IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 January 2024, accepted 27 March 2024, date of publication 3 April 2024, date of current version 15 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3384840

RESEARCH ARTICLE

An Efficient Hybrid Feature Selection Technique Toward Prediction of Suspicious URLs in IoT Environment

SANJUKTA MOHANTY[1], ARUP ABHINNA ACHARYA[1], TAREK GABER[2,3], NAMITA PANDA[1], ESRAA ELDESOUKY[4,5], AND IBRAHIM A. HAMEED[6], (Senior Member, IEEE)

[1]School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha 751024, India
[2]School of Science, Engineering, and Environment, University of Salford, M5 4WT Salford, U.K.
[3]Faculty of Computers and Informatics, Suez Canal University, Ismailia 41522, Egypt
[4]Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[5]Department of Computer Science, Faculty of Computers and Informatics, Suez Canal University, Ismailia 41522, Egypt
[6]Department of ICT and Natural Sciences, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, 6009 Ålesund, Norway

Corresponding authors: Sanjukta Mohanty (mailtorani.sanjukta@gmail.com) and Ibrahim A. Hameed (ibib@ntnu.no)

This work was supported by Prince Sattam Bin Abdulaziz University under Project PSAU/2023/R/1445.

ABSTRACT With the growth of IoT, a vast number of devices are connected to the web. Consequently, both users and devices are susceptible to deception by intruders through malicious links leading to the disclosure of personal information. Hence, it is essential to identify suspicious URLs before accessing them. While numerous researchers have proposed several URL detection approaches, the machine learning-based technique stands out as particularly effective because of its ability to detect zero-day attacks; however, its success depends on the type and dimension of features utilized. In earlier research, only the lexical features of URLs were employed for classification to attain high detection speeds. However, this approach does not allow for the retrieval of comprehensive information about a website. Hence, to enhance the security of IoT devices, both lexical and page content-based features of URLs must be considered. To improve the performance of the model, the researchers extract informative features using different Feature Selection Techniques (FSTs), including filter and wrapper methods. However, challenges such as the demand for more resources, time, and handling of high-dimensional datasets encountered by individual FSTs have driven the development of hybrid FSTs. Nevertheless, the combination of a filter-based FST and a wrapper search-based Genetic Algorithm (GA) is used in the identification of malicious URLs as well as the detection of malicious links in the IoT devices research studies. Therefore, the proposed approach leverages the advantages of a variety of features and explores a hybrid FST to produce the optimal feature subset to evaluate the boosting estimators with specific hyperparameter configurations. Our proposed approach effectively fills the research gap associated with previous methodologies research 99% while keeping the computational costs minimal, making it suitable for resource-constrained devices in detecting malignant URLs.

INDEX TERMS Boosting estimators, feature selection technique (FSTs), genetic algorithm (GA), Internet of Things (IoT), suspicious URLs.

I. INTRODUCTION

Nowadays, the rapid advancement of communication technology, coupled with sophisticated techniques on the web, has promoted new e-commerce opportunities. As a result, most businesses are shifting online as they get reliable infrastructure like huge cloud storage, lucrative platforms, and a large target market [1]. Therefore, the expansion of the internet has led to an increase in the global number of internet users, reaching nearly 5.18 billion, according to the first quarter report of Statista 2023 [2]. However, this technological progress encourages intruders to organize illegitimate actions against organizations, companies, and governments and manipulate unsuspicious users through

The associate editor coordinating the review of this manuscript and approving it for publication was Kah Phooi (Jasmine) Seng.

phishing [3], [4], [5]. Intruders employ various attack strategies, often utilizing compromised URLs as their weapon. These URLs, once compromised, transform into malicious URLs, serving as hosts for a range of unsolicited content, including drive-by downloads, phishing schemes, malware dissemination, defacement, spam, cryptojacking, and IoT malware. They skillfully lure innocent users into becoming victims of various scams, such as malware installations, financial losses, theft of sensitive information, and cryptocurrency fraud [4], resulting in substantial annual losses. When suspicious URLs are redirected to IoT devices (tablets, mobile phones, smart watches) whose screens are very small enough even to read the address bar information, cyber offenses are easily and quickly carried out by hackers [43]. According to the Cyber Threat Report 2022, 10.4 million malware attacks occur annually, and Google's Transparency report shows that 3.165 million websites were declared as dangerous in the first quarter of 2023, and 2.1 million were phishing websites [6]. In the year 2023, the Security Export Insights report [7], declares that 12.8 million websites are infected with different malware worldwide. The IBM Security X-Force Threat Intelligence Index recorded that 41% of attacks are used for phishing and 16% of attacks abuse valid accounts. [8], Sonicwall Threat Report [9] states 112.3 million IoT malware attacks and 139.3 million cryptojacking attacks spiked in the year 2022, and it is increasing year over year. As the different security threats rise exponentially, it becomes crucial to distinguish and act on such attacks early. Researchers have suggested many solutions, like blacklisting, signature-based detection schemes, and machine learning techniques to identify malicious URLs [4]. The most widely deployed technique is the traditional blacklisting-based approach, which maintains a central database that contains known suspicious URLs [10]. Thus, identifying new malicious websites becomes tedious. The signature-based detection scheme scans the signature assigned to the malicious websites and raises flags if any abnormal behaviors are detected. As it is implemented only for classifying the executable code, this proposed approach cannot identify malicious websites accurately [11]. Another more promising and intelligent approach is the classification approach based on Machine Learning (ML) [12], [13], [14], [15], [44], which utilizes pre-trained features to accurately predict malicious URLs. The various classification techniques employed for detecting malicious URLs encounter a range of issues and challenges, including, among other things, high dimensionality, lengthy training and testing times, as well as low detection rates. Furthermore, most of the existing research has focused on enhancing machine learning classifiers rather than feature selection [13]. Researchers have used a variety of feature selection techniques to choose the most relevant, redundant-free, and predictive features in order to address these issues and challenges. The individual feature selection technique has its pros and cons. The filter-based FST evaluates the features without the use of machine learning classifiers, which inhibits it from identifying the relevant and efficient minimal feature subsets required for predicting malicious URLs [19]. On the other hand, the wrapper-based FST consumes more time and requires additional computational resources (storage, preprocessing, and training time) when dealing with large datasets, as it has to train a new estimator for each subset [11]. Still, it provides the most influential and significant features and hence enhances classification performance [20]. To enhance the performance of the machine learning mode, extracting the optimal sets of URL features is essential. These optimal URL feature subsets must be derived from the high-dimensional URL dataset. Extracting the optimal URL feature subset involves fusing the combined capabilities of both filter-based FST and wrapper-based FST. The combined feature selection approach has been employed in prior research studies related to email classification [21], intrusion detection [22], social bot identification [23], and XSS attack prediction [24]. In these studies, the error rate tends to be higher while model performance is lower. However, this approach has not yet been explored in the context of malicious URL detection or the identification of malicious links within IoT devices.

Therefore, our proposed approach combines both the filter FST and the wrapper FST to address the challenges posed by individual feature selection methods. These challenges include increased training time, additional preprocessing overhead, a low true positive rate, and poor classification performance in terms of accuracy within the existing frameworks. Designing a proper model for the classification of malignant URLs is also a time-intensive task. Few studies have been conducted on the feature selection technique, particularly focusing on GA [19], [25], [26]. However, the resulting model's detection accuracy remains moderate, ranging from 89% to -95%. The combination of Mutual Information Gain (MIG) from the filter-based FST with search-based GA for selecting the relevant features in the realm of malicious URL classification remains unexplored in any research studies. The objective of our proposed approach is to identify the optimal subset of features that can yield higher detection accuracy and precision in the detection of malicious URLs on resource-constrained devices. All experiments use a publicly accessible URL dataset from Kaggle [27], which was also the basis for our prior study [15]. This dataset encompasses both lexical and page-content-based URL features. The experimental outcomes underscore the efficacy of the proposed approach, achieving satisfactory results with smaller feature sets with an accuracy of 98.3% and a precision rate close to 99%. The smallest and most representative feature sets obtained from the proposed hybrid feature selection technique are the most essential and valuable requirements in IoT devices due to their relatively low computation power and resources. Hence, the proposed research is claimed to be suitable for tracking unauthorized

access, data leakage, and other security vulnerabilities of IoT devices.

The contributions of the proposed approach are as follows:

1) It focuses on a novel combined searching strategy by fusing filter and wrapper FST to select the optimal feature subset. This enhancement aids the machine learning classifiers to effectively handle higher-dimensional datasets as well as the small feature sets that can be used in IoT devices as they need low computation power and resources.
2) The essential features obtained from hybrid feature selection techniques are evaluated through ensemble classifiers with effective hyperparameter optimization to enhance the overall classification performance.
3) Simulation results are evaluated, analyzed, and compared against outcomes from existing research as well as the ground truth, resulting in significantly improved results.

The remainder of the paper is organized as follows: Section II provides a review of existing studies. The proposed detection methodology and some background concepts are presented in Section III. Section IV represents the experimental result and analysis, and a comparison between the proposed approach and the existing studies is outlined. The conclusion, limitations, and future aspects of the study are depicted in Section V.

II. REVIEW OF EXISTING STUDIES

In recent years, numerous techniques have been explored in the literature to identify suspicious websites. However, only a few research on predicting malicious URLs within the context of the IoT environment has been conducted. In this section, we provide an overview of detection approaches against malicious websites, including features, feature selection methods, classification techniques, and evaluation metrics. In addition, the limitations of existing feature selection techniques are presented by various researchers. This discussion is particularly significant due to the vital role of feature selection in enhancing the detection of suspicious URLs.

