DOI: http://dx.doi.org/10.26483/ijarcs.v17i1.7407

Volume 17, No. 1, January-February 2026

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

DEVELOPING A PREDICTIVE MODEL FOR PHISHING WEBSITE DETECTION USING APACHE SPARK: A SURVEY

Faisal Abdullah Althobaiti
Department of Information Technology
King Abdulaziz University,
Jeddah, Saudi Arabia
ORCID - https://orcid.org/0009-0003-4517-1524

***Abstract*:** The CB-based URL classifier - as used in this study - stands as evidence of its superior performance, outperforming both the Random Forest and Logistics Regression. The CatBoost algorithm exhibited a high level of competence in discerning the nuances of phishing URLs, thereby elevating the bar for detection accuracy. This model's effectiveness extends beyond the traditional approaches and offers users a real-time shield against phishing websites, fostering a more secure network experience. Also, we were able to address the limitations of sci-kit learn, thereby ushering in improvements in terms of model training efficiency; also, leveraging Apache Spark in combination with Sk-dist paved the way for a more streamlined, responsive, and scalable phishing detection mechanism. This study not only contributes an innovative phishing URL detection model but also underscores the ongoing evolution in the cybersecurity landscape. As the digital realm continues to develop, the symbiosis between advanced machine learning algorithms and powerful frameworks like Apache Spark becomes pivotal in ensuring the resilience of our networks against ever-evolving threats. Through continuous refinement and exploration, the path toward a more secure online ecosystem unfolds, driven by the commitment to stay one step ahead in the ceaseless cat-and-mouse game of cybersecurity.

Keywords: Python, Machine learning, Apache Spark, CatBoost, Random Forest, Logistic Regression, Phishing detection.

I. INTRODUCTION

In our rapidly evolving world and the consistent developments in the digital realm, the insidious rise of cyber-attacks has become one of great concern, with Phishing attacks being one of the major forms of such attacks. These attacks not only compromise the security infrastructure underpinning online interactions but also underscore the urgent need for adaptive and robust detection systems. Characterized by a cunning fusion of social engineering and sophisticated technological deception, phishing exploits vulnerabilities to extract sensitive user information.

However, traditional defense strategies, particularly those relying on static blacklists, find themselves outpaced by the dynamic nature of contemporary phishing campaigns. The emergence of novel malicious URLs at an alarming rate presents a significant challenge to timely and effective identification [1]. Recognizing this vulnerability, researchers are turning towards machine learning (ML) methodologies as a beacon of hope, offering the promise of heightened accuracy and responsiveness in the face of evolving cyber threats.

This study, however, aims to address the shortcomings of current defense strategies against phishing attacks by leveraging the accuracy and adaptive nature of machine learning (ML) methodologies. By exploring the potential of ML in enhancing accuracy and responsiveness, the goal is to develop a predictive model that can adapt to the ever-changing tactics of cybercriminals.

To achieve this, the study delves into the intricacies of phishing detection, focusing on the utilization of Apache Spark and the sk-dist Python package to streamline processes, improve efficiency, and contribute to a more secure online environment.

In the subsequent sections, this paper will provide an in-depth exploration of the background of phishing attacks, highlighting their evolving nature and the challenges posed by conventional defense mechanisms. To further elucidate the problem statement, the need for a paradigm shift in detection strategies needs to be emphasized.

The study's primary aim, centered around harnessing the potential of ML, Apache Spark, and sk-dist, will be discussed in detail. Finally, the paper will conclude by presenting insights into the experimental setup, methodology, and anticipated contributions to the field of cybersecurity.

A. Core Concepts

1) Phishing Attack: Phishing attacks represent a complex and pervasive form of cyber threats or social engineering, targeting unsuspecting users through deceptive tactics to reveal sensitive information. A total of $245.7 million (USD) was lost by Singaporeans to scammers through various online scams such as e-commerce scams, job scams, and phishing scams [2]. The majority of these attacks comprise email and URL-based deceptions to more sophisticated social engineering techniques. The goal is consistent: to trick users into divulging touchy data such as usernames, passwords, and financial credentials. The increasing sophistication of phishing attacks, including the rapid generation of novel malicious URLs, poses a significant challenge to timely and effective identification.

2) Machine Learning: Researchers recognized the limitations of traditional defense strategies and found a need to turn to machine learning (ML) methodologies as a more promising solution, as ML offers the potential for heightened accuracy and responsiveness in the face of evolving cyber threats. By leveraging the power of ML algorithms, it becomes possible to analyze and identify patterns in large

datasets, providing a more adaptive and proactive approach to phishing detection [3]. However, constructing robust classification models remains a time-intensive process, especially when dealing with an extensive set of features crucial for capturing the nuanced characteristics of phishing attempts.

Machine Learning algorithms can be used to evolve mathematical models and are often categorized as [4]:

1) *Supervised Learning:* Supervised Learning algorithms use input and output training sets to train a model. After the training data is processed, it builds a function that maps new data on expected output values.
2) *Unsupervised Learning:* On the other hand, learning of unsupervised algorithms is another ML paradigm that uses unlabeled data to predict the output. It classifies the training data based on similar features.

3) Apache Spark: Addressing the bottleneck in constructing classification models is at the core of our research, and to overcome this challenge, we turn to the formidable capabilities of Apache Spark. As a powerful big data processing framework, Apache Spark offers the potential to streamline the feature selection process and optimize model-building times. By harnessing the distributed computing power inherent in Apache Spark, our aim is to develop a predictive model that not only amplifies the efficiency of phishing detection but also achieves unparalleled scalability. In the subsequent sections, we delve into an in depth exploration of our methodology, navigating through the intricacies of dataset collection, feature engineering, and the reasonable selection of machine learning models. Our intent is to contribute substantively to the evolving landscape of intelligent cybersecurity solutions, demonstrating the synergy between machine learning paradigms and the robust capabilities of Apache Spark [5].

