frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 21 January 2026
DOI 10.3389/fcomp.2025.1687867

OPEN ACCESS

EDITED BY
Zainab Loukil,
University of Gloucestershire, United Kingdom

REVIEWED BY
Faisal Ahmad,
Workday Inc., United States
Abdul Karim,
Hallym University, Republic of Korea

*CORRESPONDENCE
Emma Qumsiyeh
e.qumsiyeh@paluniv.edu.ps

RECEIVED 18 August 2025
REVISED 23 November 2025
ACCEPTED 26 November 2025
PUBLISHED 21 January 2026

CITATION
Alazaidah R, BaniSalman M, Alqawasmi KE, Abu Zaid A, Hazaimeh Y, Alshraiedeh FS and Qumsiyeh E (2026) Identifying key features for phishing website detection through feature selection techniques.
Front. Comput. Sci. 7:1687867.
doi: 10.3389/fcomp.2025.1687867

Identifying key features for phishing website detection through feature selection techniques

Raed Alazaidah[1], Mohammad BaniSalman[1], Khaled E. Alqawasmi[1], Ali Abu Zaid[1], Yousuf Hazaimeh[1], Fuad Sameh Alshraiedeh[1] and Emma Qumsiyeh[2]*

[1]Faculty of Information Technology, Zarqa University, Zarqa, Jordan, [2]Faculty of Engineering and Information Technology, Palestine Ahliya University, Bethlehem, Palestine

Over the past few years, phishing has evolved into an increasingly prevalent form of cybercrime, as more people use the Internet and its applications. Phishing is a type of social engineering that targets users' sensitive or personal information. This paper seeks to achieve two main objectives: first, to identify the most effective classifier for detecting phishing among 40 classifiers representing six learning strategies. Secondly, it aims to determine which feature selection method performs best on websites with phishing datasets. By analyzing three unique datasets on phishing and evaluating eight metrics, this study found that Random Forest and Random Tree were superior at identifying phishing websites compared with other approaches. Similarly, GainRatioAttributeEval, along with InfoGainAttributeEval, performed better than the five alternative feature selection methods considered in this study.

KEYWORDS

classification, phishing websites, machine learning, feature selection, URL analysis

1 Introduction

Due to the widespread use of online services like e-commerce and social media and the increased access afforded by the Internet, users are increasingly susceptible to cyberattacks targeting sensitive information, such as usernames or credit card details. One popular method used by attackers is called phishing, which uses fraudulent websites that appear authentic and trick individuals into divulging their private data (Athulya and Praveen, 2020). This can be accomplished using email or text messages designed solely for this purpose; even communication between clients and companies may contain such deceptive links. Typically motivated by financial gain, malware infections on user machines, or identity theft, most phishing attempts involve these motives.

Recent findings indicate a dramatic increase in unique reported instances, exceeding 199 thousand detections in December 2020 alone—an alarming statistic compared with the Anti-Phishing Working Group's results from previous years (APWG, 2021). Moreover, since the early days of the pandemic in March last year, when global COVID-19 fears were high, scammers have frequently issued phony certificates containing the words "COVID" or "corona." These scammers have increasingly relied on digital certification policies and HTTPS protocols rather than on traditional tactics (Warburton, 2020).

Broadly, there are two ways to identify phishing: through user knowledge or anti-phishing software. Due to the realism of phishing emails and websites, many users find it challenging to detect them. Consequently, accurate software solutions for detecting these threats have become increasingly necessary. Software-based detection strategies include blocklisting, heuristics, and machine learning (Athulya and Praveen, 2020). Previous studies using machine learning often relied on numerous features to achieve high accuracy; however, extracting these features is not always possible in real-time scenarios, requiring more resilient solutions.

The purpose of this paper is to support the worldwide effort to combat phishing scams by leveraging advanced machine learning techniques to predict fraudulent websites accurately.

Numerous classification models have been proposed and employed to identify phishing websites, claiming superiority over other approaches (Alazaidah et al., 2018). Moreover, this study aims to determine the most suitable classification method (classifier) for phishing datasets. To obtain a comprehensive overview of the findings, more than 40 classifiers across six learning strategies are evaluated using several metrics, including accuracy, precision, recall, and F1-measure.

Feature selection is one of several necessary preprocessing steps when creating any machine learning (ML)-based learning model. Its purpose is to identify relevant features that aid in constructing intended models by selecting non-redundant consistent attributes (Alluwaici M. et al., 2020). The feature selection procedure always prioritizes characteristics that closely align with the objective qualities of the dataset's attributes (Alluwaici M. et al., 2020).

To achieve the goal, 40 classifiers from six well-known learning strategies were selected for assessment. The evaluation phase encompasses eight diverse, commonly used metrics, including accuracy, precision, recall, and AUC. Besides, it aims to implicitly identify the best learning strategy among those considered using four distinct evaluation indicators: accuracy, precision, recall, F-Measure, MCC, PRC area, and ROC-Area (receiver operating characteristics).

The second objective of this study is to determine the optimal feature selection technique for predicting phishing websites. To achieve this objective, five commonly used feature selection methods were assessed and compared with identical classifiers used in the first goal across three evaluation metrics: accuracy, precision, and recall.

The remaining sections of the paper are structured as follows: Section 2 reviews the current literature on implementing ML techniques for phishing. In Section 3, we present our methodology, results, and discussion. Finally, concluding remarks and future directions are proposed in Section 4.

2 Related research

In this section, we examine prior research that has used machine learning techniques to detect phishing. In their study on fuzzy rough set feature selection, Zabihimayvan and Doran (2019) used multiple features to construct a model intended to detect fraudulent activity attempts by criminals intentionally sidestepping existing anti-phishing measures on Iranian banking websites. They trained and tested their system using fuzzy experts, achieving an accuracy of around 88%. Still, they acknowledged that there is scope for optimizing feature selection during the training/testing phases, which could increase predictive power while reducing prediction time.

A different approach was taken by Cui (2019), leveraging data analytics across multiple search engines as its source material identifying idle URLs previously exposed through popular searches or internal links shared between identified related sites along with additional input from frequently visited pages from URL structural similarity evaluation utilizing twelve (12) distinct characteristics depicting intra-relatedness/popularity degrees among entered site structures and components; altogether building classifiers resulting overall classification rates exceeding nearly ninety-five percent success rate coupled at about one-and-a-half false positives per classifying session—however may overlook obfuscated content when analyzing linked materials such as domain name variations generated algorithmically/hosted solely off malicious web domains themselves/limited character string-denser link shortening platforms commonly employed against undetected trapping activities.

Gandotra and Gupta (2021) compared various ML techniques using a 30-feature set comprising approximately 5,000 phishing websites and over 6,000 authentic webpages. This study found that incorporating feature selection enables faster creation of effective phishing detection models while maintaining accuracy. Notably, their results highlight that random forest classification (RF) achieves superior accuracy regardless of whether feature selection is used.

Detecting phishing attempts using ML often involves analyzing lexical features of URLs. This method, pioneered by Abutaha et al. (2021), was intended for use as a browser plug-in that scrutinizes a webpage's URL to alert users before they visit it. To test the efficacy of this technique, over one million legitimate and fraudulent URLs were used in experiments that extracted 22 variables, which were reduced to 10 key ones.

Findings revealed an accuracy rate of 99.89% when combined with SVM classification, surpassing the RF classifier, gradient boosting classifier (GBC), and neural network approaches trialed alongside it.

Chapla et al. (2019) proposed a fuzzy-logic-based framework for detecting phishing websites, using a dataset containing both legitimate and fraudulent URLs. The model achieved 91.4% accuracy but was limited by a small sample size of 1,000 features focused solely on URL-related attributes; as a result, it is less effective at identifying other bypass techniques.

The author in Tan (2018) improved the performance of their phishing URL detection system by using lexical features. A model proposed in Chiew et al. (2019) achieved high accuracy while being independent of third-party services and source code analysis, thereby requiring less processing time. Meanwhile, authors in Abdelhamid et al. (2014) sought to enhance the accuracy of phishing detection systems through feature selection and an ensemble learning approach, achieving 95% accuracy in their experiments.

In yet another effort detailed in article (Su et al., 2023), an innovative approach used seven distinct machine learning algorithms for detecting potential risks posed by various unwanted

attacks, including those utilizing zero-day exploits, with selected implemented security features overcoming issues such as language dependency or reliance on external parties during real-time monitoring operations without issue!

