Securing Personal Data on Smartphones: Addressing APT Challenges and Phishing Threats

[1]Ali Hodroj
[1]Makram W. Hatoum, https://orcid.org/0000-0002-4729-3547
[1]Khouloud Samrouth
[1]Ali El Attar

[1]*Cybersecurity and Forensics Department, Arab Open University Beirut, Lebanon*

Corresponding Author: *Ali Hodroj (email: arh013lb@aou.edu.lb)

Abstract— **The widespread integration of smartphones into daily life has revolutionized communication, work, and information access, but it has also made them prime targets for cybercriminals. One significant threat comes from Advanced Persistent Threats (APTs), which involve sophisticated, prolonged cyber intrusions. A common attack vector for APTs is phishing, where victims are deceived into clicking on malicious URLs delivered through SMS, email, WhatsApp, and phone calls. These URLs lead users to cloned websites that look like authentic platforms, deceiving them into disclosing sensitive information including login passwords and personal details. This study examines smartphone security vulnerabilities, with a focus on URL phishing attacks. Our research is organized into two major areas. First, we create a new dataset and use a rigorous feature extraction approach. Second, we present a robust mitigation strategy based on deep learning techniques. Our methodology uses three deep learning models: Convolutional Neural Network (CNN), Deep Neural Network (DNN), and Long Short-Term Memory (LSTM)—each assessed for its effectiveness. The findings underscore the importance of a well-curated dataset and careful feature selection in achieving high performance. The DNN model demonstrated the highest accuracy, the CNN excelled in true positive rate, and the LSTM provided balanced performance. Compared to traditional methods, these deep learning models significantly enhance the detection of phishing attacks, highlighting the crucial role of high-quality datasets in improving model accuracy and robustness.**

Keywords— APT; Deep learning; Privacy; Security; Smartphone; URL phishing attacks.

I. INTRODUCTION

The fast incorporation of smartphones into every aspect of our lives has transformed how we interact, work, and obtain information [1]. However, this widespread adoption has also raised critical concerns regarding the security and privacy of the vast amounts of personal data stored on these devices [2]. In the current era of information technology, the intersection of privacy, technology, and personal data on smartphones has become a focal point of significant consideration. The rapid advancement in communication technologies has transformed smartphones into intelligent, efficient, and indispensable tools for daily tasks, making them integral to our lives. However, this growth has concurrently increased the prevalence of security risks and threats to networked devices and their data [3]-[6]. One significant threat is posed by APTs, characterized by strategic and prolonged cyber intrusions orchestrated by sophisticated entities [7].

APTs unfold in three key phases: information gathering, social engineering, and malware deployment. It also introduces a novel perspective on smartphone vulnerabilities by focusing on application-based attacks. This attack model incorporates both attack modeling and detection methods to address a newly identified vulnerability arising from the execution of apps on smartphones. Specifically, the attack model centers on an end-user-vulnerable application that serves as the initiator of the attack. This insecure program is quietly deployed in the background, remaining hidden from the actual user's view while getting into sensitive data [8]. The primary objective of APTs extends beyond merely infecting devices; it aims to gain access to particular targets for purposes such as cyber espionage, data theft, or sabotage [9]. Muhammad *et al.* delve into various aspects of cybersecurity, encompassing APTs, malware classification, sensor exploitation, side-channel attack visualization, and an exhaustive list of families of malware prevalent on the Google Store [10].

Xiang *et al.* [11] discusses APTs and introduces a specific type that differ from traditional APTs by targeting mobile devices in addition to personal computers (PCs). Experiments conducted using three months of DNS records from a university campus network show that the proposed strategy improves the identification percentage of cross-platform APTs by more than 15%.

Attackers frequently employ phishing techniques to exploit users by sending deceptive emails, texts, or instant messages that look to come from reputable sources. These communications frequently include harmful links or files. When users click on these links or provide personal information, attackers gain unauthorized access to their mobile devices or steal credentials, enabling identity spoofing.

zulkefli *et al.* [12] investigated the vulnerability of Smartphones to APT attacks, emphasizing the role of spear phishing in compromising security. The paper introduced a machine learning-based detection system, that achieved good accuracy in identifying phishing URLs. They proposed a method for preventing spear phishing attacks by examining the similarity between URLs in browser history and received messages.

Phishing attacks aim to deceive unsuspecting victims through various communication channels, including WhatsApp, email, SMS, other communication tools, and cellphone calls. These deceptive messages are often disguised so they seem to come from familiar and reliable sources, like friends or colleagues, with the ultimate goal of extracting valuable information from the targeted individuals [13]. The objective is to prompt victims to click on URLs that lead to cloned web portals mimicking legitimate platforms like company intranets, banks, and popular social media platforms including Instagram, Facebook, Gmail, Twitter, and Yahoo. When victims try to sign in or input data on these deceptive websites, they inadvertently provide the attacker with sensitive details, including user IDs, emails, passwords, addresses, mobile numbers, dates of birth, and payment card information.

Various techniques have been proposed to enhance URL detection and mitigate phishing threats.

