IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 February 2025, accepted 5 April 2025, date of publication 15 April 2025, date of current version 25 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3561237

RESEARCH ARTICLE

RSTHFS: A Rough Set Theory-Based Hybrid Feature Selection Method for Phishing Website Classification

JAHANGGIR HOSSAIN SETU, NABARUN HALDER, (Student Member, IEEE), ASHRAFUL ISLAM, (Member, IEEE), AND M. ASHRAFUL AMIN, (Member, IEEE)

Center for Computational & Data Sciences, Independent University, Bangladesh, Dhaka 1229, Bangladesh

Corresponding author: Ashraful Islam (ashraful@iub.edu.bd)

This work was supported by Independent University, Bangladesh (IUB).

ABSTRACT Phishing is a pervasive form of cybercrime where malicious websites deceive users into revealing sensitive information, e.g., passwords and credit card details. Despite advances in cybersecurity, accurately detecting phishing websites remains challenging due to the absence of universally accepted identification parameters. This study introduces a novel feature selection method, Rough Set Theory-based Hybrid Feature Selection (RSTHFS), to enhance phishing website detection using Machine Learning (ML) techniques. Our approach was evaluated using three diverse datasets containing 2,456, 10,000, and 88,647 instances. The RSTHFS method demonstrated a significant improvement by maintaining an average accuracy rate of 95.48% while reducing the number of features by 69.11% on average. Performance was further assessed using three advanced classifiers: Light Gradient-Boosting Machine (LightGBM), Random Forest (RF), and Categorical Boosting (CatBoost), with CatBoost emerging as the most efficient, achieving the highest accuracy. Additionally, RSTHFS reduced the runtime by 61.43%, highlighting its efficiency. These findings indicate that RSTHFS is not only effective in identifying phishing websites but also accelerates ML processes, providing a reliable and swift approach to feature selection. This work contributes to the field by presenting a robust methodology that enhances the accuracy and speed of phishing detection systems.

INDEX TERMS Cyber security, feature selection, hybrid feature, machine learning, phishing, phishing websites, rough set theory, RSTHFS.

I. INTRODUCTION

In 1996, a team of cybercriminals used the term 'phishing' for the first time to describe how they tricked unwitting Netscape users into disclosing their credentials and stole their profiles on the service [1]. Phishing attacks aim to trick their targets into divulging sensitive information. Hackers imitate genuine websites to carry out a phishing attack. Then they give victims web links to these malicious websites through emails, instant messaging, social networking sites, multiplayer games, and other channels. There are several forms of phishing attacks that have surfaced, including spear phishing, drive-by downloads, cross-site scripting, malicious browsing extensions, and more [2], [3]. The most typical phishing attacks involve emailing users a malicious link and encouraging them to click on it. It is widespread on mobile platforms and personal computers since a website operates using the browser's engine and is not reliant on the underlying operating system. Phishing has become increasingly prevalent due to the growing usage of the Internet. The number of phishing attacks noticed by the Anti-Phishing Working Group (APWG) and its contributing members quadrupled during 2020, according to the phishing activity trends report. Attacks peaked in January 2021,

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Liu.

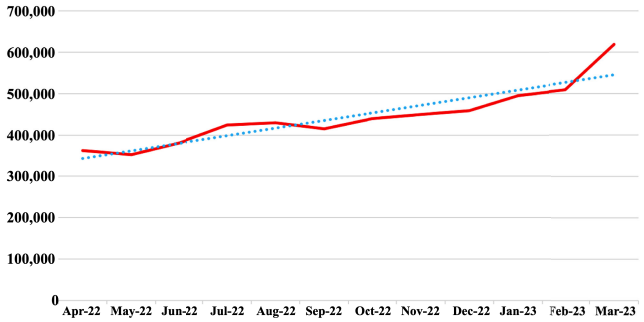

FIGURE 1. Phishing activity trend, 1st Quarter 2023 [4].

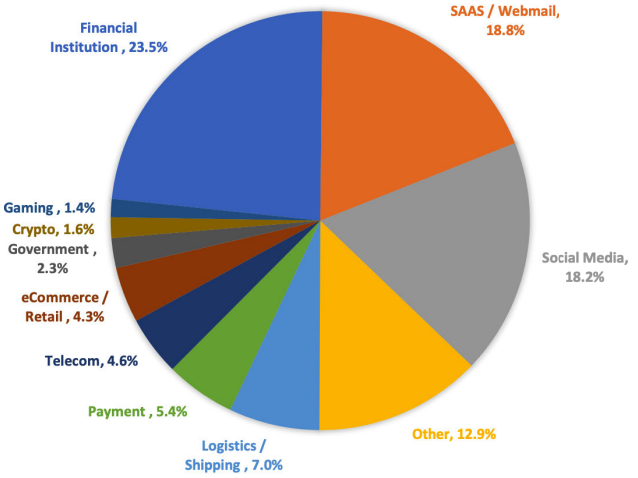

FIGURE 2. Most targeted industry, 1st Quarter 2023 [4].

when a record 245,771 new phishing sites were added in a single month [4]. The number of attacks did not go beyond 250,000 from April 2020 to March 2021. However, by March 2022, it was about 400,000.

The COVID-19 pandemic has immensely impacted lives worldwide leading to a surge in online activities, such as remote work, virtual education, and online shopping. This shift has also created new opportunities for cybercriminals to exploit users. Numerous COVID-19-related applications, e.g., iHealth, Ellume, Veritor, etc., for testing, therapies, cures, and remote work have been created and are widely used. Hackers are also targeting users through those applications [5].

The APWG reported that in the first quarter of 2023, 1,624,144 phishing attacks were detected overall. The quarterly total for phishing incidents hit one million for the first time this quarter, which was the worst APWG has ever seen. The 3rd quarter of 2022 saw 1,270,883 attacks, which was the previous high. Since the beginning of 2020, when APWG was observing between 68,000 and 94,000 attacks each month, the number of phishing attacks has more than tripled, as shown in Fig. 1 [4]. With 23.5% of all attacks, the financial sector was the most frequently targeted by phishing in 1st quarter of 2023, as depicted in Fig. 2 [4]. Software as a Service (SaaS) and webmail providers continued to be the target of 18.8% of all attacks. Notably, there was a decline in phishing incidents targeting cryptocurrency platforms, decreasing to 1.6% from the previous year's 6.6% [4].