In a recent survey, the authors [10] formulated an approach for the detection of phishing URLs in IoT devices where only 9 lexical URL features were considered for phishing attack detection. Machine learning (ML) classifiers, including Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbor (k-NN), and Logistic Regression (LR), were employed to evaluate the features and achieve an accuracy of 99% along with minimal response time. However, there are instances when a large number of features of different types are required to accurately represent a URL. Thus, in our proposed approach, we address the challenges of malicious URL detection by incorporating both lexical and page-content-based features. In [16], a phishing email detection approach has been outlined where filter FSTs such as Information Gain (IG), Correlation Feature Selection (CFS), and Chi-Square (CHI) are utilized to extract influential attributes from the feature space. Subsequently, ML algorithms such as Decision Tree (J45), Rule, JRip, and PART are evaluated using the selected features to assess their detection performance. To quantify the effectiveness of the filter methods, the classifiers were evaluated twice. Initially, the evaluation was conducted using the full set of features, a total of 47 features in the dataset, and subsequently, using the features derived from the filter methods, using 12 high-scored features. Experiments revealed that the machine learning model has provided notably improved results after passing the full set of features through the filter methods. However, the significance or quality of each feature was not analyzed thoroughly. Thabtah et al. [17] employed various filter-based feature selection techniques, such as CHI, CFS, and IG, to extract a reduced set of feature sets for identifying suspicious behaviors such as phishing websites. A cut-off point was established to differentiate between high-impact features and low-impact ones. The feature sets produced by the various FSTs were then employed to train mining algorithms such as PART, RIPPER, and C4.5, in order to measure and assess the predictive accuracy. Though this approach managed to successfully identify significant features by utilizing different cut-off points, it proved to be a time-consuming process and hence not suitable for timely or real-time detection of phishing URLs, given that data theft by intruders can occur within seconds. In [28], the authors used an attribute-based feature selection approach to separate phishing websites from real ones. They used techniques like gain ratio and relief to lower the dimension of the feature matrix by getting rid of attributes that weren't needed or relevant. The two minimal feature matrices that were created by FST were then tested using ML algorithms such as Naïve Bayes (NB), J48, and sequential minimization optimization (SMO). The J48 classifier achieved a classification accuracy of 98.1% compared to SMO, which had a classification performance of 96.42%. The J48 classifier achieved the best result with an accuracy of 98.1% outperforming SMO (96.42%). However, this approach has a limitation in the form of a very low true positive rate (TPR) of 97.2% in the J48 classifier, making it inefficient for distinguishing phishing websites from legitimate ones. To efficiently differentiate between phishing and legitimate, a light-weight dataset tailored for the IoT environment was presented by [29]. For website classification, they considered four types of lexical features: length-oriented, counting-related, http/s-related, length-oriented, counting-related, http/s-related, and Natural Language Processing (NLP)-related. To select relevant features, they employed filter-based FSTs like InfoGain, Chi-Square, and Releif. The scores from these FSTs are ensembled to obtain important features for evaluating machine learning classifiers such as KNN, RF, DT, and SMO (Sequential Minimal Optimization). Their approach achieved a phishing detection rate of 99%. The limitation of filter-based FSTs in the context of malicious URL detection or IoT is that some of the machine learning algorithms achieve high accuracy while others experience poor performance. This occurs because filter-based methods

utilize intrinsic measures to rank each feature independently of the particular machine learning classifiers.

To address this issue, [18] employed an evolutionary algorithm, Particle Swarm Optimization (PSO), as a wrapper-based feature selection method for the identification of phishing websites from safe ones. The proposed feature weighting method assigns higher weights to highly influential website features and lower weights to less important ones, effectively eliminating irrelevant features from the dataset. The results show that the wrapper-based feature weighting approach outperforms the filter-based chi-square and IG methods. The primary drawback of this approach lies in the use of classical machine learning classifiers like Back Propagation Neural Networks (BPNN), Support Vector Machine (SVM), NB, C4.5, RF, and k-NN instead of any ensemble machine learning algorithms. Another limitation that can hinder its use in real-time applications is that feature evaluation and weighting are computationally expensive in terms of time. The extended version of the feature weighting approach is used in [19], where an evolutionary GA-based search technique is considered for feature selection and feature weighting to select the most influential and pertinent features to learn a Deep Neural Network (DNN) instead of classical ML classifiers, aiming to achieve the highest classification accuracy. The experimental results demonstrated that DNN achieved the highest accuracy of 88.7%, compared to BPNN's accuracy of 87.4% in detecting phishing URLs from legitimate ones. The drawback of the DNN-based feature weighting method is that it takes a longer time for feature evaluation. Patil et al. [30] enhanced the classifiers' performance by utilizing FSTs like CFS, Latent Semantic Analysis (LSA), Information Gain Ratio (IGR), Principal Component Analysis (PCA), CHI, and wrapper feature selection (WFS) such as ranker search, greedy stepwise search, and genetic search techniques, among others, for detecting malicious web pages. Out of 117 static and dynamic features, 15 significant features were selected using FS-based methods (greedy stepwise search with the J48 classifier). These selected features were then used to train and evaluate the majority voting classifier, resulting in a detection accuracy of 99%. Additionally, they compared the proposed methodology with various anti-virus tools and anti-malware software and achieved significant results. Khonji et al. [21] classified phishing emails by evaluating several feature subset selection methods, including filters and wrapper evaluators, to extract the most effective and relevant feature subset from a feature space containing 47 features. The experiments led to the creation of a highly accurate email classifier with an F1-score of 99.1%. While wrapper-based feature selection methods outperform filter-based methods, they are computationally intensive and demanding.

Hence, to address the issues with both filter-based and wrapper-based feature selection methods, a combined FST is employed in [23] to generate an optimal feature subset for identifying social bots. The proposed approach analyzes the profile features of the bot and human accounts on Twitter to obtain the optimal web phishing features using a hybrid FST. The authors used three different filtering methods—correlation attribute (CA), cross-validation attribute evaluation (CVAE), and information gain—each on its own to score the features without involving the learning classifiers. They then combine the strength of the best-first search technique of the WFS technique to reduce the number of features produced by the filter-based FSM to produce the best-performing optimal features. Machine learning classifiers such as RF, SVM, NB, and neural networks (NN) are used to evaluate the optimal features for classifying bots from human accounts. Experimental results demonstrated that the RF classifier achieves an accuracy of 89%, a ROC of 94.3%, and a precision score of 83.9% for the bot class. Kamarudin et al. [22] combined the strengths of filter- and wrapper-based feature selection procedures to search for the best features for an intrusion detection system. The hybrid model is evaluated with optimal features generated from the CFS of filter methods along with the best-first search, genetic search, and greedy search of the wrapper-based FS procedure. The wrapper-based subset evaluation uses the RF algorithm to classify the attributes selected by the filter-based FST. Two datasets were used for testing all the experiments, and the results showed that the hybrid FST produced satisfactory outcomes in terms of detection accuracy. However, the proposed approaches in [22] and [23] show poor performance due to the use of some classical ML algorithms that might underfit the training data. Moreover, the hybrid feature selection technique is also not adopted in any of the existing research studies for malicious URL detection in the IoT context.

Although various approaches have been explored in the literature for identifying malignant websites, the threat to websites persists. No single feature selection technique alone can effectively detect a suspicious website [10], [14]. Furthermore, filter-based FSMs tend to work better with high-dimensional datasets but lack satisfactory accuracy [16], [17], [28], [29]. On the other hand, wrapper-based FSTs offer higher accuracy but require more computational overhead [18], [19], [30]. The hybrid FST, a combination of filter- and wrapper-based approaches, leverages the strengths of both approaches. Despite the adoption of several hybrid FSTs by various researchers in the literature [22] and [23], they have not achieved significant improvements in classification performance in terms of low accuracy and low false positive rate (FPR). Therefore, this paper focuses on proposing an efficient filter-wrapper-based hybrid FST that extracts an optimal feature subset. This subset can effectively train and evaluate the ensemble ML classifiers to efficiently detect malicious URLs in the context of IoT.

III. PROPOSED METHODOLOGY

This section outlines how the proposed approach, utilizing an ensemble machine learning technique, enhances the identification of malicious websites through the incorporation of hybrid feature selection methods. The proposed

detection approach is illustrated in Fig. 1. It consists of two main phases: the preprocessing phase and the detection phase. The preprocessing phase involves dataset preparation, feature extraction, and feature selection. The detection phase incorporates various machine learning classifiers.

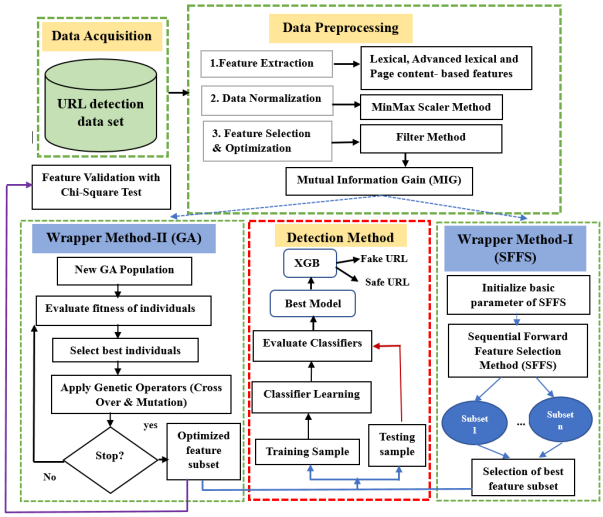

FIGURE 1. Schematic framework of the proposed approach.

A. DATASET PREPARATION

This research employs a dataset proposed in [27]. The dataset is split into two subsets: training and testing sets, with an 80:20 ratio. This implies that 80% of the data set is utilized for training, while the remaining 20% of the data set around 2000 samples is set aside for testing (i.e., holdout validation). The detailed descriptions of the dataset are explained in Section B. The categorization of the dataset features is explained in Table 1. Out of 10,000 samples, 5000 were collected from the benign corpus, and the rest of the samples were collected from phishing websites.

B. FEATURE EXTRACTION

In this phase, the website features used to analyze URLs as either malicious or benign are broadly categorized into three types: lexical features, advanced lexical features, and page-content-based features, as shown in Table 1. A total of 48 features were extracted from the 10,000 phishing and benign websites. The phishing URLs were sourced from PhishTank and OpenPhish, while the benign URLs were obtained from Common Crawl archives and Alexa. The numbers of lexical features, advanced lexical features, and page-content-based features are 17, 12, and 18, respectively. These features were extracted from various sources, including the web page URL, HTML source code, and third-party services such as domain registries, WHOIS records, and search engines. The lexical and page-content features are referred to as internal features, as they can be directly derived from the web page itself. On the other hand, the advanced lexical features are termed external features as they are derived from third-party services. To automate the feature extraction process, a tool called Selenium WebDriver and Python scripts are employed to instruct the browser to load the web pages [27] and [42].

1) LEXICAL-BASED FEATURE

The structural properties of the URL define its lexical features. The rationale behind considering lexical features in the detection of malicious URLs is related to the nature of a malicious URL, which can be identified by their appearance or "look" [4]. Malicious URLs often resemble benign URLs by mimicking their names with very slight variations. Additionally, suspicious URLs' names tend to have a longer length compared to benign ones due to the presence of additional dots ('.'), slashes ('/'), ampersands ('_&'), and so on, etc. [14]. Focusing on these statistical attributes, a machine learning algorithm can be easily trained in a shorter time on lexical features to distinguish between malicious and benign URL names. Therefore, lexical features have the potential to effectively classify the URLs as either malicious or safe. In this research work, we have incorporated 12 lexical features, as depicted in Table 1.

TABLE 1. Feature extraction.