Naqvi *et al.* [14] gave an in-depth examination of current phishing attack mitigation techniques. Their paper analyzed 248 articles, spanning from early 2018 to March 2023, to summarize the existing landscape of anti-phishing measures. The review highlighted the severe consequences of successful phishing attacks, including financial losses, reputational damage, and identity theft, emphasizing the critical need to address this pervasive cyber threat. By identifying gaps and open issues in current anti-phishing methodologies, the paper significantly contributed to the ongoing discourse on enhancing cybersecurity measures against phishing attacks.

Asiri *et al.* [15] filled a significant gap in existing research by doing a thorough review of smart detection methods for HTML link phishing attempts. Their research focused on the limitations of prior surveys, emphasizing the importance of conducting a thorough investigation of deep learning algorithms in the detection of phishing. The work makes an important contribution by providing a thorough evaluation for phishing attack recognition, including data preparation, extraction of features, model building, and efficiency measurements. This extensive analysis is a great resource for practitioners as well as researchers, helping them understand and design more successful anti-phishing measures.

Shaukat *et al.* [16] contributed to the advancement of cybersecurity measures by using machine learning approaches to improve phishing website identification and classification. Their research focused on utilizing diverse features to ensure robust protection for internet users against phishing attacks. The study highlighted the effectiveness of the XGBoost algorithm, which outperformed other models in the testing phase, achieving a maximum accuracy and precision of 91%. This level of precision in categorizing phishing websites demonstrates machine learning algorithms' effectiveness in improving defenses.

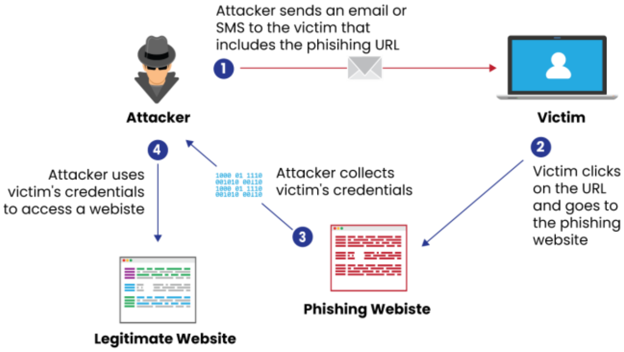
Attacker sends an email or SMS to the victim that includes the phisihing URL
1
Attacker
Victim
4
Attacker uses victim's credentials to access a webiste
Attacker collects victim's credentials
2
Victim clicks on the URL and goes to the phishing website
3
Phishing Webiste
Legitimate Website

Fig. 1. URL Phishing attack.

Lakshmi *et al.* [17] introduced an innovative method for detecting phishing websites using deep learning techniques enhanced by the Adam optimizer. To efficiently identify fraudulent websites, their suggested model uses a set of features with several parameters. The study revealed that traditional machine learning methods are significantly outperformed by the deep learning model, in accurately classifying phishing websites. This advancement emphasizes the enhancement of cybersecurity measures against phishing threats, leveraging the capabilities of deep learning frameworks.

Korkmaz *et al.* [18] underscored the escalating threat of phishing attacks within the broader landscape of cyber threats. It highlighted the evolution of phishing methods, emphasizing the financial consequences and the increasing number of phishing sites. The proposed system focused on URL analysis to improve detection capabilities using machine learning frameworks. The main advantage is its capability to tackle zero-day attacks and eliminate dependencies on third-party services or blacklist updates.

A recurrent neural network (RNN) technique is used to detect phishing URLs [19]. The researchers tested their strategy on a dataset of thousands of harmful and real sites, attaining high accuracy in a short period.

Zhang *et al.* [20] developed PhishTrim, an efficient method for detecting phishing URLs that relies on deep feature learning. Their investigations showed that PhishTrim has a high ability to identify new phishing attacks and operates very well on huge datasets.

Sahingoz *et al.* [21] developed an immediate phishing detection solution that utilizes various classification algorithms in conjunction with features derived from natural language processing (NLP). They categorized the features into two groups: word vectors, that analyze word usage in URLs lacking additional operations, and NLP-based features, that are mainly determined by humans. Although the results were satisfactory regarding detection rates, the researchers suggested that incorporating modern learning technologies could further enhance system efficiency.

Aldakheel *et al.* suggested a precise classification strategy with a CNN model to differentiate legitimate websites from phishing sites effectively [22]. Their findings underscored the effectiveness of the CNN model in enhancing classification accuracy, demonstrating its potential to significantly improve detection rates.

Sadique *et al.* developed a framework for detecting phishing URLs in real-time by employing online learning to tackle the limitless expansion of URL space [23]. Additionally, they incorporated delayed feature collection and selective sampling, which greatly enhanced the performance of the system.

Our methodology represents a concerted effort to harness the transformative power of deep learning in fortifying cybersecurity defenses against phishing attacks.

The structure of this paper is as follows: Section II outlines the research methodology, Section III discusses the experiments and analysis, Section IV provides a comparison with current studies and theories, and Section V concludes the paper while addressing potential future work.

II. RESEARCH METHODOLOGY

In this part, we delve into the complexities of our phishing detection strategy, emphasizing the critical significance of datasets and deep learning approaches in strengthening cybersecurity measures. The robustness of our methodology hinges on three key components: the new meticulously curated dataset, the judicious selection of features, and the implementation and comparison of performance of three distinct deep learning models—DNN, CNN, and LSTM.

