed that although the performance of these four models is nearly identical on DS1, the baseline models require significantly more training time to achieve the same level of accuracy, precision, recall, and F1-score as AccuRapidGuard. The training times of PDRCNN, CNN-Fusion, and BiGRU are approximately 4.4 times, 1.6 times, and 4 times longer than that of AccuRapidGuard, respectively. This further emphasizes the exceptional efficiency of AccuRapidGuard in utilizing computational resources. Additionally, in terms of testing, the inference times of PDRCNN, CNN-Fusion, and BiGRU are 0.267, 0.724, and 0.297 times that of AccuRapidGuard, respectively, highlighting the high availability and strong generalization of AccuRapidGuard to unseen data.

Furthermore, the ratios of training time and inference speed between AccuRapidGuard and the other baseline models remain relatively constant. Since the structure and ratios of training sets, validation sets, and test sets in DS1 and DS2 are similar, the efficiency of model training and testing is not strongly influenced by the specific characteristics of the data itself.

Table 5. Efficiency on DS2

Model	Training time(s)	Training time radio to AccuRapidGuard	Test time(s)	Inference speed(IPS)	Inference speed radio to AccuRapidGuard
AccuRapidGuard	1351.654	1.000	2.326	67.028	1.000
PDRCNN	5752.747	4.256	9.218	16.916	0.252
CNN-Fusion	2118.321	1.567	3.138	49.696	0.741
BiGRU	5106.883	3.778	7.335	21.260	0.317



(a) (b)

Figure 5. Traning time and inference speed on DS2

4. Conclusion

In conclusion, AccuRapidGuard presents a deep learning-based model for phishing website detection that achieves outstanding performance and efficiency. The model not only demonstrates comparable or superior accuracy to other state-of-the-art models but also exhibits exceptional efficiency, with reduced training time and significantly improved inference speed. This combination of high performance and efficiency makes AccuRapidGuard highly valuable in real-world applications.

By efficiently extracting crucial local and contextual features from suspicious website URLs using parallel CNN layers and an RNN layer with attention mechanism, AccuRapidGuard ensures accurate classification within limited runtime. Its ability to deliver accurate results while requiring less training

DOI: 10.54254/2755-2721/73/20240376

time sets it apart from existing models. Additionally, the model's improved inference speed enables rapid and efficient detection of phishing websites, making it highly suitable for time-sensitive environments.

The remarkable efficiency of AccuRapidGuard, coupled with its impressive performance, makes it a compelling solution for phishing website detection. Its practical value lies in its ability to achieve comparable or better accuracy than state-of-the-art models while reducing training time and improving inference speed. This breakthrough in efficiency enhances the model's usability and effectiveness, contributing to enhanced user security and minimizing potential risks.

In summary, AccuRapidGuard represents a significant advancement in the field of phishing website detection. Its combination of superior performance, demonstrated by its accuracy, and outstanding efficiency, as evidenced by reduced training time and improved inference speed, positions it as a highly valuable and practical solution. AccuRapidGuard has the potential to make a substantial impact in the cybersecurity domain and holds promise for widespread adoption and deployment in various real-world scenarios.

References

[1] G. Aaron, “Phishing activity trends report, 4nd quarter 2022.”[Online].Available:https://docs.apwg.org/reports/apwg_trends_report_q4_2022.pdf

[2] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences,* vol. 35, no. 2, pp. 590-611, 2023, doi: 10.1016/j.jksuci.2023.01.004.

[3] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555,* 2014.

[4] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259,* 2014.

[5] L. Tang and Q. H. Mahmoud, "A survey of machine learning-based solutions for phishing website detection," *Machine Learning and Knowledge Extraction,* vol. 3, no. 3, pp. 672-694, 2021.

[6] C. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, "Applications of deep learning for phishing detection: a systematic literature review," *Knowledge and Information Systems,* vol. 64, no. 6, pp. 1457-1500, 2022.

[7] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," *arXiv preprint arXiv:1606.01781,* 2016.

[8] T. Feng and C. Yue, "Visualizing and Interpreting RNN Models in URL-based Phishing Detection," presented at the Proceedings of the 25th ACM Symposium on Access Control Models and Technologies, 2020.

[9] W. Wang, F. Zhang, X. Luo, and S. Zhang, "PDRCNN: Precise Phishing Detection with Recurrent Convolutional Neural Networks," *Security and Communication Networks,* vol. 2019, pp. 1-15, 2019, doi: 10.1155/2019/2595794.

[10] M. Hussain, C. Cheng, R. Xu, and M. Afzal, "CNN-Fusion: An effective and lightweight phishing detection method based on multi-variant ConvNet," *Information Sciences,* vol. 631, pp. 328-345, 2023, doi: 10.1016/j.ins.2023.02.039.