Lexical based feature	Advanced lexical based feature	Page content-based feature
NumDots	SubdomainLevel	PctExtHyperlinks
UrlLength	IpAddress	PctExtResourceUrls
NumDash	DomainInSubdomains	ExtFavicon
AtSymbol	DomainInPaths	InsecureForms
TildeSymbol	HttpsInHostname	RelativeFormAction
NumUnderscore	FrequentDomainNameMismatch	AbnormalFormAction
NumQueryComponents	SubdomainLevelRT	PctNullSelfRedirectHyperlinks
NumAmpersand	PathLevel	FakeLinkInStatusBar
NumPercent	NumDashInHostname	RightClickDisabled
NumQueryComponents	HostnameLength	PopUpWindow
NumNumericChars	SubdomainLevel	SubmitInfoToEmail
NoHttps	PathLength	IframeOrFrame
RandomString		MissingTitle
QueryLength		ImagesOnlyInForm
DoubleSlashInPath		PctExtResourceUrlsRT
NumSensitiveWords		AbnormalExtFormActionR
EmbeddedBrandName		ExtMetaScriptLinkRT
		PctExtNullSelfRedirectHyperlinksRT

2) ADVANCED LEXICAL-BASED FEATURE

The lexical characteristics are directly obtained from the URL string without requiring significant domain knowledge. However, to extract more informative features, researchers have proposed advanced lexical features, such as domain features, directory-related features, file name features, and argument- or parameter-list-based features depicted in Table 1. The reason for considering advanced lexical features is that, unlike genuine websites, suspicious websites are not registered with reputable host centers and tend to exist for a short period of time. Therefore, having information about

the URL's registration date and expiration date is essential for identifying a suspicious URL.

3) PAGE-CONTENT BASED FEATURE

Web page content features are acquired by downloading the entire web page, including the HTML source code-based features and Java script-based features. The rationale behind incorporating page-content-based features is that they provide a wealthy information about a particular web page compared to URL-based features, which can ultimately lead to improved detection classifiers. In our research paper, we have utilized 18 page-content features, as represented in Table 1.

C. FEATURE SELECTION

In data engineering, feature selection is an essential step commonly used for high-dimensional data to minimize the search space [32] by eliminating irrelevant attributes, thus retaining only correlated variables or input features with the target variables or features [31]. FSTs derive an optimal subset of features that can enhance predictive power of the machine learning algorithms while reducing dataset dimensionality. This reduction in dimensionality not only lowers the computational cost of the model but also facilitates deployment and speeds up the overall process by shortening training time and inference time and lowering the chances of overfitting. FSTs are generally categorized into three types: filter method, wrapper method, and hybrid method [10]. In the filter method type, the relevant features are assessed by considering the statistical characteristics of the data rather than involving machine learning algorithms. Examples of filter-based FSTs include Mutual Information Gain (MIG), CFS, and CHI. In contrast to wrapper methods, the filter method is characterized by its low computational cost and scalability. The wrapper approach, on the other hand, collaborates closely with machine learning algorithms to perform feature selection.

In the wrapper approach, feature selection works in conjunction with machine learning algorithms. It incurs a high computational cost when the number of attributes is too large; however, it offers greater prediction accuracy [19], [33]. Wrapper methods can be further categorized into two groups: Sequential Selection Strategy and Randomized Algorithms [34]. Examples of sequential selection strategy methods include best-first search, sequential forward selection (SFS), and backward elimination. Evolutionary and swarm intelligence algorithms fall under randomized algorithms. In this research, an evolution-based GA is employed in the proposed approach for the detection of malicious URLs. Also, the SFS method is used to compare the efficiency of the evolutionary GA.

To overcome the limitation of individual FST, the advantages of both filter-based FST and wrapper-based FST are combined to create a hybrid FST (HFST). This can effectively distinguish malicious URLs from benign ones by extracting the optimal feature subsets. The strength of hybrid FST encompasses proper utilization of available data, providing better results [41] while being computationally inexpensive compared to the wrapper method, effective feature dependencies, reduced susceptibility to over-fitting compared to the wrapper method, and improved classification performance. A comparative study of existing work for suspicious URL detection based on FST (filter, wrapper, and hybrid FST) and their strengths and weaknesses are discussed in Table 2.

In recent years, there has been a focus in the literature on combining the filer- and wrapper-based methods to enhance classification accuracy. Unfortunately, existing studies employing hybrid feature selection have struggled to effectively identify the significant, relevant, and essential features from the original dataset. To address these limitations of current hybrid approaches, the proposed method introduced an efficient and impactful HFST that aimed to select the most informative essential features from the pool of features, as they are crucial and significant for performing an accurate URL classification.

In the initial phase of implementation, we have combined the MIG from the filter method with the sequential forward feature selection of the wrapper method. This aims to obtain a more informative subset of features for effective malicious URL detection. In phase II, we integrate MIG with a GA evolution-based searching technique to further optimize the subset of features, resulting in improved performance for detecting suspicious URLs.

1) HYBRID FEATURE SELECTION APPROACH-I (MIG-SFS)

Mutual Information Gain (MIG): In the proposed approach, MIG is utilized as an initial phase of the filter method to reduce the dimensionality of the dataset. MIG is employed to identify the most significant features by computing the entropy, which represents the level of uncertainty associated with a random variable. A feature with lower entropy indicates higher information gain in terms of the class variable and is selected as the best-split attribute, thus minimizing the information required to classify the data within the resulting partitions [30]. The mutual information between two random features determines how much information about one feature can be extracted using the knowledge of another feature. The mutual information gain `MIG(F\D)` can be calculated by subtracting the conditional entropy `E(D\F)` from the entropy `E(D)` of dataset D, for the given feature F, as explained by equation (1).

$$\texttt{MIG(F\D)} = \texttt{E(D)} - \texttt{E(D\F)} \tag{1}$$

where `MIG(F\D)` represents the final information gain of a feature F in dataset D, E(D) denotes the entropy of dataset D and is described in equation (2).

$$E(D) = -\sum_{i=1}^{t} p_i \log_2 p_i \tag{2}$$

TABLE 2. Comparative analysis of different feature selection techniques (FST).

FST Types	Strength	Weaknesses
Filter (MIG) [16], [17], [23], [28], [29], [30]	• Independent of ML classifiers • Computationally efficient. • Interpretation is simple • Faster than SFS	• Adopt univariate FST, therefore can't handle redundant features. • Lacks satisfactory prediction accuracy. • Don't bother about the biases of the classifiers. • May lead to overfitting issues.
Wrapper (SFS, Greedy search, GA, etc.) [19], [21], [22], [18], [30]	• The importance of a feature is measured based on its usefulness. • Avoid the biases. • Prediction accuracy is high. • Flexible and straightforward. • The GA-based FST avoids local optima as has the capability to jump towards the promising part of the search space. • Generates best optimal subset of feature. • Improve the classification performance • Overfitting issues are reduced. • Global optima is easily discovered.	• SFS is slower than MIG. • Computationally expensive as run the classifier repeatedly to assess the quality of the feature subset. • The feature included in the selected subset, can't be changed, hence leading to local optima in the case of SFS. • Computational cost is reasonable in GA-based FST because fitness function evaluation takes a little more time. • Sometimes premature convergence occurs. • Scalability issue arises with complexity.
Hybrid FST [22], [23], [41]	• Achieves both comparable accuracy to the wrapper and efficiency to the filter. • Less prone to over-fitting compared to wrapper. • Less computationally intensive as compared to wrapper. • Able to generate a stable model.	• Complexity issues arise because a small change in the data may result in different subsets of feature.

where 't' denotes the number of classes (benign and malignant). p_i determines the estimated probability of a given class (benign or malignant class) and it is calculated as in equation (3)

$$p_i = \frac{s_i}{S} \tag{3}$$

where s_i represents the total number of samples or instances that belongs to class i and S is the total number of samples in the dataset. Then the conditional entropy $\mathtt{E(D\backslash F)}$ of the dataset D after a split occurs for the feature F, can be estimated as in equation (4)

$$\mathtt{E(D\backslash F)} = -\sum_{b=1}^{n} \frac{S_b}{S} \sum_{i=1}^{t} p_{ib} \log_2 p_{ib} \tag{4}$$

where n states the total number of branches created after the split by the feature F on dataset D and p_{ib} explains the estimated probability of class i of branch b. So, the conditional probability can be summarized as the weighted sum of the entropy of each branch b after a split.

According to equation (1), the features are ranked according to their highest information gain. These features are considered for further processing by passing them to the SFS algorithm of wrapper method-I because, though the MIG FST is easy and simple, aggregating all the best features individually into a subset does not guarantee that the feature subsets are predictive for categorizing the websites as malicious or benign.

Sequential Forward Selection (SFS) Method: While MIG is widely used as a popular filter method to assess the relevance of features, it does have its own limitations. Particularly, MIG struggles with accurately distinguishing the attributes with a high number of distinct values, and it can lead to overfitting issues. Moreover, since IG selects features in a univariate manner, it may not effectively handle interdependencies among relevant features [35]. To address these limitations, the features from MIG are subjected to further evaluation and reduction using SFS from the wrapper FSM. This process aims to generate the best feature subset. SFS is a heuristic search technique that reduces an original d-dimensional feature space to a k-dimensional feature subspace where k<d. It iteratively adds variables to an empty set $Q_0 = \emptyset$ until the addition of additional features no longer improves the criterion or the objective function.

TABLE 3. Features selected from the MIG feature selection technique.

Feature selected from MIG	Feature Names
33	'NumDots', 'PathLevel', 'UrlLength', 'NumDash', 'NumDashInHostname', 'TildeSymbol', 'NumUnderscore', 'NumQueryComponents', 'NumAmpersand', 'NumNumericChars', 'IpAddress', 'DomainInSubdomains', 'HostnameLength', 'PathLength', 'QueryLength', 'DoubleSlashInPath', 'NumSensitiveWords', 'EmbeddedBrandName', 'PctExtHyperlinks', 'PctExtResourceUrls', 'InsecureForms', 'RelativeFormAction', 'AbnormalFormAction', 'PctNullSelfRedirectHyperlinks', 'FrequentDomainNameMismatch', 'RightClickDisabled', 'SubmitInfoToEmail', 'IframeOrFrame', 'UrlLengthRT', 'PctExtResourceUrlsRT', 'AbnormalExtFormActionR', 'ExtMetaScriptLinkRT', 'PctExtNullSelfRedirectHyperlinksRT'

In the proposed approach, the performance of the logistic regression (LR) in terms of accuracy is considered an objective function. The algorithm begins by calculating the objective function for the single, very first-best feature. Pairs of features are then formed by combining the remaining features with the best feature, and then the best feature pair is selected. This process continues sequentially, where the best features are incrementally added to the subset as $x^+ =$ arg max J(Q_k+x). The set Q_{k+1} is updated to $Q_k + x^+$, and the value of k is incremented. This procedure continues until the termination criteria are met. The overall feature selection process of MIG-SFFS is explained in Algorithm 1.

Algorithm 1 Hybrid Feature Selection Technique-1 (MIG-SFS)

Input: Dataset is depicted as D, the feature set is presented as $F = f_0, f_1, f_2, \ldots, f_n$, the size of F is n and the target feature set size is s

Output: the optimal subset of features

MIG_procedure (F):

1) The entropy for feature F is calculated by using equation (2).
2) The conditional entropy after the split for the dataset is estimated in equation (4)
3) The information gain for attribute F is computed by equation (1).
4) Similarly, the MIG of every feature is calculated and the best scored of features F are selected from the dataset D
5) best-scored selected features $F = f_0, f_1, f_2, \ldots, f_n$ pass to wrapper FST of SFS.