Rahman et al.'s research also explored machine learning classifiers' ability concerning various datasets related to phishing practices (Gandotra and Gupta, 2021). This initiative likewise demonstrated equivalent results, with gradient boosting trees (GBT) outperforming all metrics and achieving higher success rates than other methods, such as random forest (RF).

OFS-NN was proposed by Sahingoz et al. (2019) and combines optimal feature selection with a neural network to mitigate overfitting by using a new metric, the feature validity value (FVV). Experimental results on two datasets demonstrated that FVV outperformed information gain and optimal feature selection across various categories, including specific features such as abnormal, domain, HTML/JavaScript, and even address-bar features. The OFS-NN model achieved an overall accuracy of 0.945; however, among the feature types used for detection, the highest accuracy, 0.903, was observed with "address bar," while the lowest, around half accurate at 0.562, was observed with HTML/JavaScript.

Another phishing detection system was introduced by Sahingoz et al. (2019), which comprises 40 NLP-based traits, along with additional hybrid characteristics derived from word vectorization, totaling about 1,700 more relevant aspects.

In their study, the authors compared seven distinct algorithms offering diverse options but ultimately determined random forest's implementation made using solely natural language processing delivered the most superior performance, scoring almost perfect precision statistics, peaking up to staggering score amounts nearing practically zenith level, i.e., tracing fraudulent websites based upon this criterion managed to reach correct outcomes nearly 98 percent times—rendering maximum efficacy amongst all tested methodologies researched herein.

In Alazaidah et al. (2024), the authors conduct a comparative analysis of 24 classifiers across two datasets using several evaluation metrics. The results revealed the superiority of the random forest, filtered classifier, and J48 classifiers. The author suggests considering additional classification models with different learning strategies, as well as more datasets and evaluation metrics.

The research in Aljofey et al. (2025) proposed a hybrid methodology that combines URL character embeddings with several handcrafted features. Three datasets were used in this work: two are benchmarks, and the third was collected and preprocessed by the authors. The results showed excellent performance across accuracy and other evaluation metrics.

Several deep learning optimization techniques were used in Barik et al. (2025) to improve phishing prediction on websites. The authors used standardization and variational autoencoder techniques in the preprocessing step, and an enhanced grid search optimizer to improve accuracy. The results showed superior performance across accuracy, precision, and F1-score metrics. Unfortunately, utilizing one dataset only does not help in generalizing the finding of the conducted research. Several other related research works could be find in Ganjei and Boostani (2022), Gareth et al. (2023), Ni et al. (2022), Nti et al. (2022), Rashid et al. (2020), Srivastava (2014), Ubing et al. (2019).

Throughout this literature review, random forests perform comparatively better than their counterparts in detecting phishing using machine learning. However, gradient boosting machines (GBM) were frequently not a subject of comparison, affecting project linearity and requiring deeper exploration, while lackluster attempts, such as minimal input/no-noise coefficient data filtering, were still in early phases, indicating that extensive future research remains vital.

3 Research methodology

The methodology employed in this paper is depicted in Figure 1. The first phase in Figure 1 involves collecting the datasets. Afterward, the datasets are cleaned and preprocessed. Then, several feature selection techniques are trained on the pre-processed datasets and evaluated. Next, 40 classification models are trained on the datasets using the selected features from the previous step. These classifiers are compared using several well-known evaluation metrics.

The description of three website phishing datasets used in this research is provided in Section (A), while Sections (B, C, and D) evaluate the performance of feature selection and machine learning algorithms on these datasets.

Moreover, Section 4 considers which classification model is most appropriate for phishing website datasets. Therefore, three datasets are considered in this section.

In addition to that, Section 5 evaluates and identifies the best among five renowned feature selection methods, as well as identifying the most efficient classifiers, which are outlined in Section 6 before finally discussing primary results obtained from these sections' analyses at length.

In addition, 40 classifiers from *six* learning strategies are evaluated and contrasted in terms of their predictive efficacy across the *three* datasets under consideration. These examined classifiers encompass:

Random tree, random forest, REPTree, DecisionStump, HoeffdingTree, LMT, J4B, and REPTree from the Trees learning strategy; BayesNet, NaiveBayesUpdateable, and NaiveBayes from the Bayes learning strategy. Logistic, MultilayerPerceptron, SimpleLogistic, VotedPerceptron, and SMO from the Functions strategy. IBK, KStar, and LWL from the lazy learning strategy; AdaBoostM1, AttributeSelectedClassifier, Bagging, ClassificationViaRegression, FilteredClassifier, IterativeClassifierOptimizer, LogitBoost, MultiClassClassifier, MultiClassClassifierUpdateable, RandomCommittee, RandomizableFilteredClassifier, RandomSubSpace, Stacking, WeightedInstancesHandlerWrapper, vot, and CVParameterSelectionr from the Meta learning strategy; DecisionTable, JRip, OneR, PART, and ZeroR learning strategy. Finally, InputMappedClassifier from the misc learning strategy.

The WEKA software's default settings are utilized for all classification models. This renowned data analysis tool, also known as (Waikato Environment for Knowledge Analysis), is frequently used (Rao et al., 2020). The outcome validation process uses 10-fold cross-validation to ensure the results.

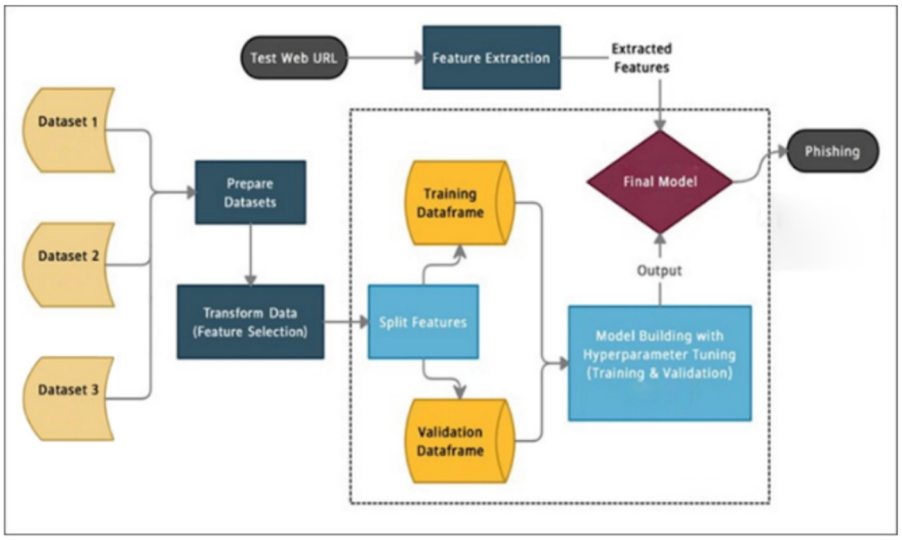

FIGURE 1
Research methodology workflow diagram.

To compare the considered classification models, six performance metrics were analyzed: Accuracy, precision, recall, F-measure, MCC (Matthews correlation coefficient), ROC Area, and PRC Area. Next up are the equations needed to calculate these metrics.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \; TP\ rate = \frac{TP}{TP + FN} \; FPrate = \frac{FP}{FP + TN} \; Precision = \frac{TP}{TP + FP} \; Recall = \frac{TP}{TP + FN}$$

Accuracy is a metric that indicates how frequently a machine learning model predicts the correct outcome. The number of right guesses divided by the total number of forecasts yields accuracy (Alzyoud et al., 2024; Alazaidah et al., 2023a,b).

Precision is a metric that indicates how often a machine learning model correctly predicts the positive class. Precision can be calculated as the number of correct positive predictions (true positives) divided by the total number of positive predictions made by the model (including true and false positives).

Recall is a metric that indicates how often a machine learning model accurately detects positive examples (true positives) from all actual positive samples in the dataset. Divide the number of true positives by the number of positive cases to determine recall. The latter includes true positives (correctly identified cases) and false negatives (missed cases) (Al-Batah et al., 2023; Pei et al., 2022).