Central to our methodology is the comprehensive dataset meticulously curated for training and evaluation purposes. We provide insights into the composition and characteristics of the dataset, shedding light on its pivotal role in facilitating the training and validation of the deep learning models. Furthermore, we elucidate the rationale behind the features selection tailored to identify the subtle nuances indicative of phishing attempts. The implementation section elucidates the architecture and intricacies of the DNN, CNN, and LSTM models deployed in our study. Each model is meticulously designed to exploit the unique strengths of its underlying architecture, thereby enhancing its efficacy in detecting phishing attempts. Through detailed exposition, we offer a glimpse into the inner workings of these models, encompassing layers, activation functions, and optimization techniques. A cornerstone of our methodology is the comparative analysis of the three deep learning models, wherein we scrutinize their performance across a spectrum of evaluation metrics. By juxtaposing the results obtained from each model, we endeavor to identify the most effective approach to phishing detection. This evaluative analysis not only clarifies each model's strengths and weaknesses but also provides useful insights for future research.

A. *Dataset Generation*

The dataset is the most important phase in deep learning. Several datasets are used for phishing attack detection, but an up-to-date dataset will greatly enhance the accuracy of the deep learning model. Selecting precise and pertinent values for a dataset stands as a pivotal and fundamental task. Hence, we meticulously curated our dataset from reliable sources. This involved a meticulous two-step process:

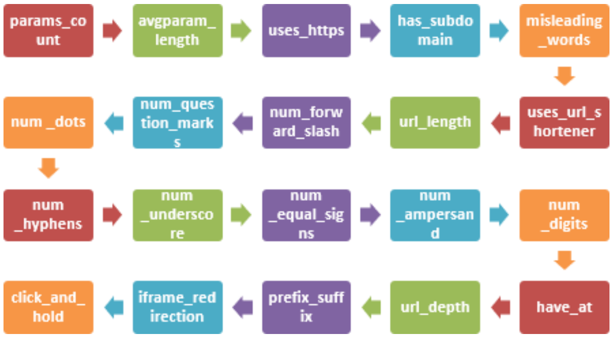

Fig. 2. Features Extracted.

Step 1: Initially, we assembled the foundational dataset. This compilation comprised 5,000 phishing URLs sourced from the reputable platform Phishtank.org [24], alongside an equivalent count of 5,000 legitimate websites drawn from a fresh dataset available on Kaggle [25]. At this stage, the dataset was composed of solely two columns: URL and Label.

Step 2: Subsequently, we embarked on feature extraction. Leveraging Python, we imported the foundational dataset and meticulously generated 20 distinctive features for each website. This comprehensive process culminated in the creation of an enriched dataset, poised for further analysis and application.

The constructed dataset consists of URLs along with specific features extracted from each URL. Each row in the dataset corresponds to a single URL, with the resources extracted for that URL listed in the respective columns. The dataset also includes a label that indicates whether the URL is considered legitimate or phishing. This extensive set of features provides a robust foundation for training models aimed at detecting phishing URLs.

By ensuring that our dataset is both current and meticulously curated, we can enhance the accuracy and efficiency of our deep learning models. The enriched dataset, with its detailed feature set, serves as a critical component in the creation of efficient phishing detection systems.

1) Features Descriptions:

In this section, we provide detailed descriptions of each feature employed in our dataset, presented in Fig. 2. Understanding these features is essential to understanding the composition of the dataset and the factors considered in our analysis.

1) **Params_count:** This feature represents the number of parameters in the URL query string. Parameters are key-value pairs separated by "&" in a URL after the "?" symbol.
2) **Avgparam_length:** This feature calculates the average length of parameters in the URL query string. It is determined by dividing the overall length of all parameters by the number of parameters.
3) **Uses_https:** This feature indicates whether the URL uses the HTTPS protocol for secure communication. A value of 1 indicates HTTPS is used, while 0 indicates it is not.

4) **Has_subdomain:** This feature indicates whether the URL contains a subdomain. A subdomain is a prefix to the domain name, such as “subdomain.example.com”.
5) **Misleading_words:** This feature checks the URL for misleading words that may mislead users into believing the site is legitimate when it is not.
6) **Uses_url_shortener:** This feature indicates whether the URL uses a URL shortener service, which is often used to disguise the real destination of the URL.
7) **URL_length:** This feature indicates the length of the URL.
8) **Num_forward_slash:** It records the number of forward slashes ("/") in the URL, which can be used to determine the depth of the path.
9) **Num_question_marks:** It counts the number of question marks (“?”), which indicate the beginning of the query string.
10) **Num_dots:** This feature tallies the number of dots (“.”), which is used to separate domain levels (e.g., example.com).
11) **Num_hyphens:** This feature checks the number of hyphens (“-”) in the URL, which is used in domain names and paths.
12) **Num_underscore**: It records the number of underscores (“_”) in the URL, which is sometimes used in domain names and paths.
13) **Num_equal_signs:** This feature measures the total of equal signs (“=”) in the URL, which is used in query string parameters.
14) **Num_ampersand:** This feature records the ampersands (“&”), which is used to separate multiple query string parameters.
15) **Num_digits:** This feature records the numerical digits in the URL, which can be an indicator of phishing URLs.
16) **Have_at:** This feature indicates whether the URL contains the “@” symbol. It is typically used in email addresses.
17) **URL_depth:** This feature represents the depth of the URL path, calculated as the number of directories in the path.
18) **Prefix_suffix: This** feature checks for the presence of prefix or suffix characters in the URL, such as “-” or “_”, which are sometimes used to mimic legitimate URLs.
19) **Iframe_redirection:** This feature indicates whether the URL uses iframe redirection, which is a technique used to load content from another source into a webpage.
20) **Click_and_hold:** This feature checks for the presence of tapping and holding events in the URL, which can be used to deceive users by displaying a different URL than the actual destination.