SFS_procedure(F):

Input: MIG feature set F, F' ← F, k = n

1) **For** i = 1 ← n - s
2) The sample data of F' is split into two parts as: x_train, x_test variables and x_train is trained with features filtered out from the MIG
3) Classification performance is computed with accuracy AC_k on x_test.
4) Add the best feature f ← arg max J $((F'_k + \{f_i\})), f_i \epsilon F'_k$
5) Update $F'_{k+1} \leftarrow F'_k + \{f_i\}$
6) k ← k + 1
7) end
8) return $F'_{k+1} \leftarrow f_x | x = 1, 2, \ldots, k, ; f_i \epsilon F'$, where k = (0, 1, 2, …, n - s)

During the selection process of the best attribute subset, the accuracy measurement of the different subsets is computed and summarized in Table 4. It becomes evident that the combination of features in the 30^{th} subset assessed by the LR classifier provides the highest accuracy of 93.72%. After the 30^{th} subset, the performance declines, as shown in Fig. 2. Consequently, the 30^{th} subset, which consists of a combination of 30 URL features, is selected as the most effective feature subset for training the ensemble classifier aimed at identifying suspicious URLs.

TABLE 4. Various subsets of features and the performance of feature subsets of SFS.

No. of features	Subset of features	Accuracy (%)
1	(32)	0.71325
2	(0, 32)	0.75675
3	(0, 24, 32)	0.79925
4	(0, 19, 24, 32)	0.819875
5	(0, 19, 23, 24, 32)	0.83025
6	(0, 1, 19, 23, 24, 32)	0.83575
.	.	.
.	.	.
29	(0, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 1...)	0.936625
30	(0, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 1...)	0.93725
31	(0, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 1...)	0.936875
32	(0, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 1...)	0.936375
33	(0, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 1...)	0.936

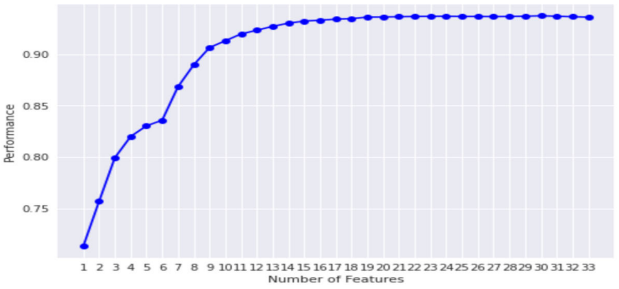

FIGURE 2. Accuracy measurement of different subsets of features.

2) HYBRID FEATURE SELECTION APPROACH-II (MIG-GA)

This section elaborates on the GA for obtaining the optimal feature subset, as the details of MIG are covered in Section C. The selected high-ranked features derived from MIG are passed into the genetic search algorithm. The algorithm aims to determine the most effective combination of features for evaluating the machine learning classifiers in the context of URL detection.

Genetic Optimization Algorithm: The primary objective of the proposed technique is to obtain the optimal solutions, especially the best feature subset, for the identification of suspicious URLs. This goal is achieved by utilizing metaheuristic algorithms, which are known for providing the most favorable solutions (i.e., an optimized one) within a finite time frame. The genetic algorithm is an evolutionary computation technique mainly used for searching. Its selection for relevant feature selection is justified by its rapid convergence and its utilization of diverse hyperparameters such as cross-over and mutation [36]. Within the genetic algorithm framework, the search starts with an initial population of features or individuals that represent the potential solutions for evaluating the classifiers [37]. These features undergo a repeated process of modification through GA's operators, where the fittest features or individuals are selected and recombined to create improved new features. These enhanced features are subsequently used in the following

iterations of the process. Each iteration is known as a generation within the evolution context. This iterative process continues until certain criteria are met. The termination criteria could be to reach a maximum number of iterations or generations or to reach a certain fitness value of the objective function. These two criteria ensure that the features within the dataset or population must attain an acceptable level of accuracy before the process concludes.

In the proposed approach, the assessment of features within the population is achieved through the use of the fitness function of GA. Generally, the fitness function for GA-based feature selection employs a classification performance measure, such as the accuracy of the machine learning model used. In this context, we have considered a decision tree classifier for fitness evaluation. As a result, the genetic algorithm-based feature selection is categorized under the wrapper approach. Then three vital GA operators—selection, crossover, and mutation—play a significant role in influencing the fitness value. The selection method involves selecting a pair of features or individuals with the highest fitness values for reproduction. In the crossover operation, the genes are swapped between two chromosomes to be replicated. The mutation process changes the gene, replacing its value with a randomly generated value. The overall GA procedure is repeated until it reaches either the maximum number of iterations or generations or when reaching a certain fitness value, as specified by the converging criteria for the proposed approach to predicting malicious URLs. In the context of malicious URL detection, the search space for the feature selection problem contains all possible subsets of features. Within a population, each feature subset is characterized by an individual or a chromosome. The number of genes that are present in the individual, or the chromosome, resembles the total number of features. The features in the individual are encoded with a binary value of '1' or '0'. The binary value '1' is set if the feature is selected. Otherwise, '0' is represented in Fig. 3.

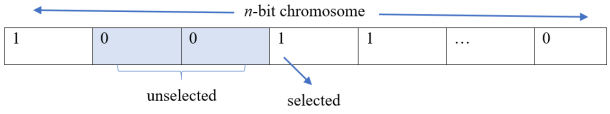

FIGURE 3. Representation of binary chromosome with n-dimensional features.

The procedure of optimal feature subset selection of hybrid approach MIG-GA is presented in Algorithm 2.

The parameters of GA with default values are represented in Table 5. The maximum number of generations is set at 50 as the primary terminating criteria of the algorithm. The number of populations represents the number of chromosomes. In the proposed approach, three different types of population have been set, such as 10, 20, and 30 variables or chromosomes. For determining the fitness value, DT with a hyperparameter of max_depth = 15 is set to obtain the fittest individual. In the proposed approach, due to its simplicity and efficiency, a tournament selection scheme is considered, which preserves diversity by giving a chance to all the chromosomes to be chosen. The size parameter is set to 3, which means three chromosomes are competing with each other. A single-point cross-over strategy with a probability of 0.8 is employed for obtaining the children from the parent. Similarly, bit-flip mutation (i.e., the mutation operation) with a probability of

Algorithm 2 Hybrid Feature Selection Technique-II (MIG-GA)

Input: Dataset D with the feature set X = $x_0, x_1, x_2, \ldots, x_n$, the feature set size is n and the target feature set size is s
Output: the optimal feature set

MIG_procedure (F):

1) The entropy for feature F is calculated by using equation (2).
2) The conditional entropy after the split for the dataset is estimated in equation (4)
3) The information gain for attribute F is computed by equation (1).
4) Similarly, the MIG of every feature is calculated and the high-scored MIG of feature F is selected from the dataset D
5) High-scored selected features F ← $\{f_0, f_1, f_2, \ldots, f_n\}$ pass to wrapper FST of GA.

GA_ procedure(F):

Input: IG feature set F, numPop ← F
Number of generation (numGen ← 50)
Numbers of populations (numPop) are set as 10,20 and 30 for three different experiments
Count ← 0
The sample data of F is split into two parts as: *x_train_mi*, *x_test_mi* variables and *x_train_mi* is trained with features filtered out from the MIG
Classification performance is computed with accuracy AC_k on *x_test_mi*.
Output: optimal feature set.

1) While (count < numGen)
2) {
3) For i ← 1 to numPop
4) {
5) testAccuracy ← getFitness (individual, *x_train_mi*, *x_test_mi*, *y_train*, *y_test*)
6) Parents [P(i), P(i+1)] ← selection_proc (numPop, size) //Tournament selection scheme
7) New_offspring[i] ← mate (P(i), P(i+1)) // single point cross over
8) M[i] ← Mutate (New_offspring [i]) // bit-flip mutation
9) }
10) Replace numPop with New_offspring$_i$, . . . , New_offspring$_{numPop}$
11) }

0.2 is applied to the chromosomes to maximize the diversity of the population and improve the quality of each population member for distinguishing the malignant URLs from the benign URLs [36].

TABLE 5. GA parameters.

Parameters	Value	Parameter	Value
Number of generations	50	Selection scheme	Tournament (size =3)
Number of populations	10, 20, 30	Selected feature	1
Fitness function	DT	Unselected feature	0
Cross over probability	0.8	Population type	Bit sequence
Mutation probability	0.2	Chromosome length	33

D. DETECTION PHASE

This section provides an explanation of the supervised ML algorithms used for evaluating the features obtained from both HFSTs. Two types of ensemble classifiers, namely bagging and boosting trees, along with one traditional machine learning algorithm, are considered for evaluation. The justification for utilizing ensemble estimators is their capacity to significantly enhance the model's performance by combining multiple weak base learners, either in a parallel or sequential manner. These learners are trained on different samples to collectively achieve the models' objectives. Throughout the experiments conducted on the proposed approach for the detection of malicious URLs, the XGBoost Classifier (XGBC) outperformed all the other classifiers. Due to this superior performance and also considering space limitations, we will only focus on explaining the XGBC among the boosting classifiers.

1) EXTREME GRADIENT BOOST CLASSIFIER (XGBC)

XGBoost is an open-source scalable end-to-end tree boosting system, also known as gradient boosted decision trees (GBDT), that provides parallel tree boosting and is widely used by data scientists to achieve efficiency, portability, flexibility, and more accuracy [9], [38]. In the proposed approach, XGBC, a tree-based ensemble method, yields an ensemble machine learning model using a decision tree as the sequential base learner. The XGB model builds the decision trees in a sequential manner so that each weak learner draws influence from the previous weak learner, and it minimizes the loss function by scaling back the errors of the base learner before it. The decision tree used in XGBC can be defined as in equation (5).

$$\sum_{k=1}^{K} f_k f_k \epsilon F, k = 1, 2, \ldots K. \tag{5}$$

where K denotes the number of decision trees and f_k represents the prediction of the tree. When all the trees are combined, the resultant prediction of trees can be stated as in equation (6).

$$R_1 = \sum_{k=1}^{K} f_k(x_i) \tag{6}$$

Here, R_i represents the one-dimensional result, $R_i \epsilon 0, 1$, benign URLs are represented as 0 and suspicious URLs are represented as 1 on the n-dimensional input vector $\vec{x} = x_1, x_2, \ldots, x_i, x_i$ shows independent feature vector of i^{th} data points and $(\hat{R}_i)$ is the sum of predictions from all the decision trees. Each leaf j of the decision tree is assigned a weight w_j and its value is computed by minimizing the objective function as in equation (3).

$$Obj = l + \Omega \tag{7}$$

where 'l' denotes the loss function and 'Omega' represents a regularization term penalizing the complexity of the model.