MCC is the best single-value classification metric for summarizing a confusion or error matrix. A confusion matrix has four entities:

- True positives (TP)
- True negatives (TN)
- False positives (FP)
- False negatives (FN)

And is calculated by the formula:

$$MCC = \frac{TN \times TP - FN \times FP}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

F-measure is an alternative machine learning evaluation metric that assesses the predictive skill of a model by elaborating on its class-wise performance rather than its overall performance, as done by accuracy. The F1 score combines two competing metrics—precision and recall—of a model, making it widely used in recent literature.

$$F - measure = \frac{2^{*}\ Recall\ ^{*}\ Precision}{Recall\ +\ Precision}$$

ROCArea: a metric that graphically assesses classifier performance across varying thresholds by plotting the false positive rate on the x-axis and the true positive rate on the y-axis.

True Positives (TPs): instances in which the model correctly identifies examples.

True Negatives (TNs): represent cases where the model correctly recognizes and labels negative examples.

False Positives (FPs): occur once the model mistakenly identifies examples as positive. In words, these are instances where negative examples are mistakenly labeled as "positive."

False Negatives (FNs): arise when positive examples are incorrectly classified as negative. These are cases in which positive examples are incorrectly labeled as "negative."

3.1 Description of datasets

In the study, three datasets are available for download from the UCI repository. The first dataset, a binary classification set, contains 11,055 instances with 30 integer features. Most

TABLE 1 Datasets characteristic.

Name	Instances	Features	No. of classes	Feature type	References
DS1	11,055	30	3	Integer	Su et al., 2023
DS2	10,000	18	3	Integer	Alluwaici M. A. et al., 2020
DS3	2,670	13	2	Integer	Mohammad et al., 2015

TABLE 2 Categories of features for the two datasets.

Dataset code	Feature category	Feature examples
DS1	URL based	having_IP_Address, URL_Length, HTTPS_token, etc.
	Abnormal based	Request_URL, URL_of_Anchor, Links_in_tags, etc.
	HTML/js Based	Redirect, on_mouseover, RightClick, popUpWindow, etc.
	Domain based	DNSRecord, web_traffic, Page_Rank, Google_Index, etc.
DS2	HTML/JS based, URL based	Redirect, on_mouseover, RightClick, popUpWindow, etc.
DS2	URL based	NumDots, UrlLength, AtSymbol, etc.
	Abnormal	AbnormalExtFormAction, ExtMetaScriptLinkRT, etc.
	HTML/Js Based	RightClickDisabled, ExtFavicon, PopUpWindow, etc.

of these features are binary. On the other hand, the second dataset comprises three class labels, supports multiclassification, and provides nine integer-type features and 10,000 examples; the third dataset comprises two class labels, consists of 13 integer-type features, and provides 2,670 instances. Table 1 presents the distinguishing qualities of both sets for quick reference. This research focuses on the first two datasets, which are the largest and have 3 class labels, while the third dataset is relatively small with only two classes: selection and understanding.

This step focused on collecting datasets and understanding the attributes. Three datasets, denoted DS1, DS2, and DS3 (Su et al., 2023; Alluwaici M. A. et al., 2020), and DS3 (Mohammad et al., 2015), were selected, as they have different numbers of features and only some are common. Table 2 summarizes the feature categories across the three datasets. DS1, DS2, and DS3 contain both internal features (i.e., derived from webpage URLs and HTML/JavaScript source code available on the webpage itself) and external features (i.e., obtained from querying third-party services such as DNS, search engines, and WHOIS records). DS2 only contains internal features (Mohammad et al., 2015).

3.2 Data preparation

Data preprocessing involves operations such as handling missing values, removing outliers, and eliminating redundant information. As stated in reference (Alazaidah et al., 2023a), the DS1, DS2, and DS3 datasets were free of missing data but required cleaning before use. For instance, the HttpsInHostname attribute in DS3 had all values set to 0, making it unnecessary for analysis.

To identify common attributes across these datasets (DS1-DS2-DS3), the authors checked their descriptions available in references (Mohammad et al., 2015) and (Alzyoud et al., 2024). The authors' citations for each dataset feature significantly simplified this preprocessing step.

It was noted that some feature pairs captured similar information expressed in different formats, such as UrlLength, which is numeric, and its counterpart, "UrlLengthRT," which is categorical. In cases where those occurred only once, they would be mapped to the same variable, URL_Length, found solely in dataset DP1; otherwise, they would remain separate. Ultimately, after scrutinizing these intricate details across variables, we discovered a match between 18 key attributes among the three aforementioned sources (as shown in Table 3).

3.3 Feature selection

The significance of independent features was assessed using *P*-values, with a threshold of 0.05 to identify statistically significant features.

To begin with, the Spearman rank-order correlation method assessed collinearity between feature pairs. In Figure 2, we show the correlation matrix for the DS1-2-3 matching feature, with the pop-up window and on-mouse-over having the highest observed value at 0.73, followed by the pop-up window and favicon pair, which had a corresponding score of 0.66. Most pairs showed small or negligible correlations.

To identify multicollinearity—where three or more variables converge even when no two have high individual similarities—the Variance inflation factor (VIF) scores were used (Ubing et al., 2019).

Each trait received its VIF rating calculated as follows:

$$VIF_i = \frac{1}{1 - R_i^2}$$

R_i^2 =Unadjusted coefficient of determination for regressing the ith independent variable on the remaining ones.

Based on VIF analysis, in addition to *p*-values, the combined DS1-2-3 data identified 15 features as noteworthy and independent.

TABLE 3 The matched features between ds1, ds2 and ds3 dataset with the features after feature selection.

DS1	DS2	DS3	DS1-1-2-3
having_IP_Address	IpAddress	IP_Address	
having_Sub_Domain	SubdomainLevel*	Sub_Domain	√
Links_pointing_to_page	PctExtHyperlinks*	Links _to_page	√
Submitting_to_email	SubmitInfoToEmail	Submitting_to_email	√
double_slash_redirecting	DoubleSlashInPath	double_redirecting	√
URL_Length	UrlLength*	URL_Length	√
Favicon	ExtFavicon	Favicon	√
Prefix_Suffix	NumDashInHostname*	Prefix_Suffix	√
SFH	AbnormalFormAction	SFH	√
Iframe	IframeOrFrame	Iframe	√
having_At_Symbol	AtSymbol	_At_Symbol	√
SSLfinal_State	NoHttps	SSLfinal_State	
on_mouseover	FakeLinkInStatusBar	on_mouseover	
URL_of_Anchor	PctNullSelfRedirectHyperlinks*	URL_of_Anchor	√
popUpWidnow	PopUpWindow	popUpWidnow	
Request_URL	PctExtResourceUrls*	Request_URL	√
RightClick	RightClickDisabled	Right_Click	
Links_in_tags	'ExtMetaScriptLinkRT*	Links _tags	√

* indicates numeric features, √ indicates selected features.

This process used various Python packages, including statsmodels to calculate VIF scores and *p*-values, scikit-learn to build logistic regression models, and Matplotlib and Seaborn to generate visualizations.

For the feature selection and ranking step, four techniques have been considered and evaluated. The first technique is called Correlation Attribute Evaluator (CAE). CAE measures the linear correlations between the input features and the output feature (class) and is usually implemented using Pearson's correlation coefficient. The second technique is the Gain Ratio Attribute Evaluator (GRAE). This technique assesses feature significance by measuring each feature's gain ratio relative to the class label. The third technique is dubbed the Information Gain Attribute Evaluator (IGAE). IAGE measures how a feature is worth based on the value of information gain for this feature with respect to the class label. The last technique is the Principal Components Analysis (PCA). This technique aims to reduce data dimensionality by transforming a large dataset into a smaller one with low-correlated features.

4 Comparative analysis amongst the classification models in the domain of website phishing

This section describes the process of determining the ideal classification model for phishing datasets. To attain this objective, three distinct sets of data cognate to phishing have been analyzed in detail. Table 4 outlines the highlighted attributes associated with these datasets, all of which can be obtained from the UCI repository with ease.

The results of using 40 classifiers on the phishing website dataset 1 (DS1) are presented in Table 4 and analyzed with respect to accuracy and pre-session metrics. The data reveal that IBK achieves the highest accuracy, whereas RandomCommittee achieves outstanding accuracy and precision.

Evaluating learning strategies indicates that Lazy achieves optimal accuracy, while RandomCommittee yields superior precision.

The Recall and MCC metric results for the phishing website dataset after applying 40 classifiers are outlined in Table 5. The table shows that random forest classification models have produced superior results when evaluated against these criteria.