Machine Learning (ML) is essential in combating phishing because it can handle large data and uncover complex patterns that humans struggle to recognize [6]. The web phishing dataset is characterized by numerous attributes, also referred to as a high-dimensional dataset [6]. ML algorithms suffer from overfitting in high-dimensionality spaces, even though they are meant to build models from specified dimensions [7]. Noisy features, such as needless and redundant ones, contribute to overfitting, where a classifier learns even the outliers, leading to poor performance [8]. Reducing these features not only improves accuracy but also reduces the requirement for large memory capacities and a strong Central Processing Unit (CPU). By eliminating redundant and irrelevant features, feature selection technique enhances classification model quality, simplifies understanding, and boosts the efficiency of inductive learners. ML can extract knowledge from datasets without human intervention, spot similarities, and predict outcomes. Supervised ML is the category under which the majority of ML algorithms for phishing detection fall. Here, a classifier attempts to learn certain traits of numerous authentic and phishing websites to predict an outcome. Based on anti-phishing software, ML algorithms collect features from Uniform Resource Locator (URL)s, hyperlink information, page content, digital certificates, website traffic, and other sources [1]. However, the feature set, training data, and type of ML algorithms affect the accuracy of the anti-phishing solution.

This study focuses on enhancing phishing website detection through the Rough Set Theory (RST)-based Hybrid Feature Selection (RSTHFS) framework. Effective feature selection is crucial for improving ML model performance, minimizing training and testing time, and addressing overfitting issues, particularly in phishing detection where not all website attributes are equally informative. The majority of existing research has utilized single-based feature selection techniques, e.g., Principal Component Analysis (PCA), Pearson correlation coefficient, and others [9], [10], yet these methods struggle with scalability when applied to large datasets. Also, existing filter-based feature selection methods, e.g., Chi-Square test, Information Gain (IG), Mutual Information (MI), Pearson correlation coefficient, Spearman's rank correlation, ReliefF algorithm, lack a systematic approach to determine the optimal cut-off rank. Most studies rely on arbitrary thresholds or predefined ranking criteria [10], [11], [12], [13], [14], [15], [16], [17], which may not always yield the most informative feature subset. Addressing these research gaps, this study proposes a hybrid feature selection method combining a Cumulative Distribution Function gradient (CDF-g) and RST. This approach aims to determine a stable, efficient feature subset,

reducing overfitting and training time of ML algorithms, and memory usage. The methodology employs CDF-g, a dynamic and data-driven technique for feature selection, which is used to generate the initial feature subset. and RST for baseline feature aggregation. We termed the entire feature selection framework as the Rough Set Theory-based Hybrid Feature Selection (RSTHFS). This method is assessed through advanced classifiers e.g., Light Gradient-Boosting Machine (LightGBM), Categorical Boosting (CatBoost), and Random Forest (RF) across three datasets, i.e., Mendeley 2018 [18], Mendeley 2020 [19], and phishing website detection from UCI ML [20]. Out of three classifiers, CatBoost was revealed to be the most efficient, consistent with the maximum accuracy rate. Furthermore, an additional benefit of RSTHFS was observed in its ability to decrease the runtime by an average of 61.43%, underscoring the method's efficiency.

II. RELATED WORKS

Researchers presented a unique strategy that uses an automatically updated whitelist of trustworthy websites each user visits to defend against phishing attacks [21]. Their testing findings demonstrate that the suggested technique can defend against phishing attacks since it has a True Positive (TP) rate of 86.02% and a False Negative (FN) rate of less than 1.48%. Blacklists are essential for defending online users against phishing scams. Blacklists' efficacy is influenced by various factors, including their size, scope, frequency, and accuracy of updates. Google Safe Browsing, OpenPhish (OP), and PhishTank are the three main phishing blacklists [22]. Although list-based systems offer a quick access time, they have a poor detection rate [23].

Phishing websites resemble their respective genuine websites to trick people into thinking they are visiting the right one. The choice is made using a feature set that includes text content, text format, HyperText Markup Language (HTML) elements, images, and more via visual similarity-based phishing detection algorithms [21], [24], [25], [26], [27]. One study combines a target website locator with an image and Cascading Style Sheets (CSS) [28] whereas another study employs database auto-updates and hue information [29]. Detecting phishing websites may be done thoroughly by looking for subspecies with similar colors. Abdelnabi et al. [24] present VisualPhishNet, a novel framework for similarity-based phishing detection that can develop profiles for websites using a similarity measure that can be applied to pages with different visual aesthetics. Despite multiple prior attempts, similarity-based detection algorithms do not provide reputable websites with enough security, especially against hidden phishing pages.

Sabahno and Safari [30] presented a Support Vector Machine (SVM)-based classification model that employs an improved spotted hyena optimization method for phishing website detection. With 11,055 instances and 30 features, the UCI repository was used for the study. Multiple ML algorithms, e.g., SVM, Convolution Neural Network (CNN), Regression Tree, MLP, and K-Nearest Neighbors (KNN), were compared with feature selection methods, e.g., PCA, Recursive Feature Elimination (RFE), and uni-variate feature selection, by Mourtaji et al. [11]. A single-source dataset of 40,000 occurrences and 37 features was obtained from PhishTank for the study. Nevertheless, the study required a large amount of computing effort and ignored mentioning the train and test split ratio as well as runtime analysis.

In [10] study, Gupta et al. proposed the use of RF, KNN, SVM, and Logistic Regression (LR) for detecting phishing websites with a small number of features. They used RF Score and Spearman correlation for feature selection and a single-source dataset from ISCXURL-2016 with 19,964 instances and 9 features. However, the study did not use Domain Name Service (DNS)-based features and used an old dataset. Hannousse and Yahiouche compared multiple ML algorithms, including RF, Decision Tree (DT), LR, and SVM, with feature selection techniques e.g., IG, Pearson correlation coefficient, and relief rank [12]. The study used a combination of datasets, including PhishTank, Alexa, OP, and Yardex, with 11,430 instances and 87 features. Moedjahedy et al. applied four classification algorithms, RF, DT, SVM, and Adaptive Boosting (AdaBoost), for phishing detection [31]. The study used a dataset of 10,000 URLs and 48 features from Mendeley. The study used a single-sourced dataset from the UCI repository with 11,000 URLs and 30 features. However, no feature selection or runtime analysis was conducted. Wei and Sekiya used RF for phishing URL detection, and the study's focus was on feature importance [32].

In [33] study, Jha et al. have employed three popular classifiers, namely Naïve Bayes (NB), RF, and SVM, for their research work. The study aims to develop a predictive model for a specific problem using a dataset that was sourced from Mendeley. The dataset used in the study contains 13,410 instances and 111 features. The classification accuracy achieved by the models is 0.95, indicating the high performance of the classifiers on the given dataset. Penta et al. [34] conducted a study on phishing detection using three classifiers, KNN, SVM, and NB. The dataset used for the study was obtained from PhishTank, and it consisted of 40,000 instances with 37 features. The study reported an accuracy of 96.67%. The study conducted by Al-Tamimi and Shkoukani [13] utilized an RF classifier with an extra tree classifier for feature selection on a dataset sourced from Kaggle. The dataset consisted of 11,055 instances with 30 features, and the achieved accuracy was 97.1%.