An “Extract Features (URL)” function is developed to retrieve the mentioned features from a given URL to facilitate phishing detection in a very accurate method. It begins by initializing an empty dictionary called to store all extracted features. The function then parses the URL to count the number of parameters in the query string and calculates the average length of those parameters. It checks whether the URL uses HTTPS and contains a subdomain, scans for misleading words often associated with phishing identifies the use of URL shortening services, and calculates the total URL length.
The function counts occurrences of specific characters such as '=', '-', '_', and '&', and checks for the presence of the '@' symbol. It calculates the URL path depth, checks for prefix or suffix characters, and examines whether the URL contains iframe redirection and click-and-hold events. Finally, the function returns the extracted features as a dictionary, providing a comprehensive dataset for phishing URL detection.

B. *Mitigation Strategy*

After building the data from trusted resources, it’s time to implement the method to obtain high accuracy and other metrics, thus confirming the success of the models. The implementation of the method was carried out on Google Colab [26]. We’ll delve into the implementation and comparison of three distinct deep learning models: DNN, CNN, and LSTM. The dataset in all models is split into 70% training data and 30% testing data.

DNN Implementation: The DNN model preprocesses the dataset by eliminating the ‘URL’ column, shuffling the data, and standardizing the features. It consists of an input layer with 20 neurons (corresponding to the number of features), a hidden layer with 64 neurons using ReLU activation, and a dropout layer with a 0.2 rate to mitigate overfitting. For binary classification, the output layer contains two neurons, utilizing a sigmoid activation function, as illustrated in Fig. 3. The Adam optimizer and Sparse Categorical Cross Entropy loss are used to compile the model, which is trained for 10 epochs.

CNN Implementation: The CNN model includes two convolutional layers, each succeeded by max-pooling layers, which assist the model in learning spatial hierarchies of features. The two convolutional layers comprise 64 filters, but the first one with a kernel size of 10, whereas the second one with a kernel size of 5.

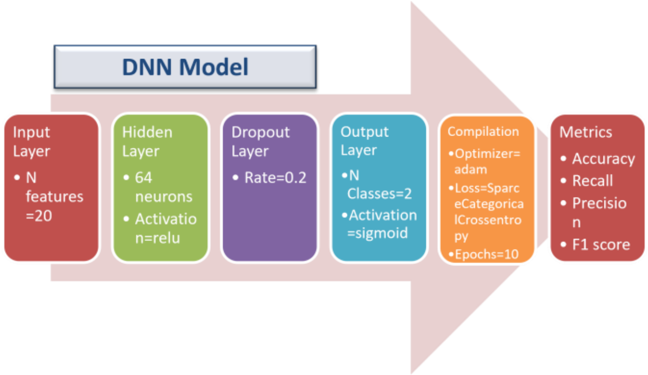

Fig. 3. DNN-Model.

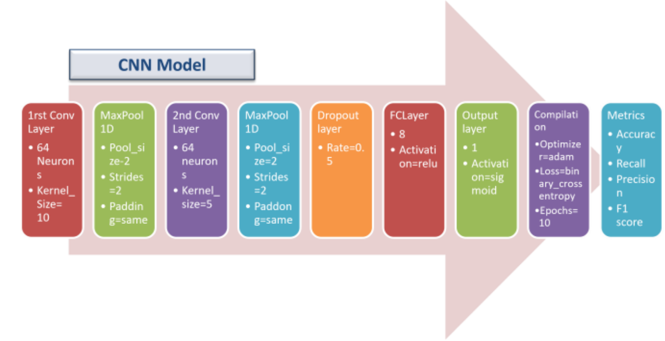

Fig. 4. CNN-Model.

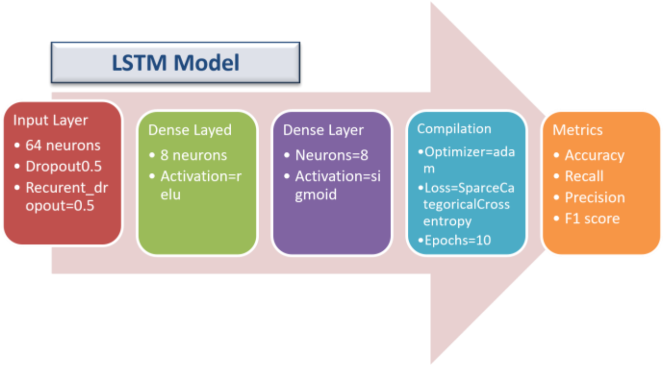

Fig. 5. LSTM-Model.