Additionally, Tree outperforms other learning strategies on both precision and MCC metrics in this dataset (DS1).

A comparative analysis of 40 classifiers on the phishing dataset, in terms of accuracy and precision, is presented in Table 5.

Random forest outperforms the other considered classifiers in accuracy and precision on the phishing dataset (DS1), as shown in the table.

Moreover, among the eight learning strategies assessed using these two measures, the Functions Tree strategy yields better outcomes than its counterparts.

The precision metrics obtained from applying 40 classifiers to the phishing dataset are shown in Table 6. According to the table, among all classification models, the RandomCommittee learning strategy achieves the highest precision. Similarly, for the Random Forest metric, based on Table 6 and the Trees learning strategy,

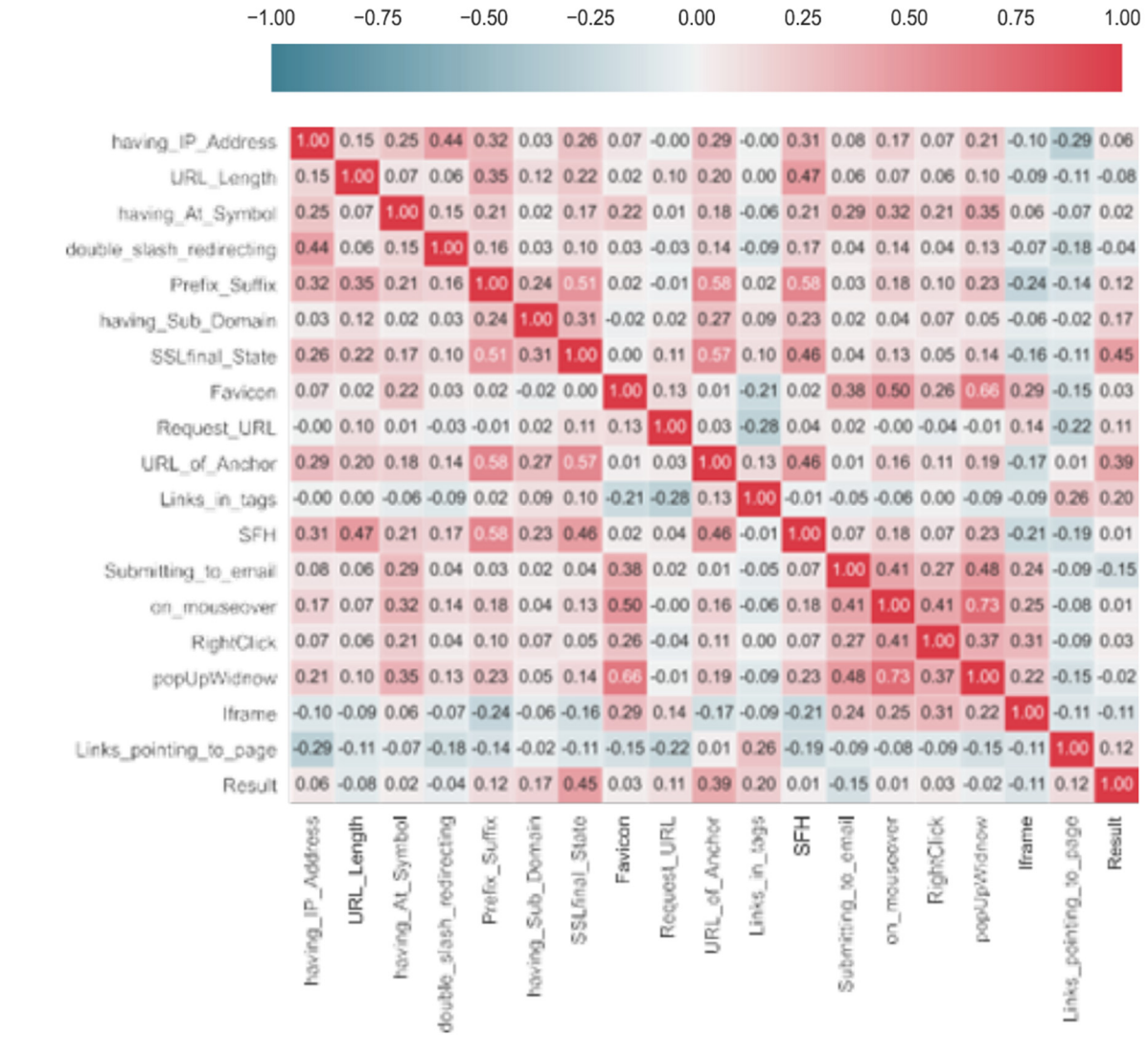

FIGURE 2
Spearman correlation heatmap based on the merged Dataset 1, 2, and besides 3 datasets, showing some collinearity between the different features (note that Result is the class attribute).

we can see that the Random Forest classification model delivers superior outcomes.

In conclusion, regarding optimizing the precision metrics shown in Table 6, function learning is our preferred approach, yielding the best results compared to other available strategies.

In Table 7, the random forest classification models achieve the best recall and MCC results on the phishing dataset (DS1). The random forest classifier belongs to the Tree learning strategy.

Moreover, regarding the best learning strategy, Table 7 shows that the tree learning strategy achieves the best results for the recall and MCC metrics.

According to Table 8, the classifier in the tree learning strategy, random forest, has the highest precision metric. Additionally, when it comes to the accuracy metric and other compared classifiers, this same classifier performs best again. Furthermore, among the seven considered learning strategies, Tree stands out as achieving superior results across comparisons.

The outcomes of the 40 classifiers applied to the phishing website dataset, with respect to recall and MCC, are shown in Table 9.

Analysis of Table 9 indicates that, among all classification models, the random tree classifier achieved the highest accuracy and precision on the given dataset (DS2). Additionally, compared with other learning strategies exhibited by the remaining classifying algorithms in Table 9, the tree strategy was found to outperform others in terms of efficient data processing.

The results from implementing 40 classifiers on the phishing website dataset (DS2) are shown in the table, including accuracy and precision metrics.

The Random Forest model, a tree-based learning strategy, achieves higher accuracy and precision than other classification models, as shown in Table 10.

TABLE 4 Comparative analysis of 40 classifiers utilizing feature selection via CAE, on dataset DS1.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	random tree	90.502	0.905	0.905	0.905	0.807	0.965	0.961
	Random forest	90.664	0.907	0.907	0.906	0.811	0.973	0.974
	REPTree	89.561	0.897	0.896	0.895	0.789	0.961	0.962
	DecisionStump	84.730	0.877	0.847	0.841	0.714	0.823	0.810
	HoeffdingTree	88.801	0.890	0.888	0.887	0.774	0.937	0.939
	LMT	90.610	0.906	0.906	0.906	0.810	0.971	0.971
	J4B	90.031	0.901	0.900	0.900	0.798	0.960	0.958
Avg		89.271	0.897	0.892	0.891	0.786	0.941	0.939
Bayes	BayesNet	87.535	0.876	0.875	0.875	0.747	0.947	0.951
	NaiveBayes	87.535	0.876	0.875	0.875	0.747	0.947	0.951
	NaiveBayesUpdateable	55.694	0.557	1.000	0.715	0.500	0.500	0.506
Avg		76.921	0.767	0.916	0.821	0.664	0.798	0.802
Functions	Logistic	88.647	0.888	0.886	0.886	0.771	0.954	0.956
	SGD	88.738	0.889	0.887	0.887	0.772	0.882	0.842
	SimpleLogistic	88.629	0.889	0.886	0.885	0.771	0.953	0.956
	SMO	88.955	0.891	0.89	0.889	0.777	0.883	0.845
	VotedPerceptron	88.358	0.886	0.884	0.883	0.765	0.88	0.84
Avg		88.666	0.888	0.886	0.886	0.771	0.910	0.887
Lazy	IBK	90.755	0.908	0.908	0.907	0.812	0.973	0.973
	Kstar	90.393	0.905	0.904	0.903	0.806	0.97	0.972
	LWL	84.730	0.877	0.847	0.841	0.714	0.945	0.947
Avg		88.652	0.891	0.886	0.885	0.773	0.927	0.913
Meta	AdaBoostM1	87.435	0.876	0.874	0.873	0.746	0.938	0.941
	AttributeSelectedClassifier	87.363	0.876	0.874	0.873	0.745	0.935	0.936
	Bagging	89.977	0.901	0.900	0.899	0.797	0.967	0.969
	ClassificationViaRegression	89.036	0.892	0.890	0.89	0.778	0.959	0.961
	FilteredClassifier	90.031	0.901	0.900	0.900	0.798	0.96	0.958
	IterativeClassifierOptimizer	87.806	0.880	0.878	0.877	0.754	0.948	0.951
	LogitBoost	87.806	0.880	0.878	0.877	0.754	0.948	0.951
	MultiClassClassifier	88.647	0.888	0.886	0.886	0.771	0.954	0.956
	MultiClassClassifierUpdateable	88.738	0.889	0.887	0.887	0.772	0.882	0.842
	RandomCommittee	90.755	0.908	0.908	0.907	0.812	0.971	0.969
	RandomizableFilteredClassifier	90.230	0.902	0.902	0.902	0.802	0.966	0.966
	RandomSubSpace	89.027	0.893	0.890	0.889	0.779	0.957	0.959
	Stacking	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	WeightedInstancesHandlerWrapper	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	vot	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	CVParameterSelection	55.694	0.557	0.557	0.715	0.500	0.500	0.506
Avg		80.601	0.807	0.805	0.845	0.706	0.836	0.836
Rules	DecisionTable	88.177	0.883	0.882	0.881	0.76	0.95	0.952
	JRip	89.271	0.895	0.893	0.892	0.784	0.904	0.890