A Deep Neural Network (DNN)-based classification model has been proposed by Lakshmi et al. [35] as a means of identifying phishing websites. The UCI repository provided a single-sourced dataset with 11,000 occurrences and 30 features for the research. Pavan et al. proposed in [14]

combining a binary bat algorithm and a CNN with swarm intelligence to identify phishing websites. They made use of a single-source dataset consisting of 30 features and 11,055 instances from Kaggle. Singh and Mishra proposed the use of the Classical RST for detecting phishing websites [15]. They used a single-source dataset from Mendeley with 10,000 instances and 48 features. Al-Ahmadi et al. used CNN and Long Short-Term Memory (LSTM) for phishing URL detection [36]. The study used a dataset of 40,000 URLs and 37 features from PhishTank. However, the authors did not provide any information regarding runtime analysis and the train-test split ratio. Also, no feature selection was performed, and the study noted high computational time requirements. Alsariera et al. used two classification algorithms, Tree and Best First Tree, for detecting phishing URLs [37]. The authors used a dataset with 13,410 URLs and 111 features from Mendeley. The feature importance was calculated using a built-in feature importance metric in RF. However, no runtime analysis or train-test split ratio was provided. Also, the study mentioned that human intervention is required for choosing the feature importance threshold value.

The most frequently employed feature selection techniques in the articles reviewed included RFE, PCA, gini coefficient, chi-squared, Pearson correlation coefficient, IG, and gain ratio. These techniques were each used separately in the majority of the evaluated research. RF performed the best average across most of the analyzed articles, with accuracy scores ranging from 94.6% to 99.57%. Algorithms including LR, SVM, extreme learning machine, DNN, Multi-Layer Perceptron (MLP), Gradient Boost, and CNN delivered good accuracy overall.

According to Adane and Beyene, while many studies have utilized individual feature selection techniques such as Principal Component Analysis, Recursive Feature Elimination, and others, only one study has explored a hybrid ensemble feature selection method [16].

Our work builds on this by introducing a novel hybrid approach that integrates Rough Set Theory, which has not been previously explored in this context.

III. MATERIALS AND METHODS

A. DATASET DESCRIPTION

This study utilizes three popular open-source phishing website datasets [18], [19], [20] to experiment. Utilizing an array of three datasets, the results of the study become more reliable and broadly applicable, providing a better comprehension of the proposed methods and how well they work in various phishing website scenarios. The labels in these datasets are binary (phishing or legitimate). The comprehensive information regarding the datasets is provided below, and a summary is outlined in Table 1.

1) DATASET 1 - MENDELEY 2018 [18]

48 features were obtained from 5,000 authentic and 5,000 fraudulent websites to create this dataset. An enhanced feature extraction technique that makes use of the browser automation process is more dependable and accurate than the regular expression-based parsing technique. PhishTank and OP are the sources for the dataset's phishing homepage. The authentic website also has Alexa and Common Crawl as sources.

2) DATASET 2 - UCI ML REPOSITORY [19]

This dataset contains 30 features extracted from 2,456 websites. The authors highlighted in this dataset the key features that are reliable and efficient in identifying phishing websites. Additionally, they suggested some features. The authors collected the dataset from the PhishTank and MillerSmiles websites.

3) DATASET 3 - MENDELEY 2020 [20]

This dataset contains 111 features extracted from 88,647 websites. The authors created a database of 30,647 verified phishing sites from the PhishTank website. On the other hand, the authors gathered 58,000 valid website domain names for site collection from the Alexa ranking website.

B. PROPOSED METHODOLOGY

Fig. 3 shows the overview of the proposed methodology for this study. In this methodology, the phishing dataset went into some pre-processing steps to make it suitable for ML classifiers. The data pre-processing steps contain duplicity checking, null value checking, encoding labels, class distribution checking, etc. In duplicity checking, duplicate entries were eliminated so that the classifiers do not overlearn from the duplicate entries. Null value checking was conducted because most of the ML classifiers cannot deal with the null value in a dataset. Mean imputation was conducted to fill up those places. Data encoding was performed to convert from string to numerical values for the target class. After all these pre-processing steps, the processed dataset was fed into the feature selection technique which reduced the number of features without any human intervention. Then, those reduced features were used for training and testing the classifiers. The train-test split ratio was 80:20. The classifiers were evaluated based on test data in terms of classification metrics, e.g., runtime analysis, accuracy, sensitivity or recall, precision, and F1-score. Afterward, a comparative analysis was performed among those classifiers' results to find the most suitable model for the phishing website detection task.

C. ROUGH SET THEORY-BASED HYBRID FEATURE SELECTION (RSTHFS)

Sampling is the process of concluding a sample using a small sample. Stratified sub-sampling is carried out by splitting the total sample into homogenous groups, namely, strata [38], as shown in Fig. 3. Proportionate stratified random sampling refers to random samples from stratified groupings that are proportionate to the population [38].

TABLE 1. Summary of the datasets used in the experiment.

Dataset	No. of instances	No. of features	Missing Values	Published Date
Dataset-1 Mendeley 2018 [18]	10,000	48	813	24 March 2018
Dataset-2 UCI ML Repository [19]	2,456	30	316	25 March 2015
Dataset-3 Mendeley 2020 [20]	88,647	111	615	24 September 2020

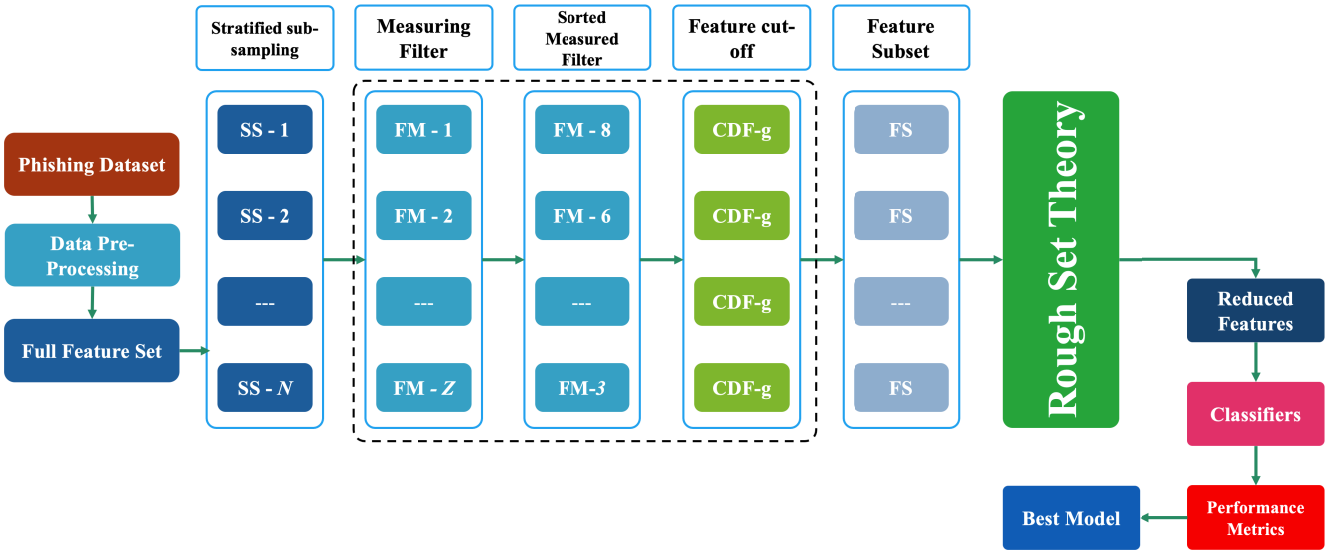

FIGURE 3. An overview of the proposed methodology.