Max-pooling layers followed the convolutional layers, each with a pool size of 2 and strides of 2, which reduce the spatial dimensions of features. To avoid overfitting, an abandonment layer with a dropout rate of 0.5 is introduced after the second convolutional layer, as illustrated in Figure 4. The dropout layer's output is flattened into an array with one dimension before being transferred to a dense layer with 8 neurons and a ReLU activation function. The exit layer contains one neuron that calculates if a URL is phishing or authentic.

LSTM Implementation: The LSTM model is a type of recurrent neural network (RNN) appropriate for sequencing data, such as time series or text data. The input data is reshaped to include a time step dimension of 1 as LSTM expects input data to be in the form of [samples, time steps, characteristics].
The LSTM layer has 64 units, which allows learning complex patterns in the input data, as shown in Fig. 5.
To prevent overfitting, a 0.5-dropout layer is inserted after the LSTM layer. The output of the dropout layer is routed to a dense layer with eight neurons. The output layer contains one neuron with a sigmoid activation function, as in other models.

All models are evaluated based on precision, accuracy, F1 score and recall using confusion matrices.

III. EXPERIMENTS AND ANALYSIS

In this section, we look at the experiments performed to assess the effectiveness of three different deep-learning models. We provide a full analysis of the outcomes obtained from these experiments, highlighting the strengths and weaknesses of each model in classifying phishing and legitimate URLs. Through this analysis, our goal is to understand the effectiveness of these deep learning architectures for this particular task.
To ensure robust training and mitigate overfitting, we employed a validation set throughout the training process. This allowed us to continuously evaluate the model's performance on unseen data and make adjustments as needed to avoid overfitting. Following training, we evaluated the performance of each model—DNN, CNN, and LSTM—on a separate testing dataset. In all the experiments, we adjusted the number of neurons in each model from 16 to 128 to conduct internal comparisons.

Starting with the **DNN model**, it delivered outstanding results, showing low loss and high accuracy on the training and validation sets.
After completing the training and validation phases, we evaluated the model on the test set, where it achieved high accuracy and low loss, indicating strong generalization to unseen data. The accuracy improved as the number of neurons in the model increased, reaching a peak value of 99.6% with 64 neurons, as shown in Table I. However, increasing the number of neurons to 128 did not result in any significant improvement, as the accuracy remained roughly constant.
Thus, after extensive testing, we determined that 64 neurons provided the optimal balance, consistently delivering the highest accuracy across the model. This finding suggests that 64 neurons are ideal for our specific application and dataset, offering the best performance without unnecessary complexity.
The results suggest that the DNN model is proficient at classifying phishing and legitimate URLs and is likely to perform well in real-world scenarios. The performance evaluation of the DNN model revealed promising outcomes, as shown in Table I.
It attained the highest accuracy, signifying its proficiency in classifying URLs accurately. Notably, the precision of 99.73% underscores the model’s remarkable precision in identifying phishing URLs. Moreover, a recall of 99.47% suggests the model’s capability to accurately identify actual phishing URLs, capturing 99.47% of all such URLs present in the dataset.
The model attained an outstanding F1 score of 99.6%, reflecting its ability to effectively balance precision and recall, thus minimizing both false positives and false negatives.

Table I. Result of Metrics in DNN

Metrics					Confusion Matrix			
Neurons	Accuracy	Precision	Recall	F1 Score	TP	TN	FP	FN
16	99.3	99.06	99.52	99.29	1480	1499	14	7
32	99.4	99.73	99.14	99.43	1509	1474	4	13
64	99.6	99.73	99.47	99.6	1512	1476	4	8
128	99.47	99.93	98.99	99.46	1475	1509	1	15

Table II. Result of Metrics in CNN

Metrics					Confusion Matrix			
Neurons	Accuracy	Precision	Recall	F1 Score	TP	TN	FP	FN
16	99.17	99.53	99.82	99.17	1509	1466	7	18
32	99.33	99.19	99.46	99.32	1478	1502	12	8
64	99.73	99.66	99.8	99.73	1510	1482	5	3
128	99.23	99.87	99.6	99.32	1541	1438	8	13

Table III. Result of Metrics in LSTM

Metrics					Confusion Matrix			
Neurons	Accuracy	Precision	Recall	F1 Score	TP	TN	FP	FN
16	99.55	99.8	99.31	99.55	1008	983	2	7
32	99.44	99.89	99.07	99.43	967	1022	2	9
64	99.55	99.9	99.22	99.56	1026	965	1	8
128	99.4	99.5	99.3	99.4	1003	985	5	7

The **CNN model** also demonstrates strong performance, exhibiting high accuracy and minimal loss across both training and validation datasets. Following training and validation, the model is evaluated on the test set, where it sustains its exceptional performance, indicative of its robust generalization capability to unseen data. Increasing the number of neurons in the model led to an improvement in accuracy, reaching a peak value of 99.73% with 64 neurons, as shown in Table II. These findings underscore the effectiveness of the CNN model in feature extraction from URLs, enabling accurate discrimination between phishing and legitimate URLs.