Hence the objective function can be redefined as in equation (8)

$$Obj = (-\frac{1}{N}\sum_{i=1}^{N} R_i \log(\hat{R}_i) + (1 - R_i)\log(1 - \hat{R}_l)) + (\gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} {w_j}^2) \tag{8}$$

where $(\hat{R}_i)$ represents the computed prediction of the proposed model on input $\vec{x}$, R_i denotes the actual value for input/output prediction $\vec{x}, R_i$. T shows the number of leaves on the DT and w_j is the score or value of a j^{th} leaf. The hyperparameters γ and λ are used to calculate the degree of regularization because λ reduces the risk of overfitting and γ penalizes each tree n for growing additional leaves. To achieve the objective of equation (4), equation (8) can be redefined as

$$\hat{R}_l^{(t)} = \sum_{i=1}^{N} l(R_i.\hat{R}_l^{(t-1)} + f_t(x_i)) + \sum_{i=1}^{t} \Omega(f_i) \tag{9}$$

where $f_t(x_i)$ expresses the current prediction, R_i, $(\hat{R}_l)^{(t-1)}$ denotes the t^{th} and the preceding steps and $\hat{R}_l^{(t)}$ represents model training at round t.

The optimal value of w_j is obtained by calculating the first and second-order gradients of the loss function, and a good tree structure is found by the greedy method, considering the best splitting point for adding new leaves. At each iteration, splitting occurs by the algorithm, and a leaf node is changed to an internal node. Let I denote the set of indexes of data points, and I_R and I_L represent the left and right trees, respectively. Gain (G) is calculated by subtracting the value of the previous leaf from the sum of the values of the left and right leaves, as in equation (10).

$$G = -\frac{1}{2}[\left[\frac{(\sum_{i\epsilon I_L} g_i)^2}{\sum_{i\epsilon I_L} h_i + \lambda} + \frac{(\sum_{i\epsilon I_R} g_i)^2}{\sum_{i\epsilon I_R} h_i + \lambda} - \frac{(\sum_{i\epsilon I} g_i)^2}{\sum_{i\epsilon I} h_i + \lambda}\right] - \gamma] \tag{10}$$

where g_i and h_i are first-order and second-order loss functions and γ is the regularization parameter. After obtaining the G value, to minimize the complexity of the tree, the tree pruning process is adopted. In the tree pruning process, if the value of G>0, the new leaves are kept. Otherwise, the current leaves are deleted and other new leaves will be trained. The proposed approach explains the fundamentals of XGBC. A detailed description of the split finding algorithm, weighted quantile sketch, etc., can be found in [24] and [39].

E. PERFORMANCE MEASURES

To measure the effectiveness of the ensemble classifiers, some statistical evaluation metrics are given in equations (11) to (14) to detect malicious URLs. All the statistical measurements are calculated from the confusion matrix.

$$ClassificationAccuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$Recall \quad or \quad DetectionRate(DR) = \frac{TP}{TP + FN} \tag{12}$$

$$Precision(P) = \frac{TP}{TP + FN} \tag{13}$$

$$F1{-}Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{14}$$

Here, TP, FP, TN, and FN are True Positive, False Positive, True Negative and False Negative, respectively. In this research study, all malicious URLs are treated as positive, benign URLs are stated as negative, and the values assigned for both the positive and negative classes are '1' and '0', respectively. TP defines the number of malicious websites correctly classified as malicious from the total malignant and benign websites. FP denotes the benign websites misclassified as malicious from the total number of safe websites. TN represents the number of safe websites correctly classified as safe out of the total number of benign websites. FN states the number of malicious websites misclassified as benign websites from the total number of malicious websites. AUC- Area under ROC (Receiver Operator Characteristics) Curve is a binary classification evaluation metric that plots the TPR (True Positive Rate) against the FPR (False Positive Rate) in y and x axis, respectively, to measure the ability of a classifier for distinguishing malicious URL class from benign URL classes. The higher the value of AUC (If AUC=1) of any classifier determines the classifier can perfectly distinguish the malicious URL from the benign URL as TPR is 100% and FPR is 0 or no false positive [18]. When the AUC equals 0, the estimator will predict all the malicious classes as benign and benign classes as malicious.

IV. RESULT AND ANALYSIS

To thoroughly evaluate the proposed approach for detecting suspicious URLs, a series of comprehensive experiments have been conducted. The results from these experiments are analyzed and discussed in this section. In experiment 1, the primary subsets of features are derived using the MIG filter feature selection technique. These subsets are then utilized as input for the forward feature selection technique of wrapper methods, which generates the most relevant and significant secondary feature subsets. As part of the second experiment, the main feature subsets from MIG are utilized with the wrapper method's GA-based search optimization method. This integration aims to produce an optimal feature subset. The optimal features generated from both experiments 1 and 2 are evaluated using ensemble machine learning classifiers, namely, XGBC, Gradient boost (GB), Adaboost, Bagging and kNN, in yet another experiment. All the conducted experiments are executed within the Jupyter editor notebook of the Python environment. Additionally, graphs depicting the results are created using Python's Matplotlib data visualization package.

A. EVALUATION-1 (MIG-SFS)

In experiment 1, the performance evaluation of the proposed approach for predicting malicious URLs is conducted. The MIG of the filter-based FST is applied to the dataset. This process provides a high-scored feature order list consisting of 33 features, as outlined in Table 3. For improved computational results, we select the parameter as top percentile-n and the value of 'n' chosen with a value of 70, resulting in the selection of 33 important features out of the initial 49 features. These selected features are then subjected to further processing in the SFS of the wrapper FST. In the SFS process, the performance evaluation of each feature subset is done using the classifier used for measuring the performance of each feature subset, the LR classifier, with the parameter max_iter set to 1500. The SFS process results in a feature subset comprising the 30 most crucial features, which are then utilized for training the supervised machine learning classifiers, such as XGBC, GB, Adaboost, Bagging, and k-NN. The performance of each of the classifiers is presented in Table 6, indicating that XGBC demonstrates the highest accuracy of 98%, surpassing the other classifiers. An AUC

TABLE 6. Performance of different classifiers of evaluation-1.

Classifiers	CA (%)	Precision (%)	DR (%)	F1-score (%)
kNN	95.7	95.0	96.0	96.0
Bagging	97.4	98.0	97.0	97.0
GradientBoost	97.8	97.0	98.0	98.0
XBoost	98.0	97.0	98.0	97.0

graph is utilized to validate the effectiveness and accuracy of the proposed approach. AUC plots FPR and TPR on the x-axis and y-axis, respectively. The location of a point on the graph serves as an indicator of how accurate a classifier is. A point at the top-left corner, with coordinates at (0,1), indicates a TPR of 100% and almost zero FPR, representing a perfect classifier. In Fig. 4, the AUC curves for different classifiers, such as k-NN, Bagging, Adaboost, Gboost, and XGBC, are shown for detecting malicious URLs. From the graph, it is evident that the XGBC model exhibits the highest

classification performance, achieving an accuracy of 98%. This result indicates that the XGBC classifier outperforms other classifiers in distinguishing malicious URLs from benign ones.

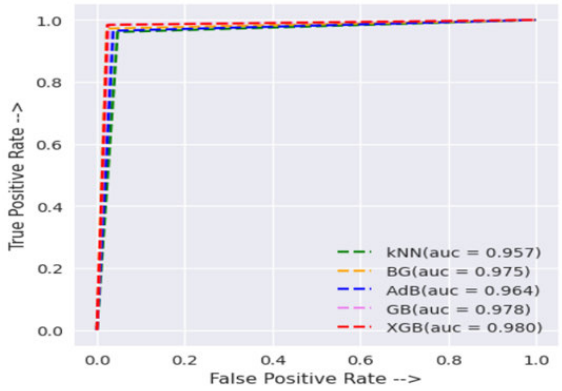

FIGURE 4. Performance of different classifiers in AUC graph.

B. EVALUATION-2 (MIG-GA)

In experiment II, the proposed approach (HFST-II) is evaluated by first applying the MIG of filter-based FST on the dataset to obtain a high-scored feature subset, as detailed in Table 2. Out of 49 initial features, 33 are selected using MIG and then input to the GA to obtain the optimal feature subsets. The decision tree classification algorithm is used on all 33 features from MIG in the GA process to get a base level of accuracy for all populations with 10, 20, or 30 chromosomes. Next, the GA is applied to the dataset of 33 features to identify a feature subset that achieves better accuracy than the baseline accuracy. Detailed descriptions of the different parameters and notations used in the GA process are represented in Table 5.

The results of machine learning classifiers for each population are represented in Tables 7, 8, 9, and 10. For each population, a feature subset is generated, along with validation accuracy and test classification accuracy. To visualize how these accuracy values change from the worst feature subset to the best subset obtained by the GA, cubic spline interpolation and continuum graphs from mathematics are employed. Interpolation is chosen due to its accurate estimation of values between two points on a curve, offering insights into accuracy changes across subsets. The testing accuracy is approachable to validation accuracy, and it is 98% in population number 20; hence, the selected feature subset can overcome the overfitting issues.

In Fig. 5a, the blue line represents the cubic spline interpolation, while the red dots represent the data points. From all the populations, it is evident that the XGBC classifier achieves the highest accuracy in population number 20, as depicted in Fig. 8. The optimal feature subset generated from this population, which is 26 features, is considered the best optimal feature subset in our proposed approach for predicting suspicious URLs, as it is outlined in Table 10.

Figures of Cubic spline interpolation and Continuum graph to plot test and validation set accuracy for POP-10 are as shown in Fig.5, 6 and 7.

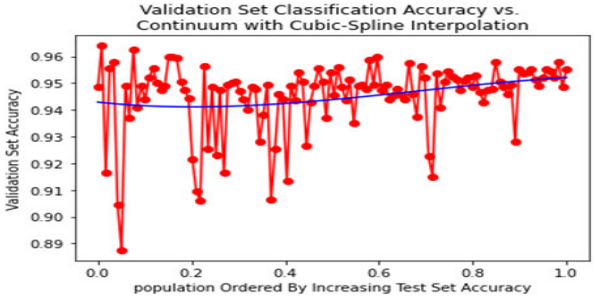

FIGURE 5. Cubic spline interpolation representation of Test and validation set accuracy for population number 10.

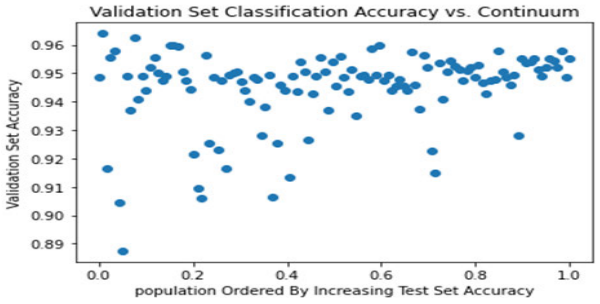

FIGURE 6. Contunuum representation of validation set accuracy for population number 10

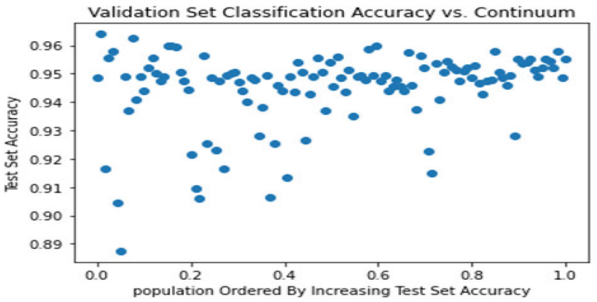

FIGURE 7. Contunuum representation of test set accuracy for population number 10.