4) Sk-dist: In the pursuit of harnessing the capabilities of distributed machine learning, this research embraces the utilization of the Sk-dist Python package. Sk-dist is an innovative framework built upon the foundations of scikit-learn and seamlessly integrated with Apache Spark (PySpark). This section elucidates the pivotal role of Sk-dist in our experimental setup and its contributions to the augmentation of machine learning workflows. Sk-dist extends scikit-learn's inherent parallelization capabilities using joblib to the distributed computing realm through PySpark. Essentially, it acts as a bridge between scikit-learn and PySpark, enabling the parallelization of meta-estimator training for enhanced scalability [6].

5) Spark MLlib: Developed natively for Spark, MLlib empowers users with scalable and distributed machine learning capabilities, providing a comprehensive suite of tools for diverse applications. Its seamless integration with Spark ensures efficient utilization of distributed computing resources, making it a reliable choice for large-scale machine learning tasks.

In this research work, we observed that Sk-dist, leveraging the foundations of scikit-learn and PySpark, introduces an additional layer of efficiency, particularly evident in meta-estimator training, making it faster than Spark MLlib.

The framework supports a range of machine learning tasks in a distributed setting [7].

II. REVIEW OF RELATED WORKS

The persistent threat of phishing attacks has prompted cybersecurity researchers to explore innovative technologies to enhance detection mechanisms. We seek to critically examine existing research efforts focused on developing predictive models for phishing website detection, emphasizing the integration of Apache Spark as a powerful distributed computing framework.

Traditional methods of phishing detection are challenged by the evolving nature of phishing attacks. The literature highlights the inadequacies of existing solutions and the need for more advanced, scalable approaches to combat phishing threats.

Machine learning (ML) algorithm integration in detecting phishing websites is a major focus in modern research, and studies emphasize the importance of analyzing features such as URLs, IP addresses, and other indicators to improve the accuracy of detection models as phishers can hide the URL and use tools like TinyUrl to make the URL appear valid [8].

Filtering these phishing emails before the users read them will help reduce the percentage of users being defrauded [9].

Moradpoor et al. [10] employed two datasets comprising 14,370 emails (benign/phishing) for their phishing email detection and classification model based on neural networks. The overall accuracies and inaccuracies of their model reached 92.2%. In a similar vein, Smadi et al. [54] introduced a model for phishing email detection, extracting 23 features. Through the comparison of various algorithms, they found that the random forest algorithm exhibited the highest accuracy, reaching 98.8%.

Patil et al. [11] introduced a method for detecting and preventing phishing websites through a machine learning approach. Initially, the URL undergoes a comparison using the Blacklist and Whitelist Approach. If the URL is present in either the Blacklist or Whitelist, it is flagged as a phishing website. Conversely, if the URL is not found in either list, the features of the URL are extracted using a Heuristic and Visual Similarity Approach. Subsequently, the researchers employ machine learning algorithms such as LR, DT, and RF to analyze the various features of URLs and webpages. The system underscores the efficiency achieved by integrating heuristic features, visual features, and a combination of blacklist/whitelist approaches with machine learning techniques. The reported results indicate that LR and DT attain an accuracy of 96.23%, while RF achieves a slightly higher accuracy of 96.58%.

Owing to the pandemic, most government and corporate activities, instructional sports, organizations, and noncommercial activities have shifted online. People are increasingly using the internet to do their every-day jobs. As a consequence, having a comprehensive phishing attack detection device with higher accuracy and reaction time has become extra crucial.

Therefore, in this work, we applied machine learning techniques to capture inherent characteristics of the email text and other features to be classified as phishing or non-phishing according to the selected datasets.

Lim Chian Fang et al. [12] analyzed and assessed the CatBoost, Random Forest, and Logistics Regression

classifiers' performances. The results yielded that CatBoost was a significantly more highly classifier than Random Forest and Logistic Regression, with up to 96% detection accuracy. Regression classifiers' performances were carried out, and Apache Spark was used to generate data and analyze it with hybridization of algorithms to enhance accuracy. Quang et al. [13] implemented deep learning techniques like DNN, CNN, LSTM, and GRU. In this research, results show that no single DL algorithm accomplished the best measures across all performance metrics and suggested future research areas related to deep learning in the phishing detection field [14].

Mughaid et al. [15] proposed to give a complete vision of what Machine learning entails and what tricks phishers use on their users with different phishing attack techniques.

Al-Ahmadi et al. [16] proposed a phishing detection model called PDGAN that relies only on a website's uniform resource locator (URL) to achieve reliable effectiveness. A long short-term memory network (LSTM) network as a generator of synthetic phishing URLs and a convolutional neural network (CNN) as a discriminator to identify whether the URLs are phishing or legitimate.

Kolla et al. [17] compared the predictive accuracy of several machine learning methods including, Decision tree, Random forest, Multilayer Perceptions, Support Vector Machines, and XGBoost for predicting phishing URLs.

Piñeiro et al. [18] proposed a web architecture based on three machine learning models to predict whether a web address has phishing or not based mainly on Random Forest, Classification Trees, and Support Vector Machine.

III. RESEARCH MODEL

A. Framework

This research focuses on the development of an advanced predictive model for the detection of phishing websites while leveraging the robust capabilities of Apache Spark to process large batches of data. The framework consists of a systematic approach that encompasses dataset collection, feature engineering, and the utilization of machine learning algorithms within the Apache Spark environment.

Apache Spark plays a pivotal role in our research, which serves as an open-source cluster computing platform designed for handling large-scale data processing. Unlike other used for processing big data, such as Hadoop and Storm, Apache Spark adopts an innovative multi-stage in-memory processing approach, which results in processing speeds that are exceptionally faster - up to 100 times - compared to traditional map-reduce processing methods. It is distinguished by its compatibility with multiple programming languages, including Java, Scala, and Python.

With a user-friendly API and shells in Python, Scala, Java, and SQL, Spark facilitates job management and query creation. In addition, Spark exhibits versatility by seamlessly running Hadoop clusters and accessing a wide array of Hadoop data sources.

At its core, Apache Spark consists of essential functionalities such as task scheduling, memory management, fault recovery, and efficient interaction with storage systems. It is also known for its implementation of Resilient Distributed Datasets (RDDs) as the primary programming abstraction. RDDs serve as a mechanism for representing distributed sets of data across multiple computing nodes, enabling efficient parallel processing of data.