The CNN model also delivered exceptional performance, boasting high precision and recall metrics, as shown in Table II. Although its accuracy slightly trails that of the DNN model, it remains impressively high, indicative of robust overall performance. Notably, the CNN model attained a remarkable recall of 99.8%, underscoring its efficacy in accurately identifying the majority of genuine phishing URLs.

The **LSTM model** demonstrates exceptional performance, characterized by high accuracy and minimal loss across both training dataset and validation dataset.

Then, the model undergoes evaluation on the test dataset, where it maintains its high performance, indicative of its robust generalization capability to new URL sequences. Increasing the number of neurons in the model led to an improvement in accuracy, reaching a peak value of 99.55% with 64 neurons, as shown in Table III. These results underscore the effectiveness of the LSTM model in capturing the sequential nature inherent in URLs. They suggest its suitability for classifying phishing and legitimate URLs based on their sequential patterns, thereby highlighting its potential for real-world application in cybersecurity tasks. The LSTM model exhibited outstanding performance, boasting high precision, recall, and accuracy metrics, as shown in Table III. It successfully achieved a harmonious balance between precision and recall, with both metrics surpassing 99%. The consistent performance of the LSTM model aligns with that of the other models, underscoring its effectiveness in detecting phishing URLs.

Overall, all three models demonstrated excellent performance, each showcasing its unique strengths. The CNN model delivered the highest accuracy, F1 score, and recall, while the LSTM model demonstrated the highest precision. These findings collectively emphasize the usefulness of deep learning models in identifying phishing URLs, affirming their potential for enhancing cybersecurity measures. The metrics, and confusion metrics results are presented in Fig. 6 and 7.

A. Dataset Analysis

Reducing the number of features in the dataset led to a decrease in accuracy, highlighting the importance of a rich feature set. Reducing features likely removed critical characteristics that distinguish phishing from legitimate URLs, hindered the model's ability to classify URLs correctly and limited its capacity to recognize complex relationships and patterns.

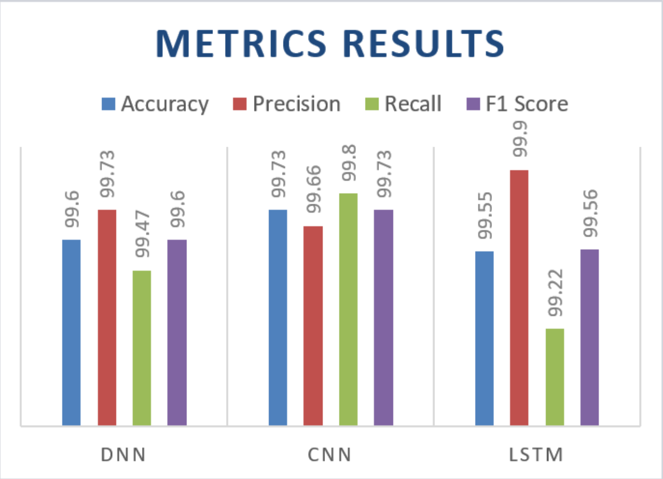

Fig. 6. Metric Results.

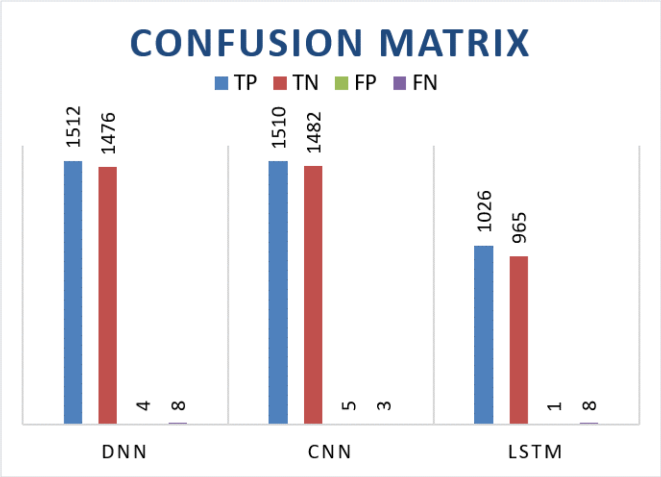

Fig. 7. Confusion Matrix Results.

Table IV. Datasets Comparison

Model	Accuracy using Kaggle Dataset	Accuracy using our Dataset
DNN	94.24	99.6
CNN	93.53	99.73
LSTM	93.26	99.55

This limitation leads to poorer performance, especially in challenging cases where subtle differences are crucial.

The success of the presented deep learning models, with an accuracy of 99.73%, is largely attributable to the precise and pertinent dataset we created from scratch. The balanced nature of the data together with the detailed feature extraction process provided a robust foundation for our model to learn effectively. Reducing the number of features demonstrated the critical role that each feature plays in maintaining high accuracy. Therefore, maintaining a rich and comprehensive feature set is essential for achieving optimal performance in phishing detection of URL.