(Continued)

TABLE 4 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	OneR	84.730	0.877	0.847	0.841	0.714	0.828	0.794
	PART	90.375	0.904	0.904	0.903	0.805	0.967	0.966
	ZeroR	55.694	0.557	0.557	0.715	0.506	0.500	0.506
Avg		81.644	0.823	0.816	0.846	0.713	0.829	0.821
Misc	InputMappedClassifier	55.694	0.557	0.557	0.715	0.500	0.500	0.506
Avg		55.694	0.557	0.557	0.715	0.500	0.500	0.506

TABLE 5 Comparative analysis of 40 classifiers utilizing feature selection via CAE, on dataset DS1.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	95.911	0.959	0.959	0.959	0.917	0.978	0.969
	Random forest	96.436	0.964	0.964	0.964	0.928	0.993	0.993
	REPTree	94.898	0.949	0.949	0.949	0.897	0.984	0.981
	DecisionStump	88.891	0.889	0.889	0.889	0.774	0.882	0.854
	HoeffdingTree	94.002	0.940	0.940	0.940	0.878	0.983	0.983
	LMT	95.766	0.958	0.958	0.958	0.914	0.989	0.988
	J4B	95.45	0.955	0.955	0.954	0.908	0.981	0.977
Avg		94.479	0.944	0.944	0.944	0.888	0.970	0.963
Bayes	BayesNet	92.772	0.928	0.928	0.928	0.853	0.981	0.982
	NaiveBayes	92.772	0.928	0.928	0.928	0.853	0.981	0.982
	NaiveBayesUpdateable	92.772	0.928	0.928	0.928	0.853	0.981	0.982
Avg		92.772	0.928	0.928	0.928	0.853	0.981	0.982
Functions	Logistic	93.369	0.934	0.934	0.934	0.866	0.985	0.986
	SGD	93.306	0.933	0.933	0.933	0.864	0.931	0.904
	SimpleLogistic	93.306	0.933	0.933	0.933	0.864	0.985	0.986
	SMO	93.315	0.933	0.933	0.933	0.864	0.931	0.904
	VotedPerceptron	93.288	0.933	0.933	0.933	0.864	0.932	0.904
Avg		88.665	0.888	0.886	0.886	0.771	0.910	0.887
Lazy	IBK	96.119	0.961	0.961	0.961	0.921	0.987	0.986
	Kstar	96.128	0.962	0.961	0.961	0.922	0.995	0.995
	LWL	88.991	0.890	0.890	0.89	0.777	0.975	0.976
Avg		88.652	0.891	0.881	0.885	0.773	0.927	0.913
Meta	AdaBoostM1	92.582	0.926	0.926	0.926	0.850	0.981	0.982
	AttributeSelectedClassifier	94.400	0.944	0.944	0.944	0.886	0.980	0.978
	Bagging	95.486	0.955	0.955	0.955	0.908	0.990	0.990
	ClassificationViaRegression	94.536	0.945	0.945	0.945	0.889	0.988	0.988
	FilteredClassifier	95.450	0.955	0.955	0.954	0.908	0.981	0.977
	IterativeClassifierOptimizer	92.736	0.927	0.927	0.927	0.853	0.981	0.982
	LogitBoost	92.736	0.927	0.927	0.927	0.853	0.981	0.982
	MultiClassClassifier	93.369	0.934	0.934	0.934	0.866	0.985	0.986

(Continued)

TABLE 5 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	MultiClassClassifierUpdateable	93.306	0.933	0.933	0.933	0.864	0.931	0.904
	RandomCommittee	96.408	0.964	0.964	0.964	0.927	0.989	0.985
	RandomizableFilteredClassifier	94.292	0.943	0.943	0.943	0.884	0.969	0.964
	RandomSubSpace	93.414	0.935	0.934	0.934	0.867	0.984	0.985
	Stacking	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	WeightedInstancesHandlerWrapper	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	vot	55.694	0.557	0.557	0.715	0.500	0.507	0.506
	CVParameterSelection	55.694	0.557	0.557	0.715	0.500	0.500	0.506
Avg		84.467	0.844	0.844	0.885	0.784	0.859	0.857
Rules	DecisionTable	92.863	0.929	0.929	0.929	0.855	0.979	0.98
	JRip	94.753	0.948	0.948	0.947	0.894	0.96	0.953
	OneR	88.891	0.889	0.889	0.889	0.774	0.886	0.845
	PART	95.585	0.956	0.956	0.956	0.911	0.985	0.966
5	ZeroR	55.694	0.557	0.557	0.715	0.506	0.511	0.506
Avg		85.557	0.855	0.855	0.887	0.788	0.864	0.85
Misc	InputMappedClassifier	55.694	0.557	0.557	0.715	0.500	0.506	0.506
Avg		55.694	0.557	0.557	0.715	0.500	0.506	0.506

TABLE 6 Comparative analysis of 40 classifiers utilizing feature selection via GRAE, on dataset DS1.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	95.640	0.956	0.956	0.956	0.912	0.981	0.974
	Random forest	96.191	0.962	0.962	0.962	0.923	0.992	0.992
	REPTree	94.744	0.947	0.947	0.947	0.893	0.984	0.982
	DecisionStump	88.891	0.889	0.889	0.889	0.774	0.882	0.854
	HoeffdingTree	93.903	0.939	0.939	0.939	0.876	0.983	0.984
	LMT	95.676	0.957	0.957	0.957	0.912	0.988	0.986
	J4B	95.106	0.951	0.951	0.951	0.901	0.983	0.98
Avg		94.307	0.943	0.943	0.943	0.884	0.970	0.964
Bayes	BayesNet	92.636	0.927	0.926	0.926	0.851	0.980	0.981
	NaiveBayes	92.645	0.927	0.926	0.926	0.851	0.980	0.981
	NaiveBayesUpdateable	55.694	0.557	0.557	0.715	0.500	0.500	0.506
Avg		80.327	0.803	0.803	0.855	0.734	0.820	0.822
Functions	Logistic	93.378	0.934	0.934	0.934	0.866	0.985	0.986
	SGD	93.514	0.935	0.935	0.935	0.868	0.933	0.906
	SimpleLogistic	93.432	0.934	0.934	0.934	0.867	0.985	0.985
	SMO	93.523	0.935	0.935	0.935	0.869	0.933	0.907
	VotedPerceptron	93.360	0.934	0.934	0.934	0.865	0.933	0.906
Avg		88.665	0.888	0.886	0.886	0.771	0.910	0.887
Lazy	IBK	95.730	0.957	0.957	0.957	0.913	0.988	0.987

(Continued)