The optimal features obtained from the GA are assessed using classifiers including k-NN, Bagging, Adaboost, Gboost, and XGBC for predicting suspicious URLs. The performance metrics for those different classifiers are presented in Tables 6, 7, 8, and 9. Notably, both the classification accuracy and precision measures of XGBC are higher (98.3%, 99%) compared to other classifiers, such as k-NN (94%, 95%), Bagging (96%, 98%), Adaboost (96%, 96%), and Gboost (97.3%, 97%). To validate the effectiveness of the proposed MIG-GA approach, an AUC graph is plotted using both the FPR and TPR and is represented in Figs. 6, 8, and 10.

Figures of cubic spline interpolation and contunuum graph to plot test and validation set accuracy for POP-20 are as shown in Fig.9, 10 and 11

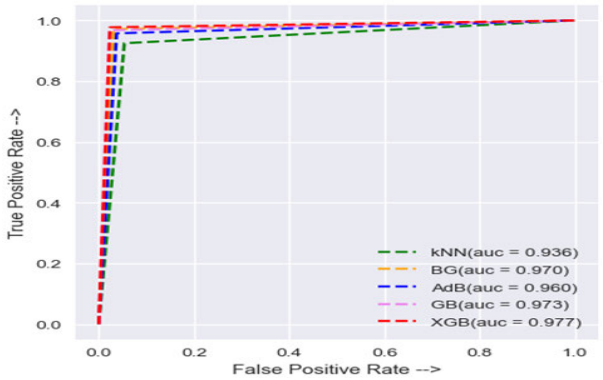

FIGURE 8. AUC graph for accuracy of different classifiers for population number 10.

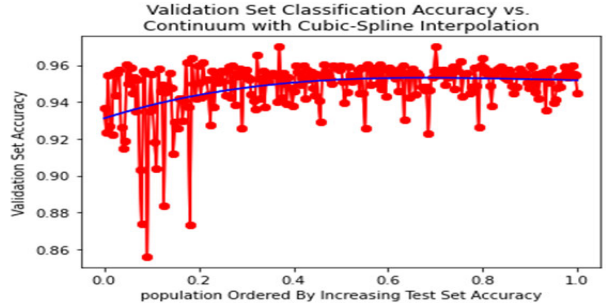

FIGURE 9. Cubic spline interpolation representation of test and validation set accuracy for population number 20.

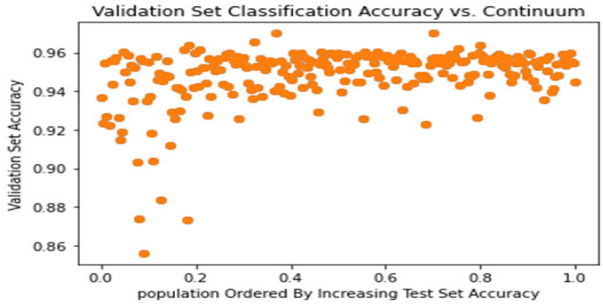

FIGURE 10. Contunuum representation of validation set accuracy for population number 20.

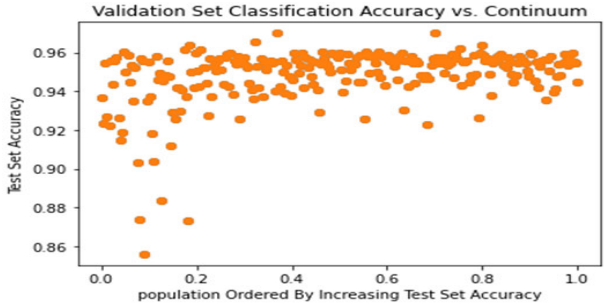

FIGURE 11. Contunuum representation of test set accuracy for population number 20.

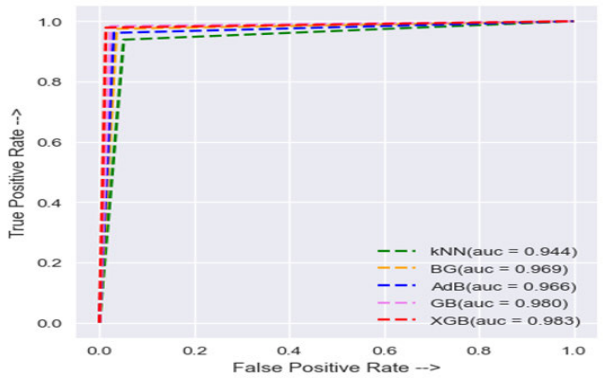

FIGURE 12. AUC graph for accuracy of different classifiers for population number 20.

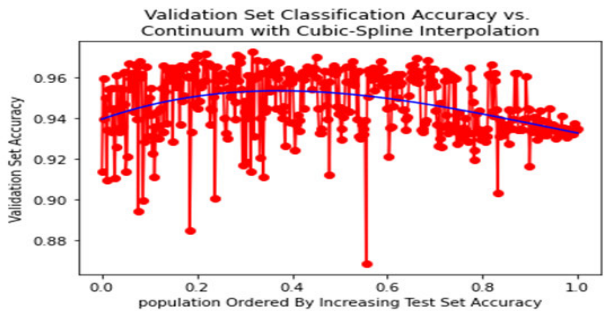

FIGURE 13. Cubic spline interpolation representation of test and validation set accuracy for population number 30.

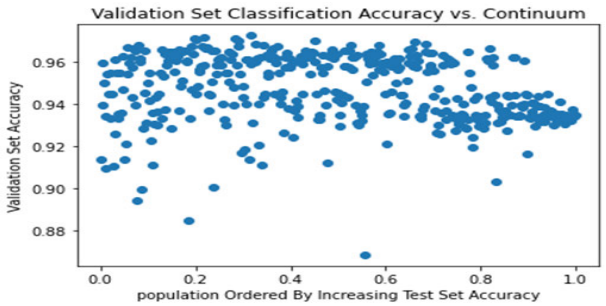

FIGURE 14. Contunuum representation of validation set accuracy for population number 30.

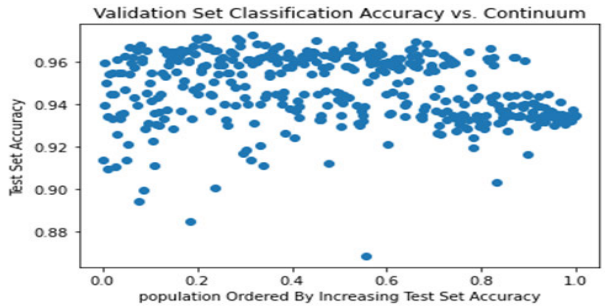

FIGURE 15. Contunuum representation of test set accuracy for population number 30.

Figures of cubic spline interpolation and contunuum graph to plot test and validation set accuracy for POP-30 are as shown in Fig.13, 14 and 15.

To validate the robustness of the MIG-GA hybrid FSM, three trials were conducted using population numbers 10, 20, and 30. In the second trial (pop-20), an optimal feature set consisting of 26 URL features was obtained when evaluated by the machine learning classifiers, resulting in the best accuracy. Tables 7, 8, 9, and 10 show different performance measures of the classifiers in each population, with no

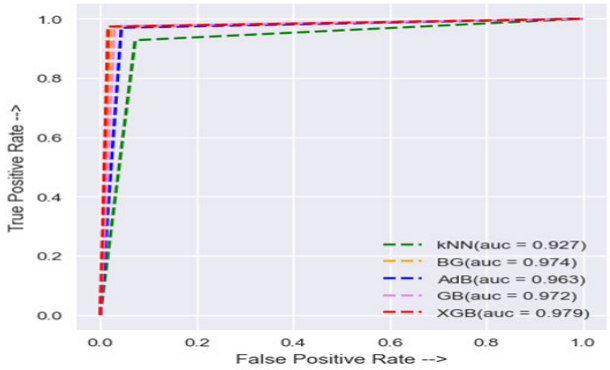

FIGURE 16. AUC graph for accuracy of different classifiers for population number 30.

features chosen and a maximum number of generations. These measures include accuracy, precision, detection rate, and F1-score. From these tables, it is evident that in pop-20, the accuracy, precision, detection rate, and F1-score of XGBC are 98.3%, 99%, 98%, and 98%, respectively, in pop-20. This indicates that the URL features selected from the pop-20 of the GA-based search technique are highly relevant, informative, and significant in detecting suspicious URLs.

TABLE 7. Classification accuracy (%) of different classifiers in pop-10, pop-20 and pop-30 of MIG-GA FST.

No. of generation	Population Size	Features subset	k-NN	Bagging	Adaboost	Gboost	XGBC
50	10	22	93.6	97.0	96.0	97.3	98.0
50	20	26	94.4	97.0	96.5	98.0	98.3
50	30	25	93.0	97.4	96.3	97.2	98.0

TABLE 8. Precision (%) measure of different classifiers in pop-10, pop-20 and pop-30 of MIG-GA FST.

No. of generation	Population Size	Features subset	k-NN	Bagging	Adaboost	Gboost	XGBC
50	10	22	95.0	98.0	96.0	97.0	98.0
50	20	26	95.0	96.0	96.0	98.0	99.0
50	30	25	93.0	98.0	96.0	97.0	98.0

TABLE 9. DR (%) measure of different classifiers in pop-10, pop-20 and pop-30 of MIG-GA FST.

No. of generation	Population Size	Features subset	k-NN	Bagging	Adaboost	Gboost	XGBC
50	10	22	93.0	97.0	96.0	98.0	98.0
50	20	26	94.0	98.0	96.0	98.0	98.0
50	30	25	93.0	97.0	97.0	98.0	97.0

TABLE 10. F1-score (%) measure of different classifiers in pop-10,20 and 30 of MIG-GA FST.

No. of generation	Population Size	Features subset	k-NN	Bagging	Adaboost	Gboost	XGBC
50	10	22	94.0	97.0	96.0	98.0	98.0
50	20	26	94.0	97.0	96.0	98.0	98.0
50	30	25	93.0	97.0	96.0	97.0	98.0

C. SIGNIFICANCE OF FEATURES OBTAINED FROM POPULATION NUMBER-20 OF MIG-GA BASED FST

To verify whether the features obtained from population number 20 of MIG-GA based FST are significant and informative in enhancing the effectiveness of detection of suspicious URLs, a Chi-Square statistical test is conducted. The justification for using the Chi-Square test in classification tasks is that it can compute the degree of independence between a pair of categorical variables [45]. The p-value (probability value) of each feature is calculated, and it is shown in Fig. 17. From Fig. 17, it is clear that the p-value of the selected features is much less than the threshold, significance level, $\alpha = 0.05$; hence, the selected features are more informative and statistically significant.