In a disproportionate sampling, the strata do not correspond to the population distribution.

Stratified subsampling guarantees that every possible sample has an equal probability of happening, in contrast to basic subsampling, which selects data at random from a population as a whole [38]. The primary advantage of stratified subsampling is in its ability to precisely capture significant demographic features within the sample. Similar to a weighted average, this sampling technique produces features in the sample that are proportional to the entire population. If subgroups cannot be generated, stratified sub-sampling is useless for populations with various features. Compared to the straightforward random sampling approach, stratification provides a lesser estimate error and improved accuracy [39]. The improvement in accuracy increases with the size of the discrepancies between the layers.

In order to reduce vector space without compromising detection precision, a subset of feature variables is chosen for ML-based phishing detection [40]. When using filter measures in feature subset selection techniques (depicted in Fig. 3), the best cut-off rank must be selected in order to accomplish the aforementioned aim [41]. Selecting a cut-off rank is necessary in order to obtain a smaller feature subset, as only features positioned above this cut-off rank are chosen. Fig. 4 shows the cut-off rank, where a discontinuous rectangle surrounds the chosen feature subset.

Assume that M is a discrete random variable and that m denotes one of the variable's potential values. The probability

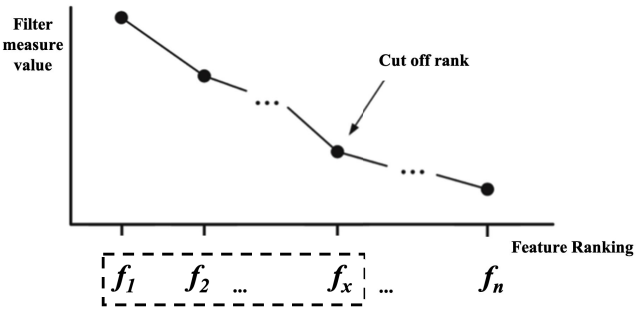

FIGURE 4. Overview of CDF-g algorithm [40].

density function, also known as the frequency function, is a mathematical expression that represents the likelihood that the discrete random variable M will take on a certain value m [40]:

$$P(M) = m \tag{1}$$

The following equation defines the Cumulative Distribution Function (CDF) of the random variable M [40]:

$$F_M(u) = P(M <= u) \tag{2}$$

The CDF for the random variable M is represented by the notation $F_M(u)$, which is a function of u. The discrete random variable is considered to reflect the value of the filter measure calculated for each feature by adapting the CDF in the feature selection context. The plateau areas in the range of

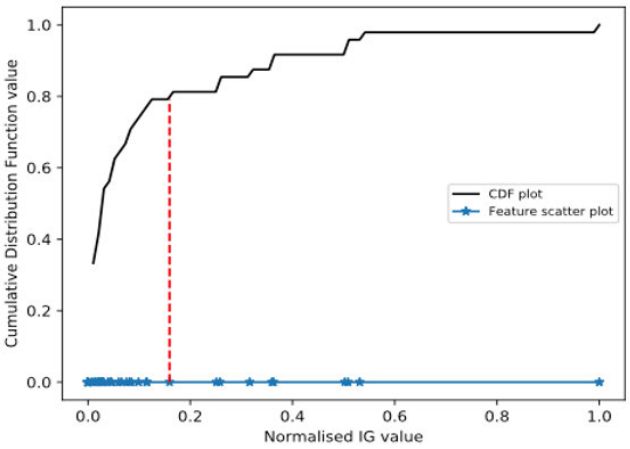

FIGURE 5. An illustration of where to find a plateau on the IG values' CDF-g curve [40].

filter measure values can be found by examining the gradient variations in the CDF curve. A curve's plateau portions are those with zero gradients, indicating that the gaps between the succeeding values are quite wide. In the context of the feature selection process, the distinction between two feature subsets that have substantially different predictive powers is represented as a plateau area inside a CDF curve of filter measure values. As a result, the method uses this idea to determine the top features' cut-off rank. As demonstrated in the following equation, central differences for the interior points and one-side (forward or backward) differences for the boundary are utilized to calculate the gradient at any point R_i on the CDF curve [40]:

$$G(R_i) = \begin{cases} \dfrac{F_M(u_{i+1}) - F_M(u_i)}{h}, & \text{if } i = 1 \\ \dfrac{F_M(u_{i+1}) - F_M(u_{i-1})}{2h}, & \text{if } 1 < i < n \\ \dfrac{F_M(u_i) - F_M(u_{i-1})}{h}, & \text{if } i = n \end{cases} \quad (3)$$

The number of points located on the CDF curve is indicated by n, and the default distance is set at $h = 1.0$.

In particular, the values of IG are calculated for the whole set of features when IG is used as a filtered measure. Following normalization, the obtained values are sorted in ascending order [40]. The sorted numbers are then used to construct and depict a CDF curve. To determine the cut-off rank, the first plateau over 50% of the cumulative value is found using the gradient of the CDF curve, and its associated normalized IG value is then mapped to the ranking of features in ascending order, as indicated with a red dotted straight line illustrated in Fig. 5. The link between the cut-off rank and the feature distribution within the range of normalized IG values is made clear by the scatter plot in Fig. 5.

Without needing any extra information, RST can be utilized to identify data relationships, and minimize, and aggregate the number of feature variables included in a dataset. In the last 10 years, RST has grown in popularity among scholars and was employed in several domains [42], [43]. It is feasible to use RST to identify the subset (referred to as a reduct) of the original feature variables that are the most relevant given a dataset with discretized feature values [44]. RST estimates a reduct without thoroughly producing all potential subsets. The rough set correlation metric is increased incrementally, starting with a null set and adding each feature one by one until the score reaches its maximum value for the dataset [44]. This process iterates until the dataset's integrity and the reduct's dependence are equal.

D. CLASSIFICATION ALGORITHMS

Choosing appropriate classification algorithms is important for achieving reliable and efficient results. This study explores three popular classifiers: RF, CatBoost, and LightGBM.