TABLE 6 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	Kstar	95.649	0.957	0.956	0.956	0.912	0.994	0.994
	LWL	89.018	0.890	0.890	0.89	0.777	0.974	0.974
Avg		93.465	0.934	0.934	0.934	0.867	0.985	0.985
Meta	AdaBoostM1	92.582	0.926	0.926	0.926	0.85	0.981	0.982
	AttributeSelectedClassifier	94.310	0.943	0.943	0.943	0.885	0.979	0.977
	Bagging	95.386	0.954	0.954	0.954	0.906	0.990	0.990
	ClassificationViaRegression	94.635	0.946	0.946	0.946	0.891	0.989	0.989
	FilteredClassifier	95.106	0.951	0.951	0.951	0.901	0.983	0.980
	IterativeClassifierOptimizer	92.736	0.927	0.927	0.927	0.853	0.981	0.982
	LogitBoost	92.736	0.927	0.927	0.927	0.853	0.981	0.982
	MultiClassClassifier	93.378	0.934	0.934	0.934	0.866	0.985	0.986
	MultiClassClassifierUpdateable	93.514	0.935	0.935	0.935	0.868	0.933	0.906
	RandomCommittee	96.408	0.964	0.964	0.964	0.927	0.989	0.985
	RandomizableFilteredClassifier	94.771	0.948	0.948	0.948	0.894	0.975	0.971
	RandomSubSpace	93.450	0.935	0.935	0.934	0.867	0.983	0.984
	Stacking	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	WeightedInstancesHandlerWrapper	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	vot	55.694	0.557	0.557	0.715	0.500	0.507	0.506
	CVParameterSelection	55.694	0.557	0.557	0.715	0.500	0.500	0.506
Avg		84.487	0.844	0.844	0.884	0.785	0.859	0.858
Rules	DecisionTable	92.971	0.93	0.930	0.930	0.858	0.978	0.978
	JRip	94.563	0.946	0.946	0.946	0.890	0.959	0.952
	OneR	88.891	0.889	0.889	0.889	0.774	0.886	0.845
	PART	95.468	0.955	0.955	0.955	0.908	0.987	0.984
	ZeroR	55.694	0.557	0.557	0.715	0.506	0.511	0.506
Avg		85.517	0.855	0.855	0.887	0.787	0.864	0.853
misc	InputMappedClassifier	55.694	0.557	0.557	0.715	0.500	0.506	0.506
Avg		55.694	0.557	0.557	0.715	0.500	0.506	0.506

TABLE 7 Comparative analysis of 40 classifiers utilizing feature selection via IGAE, on dataset DS1.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	95.649	0.956	0.956	0.956	0.912	0.978	0.969
	Random forest	96.255	0.963	0.963	0.963	0.924	0.992	0.991
	REPTree	94.853	0.949	0.949	0.949	0.896	0.983	0.980
	DecisionStump	88.891	0.889	0.889	0.889	0.774	0.882	0.854
	HoeffdingTree	93.930	0.939	0.939	0.939	0.877	0.983	0.983
	LMT	95.829	0.958	0.958	0.958	0.915	0.989	0.988
	J4B	95.630	0.956	0.956	0.956	0.911	0.985	0.982
Avg		94.434	0.944	0.944	0.944	0.887	0.970	0.963

(Continued)

TABLE 7 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Bayes	BayesNet	92.781	0.928	0.928	0.928	0.854	0.981	0.982
	NaiveBayes	92.781	0.928	0.928	0.928	0.854	0.981	0.982
	NaiveBayesUpdateable	55.694	0.559	1.000	0.715	0.500	0.500	0.506
Avg		80.419	0.805	0.952	0.857	0.736	0.820	0.823
Functions	Logistic	93.387	0.934	0.934	0.934	0.866	0.985	0.986
	SGD	93.351	0.932	0.934	0.933	0.865	0.932	0.904
	SimpleLogistic	93.351	0.934	0.934	0.933	0.865	0.985	0.986
	SMO	93.324	0.933	0.933	0.933	0.865	0.931	0.904
	VotedPerceptron	93.333	0.933	0.933	0.933	0.865	0.932	0.905
Avg		88.665	0.888	0.886	0.886	0.771	0.910	0.887
Lazy	IBK	95.829	0.958	0.958	0.958	0.915	0.988	0.986
	Kstar	95.983	0.960	0.960	0.960	0.919	0.994	0.994
	LWL	88.973	0.890	0.890	0.890	0.776	0.975	0.975
Avg		88.652	0.891	0.886	0.885	0.773	0.927	0.912
Meta	AdaBoostM1	92.582	0.926	0.926	0.926	0.850	0.981	0.982
	AttributeSelectedClassifier	94.400	0.944	0.944	0.944	0.886	0.980	0.978
	Bagging	95.404	0.954	0.954	0.954	0.907	0.990	0.990
	ClassificationViaRegression	94.436	0.944	0.944	0.944	0.887	0.988	0.988
	FilteredClassifier	95.630	0.956	0.956	0.956	0.911	0.985	0.982
	IterativeClassifierOptimizer	92.736	0.927	0.927	0.927	0.853	0.981	0.982
	LogitBoost	92.736	0.927	0.927	0.927	0.853	0.981	0.982
	MultiClassClassifier	93.387	0.934	0.934	0.934	0.866	0.985	0.986
	MultiClassClassifierUpdateable	93.351	0.934	0.934	0.933	0.865	0.932	0.904
	RandomCommittee	90.755	0.908	0.908	0.907	0.812	0.971	0.969
	RandomizableFilteredClassifier	90.230	0.902	0.902	0.902	0.802	0.966	0.966
	RandomSubSpace	93.984	0.940	0.940	0.940	0.878	0.986	0.986
	Stacking	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	WeightedInstancesHandlerWrapper	55.6943	0.557	0.557	0.715	0.5	0.5	0.506
	vot	55.6943	0.557	0.557	0.715	0.5	0.509	0.506
	CVParameterSelection	55.6943	0.557	0.557	0.715	0.5	0.5	0.506
Avg		83.90095	0.839	0.839	0.878375	0.773125	0.858438	0.857438
Rules	DecisionTable	92.9986	0.93	0.93	0.93	0.858	0.981	0.981
	JRip	94.5274	0.945	0.945	0.945	0.889	0.96	0.953
	OneR	88.8919	0.889	0.889	0.889	0.774	0.886	0.845
	PART	95.4591	0.955	0.955	0.955	0.908	0.986	0.983
	ZeroR	55.6943	0.557	0.557	0.715	0.506	0.5	0.506
Avg		81.64994	0.8232	0.8166	0.8464	0.7138	0.8298	0.8216
Misc	InputMappedClassifier	55.6943	0.557	0.557	0.715	0.5	0.5	0.506
Avg		55.6943	0.557	0.557	0.715	0.5	0.5	0.506