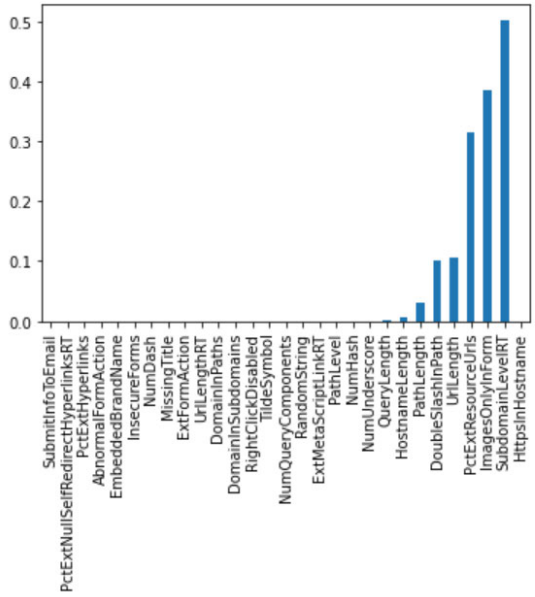

FIGURE 17. p-values of selected features.

D. COMPARISON OF THE MIG-GA WITH THE MIG-SFS HYBRID FST

This section compares the performance of two hybrid FSTs, MIG-GA and MIG-FFS, along with filter and wrapper methods for detecting malicious URLs. The classifiers k-NN, Bagging, Adaboost, Gboost, and XGBC evaluate the most significant and informative features produced by these approaches. The XGBC classifier consistently outperforms the other classifiers in terms of malicious URLs. As observed in all the experiments, we have created a comparison table showcasing the different FSTs used in the proposed approach and the performance of XGBC in terms of accuracy and precision measures. From Table 11, it is evident that the hybrid FST, the MIG-GA hybrid approach, outperforms all other FSTs in terms of accuracy and precision measures.

E. COMPARATIVE STUDY OF THE PROPOSED APPROACH WITH THE EXISTING STUDIES

The performance of ensemble estimators, considering the hybrid feature selection technique that combines both filter

TABLE 11. Comparison of MIG-GA with other FST.

FST	MIG	SFFS	GA	MIG-SFFS	MIG-GA
No. of Features	33	27	26	33	26
Accuracy (%)	97	97.5	98	98	98.3
Precision (%)	96	97	98	98	99

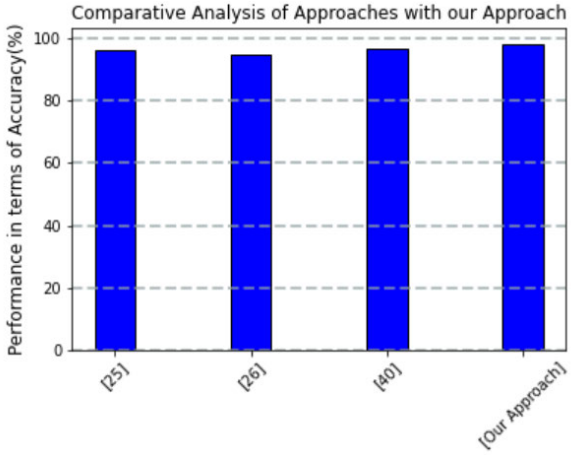

FIGURE 18. Comparison graph of the proposed approach with the existing research studies.

and wrapper feature selection methods, is compared and discussed in relation to other current research articles focused on identifying suspicious URLs in general-purpose computing systems where computational resource consumption is not an issue. In contrast, the proposed approach is oriented toward resource-constrained IoT devices. It meets the requirements of IoT devices, like utilizing a small representative data subset for maintaining computational resources and power consumption within a reasonable range. As shown in Fig. 18, it is evident that the proposed approach, based on MIG-GA based hybrid FST, produces the best optimal subset of features when evaluated by the ensemble estimators, and it outperforms all other existing approaches. Recent studies [25], [26], employed the GA-based optimization algorithm for selecting optimal features for malicious URL detection in their research, achieving classification performance of 96.45% and 95%, respectively. Additionally, the study of [40] introduced a PSO-based optimization technique to obtain substantial features and got a computation result of 96.75% accuracy and 98.15% precision, which is comparatively less than the proposed approach. The existing research [10] and [29] focused on phishing detection using a lightweight dataset and RF classifier in an IoT context and reported model performance as 99% which is the same as the proposed approach. The response time of the approach [10] is 51.5 ms but in our approach, it takes 1.79 s. Our approach includes page-content-based features of URLs along with lexical features, whereas they have considered only the lexical features of URLs, from which more information about a suspicious URL cannot always be drawn. Additionally, the XGB ensemble estimator performs better than the conventional RF classifier at addressing issues like bias-variance trade-offs and overfitting issues. Hence, the proposed hybrid approach can be claimed to be more suitable for IoT devices due to the use of a small feature set and XGB model, where the former feature leads to the requirement of fewer computational resources and the latter will guarantee an accurate prediction, though it may consume a little extra power. The performance of models in terms of the accuracy of existing work is compared with the proposed work, as shown in Fig. 18.

V. CONCLUSION

The artifacts of suspicious web pages are constantly changing by the intruder, so to prevent naïve users from browsing these websites, the identification of malignant URLs is a challenging task for the researcher as well as a vital part of a web security defense system. Therefore, selecting the optimal feature subset to improve the prediction of suspicious URLs is significant. In the proposed method, we described two hybrid FSTs, such as MIG-SFS and MIG-GA, to select the distinct and effective feature subset for classifying the web page as malignant or safe in IoT environments using a publicly available dataset. Additionally, we perform some hyperparameter tuning to improve classification accuracy. From the experimental results, it is found that the hybrid FST containing evolutionary optimization genetic algorithm procedure produces predictive features that are more informative than the features generated from SFS methods. Population number 20 of GA generates a feature subset of 26 features that, when evaluated against the ensemble classifiers, gives the best accuracy of 98.3% and precision of 99% among the other subsets of features generated from population numbers 10 and 30 also from MIG-SFS methods. Among all the classifiers, the XGBC estimator performs the best, followed by GBoost, Adaboost, Bagging, and k-NN. **Limitation:** First, the proposed approach has not been tested in a real-time system in which an enormous volume of data movement occurs every second. Secondly, to obtain the most significant features, only one optimization method has been explored, and this technique is not compared with other optimization methods, which might have a greater impact on the classification of malicious URLs. Third, four objectives have been considered: accuracy of the model, precision, predicting FPR, and reduction of dimensionality. Computational time and complexity are the two vital parameters to be explored in suspicious URL detection. Fourth, filtering out the different attacking patterns like phishing and spam from URLs has not been considered. Fifth: The approach presented in this research is suitable for IoT devices for detecting malicious URLs but has yet to be implemented in an IoT environment.

In the future, we plan to design a model capable of implementing different feature selection optimization techniques to generate significant optimal feature subsets. By evaluating these optimal feature subsets, we aim to improve accuracy, decrease computational time and complexity, and also be able to identify different types of attacks within URLs.

ACKNOWLEDGMENT

This study is partially supported by funding from Prince Sattam bin Abdulaziz University, project number (PSAU/2024/R/1445).

REFERENCES

[1] M. Alsaedi, F. Ghaleb, F. Saeed, J. Ahmad, and M. Alasli, "Cyber threat intelligence-based malicious URL detection model using ensemble learning," *Sensors*, vol. 22, no. 9, p. 3373, Apr. 2022, doi: 10.3390/s22093373.

[2] A. Petrocyan. *Number of Internet and Social Media Users Worldwide As of April 2023*. Accessed: May 20, 2023. [Online]. Available: https://www.statista.com/statistics/617136/digital-population-worldwide

[3] N. Aslam, I. U. Khan, S. Mirza, A. AlOwayed, F. M. Anis, R. M. Aljuaid, and R. Baageel, "Interpretable machine learning models for malicious domains detection using explainable artificial intelligence (XAI)," *Sustainability*, vol. 14, no. 12, p. 7375, Jun. 2022, doi: 10.3390/su14127375.

[4] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," 2017, *arXiv:1701.07179*.

[5] F. O. Catak, K. Sahinbas, and V. Dörtkardeş, "Malicious URL detection using machine learning," in *Artificial Intelligence Paradigms for Smart Cyber-Physical Systems*. Hershey, PA, USA: IGI Global, 2021, pp. 160–180.

[6] S. Cook. *Malware Statistics and Facts for 2023*. Accessed: May 20, 2023. [Online]. Available: https://www.comparitech.com/antivirus/malware-statistics-facts/

[7] C. Jones. *50 Web Security Stats You Should Know in 2023*. Accessed: May 22, 2023. [Online]. Available: https://expertinsights.com/insights/50-web-security-stats-you-should-know/

[8] IBM. *IBM Security X-Force Threat Intelligence Index 2023*. Accessed: May 25, 2023. [Online]. Available: https://www.ibm.com/reports/threat-intelligence#320478

[9] SonicWall. *SonicWall Cyber Threat Report 2023*. Accessed: May 20, 2023. [Online]. Available: https://www.sonicwall.com/2023-cyber-threat-report/

[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, doi: 10.1016/j.comcom.2021.04.023.

[11] H. B. Kazemian and S. Ahmed, "Comparisons of machine learning techniques for detecting malicious webpages," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1166–1177, Feb. 2015, doi: 10.1016/j.eswa.2014.08.046.

[12] S. Mohanty, A. A. Acharya, L. Sahu, and S. K. Mohapatra, "Hazard identification and detection using machine learning approach," in *Proc. 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Madurai, India, May 2020, pp. 1239–1244, doi: 10.1109/ICICCS48265.2020.9121048.

[13] L. Sahu, S. Mohanty, S. K. Mohapatra, and A. A. Acharya, "Malignant web sites recognition utilizing distinctive machine learning techniques," in *Proc. Comput. Netw., Big Data IoT*, vol. 66. Singapore: Springer, 2020, pp. 497–506, doi: 10.1007/978-981-16-0965-7_39.

[14] S. Mohanty, A. A. Acharya, and L. Sahu, "Improving suspicious URL detection through ensemble machine learning techniques," in *Society 5.0 and the Future of Emerging Computational Technologies*. Boca Raton, FL, USA: CRC Press, 2022, ch. 12, pp. 229–248.

[15] S. Mohanty and A. A. Acharya, "MFBFST: Building a stable ensemble learning model using multivariate filter-based feature selection technique for detection of suspicious URL," *Proc. Comput. Sci.*, vol. 218, pp. 1668–1681, Jan. 2023, doi: 10.1016/j.procs.2023.01.145.

[16] 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.*, Amman, Jordan, Dec. 2014, pp. 125–132, doi: 10.1109/ACSAT.2014.29.

[17] F. Thabtah and N. Abdelhamid, "Deriving correlated sets of website features for phishing detection: A computational intelligence approach," *J. Inf. Knowl. Manage.*, vol. 15, no. 4, Dec. 2016, Art. no. 1650042, doi: 10.1142/s0219649216500428.

[18] W. Ali and S. Malebary, "Particle swarm optimization-based feature weighting for improving intelligent phishing website detection," *IEEE Access*, vol. 8, pp. 116766–116780, 2020, doi: 10.1109/ACCESS.2020.3003569.