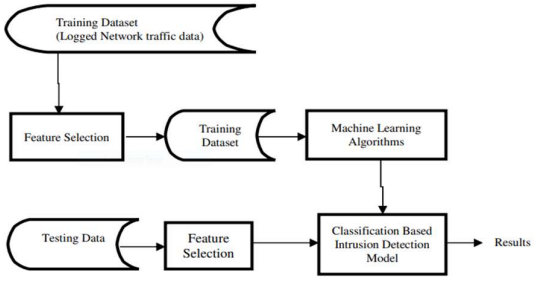

Fig. 1. A Framework for Rapid and Effective Detection of Cybersecurity Intrusions [19].

Machine Learning Algorithms

1) Logistic Regression (LR)

Logistic Regression (LR) serves as a foundational classification algorithm for categorizing data into a binary set of classes. The LR output, represented by the probability value, undergoes transformation through the logistic sigmoid function, ensuring the conversion of real values into a range between 0 and 1. The logistic sigmoid function is expressed as:

$$S(x) = 1 \div (1 + e^{-x})$$

In this expression, S(x) denotes the output of the logistic sigmoid function, and e signifies the mathematical constant approximately equal to 2.71828. The sigmoid function's characteristic S-shaped curve ensures the mapping of input values to a probabilistic range, making Logistic Regression particularly suitable for binary classification tasks [20].

2) Random Forest (RF)

Random Forest (RF) operates on the principle of aggregating outputs from a multitude of randomized decision trees crafted in the training phase. The Gini Index serves as the criterion for node branching in decision trees during the classification process. The mathematical expression for the Gini Index is articulated as follows:

$$\text{Gini Index} = 1 - \Sigma_{i=1}^{c} (Pi)^2$$

Here, c represents the number of classes in the classification task, and Pi denotes the probability of belonging to the i-th class [21]. The Gini Index provides a measure of impurity or disorder within a set of data, guiding the construction of decision trees in the Random Forest ensemble. A lower Gini Index signifies a more homogeneous and pure dataset, aligning with the objective of enhancing the predictive accuracy of the Random Forest model.

3) CatBoost (CB)

CatBoost (CB) distinguishes itself through the implementation of the Ordered Target Statistic (OTS) and Order Boosting (OB) techniques, showcasing its prowess in handling datasets rich in categorical data. By incorporating random permutations of training examples, CB adeptly mitigates prediction shifts caused by target leakage common in conventional gradient boosting algorithms. The fundamental building blocks of CB are binary decision trees, contributing to its robust predictive capabilities. The algorithm's estimated output equation encapsulates its innovative approach to handling categorical features, making it a valuable asset in scenarios where such data intricacies

play a pivotal role. This strategic use of OTS and OB reinforces CB's position as an advanced and adaptable model, particularly well-suited for applications demanding precise handling of categorical data nuances. This aligns with the objective of enhancing the predictive accuracy of the Random Forest model [22].

The equation for Estimated Output $= Z = H(x_i) = \Sigma_{j=1}^{J} C_j 1_{\{x \epsilon R_i\}}$

B. Classification Process

The classification process involves utilizing the most accurate machine learning algorithm to construct a URL Classifier. The URL features are extracted and categorized into four main parts:

TABLE I. DATASET FEATURES

Type	No	Feature	Name	Description	Value
Address bar-based	1	IP address	UsingIP	Having IP address in URL	−1, 1
	2	URL length	LongURL	Long URL to hide the suspicious part	−1, 0, 1
	3	Shortening service	ShortURL	Using URL shortening services “TinyURL”	−1, 1
	4	@ Symbol	Symbol@	URL’s having @ symbol	−1, 1
	5	“//” redirecting	Redirecting//	Having “//” within URL path for directing	−1, 1
	6	Prefix suffix	PrefixSuffix	Adding prefix or suffix separated by (-) to the domain	−1, 1
	7	Sub domain	SubDomains	Sub domain and multi sub domain	−1, 0, 1
	8	SSL final state	HTTPS	Existence of HTTPS and validity of the certificate	−1, 0, 1
	9	Domain registration	DomainRegLen	Expiry date of domains/Domain registration length	−1, 1
	10	Favicon	Favicon	Favicon loaded from a domain	−1, 1
	11	Port	NonStdPort	Using non-standard port	−1, 1
	12	HTTPS token	HTTPSDomainURL	The existence of HTTPS token in the domain part of URL	-1, 1
	13	Request URL	RequestURL	Request URL within a webpage/Abnormal request	−1, 1
	14	URL of anchor	AnchorURL	URL within tag/Abnormal anchor	−1, 0, 1
	15	Links in tags	LinksInScriptTags	Links in <meta>, <script>, and <link> tags	−1, 0, 1
Abnormal-based	16	SFH	ServerFormHandler	Server Form Handler	−1, 0, 1
	17	Email	InfoEmail	Submitting information to E-mail	−1, 1
	18	Abnormal URL	AbnormalURL	Host name is included in the URL/Whois	−1, 1
	19	Redirecting	WebsiteForwarding	Number of times a website has been redirected	0, 1
HTML and JavaScript-based	20	On mouseover	StatusBarCust	On mouse over changes status bar/Status bar customization	−1, 1
	21	Right click	DisableRightClick	Disabling right click	−1, 1
	22	Pop-up window	UsingPopupWindow	Using Pop-up window	−1, 1
	23	Iframe redirection	IframeRedirection	Using Iframe	−1, 1
	24	Age of domain	AgeofDomain	Minimum age of a legitimate domain is 6 months	−1, 1
	25	DNS record	DNSRecording	Existence of DNS record for the domain	−1, 1
	26	Website traffic	WebsiteTraffic	Being among top 100,000 in Alexa rank	−1, 0, 1
Domain-based	27	Page rank	PageRank	Having a page rank greater than 0.2	−1, 1
	28	Google index	GoogleIndex	Website indexed by Google	−1, 1
	29	Link reference	LinksPoitingTo Page	Number of links pointing to a page	−1, 0, 1
	30	Statistical report	StatsReport	Top 10 domain and top 10 Ips from PhishTank	−1, 1
		Result	class	Phishing or legitimate	−1, 1

To comprehensively assess the performance of the machine learning classification models, a set of well-established evaluation metrics based on the confusion matrix was employed. The confusion matrix is particularly effective in scenarios where the classification output comprises more than two classes. The matrix encompasses four key values contributing to the calculation of accuracy, precision, recall, and F1 score [23].