1) RANDOM FOREST (RF)

RF employs a collective training approach by constructing numerous decision trees. In this ensemble method, each tree contributes a singular vote towards assigning the most prevalent class determined by the input data [45]. RF excels in handling complex relationships and reducing overfitting [46].

2) CATEGORICAL BOOSTING (CATBOOST)

CatBoost, designed for categorical feature support, uses a DT boosting approach and requires minimal hyperparameter tuning [47]. CatBoost utilizes ordered target statistics and order-boosting methods. Both methods involve employing random permutations of the training examples to counteract the prediction shift induced by a particular form of target leakage present in all current gradient-boosting algorithms [47].

3) LIGHT GRADIENT-BOOSTING MACHINE (LIGHTGBM)

LightGBM, a gradient-boosting framework, focuses on leaf-wise growth, demonstrating efficiency with large datasets and high accuracy [48].

The selection of these classifiers is justified based on their robustness, adaptability to diverse data characteristics, effectiveness with categorical features for CatBoost, and efficiency with large-scale datasets for LightGBM, ensuring a comprehensive and effective approach to phishing website detection.

E. PERFORMANCE METRICS

1) ACCURACY

Accuracy refers to the percentage of correctly classified instances out of all instances, in contrast to the error value, which considers misclassified instances instead of correctly classified ones [49]. The equation for accuracy is as follows [49]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (4)$$

TABLE 2. The optimal feature set from Dataset 1 (Feature names and descriptions are available in [18]).

Features	Description
NumDash	Dashes in the URL of the webpage.
NumNumericChars	The webpage URL's numeric character count.
PctExtHyperlinks	Proportion of external links found in the HTML source code.
PctExtResourceUrls	HTML source code percentage of external resource URLs.
PctNullSelfRedirectHyperlinks	Proportion of the current URL's hyperlink fields that are empty, anomalous, or self-redirecting.
FrequentDomainNameMismatch	The count of mismatches in the domain name value.
ExtMetaScriptLinkRT	Proportion of external URLs contained in meta, script, and link tags.
PctExtNullSelfRedirectHyperlinksRT	Proportion of backlinks with distinct domain names found in the HTML source code.

TABLE 3. The optimal feature set from Dataset 2 (Feature names and descriptions are available in [19]).

Features	Description
having_IP_Address	The search field will display websites with a given port number and IP address in the format.
Shortining_Service	URL is abbreviated, potentially leading to fraudulent websites.
Prefix_Suffix	The "-" sign on links allows users to add prefixes and suffixes to the website, giving the impression that it is a legitimate one.
having_Sub_Domain	Where the top-level and second-level domains are removed from the base of subdomains.
SSLfinal_State	Website host is part of a leading phishing IP network.
Domain_registeration_length	How many more years/months are the domain registration valid for?
HTTPS_token	Using the HTTPS protocol or not, as well as if the certificate was issued by a reliable source and is one year old.
Request_URL	Whether they are from another website or not, the pictures, animations, and other graphical content on this page.
URL_of_Anchor	A different domain than this one is referenced via anchor tags.
Links_in_tags	Website tags may contain connections to other domains or the same domain as the website itself.
SFH	Value of SFH if "about:blank" is present. It implies that it does not specify how the information given will be treated.
web_traffic	Popularity of the website.

where,
True Positive (TP): The proportion of phishing websites that were successfully identified as such.
False Positive (FP): The proportion of phishing websites that have been mistaken for real websites.
True Negative (TN): The proportion of reliable websites that are designated as such.
False Negative (FN): The proportion of legitimate websites that have been labeled as phishing sites.

2) SENSITIVITY OR RECALL

Recall or sensitivity is a measure of the proportion of TP instances that are correctly identified by the ML system [49]. It determines how accurately the system can label genuinely reliable websites. The formula for Recall or sensitivity is as follows [49]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

3) PRECISION

Precision considers positive instances. It determines the proportion of accurately projected positive instances to all anticipated positive instances [49]. It addresses the question of how many verified websites are genuinely authentic. Precision values are linked to low FP rates [49]. The mathematical formula for Precision is as follows [49]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

4) F1-SCORE

The F1-score is the harmonic mean between Precision and Recall and takes into account both FN and FP [49]. The formula for the F1-score is as follows [49]:

$$\text{F1} - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

5) RUNTIME ANALYSIS

The run-time analysis is defined as the amount of time an algorithm takes to classify samples in test data. The low computational time for the classifier is another benefit of fewer features [40].

IV. EXPERIMENTAL RESULT AND ANALYSIS

Through the utilization of the proposed RSTHFS technique, we were able to discern a set of optimal features, that effectively exploits contemporary phishing schemes and makes a substantial contribution to the enhancement of phishing detection. The ultimate optimal set of features comprises 10 distinct features from Dataset 1, 12 features from Dataset 2, and 40 features from Dataset 3. The optimal feature set of Dataset 1, 2, and 3 are detailed in Table 2, 3, and 4.

Tables 5, 6, and 7 demonstrate the result for classification metrics and the runtime of the classifiers on each distinct dataset individually.

In Dataset 1, the RSTHFS technique significantly reduces the feature set by 83.33%. The full feature set contains

TABLE 4. The optimal feature set from Dataset 3 (Feature names and descriptions are available in [20]).

Features	Description
qty_slash_url	How many "/" indications are in the URL
length_url	Character count in the URL
qty_dot_directory	How many "." indications are in the URL
qty_hyphen_directory	How many "-" indications are in the URL
qty_underline_directory	How many "_" indications are in the URL
qty_slash_directory	How many "/" indications are in the URL
qty_questionmark_directory	How many "?" indications are in the URL
qty_equal_directory	How many "=" indications are in the URL
qty_at_directory	How many "@" indications are in the URL
qty_and_directory	How many "&" indications are in the URL
qty_exclamation_directory	How many "!" indications are in the URL
qty_space_directory	How many " " indications are in the URL
qty_tilde_directory	How many "∼" indications are in the URL
qty_comma_directory	How many "," indications are in the URL
qty_plus_directory	How many "+" indications are in the URL
qty_asterisk_directory	How many "*" indications are in the URL
qty_hashtag_directory	How many "#" indications are in the URL
qty_dollar_directory	How many "$" indications are in the URL
qty_percent_directory	How many "%" indications are in the URL
directory_length	How many directory characters
qty_dot_file	How many "." indications are in the file name
qty_hyphen_file	How many "-" indications are in the file name
qty_underline_file	How many "_" indications are in the file name
qty_slash_file	How many "/" indications are in the file name
qty_questionmark_file	How many "?" indications are in the file name
qty_equal_file	How many "=" indications are in the file name
qty_at_file	How many "@" indications are in the file name
qty_and_file	How many "&" indications are in the file name
qty_exclamation_file	How many "!" indications are in the file name
qty_space_file	How many " " indications are in the file name
qty_tilde_file	How many "∼" indications are in the file name
qty_comma_file	How many "," indications are in the file name
qty_plus_file	How many "+" indications are in the file name
qty_asterisk_file	How many "*" indications are in the file name
qty_hashtag_file	How many "#" indications are in the file name
qty_dollar_file	How many "$" indications are in the file name
qty_percent_file	How many "%" indications are in the file name
file_length	How many file name characters
asn_ip	Autonomous System Number
time_domain_activation	Days needed for domain activation