[19] W. Ali and A. A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," *IET Inf. Secur.*, vol. 13, no. 6, pp. 659–669, Nov. 2019, doi: 10.1049/iet-ifs.2019.0006.

[20] W. Ali, "Phishing website detection based on supervised machine learning with wrapper features selection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 9, pp. 72–78, 2017.

[21] M. Khonji, A. Jones, and Y. Iraqi, "An empirical evaluation for feature selection methods in phishing email classification," *Int. J. Comput. Syst. Sci. Eng.*, vol. 28, no. 1, pp. 37–51, 2013.

[22] M. H. Kamarudin, C. Maple, and T. Watson, "Hybrid feature selection technique for intrusion detection system," *Int. J. High Perform. Comput. Netw.*, vol. 13, no. 2, p. 232, Jan. 29, 2019, doi: 10.1504/ijhpcn.2019.097503.

[23] E. Alothali, K. Hayawi, and H. Alashwal, "Hybrid feature selection approach to identify optimal features of profile metadata to detect social bots in Twitter," *Social Netw. Anal. Mining*, vol. 11, no. 1, pp. 1–15, Sep. 2021, doi: 10.1007/s13278-021-00786-4.

[24] F. M. M. Mokbal, W. Dan, W. Xiaoxi, Z. Wenbin, and F. Lihua, "XGBXSS: An extreme gradient boosting detection framework for cross-site scripting attacks based on hybrid feature selection approach and parameters optimization," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102813, doi: 10.1016/j.jisa.2021.102813.

[25] S.-J. Bu and H.-J. Kim, "Optimized URL feature selection based on genetic-algorithm-embedded deep learning for phishing website detection," *Electronics*, vol. 11, no. 7, p. 1090, Mar. 2022, doi: 10.3390/electronics11071090.

[26] M. T. Suleman and S. M. Awan, "Optimization of URL-based phishing websites detection through genetic algorithms," *Autom. Control Comput. Sci.*, vol. 53, no. 4, pp. 333–341, Sep. 2019, doi: 10.3103/s0146411619040102.

[27] L. Choon, (2018), "Phishing dataset for machine learning: Feature evaluation," *Mendeley Data, V1*, doi: 10.17632/h3cgnj8hft.1. Accessed: Jun. 10, 2022. [Online]. Available: https://www.kaggle.com/shashwatwork/phishing-datasetfor-machine-learning

[28] M. Aydin, I. Butun, K. Bicakci, and N. Baykal, "Using attribute-based feature selection approaches and machine learning algorithms for detecting fraudulent website URLs," in *Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Las Vegas, NV, USA, Jan. 2020, pp. 0774–0779, doi: 10.1109/CCWC47524.2020.9031125.

[29] L. Bustio-Martínez, M. A. Álvarez-Carmona, V. Herrera-Semenets, C. Feregrino-Uribe, and R. Cumplido, "A lightweight data representation for phishing URLs detection in IoT environments," *Inf. Sci.*, vol. 603, pp. 42–59, Jul. 2022, doi: 10.1016/j.ins.2022.04.059.

[30] D. R. Patil and J. B. Patil, "Malicious web pages detection using feature selection techniques and machine learning," *Int. J. High Perform. Comput. Netw.*, vol. 14, no. 4, p. 473, Sep. 2019, doi: 10.1504/ijhpcn.2019.102355.

[31] K. D. Rajab, "New hybrid features selection method: A case study on websites phishing," *Secur. Commun. Netw.*, vol. 2017, pp. 1–10, Mar. 2017, doi: 10.1155/2017/9838169.

[32] S. C. Yusta, "Different metaheuristic strategies to solve the feature selection problem," *Pattern Recognit. Lett.*, vol. 30, no. 5, pp. 525–534, Apr. 2009, doi: 10.1016/j.patrec.2008.11.012.

[33] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[34] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021, doi: 10.1109/ACCESS.2021.3056407.

[35] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," in *Data Classification: Algorithms and Applications*, vol. 53. Boca Raton, FL, USA: CRC Press, 2014, p. 37.

[36] A. K. Shukla, P. Singh, and M. Vardhan, "A new hybrid feature subset selection framework based on binary genetic algorithm and information theory," *Int. J. Comput. Intell. Appl.*, vol. 18, no. 3, Sep. 2019, Art. no. 1950020, doi: 10.1142/s1469026819500202.

[37] U. S. Chanu, K. J. Singh, and Y. J. Chanu, "A dynamic feature selection technique to detect DDoS attack," *J. Inf. Secur. Appl.*, vol. 74, May 2023, Art. no. 103445, doi: 10.1016/j.jisa.2023.103445.

[38] M. Mittendorf, U. D. Nielsen, and H. B. Bingham, "Data-driven prediction of added-wave resistance on ships in oblique waves—A comparison between tree-based ensemble methods and artificial neural networks," *Appl. Ocean Res.*, vol. 118, Jan. 2022, Art. no. 102964, doi: 10.1016/j.apor.2021.102964.

[39] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[40] W. Bo, Z. B. Fang, L. X. Wei, Z. F. Cheng, and Z. X. Hua, "Malicious URLs detection based on a novel optimization algorithm," *IEICE Trans. Inf. Syst.*, vol. 104, no. 4, pp. 513–516, Apr. 2021, doi: 10.1587/transinf.2020edl8147.

[41] D. Jain and V. Singh, "Feature selection and classification systems for chronic disease prediction: A review," *Egyptian Informat. J.*, vol. 19, no. 3, pp. 179–189, Nov. 2018, doi: 10.1016/j.eij.2018.03.002.
[42] T. Gaber, A. Tharwat, V. Snasel, and A. Hassanien, "Plant identification: Two dimensional-based vs. One dimensional-based feature extraction methods," in *Proc. 10th Int. Conf. Soft Comput. Models Ind. Environ. Appl.* Switzerland: Springer, May 2015, pp. 375–385, doi: 10.1007/978-3-319-19719-7_33.
[43] T. Gaber, A. El-Ghamry, and A. E. Hassanien, "Injection attack detection using machine learning for smart IoT applications," *Phys. Commun.*, vol. 52, Jun. 2022, Art. no. 101685, doi: 10.1016/j.phycom.2022.101685.
[44] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "Phishing website detection from URLs using classical machine learning ANN model," in *Proc. Int. Conf. Secur. Privacy Commun. Syst.* Cham, Switzerland: Springer, Nov. 2021, pp. 509–523, doi: 10.1007/978-3-030-90022-9_28.
[45] R. Singhal and R. Rana, "Chi-square test and its application in hypothesis testing," *J. Pract. Cardiovascular Sci.*, vol. 1, no. 1, pp. 69–71, 2015.

SANJUKTA MOHANTY received the M.Tech. degree in computer science and engineering from KIIT Deemed to be University, Bhubaneswar, Odisha, India, in 2011, where she is currently pursuing the Ph.D. degree in computer science and engineering. Since 2011, she has been an Assistant Professor with Odisha University of Technology and Research (OUTR), Odisha. She is the author of more than 25 articles. Her research interests include computer architecture, web technology, cyber security, artificial intelligence, data mining, machine learning, data science, big data analytics, and the IoT. She is a reviewer of many conferences and journals.

ARUP ABHINNA ACHARYA received the B.Tech. degree in computer science and engineering from Utkal University, Odisha, India, and the M.Tech. and Ph.D. degrees in computer science and engineering from KIIT Deemed to be University, Bhubaneswar, Odisha, in 2006 and 2016, respectively. He is currently an Associate Professor and the Dean of the School of Computer Engineering, KIIT Deemed to be University. Apart from teaching he holds many administrative positions in the University, such as the Program Head of Information Technology, the Dean of Academics, and the Senior Superintendent of Hostels to name a few. There are more than 50 number of publications are there to his credit in many international and national level journals and proceedings. He has 20 years of experience in teaching undergraduate and postgraduate courses. His research interests include object-oriented systems, software testing, software cost estimation, security testing, data mining, and data analytics. He is a member of ISTE and IET and a reviewer of many conferences and journals.

TAREK GABER received the Ph.D. degree in computer science (information security) from The University of Manchester, Manchester, U.K., in 2012. He was a Postdoctoral Researcher with the Faculty of Electrical Engineering and Computer Science, VSB—Technical University of Ostrava, Ostrava, Czech Republic. He is currently a Senior Lecturer with the University of Salford, U.K., and a Professor with Suez Canal University, Egypt. He has worked in many universities, including the Faculty of Computers and Informatics, Suez Canal University; the Faculty of Computers and Information Sciences, Ain Shams University; and the School of Computer Science, The University of Manchester. He has more than 80 publications in international journals, conferences, and book chapters. He has successfully supervised many M.Sc. (by Research) and Ph.D. students. He is supervising M.Sc. and Ph.D. students in information security and machine learning. In addition, he has published five edited books. His research interests include cybersecurity, machine learning, the Internet of Things, and biometric authentication. He is a member of the Scientific Research Group in Egypt (SRGE). He has served as the co-chair and a PC member for many international conferences. He reviewed many scientific papers and participated in many scientific events (national/international conferences and workshops). He also served as the Lead Guest Editor in many SCI international journals, including *Applied Sciences*, *Wireless Communications and Mobile Computing*, *Sustainability*, and *Electronics*. He served as a keynote speaker at many international conferences.

NAMITA PANDA received the B.Tech. degree in computer science and engineering from BPUT, Odisha, India, in 2003, and the M.Tech. and Ph.D. degrees in computer science and engineering from KIIT Deemed to be University, Bhubaneswar, Odisha, in 2020. She is currently an Assistant Professor with the School of Computer Engineering, KIIT Deemed to be University. There are more than 25 number of publications are there to her credit in many international and national level journals and proceedings. She has 18 years of experience in teaching undergraduate and postgraduate courses. Her research interests include object-oriented systems, software testing, security testing, data mining, operating systems, and computer architecture. She is a member of ISTE and a reviewer of many conferences and journals.

ESRAA ELDESOUKY received the M.Sc. degree in computer science and technology from Hunan University, in 2011, and the Ph.D. degree from the College of Information Science and Engineering, Hunan University, in 2015. She is currently an Assistant Professor with the College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, and an Associate Professor with Suez Canal University, Egypt. Her current research interests include vehicular networks, wireless sensor networks, machine learning, and optimization algorithms. She served as a Keynote Speaker for Saudi International Telecommunications Conference, Riyadh International Convention and Exhibition Center, Riyadh, Saudi Arabia.

IBRAHIM A. HAMEED (Senior Member, IEEE) received the first Ph.D. degree in industrial systems and information engineering from Korea University, Seoul, South Korea, and the second Ph.D. degree in mechanical engineering from Aarhus University, Aarhus, Denmark. He is currently a Professor with the Department of ICT and Natural Sciences, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology (NTNU), Norway. He has authored more than 200 journals and conference papers. His current research interests include artificial intelligence, machine learning, optimization, and robotics. He is a Secretary of the IEEE Norway Chapter and the Elected Chair of the IEEE Computational Intelligence Society (CIS) Norway Section.