Within the context of the confusion matrix, four key components play distinct roles in the evaluation process. True Positive (TP) instances signify the accurate prediction of positive values, while False Positive (FP) occurrences denote situations where negative values are predicted incorrectly as positive. Conversely, False Negative (FN) instances represent the inaccurate prediction of positive values as negative, and True Negative (TN) cases indicate accurate predictions of negative values [24].

These components collectively contribute to the formulation of crucial evaluation metrics, each shedding light on different facets of the model's performance.

C. Evaluation Metrics Formulas

To comprehensively assess the performance of the machine learning classification models, a set of well-established evaluation metrics based on the confusion matrix was employed. The confusion matrix is particularly effective in scenarios where the classification output comprises more than two classes. The matrix encompasses four key values contributing to the calculation of accuracy, precision, Recall, and F1 score.

1) *Accuracy:* as a holistic measure, is derived from the ratio of correctly predicted data to the total dataset size. The formula for accuracy is expressed as (TP + TN) / (TP + FP + TN + FN).
2) *Precision:* a metric focusing on the percentage of actual positive classes relative to the total predicted positive classes, is calculated using the formula Precision = TP / (TP + FP)
3) *Recall:* also known as sensitivity, signifies the percentage of correctly predicted positive values concerning the actual count of positive data in the dataset. The recall formula is articulated as Recall = TP / (TP + FN).
4) *F1-Score:* recognized as the harmonic mean of precision and recall, offers a balanced assessment of the model's performance. The formula for F1-Score is expressed as F1-Score = 2 / ((1 / Recall) + (1 / Precision)) [25].

By employing these evaluation metrics, a nuanced and comprehensive understanding of the classification model's precision, recall, and overall predictive accuracy is achieved. These metrics collectively contribute to a robust assessment, enabling insights into the model's proficiency in distinguishing between legitimate and phishing websites [26].

D. Comparative Analysis

In the quest to identify the most adept model for phishing URL detection, an extensive and meticulous comparative analysis unfolded, focusing on three prominent machine learning classifiers: LR (Logistic Regression), RF (Random Forest), and CB (CatBoost). These classifiers, renowned for their distinct methodologies, were instrumental in both the training and testing phases, and the outcomes of this rigorous experimentation were subjected to a comprehensive examination, utilizing key evaluation metrics: accuracy, precision, recall, and F1 score.

TABLE II. COMPARATIVE ANALYSIS BETWEEN THREE ML ALGORITHM

Classifiers	Evaluation Parameter						
	AC	PC		RC		F1-Score	
		PURL	LURL	PURL	LURL	PURL	LURL
LR	93.22%	92%	94%	92%	94%	92%	94%
RF	96.43%	97%	96%	95%	97%	96%	97%
CB	97.87%	98%	98%	97%	99%	98%	98%

AC = Accuracy, CB = CatBoost, PURL = Phishing URL, LURL = Legitimate URL, LR = Logistics Regression, RF = Random Forest, RC = Recall.

E. Experimental Procedure

This research undertook an analysis of a sizable dataset encompassing a total of 11,055 URLs. The experimental framework employed three distinct classifiers, namely Logistic Regression, Random Forest, and CatBoost. To ensure a robust evaluation, the dataset underwent a systematic division into both training and testing sets, fostering an environment that simulated real-world scenarios.

The multi-faceted nature of this experiment facilitated an intricate exploration of the classifiers' capabilities, unravelling their nuanced responses to the dynamic landscape of phishing URL detection. By subjecting the classifiers to a diverse array of URLs, the experiment aimed to capture the intricacies of their adaptability and efficacy in discerning between phishing and legitimate URLs.

In the crucible of assessment, the experiment homed in on a carefully curated subset of 2,211 URLs, subjecting them to rigorous testing protocols to predict outcomes. The resulting insights, encapsulated in Table 2, stand as a repository of valuable information distilled from the exhaustive experimentation. Among the classifiers, CatBoost emerged as a standout performer, showcasing unparalleled prowess in discrimination between phishing and legitimate URLs [27].

This multi-faceted experiment not only expands our understanding of classifier performance but also underscores the significance of a meticulous experimental setup. The nuanced analysis conducted within this study contributes to the evolving landscape of phishing URL detection, offering insights that can inform the development of more effective and adaptive cybersecurity solutions.

F. Accuracy Dominance

CB rose to prominence by showcasing the highest accuracy score, an impressive 97.87%, a feat that surpassed both RF and LR. This underscores CB's proficiency in rendering accurate distinctions between phishing and legitimate URLs.

1) Precision, Recall, and F1-Score Excellence:

The ascendancy of CB extended beyond accuracy, with the classifier excelling in precision, recall, and F1-score for both phishing and legitimate URLs. With an average score of 98%, CB consistently outperformed RF and LR across every performance indicator [45].

2) Classifier Selection Rationale: Elevating CB as the Model of Choice

The decision to designate CB as the ultimate model for URL classification was not arbitrary. Rather, it was a deliberate choice rooted in its consistent and superior performance across accuracy, precision, recall, and F1-score metrics. CB's unparalleled proficiency in these dimensions solidifies its status as the most robust and effective classifier for phishing URL detection within the specific context of the UCI repository dataset.

G. Implications and Future Considerations

This study delves into not only the quantitative aspects of accuracy but also the qualitative dimensions of precision, recall, and overall F1-score. These nuanced insights hold profound implications for the selection of a classifier that not only excels in one facet but demonstrates prowess across the spectrum of evaluation metrics. As we chart the path forward, considerations for the dynamic nature of cybersecurity threats and the ever-evolving landscape of phishing attacks become imperative. Future refinements and adaptations of the chosen classifier, in this case, CB, will be crucial for maintaining efficacy and relevance in an ever-changing digital terrain. In essence, this comparative analysis serves not only as a snapshot of current classifier performance but also as a guidepost for future enhancements and advancements in the realm of phishing URL detection [28].