48 features where the number of features selected by RSTHFS is only 8. When comparing CatBoost on selected and full feature sets, a slight accuracy degradation of 1.73% occurs, accompanied by a substantial 73.34% reduction in runtime. CatBoost and LightGBM exhibit identical performance in terms of classification metrics on selected features, but CatBoost demonstrates lower runtime than LightGBM. RF, while showcasing a negligible accuracy degradation of 0.98%, observes a notable 30% reduction in runtime. RF outperforms CatBoost and LightGBM in terms of classification metrics but lags behind CatBoost in runtime, as indicated in Table 5. However, the runtime for LightGBM increases after feature selection due to its gradient-based loss computation, which may lead to longer convergence times.

For Dataset 2, the RSTHFS technique reduces the feature set by 60%. The full set of features comprises 30 features, while RSTHFS only selected 12. When comparing CatBoost on selected and full feature sets, a minor accuracy deterioration of 2.04% is observed, accompanied by a significant 54% reduction in runtime. Unexpectedly, the findings show a 73% increase in runtime for LightGBM, with only a 2.04% decrease in selected feature accuracy. RF experienced a minor accuracy loss of 2.51% and a reduction in runtime by around 11%. Notably, CatBoost outperforms LightGBM in classifying data more efficiently. In contrast to dataset 1, RF underperforms compared to CatBoost and LightGBM in all classification metrics by a small margin. However, Table 6 highlights that RF was four times slower than CatBoost in classifying test data.

Dataset 3 witnesses a substantial 64% reduction in the feature set with the RSTHFS technique. The full feature set consists of 111 features, whereas RSTHFS opted for a subset of 40 features. A minimal accuracy drop of 0.8% is observed, along with a significant 57% decrease in runtime by CatBoost. This time, LightGBM reveals a slight accuracy deterioration of 0.8% and a 15% increase in runtime. Runtime for LightGBM also increased while classifying test data for dataset 1 and dataset 2. RF saw a little accuracy deterioration of 0.83%, and the runtime was lowered by around 10%. In particular, LightGBM classifies data three times faster than CatBoost. However, RF narrowly outperforms CatBoost and LightGBM in all classification metrics. It is worth noting that RF classifies test data 4.5 times slower than LightGBM, as presented in Table 7.

In order to substantiate the efficacy of the proposed methodology, we conducted a benchmark comparison with a contemporary hybrid feature selection approach in the field of phishing detection, as presented by Chiew et al. [40]. In their work, the authors introduced a hybrid feature selection algorithm named Hybrid Ensemble Feature Selection (HEFS) and subjected it to evaluation using Dataset-1 Mendeley 2018, which aligns with the dataset employed in our study. This dataset, encompassing 10,000 instances and featuring 48 distinct attributes, has established itself as a widely adopted benchmark in the world of phishing detection research. The outcomes of the benchmarking, as depicted in Table 8, indicate that our RSTHFS framework surpasses the feature selection method introduced by Chiew et al. [40], concurrently achieving a substantial reduction of up to 83.33% in feature dimensionality. Furthermore, the runtime analysis highlights the efficiency of our proposed methodology, demonstrating a 76.29% acceleration in comparison to the approach presented by Chiew et al. [40].

Also, our study outperforms existing state-of-the-art feature selection methods for phishing website detection. Compared to eXtreme Gradient Boosting (XGBoost) with Diversity Oriented Firefly Algorithm (DOFA), which achieved 95.54% accuracy, our method using RSTHFS with CatBoost surpasses it with 96.48% accuracy [50]. Similarly, RF with PCA and RFE reported 95.38% and 94.97% accuracy, respectively, falling short of our study result [51]. The OFS-NN approach reached 96.44% accuracy but required 1.275 seconds for classification, whereas our method achieves better accuracy with an extremely low classification time of 0.0124 milliseconds [52].

TABLE 5. Performance comparison of classifiers on selected feature set and full feature set of Dataset 1 - Mendeley 2018.

Classifier	Features (n)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Runtime Analysis (ms)
CatBoost	Selected Features (8)	96.48	96.44	96.48	96.46	0.0124
	Full Feature Set (48)	98.21	98.22	98.18	98.19	0.0466
LightGBM	Selected Features (8)	96.48	96.44	96.48	96.46	0.0187
	Full Feature Set (48)	98.21	98.22	98.18	98.19	0.0162
RF	Selected Features (8)	96.64	96.61	96.63	96.62	0.0437
	Full Feature Set (48)	97.62	97.65	97.55	97.60	0.0625

TABLE 6. Performance comparison of the classifiers on selected feature set and full feature set of Dataset 2 - UCI Repository.

Classifier	Features (n)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Runtime Analysis (ms)
CatBoost	Selected Features (12)	93.71	92.99	93.59	93.27	0.0071
	Full Feature Set (30)	95.75	95.30	95.78	95.53	0.0153
LightGBM	Selected Features (12)	93.71	92.99	93.59	93.27	0.0083
	Full Feature Set (30)	95.75	95.30	95.78	95.53	0.0048
RF	Selected Features (12)	93.08	92.17	93.27	92.65	0.0276
	Full Feature Set (30)	95.59	95.27	95.41	95.34	0.0310

TABLE 7. Performance comparison of classifiers on selected feature set and full feature set of Dataset 3 - Mendeley 2020.

Classifier	Features (n)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Runtime Analysis (ms)
CatBoost	Selected Features (40)	96.26	95.90	95.92	95.91	0.2435
	Full Feature Set (111)	97.07	96.78	96.78	96.78	0.5671
LightGBM	Selected Features (40)	96.27	95.90	95.92	95.91	0.0920
	Full Feature Set (111)	97.07	96.78	96.78	96.78	0.0777
RF	Selected Features (40)	96.37	96.01	96.05	96.03	0.3505
	Full Feature Set (111)	97.20	96.93	96.93	96.93	0.3906

TABLE 8. Performance benchmarking comparison.

Feature Selection	Number of Features	Accuracy (%)	Runtime/Classification Time (ms)
HEFS + RF	10	94.6	0.0523
RSTHFS + CatBoost (Our proposed framework)	8	96.48	0.0124

It is evident from the result analysis that the RSTHFS technique consistently reduces features, and CatBoost proves advantageous in terms of both accuracy and runtime across all three datasets.