To confirm the importance of the dataset and its impact on achieving the desired results in any model, we tested a dataset from Kaggle [27] using 64 neurons in the input layers. This dataset comprises over 11,000 website URLs, each with 30 parameters and a class label that categorizes it as either a phishing website or not. The experimental results are presented in Table IV, showing the differences in accuracy between the datasets. The fact that all three models performed excellently using our dataset underscores not only their effectiveness but also the dataset quality utilised for training, validation, and testing. A high-quality dataset ensures that the models are exposed to a variety of scenarios, allowing them to learn robust patterns and make accurate predictions in real-world situations. Therefore, investing time and effort in collecting and preprocessing datasets is crucial for achieving reliable and high-performing deep learning models.

IV. COMPARISON WITH EXISTING STUDIES AND THEORIES

Compared with existing studies in the domain, we noticed that the presented “deep learning” models outperform other approaches, as shown in Table V. The presented models in our study achieved higher accuracies of 99.6%, 99.73%, and 99.55%, respectively for DNN, CNN, and LSTM, demonstrating the superiority of deep learning in detecting phishing URLs on smartphones. The high mark of accuracy, recall, precision, and F1 score of those models demonstrate their effectiveness in accurately detecting phishing URLs, thus increasing the security and protection of private and personal data on smartphones. The performance of our study emphasizes the promise of deep learning to address security challenges on mobile devices.

Table V. Studies Comparison

Ref	Approach	Accuracy	F1_Score
[19]Dutta *et al.*	RNN	97.4 %	96.4 %
[20]Zhang *et al.*	CNN	98.34 %	98.3 %
[21]Sahingoz *et al.*	Random Forest	97.98 %	98 %
[22]Aldakheel *et al.*	CNN	98.77 %	not listed
[23]Sadique *et al.*	Random Forest	87 %	not listed
Our Study	DNN	99.6%	99.73%
	CNN	99.73%	99.66
	LSTM	99.55%	99.9%

We can conclude that dataset quality is vital in influencing accuracy and overall performance of machine learning models, including deep learning models. The well-curated and representative dataset contributes significantly to training models effectively and allows them to generalize effectively to new data. Additionally, the dataset should be free from biases and anomalies that could skew the model’s learning process.

The fact that all three models performed excellently underscores not only their effectiveness but also the dataset quality used for training, validation, and testing. A high-quality dataset ensures that the models are exposed to a variety of scenarios, allowing them to learn robust patterns and make accurate predictions in real-world situations. Therefore, investing time and effort in collecting and preprocessing datasets is crucial for achieving reliable and high-performing machine learning models.

V. CONCLUSION

The proliferation of smartphones has significantly increased concerns about data privacy and security, particularly in relation to Advanced Persistent Threats (APTs) and URL phishing attacks. While existing studies have utilized deep learning and machine learning techniques to detect phishing, there is a clear need for advanced deep learning approaches tailored to smartphones, leveraging comprehensive and high-quality datasets.

Our research is centered on two main objectives: the creation of an up-to-date dataset enriched with pertinent features and a preventive framework using deep learning algorithms that adapt to evolving phishing patterns. By incorporating proactive measures, such as real-time URL authenticity assessment, we aim to halt phishing attempts before they pose a risk to users.

The research illustrates the promise of deep learning in detecting phishing URLs on smartphones. By meticulously curating datasets and extracting relevant features, we have developed robust models that exhibit high accuracy, precision, recall, and F1 scores.

Despite these promising results, several challenges remain in safeguarding personal data on smartphones. The dynamic nature of APTs and phishing techniques necessitates continuous adaptation of security measures. Additionally, future research should address the scalability of our approach to handle larger datasets and real-time detection more effectively. Use of PCA (Principal Component Analysis) to decrease the number of features. Integrating user behavior analysis and anomaly detection techniques could further enhance the accuracy and robustness of our models, providing a comprehensive solution to the evolving threats in mobile security. Additionally, potential user experience improvements should be considered. Implementing features such as warning messages or automatic blocking of detected phishing URLs can significantly enhance user safety and trust. Such features would not only alert users to potential threats but also prevent them from inadvertently accessing malicious sites.

By addressing these additional aspects, the proposed phishing detection system can offer a more comprehensive solution that not only accurately identifies phishing URLs but also integrates seamlessly into everyday mobile usage scenarios, thereby providing users with real-time, reliable protection against phishing attacks.

VI. ACKNOWLEDGEMENTS

We want to express our sincere appreciation to everyone who contributed to the success of this article. We acknowledge the Arab Open University (AOU) for providing a conducive academic environment and essential resources. In particular, we are thankful for the support from the Faculty of Computer Studies and the Cybersecurity and Forensics Department, without which this research would not have been possible.

REFERENCES

[1] Majid Hatamian and Jetzabel Serna-Olvera. Beacon alarming: Informed decision-making supporter and privacy risk analyser in smartphone applications. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, pages 468–471. IEEE, 2017.

[2] Alireza Naghizadeh, Behrooz Razeghi, Ehsan Meamari, Majid Hatamian, and Reza Ebrahimi Atani. C-trust: A trust managementsystem to improve fairness on circular p2p networks. *Peer-to-Peer Networking and Applications*, 9:1128–1144, 2016.