Going beyond a surface-level assessment of accuracy, this exploration aims to unravel the intricate nuances inherent in each model's behavior, providing a more holistic understanding of their respective strengths and weaknesses. The comparative analysis serves as a rigorous examination of three distinct models—Logistic Regression (LR), Random Forest (RF), and CatBoost (CB).

In dissecting the performance of these models, it becomes apparent that accuracy alone is an insufficient metric for a comprehensive evaluation by scrutinizing various dimensions, including precision, recall, and F1 score, and offering a nuanced perspective on the models' capabilities to not only identify phishing URLs but also to discern their specific characteristics [29].

The Logistic Regression model, while demonstrating a certain level of accuracy, reveals vulnerabilities in the face of distinct classes and outliers. The inherent sensitivity of this model to outliers and deviations from the norm highlights its limitations [30], potentially leading to inaccuracies when faced with unconventional patterns in phishing attempts.

On the other hand, the Random Forest model, known for its proficiency in handling distinct classes [31], encounters challenges in terms of interpretability and preference for features with more values. The potential for incorrect predictions arises, especially when dealing with categorical features with varying levels of cardinality.

In contrast, the CatBoost model emerges as a frontrunner, showcasing remarkable performance across various metrics. Its ability to streamline hyper-parameter adjustment [32], reduce the risk of overfitting, and evaluate and select essential features positions it as a robust choice for phishing URL detection. Moreover, CatBoost's innovative technique for analyzing categorical features adds an extra layer of sophistication, enhancing accuracy in scenarios where other models may falter [33].

Beyond a mere comparison of models, this analysis contributes to the collective understanding of their behavior in complex cybersecurity scenarios, and by acknowledging and comprehending the subtle intricacies of each model, this research paves the way for more informed decision-making in the selection and deployment of models tailored to the evolving challenges of phishing URL detection.

1) Logistic Regression (LR): Unraveling Overfitting and Sensitivity Challenges

Logistic Regression (LR) reveals its vulnerability in comparison to Random Forest (RF) and CatBoost (CB), primarily due to its susceptibility to overfitting [27]. The model's inclination to overemphasize accuracy during training poses a significant hurdle in generalizing effectively to unseen data, a pivotal factor for robust model performance. This characteristic makes LR less adept at handling diverse and dynamic patterns present in phishing URL datasets.

Moreover, LR's sensitivity to outliers exacerbates its challenges, leading to computational instability when confronted with distinct class characteristics [28]. The logistic functions inherent in LR struggle to optimally navigate scenarios with pronounced differences among classes, potentially resulting in inaccuracies in predictions. This underlines the model's limitations in capturing the nuanced features of phishing attacks, particularly those characterized by unconventional patterns and behaviors.

In the pursuit of a sophisticated and adaptable phishing URL detection model, LR's shortcomings spotlight the need for alternative algorithms like RF and CB. These alternatives, with their distinctive strengths, offer more robust solutions for discerning phishing activities in the complex and ever-evolving landscape of cybersecurity threats.

2) Random Forest (RF): Robust but Limited in Readability

Random Forest (RF) emerges as a robust performer, showcasing commendable strength in scenarios marked by distinct classes [34]. The model strategically employs effective tree-pruning methods, addressing computational challenges and enhancing efficiency. However, this robustness comes with a trade-off, particularly when compared to CatBoost (CB).

One notable limitation of RF lies in its readability. The model, while proficient in handling distinct class characteristics, struggles with presenting a clear and interpretable structure. This challenge arises from RF's approach of grouping decision trees, which, while beneficial for computations, may compromise the model's ability to discern the importance of individual features [35].

In instances where categorical features exhibit varying cardinalities, RF's tendency to neglect the significance of each feature becomes apparent. This oversight has the potential to lead to suboptimal predictions, especially when confronted with diverse datasets. The consequence of this limitation is the compromise in the model's interpretability, hindering the ability to extract meaningful insights from its decision-making process [36]. As interpretability is crucial in understanding the factors contributing to model predictions, the readability constraints of RF underscore the need for alternative approaches, such as CatBoost, which excels in balancing robustness with model interpretability.

3) CatBoost (CB): Superiority in Default Parameterization and Feature Importance

CatBoost (CB) stands as a frontrunner in the comparative analysis, propelled by the inherent strengths that set it apart in the area of phishing URL detection. A key element of CB's ascendancy lies in its adept handling of hyperparameters, obviating the need for extensive adjustments [37]. The model's default parameters exhibit commendable performance, reducing the complexity associated with hyperparameter tuning.

CB's prowess in minimizing the risk of overfitting is a standout feature, contributing to the development of flexible and accurate models. Its innovative approach to determining tree structure, particularly the calculation of leaf values, plays a pivotal role in mitigating concerns related to overfitting. This intrinsic quality ensures that CB models maintain adaptability and effectiveness across diverse datasets.

The model's sophistication extends to feature evaluation and selection, a critical aspect in the context of categorical attributes. CB introduces the Prediction Values Change method, offering a nuanced strategy for discerning feature importance. This approach surpasses traditional methods like one-hot encoding, providing a more refined and effective means of handling categorical data.

In essence, CB's ascendancy in the comparative analysis rests on a foundation of versatility, adaptability, and innovative strategies that collectively contribute to its superior performance in the complex landscape of phishing URL detection. As a frontrunner, CB not only excels in accuracy but also showcases a commitment to addressing inherent challenges in feature selection and model interpretability [38].

4) Computational Efficiency: CB's Trade-Off in Time Consumption

In the realm of phishing URL detection, the comparative analysis of Logistic Regression (LR), Random Forest (RF), and CatBoost (CB) transcends conventional metrics, offering a nuanced exploration of model behavior and computational efficiency [39]. While CB emerges as a frontrunner with superior predictive accuracy and feature importance, it introduces a trade-off in terms of computational efficiency.

CB's exceptional predictive capabilities and meticulous feature analysis contribute to its status as the most time-consuming model, requiring 54.86 seconds [40]. This extended training and testing duration can be attributed to CB's default setting of constructing 1000 trees, reflecting its commitment to thorough and nuanced analysis [41]. In contrast, RF's efficiency stems from its ability to operate on a subset of variables, making it the fastest to train. This presents a trade-off, highlighting the delicate balance between the depth of analysis and computational efficiency [42].