V. CONCLUSION

Phishing websites are on the rise, misleading website visitors into doing things that provide cybercriminals the chance to steal essential data from them. The hacker gains access to an organization's sensitive data after breaching its cyber security system. To enhance phishing website identification through ML, this study introduces the RSTHFS technique. Thorough evaluation across three distinct datasets demonstrates the remarkable effectiveness of RSTHFS, maintaining a high average accuracy of 95.48% alongside a substantial feature count reduction of 69.11%. RSTHFS contributes an additional benefit by reducing runtime by an average of 30%, emphasizing its efficiency. The study's findings unequivocally establish RSTHFS as the most effective technique for phishing website identification, offering a reliable, accurate, and expeditious method for feature selection. Conducting experiments and evaluations across various industries and sectors, e.g., finance, healthcare, e-commerce, and more, could shed light on the adaptability and generalizability of RSTHFS beyond the specific datasets used in this study.

REFERENCES

[1] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommun. Syst.*, vol. 76, no. 1, pp. 139–154, Jan. 2021.

[2] Y. A. Younis and M. Musbah, "A framework to protect against phishing attacks," in *Proc. 6th Int. Conf. Eng. MIS*, Sep. 2020, pp. 1–6.

[3] S. J. Y. Weamie, "Cross-site scripting attacks and defensive techniques: A comprehensive survey," *Int. J. Commun., Netw. Syst. Sci.*, vol. 15, no. 8, pp. 126–148, 2022.

[4] (2023). *Anti-Phishing Working Group*. Accessed: Apr. 15, 2025. [Online]. Available: https://apwg.org/trendsreports/#:~:text=Summary%20%E2%80%93%202nd%20Quarter%202023

[5] A. F. Al-Qahtani and S. Cresci, "The COVID-19 scamdemic: A survey of phishing attacks and their countermeasures during COVID-19," *IET Inf. Secur.*, vol. 16, no. 5, pp. 324–345, Sep. 2022.

[6] K. Gao, T. M. Khoshgoftaar, and A. Napolitano, "Exploring software quality classification with a wrapper-based feature ranking technique," in *Proc. 21st IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2009, pp. 67–74.

[7] M. Ashraf, F. Anowar, J. H. Setu, A. I. Chowdhury, E. Ahmed, A. Islam, and A. Al-Mamun, "A survey on dimensionality reduction techniques for time-series data," *IEEE Access*, vol. 11, pp. 42909–42923, 2023.

[8] X. Ying, "An overview of overfitting and its solutions," *J. Phys., Conf. Ser.*, vol. 1168, Feb. 2019, Art. no. 022022.

[9] A. Suryan, C. Kumar, M. Mehta, R. Juneja, and A. Sinha, "Learning model for phishing website detection," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 7, no. 27, p. e6, 2020.

[10] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," *Comput. Commun.*, vol. 175, pp. 47–57, Jul. 2021.

[11] Y. Mourtaji, M. Bouhorma, D. Alghazzawi, G. Aldabbagh, and A. Alghamdi, "Hybrid rule-based solution for phishing URL detection using convolutional neural network," *Wireless Commun. Mobile Comput.*, vol. 2021, no. 1, pp. 1–24, Jan. 2021.

[12] A. Hannousse and S. Yahiouche, "Towards benchmark datasets for machine learning based website phishing detection: An experimental study," *Eng. Appl. Artif. Intell.*, vol. 104, Sep. 2021, Art. no. 104347.

[13] Y. Al-Tamimi and M. Shkoukani, "Employing cluster-based class decomposition approach to detect phishing websites using machine learning classifiers," *Int. J. Data Netw. Sci.*, vol. 7, no. 1, pp. 313–328, 2023.

[14] P. P. Kumar, T. Jaya, and V. Rajendran, "Si-BBA—A novel phishing website detection based on swarm intelligence with deep learning," *Mater. Today, Proc.*, vol. 80, pp. 3129–3139, Feb. 2023.

[15] A. Singh and S. C. Misra, "A comparison of performance of rough set theory with machine learning techniques in detecting phishing attack," in *Advances in Computing, Informatics, Networking and Cybersecurity: A Book Honoring Professor Mohammad S. Obaidat's Significant Scientific Contributions*. Cham, Switzerland: Springer, 2022, pp. 631–650.

[16] K. Adane and B. Beyene, "Machine learning and deep learning based phishing websites detection: The current gaps and next directions," *Rev. Comput. Eng. Res.*, vol. 9, no. 1, pp. 13–29, May 2022.

[17] X. Deng, Y. Li, J. Weng, and J. Zhang, "Feature selection for text classification: A review," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 3797–3816, Feb. 2019.

[18] C. L. Tan, "Phishing dataset for machine learning: Feature evaluation," *Mendeley Data*, vol. 1, p. 2018, Mar. 2018.

[19] R. Mohammad and L. McCluskey, "Phishing websites," UCI Mach. Learn. Repository, Tech. Rep., 2015, doi: 10.24432/C51W2X.

[20] G. Vrbančič, "Phishing websites dataset," *Mendeley Data*, vol. 1, p. 2020, Sep. 2020.

[21] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Syst. Appl.*, vol. 117, pp. 345–357, Mar. 2019.

[22] S. S. Roy, U. Karanjit, and S. Nilizadeh, "Evaluating the effectiveness of phishing reports on Twitter," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, Dec. 2021, pp. 1–13.

[23] S. Vidyakeerthi, M. Nabeel, C. Elvitigala, and C. Keppitiyagama, "Demo: PhishChain: A decentralized and transparent system to blacklist phishing URLs," in *Companion Proc. Web Conf.*, Apr. 2022, pp. 286–289.

[24] S. Abdelnabi, K. Krombholz, and M. Fritz, "VisualPhishNet: Zero-day phishing website detection by visual similarity," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 1681–1698.

[25] C. C. L. Tan, K. L. Chiew, K. S. C. Yong, Y. Sebastian, J. C. M. Than, and W. K. Tiong, "Hybrid phishing detection using joint visual and textual identity," *Expert Syst. Appl.*, vol. 220, Jun. 2023, Art. no. 119723.

[26] M. Wang, L. Song, L. Li, Y. Zhu, and J. Li, "Phishing webpage detection based on global and local visual similarity," *Expert Syst. Appl.*, vol. 252, Oct. 2024, Art. no. 124120.

[27] C. Opara, Y. Chen, and B. Wei, "Look before you leap: Detecting phishing Web pages by exploiting raw URL and HTML characteristics," *Expert Syst. Appl.*, vol. 236, Feb. 2024, Art. no. 121183.

[28] R. S. Rao and A. R. Pais, "Two level filtering mechanism to detect phishing sites using lightweight visual similarity approach," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 9, pp. 3853–3872, Sep. 2020.

[29] E. S. Aung, C. T. Zan, and H. Yamana, "A survey of URL-based phishing detection," in *Proc. DEIM Forum*, 2019, pp. G2–G3.