[3] Shikah J Alsunaidi and Abdullah M Almuhaideb. Security methods against potential physical attacks on smartphones. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–6. IEEE, 2019.

[4] W. Rayes, K. Samrouth, N. Bakir. Using Blockchain and Vazka Authentication for the Security of Smart Home Devices, In *Arab ICT Conference on Digital Transformation for Sustainable Infrastructure*, Bahrain, 2024.

[5] Z. Fneish, M. El Hajj, K. Samrouth. Survey on IoT Multi-Factor Authentication Protocols: A Systematic Literature Review, *International Symposium on Digital Security and Forensics*, Tennesse, USA, 2023.

[6] A. Tawil, K. Samrouth. IEWS: a Free Open Source Intelligent Early Warning System Based on Machine Learning, *International Symposium on Digital Security and Forensics*, Tennesse, USA, 2023

[7] Ping Chen, Lieven Desmet, and Christophe Huygens. A study on advanced persistent threats. In *Communications and Multimedia Security: 15th IFIP TC 6/TC 11 International Conference, CMS 2014, Aveiro, Portugal, September 25-26, 2014. Proceedings 15*, pages 63–72. Springer, 2014.

[8] Misbah Shafi, Rakesh Kumar Jha, and Sanjeev Jain. Behavioral model for live detection of apps based attack. *IEEE Transactions on Computational Social Systems*, 2022.

[9] Sean D Kaster and Prescott C Ensign. Privatized espionage: Nso group technologies and its pegasus spyware. *Thunderbird International Business Review*, 65(3):355–364, 2023.

[10] Zia Muhammad, Zahid Anwar, Abdul Rehman Javed, Bilal Saleem, Sidra Abbas, and Thippa Reddy Gadekallu. Smartphone security and privacy: A survey on apts, sensor-based attacks, side-channel attacks, google play attacks, and defenses. *Technologies*, 11(3):76, 2023.

[11] Zongyuan Xiang, Dong Guo, and Qiang Li. Detecting mobile advanced persistent threats based on large-scale dns logs. *Computers & Security*, 96:101933, 2020.

[12] Zakiah Zulkefli, Manmeet Mahinderjit Singh, Azizul Rahman Mohd Shariff, and Azman Samsudin. Typosquat cyber crime attack detection via smartphone. *Procedia Computer Science*, 124:664–671, 2017.

[13] Akashdeep Bhardwaj, Varun Sapra, Aman Kumar, Naman Kumar, and S Arthi. Why is phishing still successful? *Computer Fraud & Security*, 2020(9):15–19, 2020.

[14] Naqvi, Bilal, Kseniia Perova, Ali Farooq, Imran Makhdoom, Shola Oyedeji, and Jari Porras. Mitigation strategies against the phishing attacks: A systematic literature review. *Computers & Security*, 2023.

[15] Asiri, Sultan, Yang Xiao, Saleh Alzahrani, Shuhui Li, and Tieshan Li. A survey of intelligent detection designs of HTML URL phishing attacks. *IEEE Access* 11: 6421-6443, 2023.

[16] Shaukat, Muhammad Waqas, Rashid Amin, Muhana Magboul Ali Muslam, Asma Hassan Alshehri, and Jiang Xie. A hybrid approach for alluring ads phishing attack detection using machine learning. *Sensors* 23(19): 8070, 2023

[17] Lakshmi, L., M. Purushotham Reddy, Chukka Santhaiah, and U. Janardhan Reddy. "Smart phishing detection in web pages using supervised deep learning classification and optimization technique ADAM." Wireless Personal Communications 118(4): 3549-3564, 2021

[18] Mehmet Korkmaz, Ozgur Koray Sahingoz, and Banu Diri. Detection of phishing websites by using machine learning-based url analysis. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2020.

[19] Ashit Kumar Dutta. Detecting phishing websites using machine learning technique. *PloS one*, 16(10):e0258361, 2021.

[20] Lei Zhang and Peng Zhang. Phishtrim: Fast and adaptive phishing detection based on deep representation learning. In *2020 IEEE International Conference on Web Services (ICWS)*, pages 176–180. IEEE, 2020.

[21] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, and Banu Diri. Machine learning based phishing detection from urls. *Expert Systems with Applications*, 117:345–357, 2019.

[22] Eman Abdullah Aldakheel, Mohammed Zakariah, Ghada Abdalaziz Gashgari, Fahdah A Almarshad, and Abdullah IA Alzahrani. A deep learning-based innovative technique for phishing detection in modern security with uniform resource locators. *Sensors*, 23(9):4403, 2023.

[23] Farhan Sadique, Raghav Kaul, Shahriar Badsha, and Shamik Sengupta. An automated framework for real-time phishing url detection. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0335–0341. IEEE, 2020.

[24] PhishTank. [Online]. Available: https://www.phishtank.org

[25] Kaggle. [Online]. Available: https://www.kaggle.com/datasets

[26] Ekaba Bisong and Ekaba Bisong. Google colaboratory. *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners*, pages 59–64, 2019.

[27] ESWAR Chand. Phishing website detector. https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector, 2020.