However, this comprehensive comparative analysis extends beyond traditional metrics like accuracy, precision, and recall. It emphasizes the need for a nuanced understanding of each model's behavior and the implications for practical applications. While CB excels in predictive power and feature importance, the optimal choice depends on the specific requirements and constraints of the intended use [43].

The decision-making process involves striking a harmonious balance between predictive accuracy, interpretability, and computational resources. The selection of LR, RF, or CB is not a one-size-fits-all scenario; instead, it's a strategic choice informed by a thorough grasp of the trade-offs inherent in each model's architecture and functionality.

As the cybersecurity landscape continues to evolve, the insights gleaned from this analysis contribute to a more informed and strategic approach to phishing URL detection. The study emphasizes the need for a tailored model selection process, aligning with the unique demands of diverse applications in the dynamic realm of cybersecurity. In conclusion, the comparative analysis of LR, RF, and CB serves as a guidepost for navigating the intricate landscape of phishing URL detection, fostering a paradigm where model selection aligns seamlessly with the multifaceted challenges posed by evolving cyber threats.

IV. DISCUSSION

Phishing scam poses a great challenge to users and can affect the growth and development of tech business entities as they will raise scepticism among users and, by extension, it's will affect economic development in countries where such occurrences are prevalent.

In this study, we took into consideration other research works that implemented several other methodologies to detect phishing scams.

Below is Table III highlighting the methodologies, strengths, and limitations of other research works in this field.

TABLE III. COMPARISON OF SEVERAL RESEARCH WORK IN THE AREA OF DETECTING PHISHING URLS USING MACHINE LEARNING AND DEEP LEARNING

Reference	Methodology	Strengths	Limitations
[30]	Dividing the hyperlink specific features into 12 different categories and used these features to train the machine learning algorithms.	High accuracy of 98.4% accuracy on logistic regression classifier.	Adaptation to dynamic web technologies
[48]	Developed a novel classification model, based on heuristic features that are extracted from URL, source code, and third-party services to overcome the disadvantages of existing anti-phishing techniques.	Zero-day phishing detection	Sensitivity to Feature Changes
[27]	Comparing the results of multiple machine learning methods for predicting phishing websites.	Adaptability to Evolving Phishing Methods	Limited Exploration of Ensemble Methods
[49]	Extraction of both email content and behavior-based features.	Comprehensive Feature Set	Assumed Effectiveness of Internet Service Provider Models and dynamic nature of UBEs

[50]	Use of List-Based Approaches	Effectively identifies phishing attempts by detecting mimicry of well-known websites.	Effectiveness depends on the selection of relevant features, making it sensitive to changes in phishing tactics.
[51]	Combining datasets, and creation of web application	Combined large dataset, low memory consumption	Inability to detect masked phishing URLs
[47]	Comparing the results of multiple models including Neural networks and K-Means	High accuracy of 97.8% of the Artificial Neural Network and Decision tree	Resource intensity
[52]	Combining ML and DL models to detect phishing URLs and Email.	Use of novel and domain-specific models like THEMIS.	Vulnerability of DL models to Adversarial Examples
[53]	Introduced a novel hybrid model (LSD) combining Logistic Regression, Support Vector Machine, and Decision Tree (LR+SVC+DT) with both soft and hard voting.	Comparative analyses demonstrated that the proposed approach (LSD model) outperformed other models, achieving the best results.	Some complex models, especially hybrid ones, might be less interpretable, making it challenging to understand their decision-making processes.

This study investigated how scammers craft phishing URLs to be able to examine the current methods for identifying phishing URLs. Scammers come up with different methods consistently to create phishing URLs to rob people of their hard-earned money. One such trick they use is to mask the URL with a URL shortener such as bit.ly to evade blacklisting. In this case, the unsuspecting user falls victim as they are unable to verify the credibility of the URL. This is the current state of phishing URL detection in cyberspace, with researchers coming up with several methods to detect these inadequacies, as shown in Fig. 2.

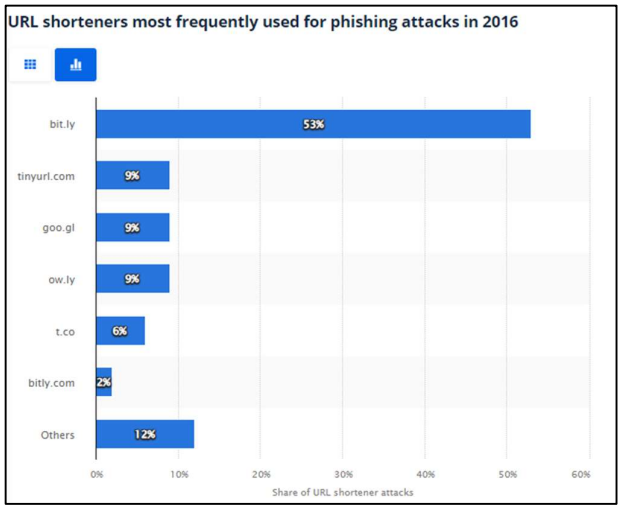

Fig. 2. Most used URL shorteners for phishing attacks [46].

The use of machine learning and deep learning has taken the stage for some years now, which prompted me to cite a number of relevant research topics in Table III above. However, one of the challenges being faced is the high computation requirement for running large models, as cited by Chawla, Ameya [47]. Therefore, I explored the use of big data tools like Apache Spark in combination with Sk-dist, and Spark MLlib.

V. Conclusion

In conclusion, this study marks a pivotal advancement in the domain of phishing URL detection, providing a nuanced exploration of machine learning models, namely Logistic Regression (LR), Random Forest (RF), and CatBoost (CB). The multifaceted analysis extended beyond conventional accuracy metrics, delving into precision, recall, and F1 score to unravel the distinctive strengths and limitations of each model.

The landscape of cybersecurity, dominated by the insidious rise of phishing attacks, demands adaptive and robust detection mechanisms. Traditional defense strategies, relying on static blacklists, fall short in the face of dynamic and sophisticated phishing campaigns. Recognizing this vulnerability, the study embraced machine learning methodologies as a beacon of hope, offering heightened accuracy and responsiveness to evolving cyber threats [38].