TABLE 8 Comparative analysis of 40 classifiers utilizing feature selection via PC, on dataset DS1.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	94.219	0.942	0.942	0.942	0.883	0.982	0.978
	Random forest	94.473	0.945	0.945	0.945	0.888	0.987	0.986
	REPTree	93.523	0.936	0.935	0.935	0.869	0.977	0.976
	DecisionStump	88.891	0.889	0.889	0.889	0.774	0.882	0.854
	HoeffdingTree	93.062	0.932	0.931	0.93	0.86	0.967	0.965
	LMT	94.373	0.944	0.944	0.944	0.886	0.986	0.986
	J4B	93.794	0.939	0.938	0.938	0.875	0.975	0.974
Avg		93.191	0.932	0.932	0.931	0.862	0.965	0.959
Bayes	BayesNet	92.356	0.924	0.924	0.923	0.845	0.972	0.974
	NaiveBayes	92.365	0.924	0.924	0.923	0.845	0.972	0.974
	NaiveBayesUpdateable	92.365	0.924	0.924	0.923	0.845	0.972	0.974
Avg		92.362	0.924	0.924	0.923	0.845	0.972	0.974
Functions	Logistic	92.682	0.927	0.927	0.927	0.852	0.976	0.977
	SGD	91.705	0.917	0.917	0.917	0.832	0.916	0.882
	SimpleLogistic	92.645	0.927	0.926	0.926	0.851	0.976	0.977
	SMO	91.714	0.917	0.917	0.917	0.832	0.916	0.882
	VotedPerceptron	92.555	0.926	0.926	0.925	0.849	0.924	0.894
vg		88.665	0.888	0.8866	0.886	0.7712	0.9104	0.8878
Lazy	IBK	94.237	0.943	0.942	0.942	0.883	0.986	0.985
	Kstar	94.165	0.942	0.942	0.941	0.882	0.986	0.986
	LWL	88.991	0.890	0.890	0.890	0.777	0.966	0.967
Avg		88.652	0.891	0.886	0.885	0.773	0.927	0.913
Meta	AdaBoostM1	92.166	0.922	0.922	0.922	0.841	0.973	0.974
	AttributeSelectedClassifier	92.935	0.931	0.929	0.929	0.858	0.961	0.960
	Bagging	93.830	0.939	0.938	0.938	0.875	0.982	0.983
	ClassificationViaRegression	93.188	0.932	0.932	0.932	0.862	0.980	0.981
	FilteredClassifier	93.794	0.939	0.938	0.938	0.875	0.975	0.974
	IterativeClassifierOptimizer	92.220	0.923	0.922	0.922	0.842	0.974	0.975
	LogitBoost	92.437	0.925	0.924	0.924	0.847	0.974	0.975
	MultiClassClassifier	92.682	0.927	0.927	0.927	0.852	0.976	0.977
	MultiClassClassifierUpdateable	93.830	0.938	0.938	0.938	0.875	0.981	0.980
	RandomCommittee	94.409	0.944	0.944	0.944	0.887	0.986	0.984
	RandomizableFilteredClassifier	90.230	0.902	0.902	0.902	0.802	0.966	0.966
	RandomSubSpace	92.691	0.927	0.927	0.927	0.852	0.973	0.974
	Stacking	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	WeightedInstancesHandlerWrapper	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	vot	55.694	0.557	0.557	0.715	0.500	0.509	0.506
	CVParameterSelection	55.694	0.557	0.557	0.715	0.500	0.500	0.506
Avg		83.574	0.836	0.835	0.875	0.766	0.856	0.857
Rules	DecisionTable	93.025	0.931	0.930	0.930	0.859	0.977	0.977
	JRip	93.306	0.934	0.933	0.933	0.864	0.945	0.936

(Continued)

TABLE 8 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	OneR	88.891	0.889	0.889	0.889	0.774	0.886	0.845
	PART	94.355	0.944	0.944	0.943	0.886	0.983	0.983
	ZeroR	55.694	0.557	0.557	0.715	0.500	0.500	0.506
Avg		85.054	0.851	0.8506	0.882	0.776	0.858	0.849
Misc	InputMappedClassifier	55.694	0.557	0.557	0.715	0.500	0.500	0.506
Avg		55.694	0.557	0.557	0.715	0.500	0.500	0.506

TABLE 9 Comparative analysis of 40 classifiers utilizing feature selection via CAE, on dataset DS2.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	88.011	0.880	0.880	0.880	0.867	0.968	0.889
	Random forest	87.893	0.879	0.879	0.879	0.866	0.984	0.913
	REPTree	55.255	0.550	0.553	0.550	0.499	0.925	0.613
	DecisionStump	17.341	0.203	0.173	0.309	0.205	0.587	0.130
	HoeffdingTree	29.076	0.341	0.291	0.268	0.224	0.754	0.289
	J4B	60.422	0.602	0.604	0.603	0.558	0.946	0.715
Avg		56.333	0.575	0.563	0.581	0.536	0.860	0.591
bayes	BayesNet	72.783	0.732	0.728	0.729	0.699	0.975	0.822
	NaiveBayes	25.797	0.295	0.258	0.242	0.187	0.740	0.27
	NaiveBayesUpdateable	25.797	0.295	0.258	0.242	0.187	0.74	0.27
Avg		41.459	0.440	0.414	0.404	0.357	0.818	0.454
functions	Logistic	27.989	0.291	0.280	0.249	0.187	0.77	0.282
	MultilayerPerceptron	36.445	0.358	0.364	0.353	0.285	0.819	0.385
	SimpleLogistic	28.107	0.294	0.281	0.250	0.188	0.769	0.281
	SMO	29.360	0.335	0.294	0.265	0.21	0.745	0.243
Avg		30.475	0.319	0.304	0.279	0.217	0.775	0.297
Lazy	IBK	87.717	0.877	0.877	0.877	0.863	0.95	0.859
	Kstar	62.781	0.642	0.628	0.625	0.590	0.941	0.698
	LWL	23.439	0.267	0.234	0.373	0.284	0.742	0.289
Avg		57.979	0.595	0.579	0.625	0.579	0.877	0.615
Meta	AdaBoostM1	17.341	0.203	0.173	0.309	0.205	0.587	0.130
	AttributeSelectedClassifier	65.404	0.667	0.654	0.647	0.615	0.966	0.775
	Bagging	64.425	0.642	0.644	0.642	0.602	0.953	0.718
	ClassificationViaRegression	56.292	0.563	0.563	0.556	0.511	0.926	0.627
	FilteredClassifier	74.623	0.745	0.746	0.744	0.716	0.977	0.863
	IterativeClassifierOptimizer	34.400	0.359	0.344	0.338	0.270	0.810	0.356
	LogitBoost	34.400	0.359	0.344	0.338	0.270	0.810	0.356
	MultiClassClassifier	27.128	0.272	0.271	0.236	0.173	0.765	0.276
	MultiClassClassifierUpdateable	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	RandomCommittee	87.874	0.878	0.879	0.878	0.865	0.980	0.936

(Continued)

TABLE 9 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	RandomizableFilteredClassifier	87.336	0.873	0.873	0.873	0.859	0.949	0.860
	RandomSubSpace	74.584	0.745	0.746	0.743	0.716	0.974	0.822
	Stacking	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	WeightedInstancesHandlerWrapper	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	vot	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	CVParameterSelection	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		43.318	0.437	0.433	0.469	0.438	0.762	0.455
Rules	DecisionTable	65.551	0.667	0.656	0.648	0.616	0.966	0.748
	JRip	45.221	0.605	0.452	0.463	0.437	0.817	0.463
	OneR	63.897	0.646	0.639	0.625	0.594	0.798	0.448
	PART	59.776	0.597	0.598	0.597	0.551	0.945	0.708
	ZeroR	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		49.661	0.5308	0.4968	0.5152	0.4882	0.805	0.4944
Misc	InputMappedClassifier	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		13.857	0.139	0.139	0.243	0.243	0.499	0.105

TABLE 10 Comparative analysis of 40 classifiers utilizing feature selection via CAE on dataset DS2.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	91.886	0.919	0.919	0.919	0.91	0.971	0.903
	Random forest	92.170	0.922	0.922	0.922	0.913	0.991	0.947
	REPTree	57.349	0.571	0.573	0.571	0.522	0.931	0.635
	DecisionStump	18.183	0.14	0.182	0.243	0.135	0.596	0.149
	HoeffdingTree	27.402	0.232	0.274	0.233	0.167	0.720	0.265
	J4B	62.301	0.621	0.623	0.621	0.577	0.951	0.741
Avg		58.215	0.567	0.582	0.584	0.537	0.86	0.606
Bayes	BayesNet	70.992	0.713	0.710	0.707	0.677	0.972	0.807
	NaiveBayes	27.216	0.230	0.272	0.231	0.164	0.72	0.265
	NaiveBayesUpdateable	27.216	0.230	0.272	0.231	0.164	0.72	0.265
Avg		41.808	0.391	0.418	0.389	0.335	0.804	0.445
Functions	Logistic	26.590	0.145	0.266	0.155	−0.008	0.716	0.245
	SGD	43.933	0.351	0.439	0.325	0.211	0.767	0.401
	SimpleLogistic	26.561	0.146	0.266	0.155	0.006	0.715	0.244
	SMO	29.477	0.185	0.295	0.06	0.004	0.708	0.217
Avg		31.640	0.206	0.316625	0.173	0.048	0.726	0.238
lazy	IBK	91.661	0.917	0.917	0.917	0.907	0.954	0.875
	Kstar	53.210	0.543	0.532	0.517	0.474	0.914	0.569
	LWL	23.576	0.446	0.236	0.159	0.187	0.724	0.250
Avg		56.149	0.635	0.561	0.531	0.522	0.864	0.564
Meta	AdaBoostM1	18.183	0.140	0.182	0.243	0.135	0.596	0.149

(Continued)