[30] M. Sabahno and F. Safara, "ISHO: Improved spotted hyena optimization algorithm for phishing website detection," *Multimedia Tools Appl.*, vol. 81, no. 24, pp. 34677–34696, Oct. 2022.

[31] J. Moedjahedy, A. Setyanto, F. K. Alarfaj, and M. Alreshoodi, "CCrFS: Combine correlation features selection for detecting phishing websites using machine learning," *Future Internet*, vol. 14, no. 8, p. 229, Jul. 2022.

[32] Y. Wei and Y. Sekiya, "Sufficiency of ensemble machine learning methods for phishing websites detection," *IEEE Access*, vol. 10, pp. 124103–124113, 2022.

[33] A. K. Jha, R. Muthalagu, and P. M. Pawar, "Intelligent phishing website detection using machine learning," *Multimedia Tools Appl.*, vol. 82, no. 19, pp. 29431–29456, Aug. 2023.

[34] U. B. Penta, B. Panda, and S. S. Gantayat, "Machine learning model for identifying phishing websites," *J. Data Acquisition Process.*, vol. 38, no. 1, p. 2455, 2023.

[35] L. Lakshmi, M. P. Reddy, C. Santhaiah, and U. J. Reddy, "Smart phishing detection in Web pages using supervised deep learning classification and optimization technique Adam," *Wireless Pers. Commun.*, vol. 118, no. 4, pp. 3549–3564, Jun. 2021.

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

[37] Y. A. Alsariera, A. O. Balogun, V. E. Adeyemo, O. H. Tarawneh, and H. A. Mojeed, "Intelligent tree-based ensemble approaches for phishing website detection," *J. Eng. Sci. Technol.*, vol. 17, pp. 563–582, Aug. 2022.

[38] J. Ongoma, D. A. Alilah, and O. Erick, "Optimal allocation in small area mean estimation using stratified sampling in the presence of non-response," Tech. Rep., 2021. [Online]. Available: https://www.sciencepublishinggroup.com/article/10.11648/j.ijsd.20210701.13

[39] I. Qabajeh and F. Thabtah, "An experimental study for assessing email classification attributes using feature selection methods," in *Proc. 3rd Int. Conf. Adv. Comput. Sci. Appl. Technol.*, Dec. 2014, pp. 125–132.

[40] K. L. Chiew, C. L. Tan, K. Wong, K. S. C. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Inf. Sci.*, vol. 484, pp. 153–166, May 2019.

[41] K. D. Rajab, "New hybrid features selection method: A case study on websites phishing," *Secur. Commun. Netw.*, vol. 2017, pp. 1–10, Feb. 2017.

[42] A. K. Das, S. Sengupta, and S. Bhattacharyya, "A group incremental feature selection for classification using rough set theory based genetic algorithm," *Appl. Soft Comput.*, vol. 65, pp. 400–411, Apr. 2018.

[43] M. Prasad, S. Tripathi, and K. Dahal, "An efficient feature selection based Bayesian and rough set approach for intrusion detection," *Appl. Soft Comput.*, vol. 87, Feb. 2020, Art. no. 105980.

[44] R. K. Bania and A. Halder, "R-HEFS: Rough set based heterogeneous ensemble feature selection method for medical data classification," *Artif. Intell. Med.*, vol. 114, Apr. 2021, Art. no. 102049.

[45] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Jun. 2001.

[46] S. J. Rigatti, "Random forest," *J. Insurance Med.*, vol. 47, no. 1, pp. 31–39, Jan. 2017.

[47] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: An interdisciplinary review," *J. Big Data*, vol. 7, no. 1, pp. 1–45, Dec. 2020.

[48] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 1–11.

[49] Ž. Vujovic, "Classification model evaluation metrics," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 599–606, 2021.

[50] L. Jovanovic, D. Jovanovic, M. Antonijevic, B. Nikolic, N. Bacanin, M. Zivkovic, and I. Strumberger, "Improving phishing website detection using a hybrid two-level framework for feature selection and XGBoost tuning," *J. Web Eng.*, vol. 22, no. 3, pp. 543–574, Jul. 2023.

[51] M. A. Daniel, S.-C. Chong, L.-Y. Chong, and K.-K. Wee, "Optimising phishing detection: A comparative analysis of machine learning methods with feature selection," *J. Informat. Web Eng.*, vol. 4, no. 1, pp. 200–212, Feb. 2025.

[52] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu, "OFS-NN: An effective phishing websites detection model based on optimal feature selection and neural network," *IEEE Access*, vol. 7, pp. 73271–73284, 2019.

JAHANGGIR HOSSAIN SETU received the B.Sc. degree in computer science and engineering from Daffodil International University, Bangladesh. He is currently pursuing the M.Sc. degree in computer science with Independent University, Bangladesh (IUB). He is currently a Laboratory Instructor in computer science courses with IUB, where he is also working with the Center for Computational & Data Sciences (CCDS) as a Graduate Research Assistant. His research interests include machine learning, pattern recognition, cybersecurity, and human-centered computing. He has published several articles in different renowned venues/journals, including IJCNN, ICMLA, and IEEE Access.

NABARUN HALDER (Student Member, IEEE) received the B.Sc. degree in computer science and engineering from Daffodil International University, Bangladesh. He is currently pursuing the M.Sc. degree in computer science with Independent University, Bangladesh (IUB). He is currently a Laboratory Instructor in computer science courses with IUB, where he is also working with the Center for Computational & Data Sciences (CCDS) as a Graduate Research Assistant. His research interests include machine learning, pattern recognition, cybersecurity, and human-centered computing. He has published several articles in different renowned venues, including IJCNN, ICMLA, and IEEE ASET.

ASHRAFUL ISLAM (Member, IEEE) received the M.S. and Ph.D. degrees in computer science from the University of Louisiana at Lafayette, USA. He is currently an Assistant Professor in computer science and engineering with Independent University, Bangladesh (IUB), where he is also the Director of the Center for Computational & Data Sciences (CCDS). His research interests include machine learning, pattern recognition, applied computational intelligence, and human-centered computing. He is an Active Member of the IEEE System, Man and Cybernetics (SMC) Society. His research works have been published in different renowned venues/journals, including ACM CHI, ACM UIST, IJCNN, ICMLA, *JMIR Human Factors*, and IEEE INTERNET OF THINGS JOURNAL.

M. ASHRAFUL AMIN (Member, IEEE) received the Ph.D. degree from the Electrical Engineering Department, The City University of Hong Kong, in 2009. He is currently an Active Contributor to the field of machine learning. He is currently a Professor with the School of Engineering, Technology & Sciences, Independent University, Bangladesh (IUB), where he established the Center for Computational & Data Sciences (CCDS) and has been the Founding Director of CCDS. His current research interests include bioinformatics, pattern recognition, augmented reality creation, and surveillance through the use of computer vision. He is an Active Member of the IEEE System, Man and Cybernetics (SMC) Society.