The comparative analysis unfolded as a meticulous dissection of LR, RF, and CB, revealing valuable insights. LR, though exhibiting accuracy, displayed vulnerabilities to outliers, compromising its reliability in unconventional phishing patterns. RF, proficient in handling distinct classes, faced challenges in interpretability and feature preference, posing risks of incorrect predictions. In contrast, CB emerged as a frontrunner, showcasing robust performance, streamlined hyper-parameter adjustment, and a unique approach to categorical feature analysis.

The practical implications of these findings are profound. CatBoost, with its superior performance, not only fortifies individual cybersecurity but also contributes to collective resilience against malicious online entities. End users stand to benefit from real-time phishing website detection, empowering them to stay vigilant in the evolving threat landscape. Moreover, cybersecurity authorities can leverage these insights to erect formidable defenses, preventing users from unknowingly navigating toward phishing websites.

However, the study acknowledges the trade-off in the time-intensive nature of training and testing datasets with CatBoost, necessitating a balanced consideration between accuracy and computational efficiency. The integration of tools like sk-dist and Apache Spark provides avenues for overcoming these challenges, paving the way for more streamlined, responsive, and scalable phishing detection mechanisms.

In this ever-evolving cybersecurity landscape, the symbiosis of advanced machine learning algorithms and powerful frameworks is pivotal [44]. This research contributes not only to an innovative phishing URL detection model but also underscores the ongoing evolution in securing digital realms. Through continuous refinement and exploration, the path toward a more secure online ecosystem unfolds, driven by the commitment to stay one step ahead in the ceaseless cat-and-mouse game of cybersecurity.

VI. ACKNOWLEDGMENTS

I extend my sincere and utmost gratitude to Prof. Tarig Ahmed for his impeccable guidance and mentorship throughout the course of this study. His expertise and unwavering support have been instrumental in shaping the trajectory of this work. I am truly fortunate to have had the opportunity to benefit from his wisdom and encouragement. Prof. Tarig's dedication to fostering academic excellence has left an indelible mark on this work, and I am grateful for the inspiration and knowledge he has shared.

VII. REFERENCES

[1] J. Milletary and C. C. Center, 'Technical trends in phishing attacks', Retrieved December, vol. 1, no. 2007, pp. 3–3, 2005.

[2] L. Tang, 'More than S$330 million lost to scammers in first half of 2023; cases continue to rise --- channelnewsasia.com', 2023. [Online]. Available: https://www.channelnewsasia.com/singapore/police-scam-cybercrime-phishing-fake-friend-malware-first-half-2023-3766796.

[3] B. Deekshitha, C. Aswitha, C. S. Sundar, and A. K. Deepthi, 'URL Based Phishing Website Detection by Using Gradient and Catboost Algorithms', Int. J. Res. Appl. Sci. Eng. Technol., vol. 10, no. 6, pp. 3717–3722, 2022.

[4] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, and A. J. Aljaaf, 'A systematic review on supervised and unsupervised machine learning algorithms for data science', Supervised and unsupervised learning for data science, pp. 3–21, 2020.

[5] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang, 'Big data analytics on Apache Spark', International Journal of Data Science and Analytics, vol. 1, pp. 145–164, 2016.

[6] M. S. O. Djediden, H. Reguieg, and Z. M. Maaza, 'A distributed intrusion detection system based on apache spark and scikit-learn library', Journal of Applied and Physical Sciences, vol. 5, no. 1, pp. 30–36, 2019.

[7] X. Meng et al., 'MLlib: Machine Learning in Apache Spark', The Journal of Machine Learning Research, vol. 17, no. 1, pp. 1235–1241, 2016.

[8] R. Mahajan and I. Siddavatam, 'Phishing website detection using machine learning algorithms', International Journal of Computer Applications, vol. 181, no. 23, pp. 45–47, 2018.

[9] M. Khonji, Y. Iraqi, and A. Jones, 'Phishing detection: a literature survey', IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.

[10] N. Moradpoor, B. Clavie, and B. Buchanan, 'Employing machine learning techniques for detection and classification of phishing emails', in 2017 Computing Conference, 2017, pp. 149–156.

[11] S. Patil and S. Dhage, 'A methodical overview on phishing detection along with an organized way to construct an anti-phishing framework', in 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 588–593.

[12] L. C. Fang, Z. Ayop, S. Anawar, N. F. Othman, N. Harum, and R. S. Abdullah, 'Url phishing detection system utilizing catboost machine learning approach', International Journal of Computer Science & Network Security, vol. 21, no. 9, pp. 297–302, 2021.

[13] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, 'Deep learning for phishing detection: Taxonomy, current challenges and future directions', IEEE Access, vol. 10, pp. 36429–36463, 2022.

[14] E. Benavides, W. Fuertes, S. Sanchez, and M. Sanchez, 'Classification of phishing attack solutions by employing deep learning techniques: A systematic literature review', Developments and Advances in Defense and Security: Proceedings of MICRADS 2019, pp. 51–64, 2020.

[15] A. Mughaid, S. AlZu'bi, A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsoud, 'An intelligent cyber security phishing detection system using deep learning techniques', Cluster Computing, vol. 25, no. 6, pp. 3819–3828, 2022.

[16] S. Al-Ahmadi, A. Alotaibi, and O. Alsaleh, 'PDGAN: Phishing detection with generative adversarial networks', IEEE Access, vol. 10, pp. 42459–42468, 2022.

[17] J. Kolla, S. Praneeth, M. S. Baig, and G. reddy Karri, 'A comparison study of machine learning techniques for phishing detection', Journal of Business and Information System (e-ISSN: 2685-2543), vol. 4, no. 1, pp. 21–33, 2022.

[18] J. L. Piñeiro and L. W. Portillo, 'Web architecture for URL-based phishing detection based on Random Forest, Classification Trees, and Support Vector Machine', Inteligencia Artificial, vol. 25, no. 69, pp. 107–121, 2022.

[19] G. P. Gupta and M. Kulariya, 'A framework for fast and efficient cyber security network intrusion detection using apache spark', Procedia Computer Science, vol. 93, pp. 824–831, 2016.

[20] M. N. Feroz and S. Mengel, 'Examination of data, rule generation and detection of phishing URLs using online logistic regression', in 2014 IEEE International Conference on Big Data (Big Data), 2014, pp. 241–250.

[21] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, 'A comparison of random forest variable selection methods for classification prediction modeling', Expert systems with applications, vol. 134, pp. 93–101, 2019.

[22] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, 'CatBoost: unbiased boosting with categorical features', Advances in neural information processing systems, vol. 31, 2018.

[23] A. A. Soofi and A. Awan, 'Classification techniques in machine learning: applications and issues', Journal of Basic & Applied Sciences, vol. 13, no. 1, pp. 459–465, 2017.

[24] T. Pietraszek and A. Tanner, 'Data mining and machine learning—towards reducing false positives in intrusion detection', Information security technical report, vol. 10, no. 3, pp. 169–183, 2005.

[25] G. S. Handelman et al., 'Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods', American Journal of Roentgenology, vol. 212, no. 1, pp. 38–43, 2019.

[26] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, 'Machine learning based phishing detection from URLs', Expert Systems with Applications, vol. 117, pp. 345–357, 2019.

[27] V. Shahrivari, M. M. Darabi, and M. Izadi, 'Phishing detection using machine learning techniques', arXiv preprint arXiv:2009.11116, 2020.

[28] C. Liu, L. Wang, B. Lang, and Y. Zhou, 'Finding effective classifier for malicious URL detection', in Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences, 2018, pp. 240–244.

[29] S. Garera, N. Provos, M. Chew, and A. D. Rubin, 'A framework for detection and measurement of phishing attacks', in Proceedings of the 2007 ACM workshop on Recurring malcode, 2007, pp. 1–8.

[30] A. K. Jain and B. B. Gupta, 'Towards detection of phishing websites on client-side using machine learning based approach', Telecommunication Systems, vol. 68, pp. 687–700, 2018.

[31] M. R. Segal, 'Machine learning benchmarks and random forest regression', 2004.

[32] L. Yang and A. Shami, 'On hyperparameter optimization of machine learning algorithms: Theory and practice', Neurocomputing, vol. 415, pp. 295–316, 2020.

[33] S. Garera, N. Provos, M. Chew, and A. D. Rubin, 'A framework for detection and measurement of phishing attacks', in

Proceedings of the 2007 ACM workshop on Recurring malcode, 2007, pp. 1–8.
[34] P. A. Câlburean et al., 'Prediction of 3-Year All-Cause Death in a Percutaneous Coronary Intervention Registry using Machine Learning: A Comparison Between Random Forest and CatBoost Algorithms', Applied Medical Informatics, vol. 43, pp. 21–21, 2021.
[35] L. Machado and J. Gadge, 'Phishing Sites Detection Based on C4.5 Decision Tree Algorithm', in 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), 2017, pp. 1–5.
[36] M. Kearns, 'Computational complexity of machine learning', in ACM distinguished dissertations, 1990.
[37] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, 'CatBoost: unbiased boosting with categorical features', in Advances in Neural Information Processing Systems, 2018, vol. 31.
[38] I. Salihovic, H. Serdarevic, and J. Kevric, 'The role of feature selection in machine learning for detection of spam and phishing attacks', in Advanced Technologies, Systems, and Applications III: Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies (IAT), 2019, vol. 2, pp. 476–483.
[39] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, 'Efficient machine learning for big data: A review', Big Data Research, vol. 2, no. 3, pp. 87–93, 2015.
[40] Y. Chen and X. Han, 'CatBoost for fraud detection in financial transactions', in 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE), 2021, pp. 176–179.
[41] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, and A. J. Aljaaf, 'A systematic review on supervised and unsupervised machine learning algorithms for data science', in Supervised and Unsupervised Learning for Data Science, Springer, 2020, pp. 3–21.
[42] J. Chen and C. Guo, 'Online detection and prevention of phishing attacks', in 2006 First International Conference on Communications and Networking in China, 2006, pp. 1–7.
[43] F. Y. Osisanwo, J. E. T. Akinsola, O. Awodele, J. O. Hinmikaiye, O. Olakanmi, and J. Akinjobi, 'Supervised machine learning algorithms: classification and comparison', International Journal of Computer Trends and Technology (IJCTT), vol. 48, no. 3, pp. 128–138, 2017.
[44] P. H. Las-Casas, V. S. Dias, W. Meira, and D. Guedes, 'A big data architecture for security data and its application to phishing characterization', in 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2016, pp. 36–41.
[45] S. Alnemari and M. Alshammari, 'Detecting Phishing Domains Using Machine Learning', Applied Sciences, vol. 13, no. 8, p. 4649, 2023.
[46] A. Petrosyan, 'URL shortener phishing usage 2016 | Statista --- statista.com', 2023. [Online]. Available: https://www.statista.com/statistics/266162/url-shortener-phishing-usage/.
[47] A. Chawla, 'Phishing website analysis and detection using Machine Learning', International Journal of Intelligent Systems and Applications in Engineering, vol. 10, no. 1, pp. 10–16, 2022.
[48] R. S. Rao and A. R. Pais, 'Detection of phishing websites using an efficient feature-based machine learning framework', Neural Computing and Applications, vol. 31, pp. 3851–3873, 2019.
[49] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, 'Applicability of machine learning in spam and phishing email filtering: review and approaches', Artificial Intelligence Review, vol. 53, pp. 5019–5081, 2020.
[50] 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.
[51] A. Deshpande, O. Pedamkar, N. Chaudhary, and S. Borde, 'Detection of Phishing Websites Using Machine Learning', International Journal of Engineering Research & Technology (IJERT), vol. 10, no. 05, 2021.
[52] D. Rathee and S. Mann, 'Detection of E-mail phishing attacks–using machine learning and deep learning', International Journal of Computer Applications, vol. 183, no. 1, p. 7, 2022.
[53] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, 'Phishing Detection System Through Hybrid Machine Learning Based on URL', IEEE Access, vol. 11, pp. 36805–36822, 2023.
[54] S. Smadi, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, 'Detection of phishing emails using data mining algorithms', in 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2015, pp. 1–8.