TABLE 10 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	AttributeSelectedClassifier	65.404	0.667	0.654	0.647	0.615	0.966	0.775
	Bagging	68.036	0.677	0.68	0.678	0.641	0.961	0.754
	ClassificationViaRegression	58.896	0.583	0.589	0.581	0.535	0.934	0.643
	FilteredClassifier	73.106	0.730	0.731	0.728	0.699	0.975	0.848
	IterativeClassifierOptimizer	33.940	0.396	0.339	0.316	0.266	0.785	0.342
	LogitBoost	33.940	0.396	0.339	0.316	0.266	0.785	0.342
	MultiClassClassifier	25.944	0.142	0.259	0.152	0.053	0.715	0.239
	MultiClassClassifierUpdateable	13.857	0.139	0.139	0.243	0.499	0.499	0.105
	RandomCommittee	91.935	0.920	0.919	0.919	0.910	0.987	0.962
	RandomizableFilteredClassifier	90.781	0.908	0.908	0.908	0.897	0.957	0.881
	RandomSubSpace	76.874	0.770	0.769	0.767	0.742	0.978	0.849
	Stacking	13.857	0.139	0.139	0.243	0.139	0.499	0.105
	WeightedInstancesHandlerWrapper	55.694	0.557	0.557	0.715	0.500	0.500	0.506
	vot	13.857	0.139	0.139	0.243	0.139	0.499	0.105
	CVParameterSelection	13.857	0.139	0.139	0.243	0.139	0.499	0.105
Avg		46.760	0.465	0.467	0.496	0.448	0.758	0.470
Rules	DecisionTable	65.394	0.668	0.654	0.645	0.614	0.967	0.755
	JRip	44.969	0.625	0.450	0.460	0.439	0.808	0.453
	OneR	63.897	0.646	0.639	0.625	0.594	0.798	0.448
	PART	62.771	0.625	0.628	0.625	0.582	0.950	0.738
	ZeroR	13.857	0.139	0.139	0.243	0.139	0.499	0.105
Avg		50.178	0.540	0.502	0.519	0.476	0.804	0.499
Misc	InputMappedClassifier	13.857	0.139	0.139	0.243	0.139	0.499	0.105
Avg		13.857	0.139	0.139	0.243	0.139	0.499	0.105

TABLE 11 Comparative analysis of 40 classifiers utilizing feature selection via GRAE, on dataset DS2.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	93.668	0.937	0.937	0.937	0.930	0.974	0.913
	Random forest	93.844	0.938	0.938	0.938	0.931	0.992	0.957
	REPTree	61.460	0.611	0.615	0.611	0.568	0.942	0.678
	DecisionStump	17.341	0.203	0.173	0.309	0.205	0.587	0.130
	HoeffdingTree	30.779	0.353	0.308	0.297	0.245	0.764	0.308
	LMT	77.745	0.778	0.777	0.777	0.752	0.976	0.852
	J4B	67.165	0.672	0.672	0.67	0.634	0.957	0.772
Avg		63.143	0.641	0.631	0.648	0.609	0.884	0.651
Bayes	BayesNet	72.959	0.731	0.730	0.729	0.699	0.975	0.828
	NaiveBayes	27.001	0.307	0.270	0.248	0.197	0.746	0.295
	NaiveBayesUpdateable	27.001	0.307	0.270	0.715	0.197	0.746	0.295
Avg		42.320	0.448	0.423	0.564	0.364	0.822	0.477

(Continued)

TABLE 11 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Functions	Logistic	30.847	0.253	0.308	0.322	0.279	0.763	0.301
	SimpleLogistic	30.739	0.251	0.307	0.317	0.274	0.763	0.301
	SMO	32.530	0.401	0.325	0.308	0.247	0.748	0.262
	MultilayerPerceptron	43.305	0.426	0.433	0.418	0.36	0.857	0.467
Avg		88.665	0.888	0.886	0.886	0.771	0.910	0.887
Lazy	IBK	93.550	0.936	0.936	0.935	0.928	0.958	0.886
	Kstar	67.009	0.67	0.67	0.665	0.631	0.955	0.734
	LWL	23.791	0.276	0.238	0.38	0.292	0.756	0.303
Avg		88.652	0.891	0.886	0.885	0.773	0.927	0.912
Meta	AdaBoostM1	17.341	0.203	0.173	0.309	0.205	0.587	0.130
	AttributeSelectedClassifier	65.404	0.667	0.654	0.647	0.615	0.966	0.775
	Bagging	70.630	0.704	0.706	0.704	0.671	0.967	0.780
	ClassificationViaRegression	62.634	0.621	0.626	0.620	0.579	0.944	0.680
	FilteredClassifier	75.543	0.754	0.755	0.754	0.727	0.978	0.873
	IterativeClassifierOptimizer	35.036	0.347	0.350	0.340	0.269	0.813	0.381
	LogitBoost	35.036	0.347	0.350	0.340	0.269	0.813	0.381
	MultiClassClassifier	30.295	0.286	0.303	0.320	0.275	0.762	0.300
	MultiClassClassifierUpdateable	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	RandomCommittee	93.707	0.937	0.937	0.937	0.930	0.987	0.965
	RandomizableFilteredClassifier	90.634	0.906	0.906	0.906	0.896	0.959	0.885
	RandomSubSpace	77.432	0.774	0.774	0.773	0.748	0.977	0.848
	Stacking	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	WeightedInstancesHandlerWrapper	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	vot	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	CVParameterSelection	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		45.186	0.452	0.451	0.491	0.462	0.765	0.470
Rules	DecisionTable	65.551	0.667	0.656	0.648	0.616	0.966	0.748
	JRip	52.329	0.641	0.523	0.536	0.506	0.861	0.533
	OneR	63.897	0.646	0.639	0.625	0.594	0.798	0.448
	PART	66.901	0.671	0.669	0.668	0.631	0.956	0.764
	ZeroR	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		81.649	0.823	0.816	0.844	0.713	0.829	0.821
Misc	InputMappedClassifier	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		55.694	0.557	0.557	0.715	0.500	0.500	0.506

TABLE 12 Comparative analysis of 40 classifiers utilizing feature selection via IGAE, on dataset DS2.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	Random tree	93.707	0.937	0.937	0.937	0.930	0.972	0.908
	Random forest	93.707	0.937	0.937	0.937	0.930	0.993	0.959

(Continued)

TABLE 12 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	REPTree	59.238	0.591	0.592	0.509	0.543	0.935	0.652
	DecisionStump	18.183	0.140	0.182	0.243	0.135	0.596	0.149
	HoeffdingTree	26.893	0.249	0.269	0.214	0.174	0.732	0.272
	LMT	78.606	0.788	0.786	0.786	0.762	0.974	0.859
	J4B	67.175	0.671	0.672	0.671	0.633	0.956	0.781
Avg		62.501	0.616	0.625	0.625	0.586	0.879	0.654
Bayes	BayesNet	72.998	0.731	0.73	0.728	0.699	0.973	0.821
	NaiveBayes	71.824	0.723	0.718	0.714	0.685	0.965	0.789
	NaiveBayesUpdateable	26.893	0.251	0.269	0.214	0.175	0.732	0.272
Avg		57.238	0.568	0.572	0.552	0.519	0.890	0.633
Functions	Logistic	27.725	0.072	0.277	0.035	0.008	0.731	0.245
	SimpleLogistic	27.676	0.078	0.277	0.032	0.01	0.73	0.244
	SMO	28.831	0.157	0.288	0.064	0.051	0.713	0.213
	MultilayerPerceptron	41.837	0.423	0.418	0.412	0.348	0.839	0.431
Avg		88.665	0.888	0.886	0.886	0.771	0.910	0.887
Lazy	IBK	93.198	0.932	0.932	0.932	0.924	0.951	0.871
	Kstar	95.649	0.957	0.956	0.956	0.912	0.951	0.994
	LWL	23.791	0.345	0.238	0.058	0.009	0.739	0.267
Avg		70.879	0.744	0.706	0.648	0.642	0.880	0.710
Meta	AdaBoostM1	18.183	0.140	0.182	0.243	0.135	0.596	0.149
	AttributeSelectedClassifier	65.404	0.667	0.654	0.647	0.615	0.966	0.775
	Bagging	71.207	0.701	0.712	0.701	0.677	0.967	0.786
	ClassificationViaRegression	64.083	0.634	0.641	0.634	0.594	0.943	0.695
	FilteredClassifier	77.363	0.771	0.774	0.771	0.746	0.979	0.885
	IterativeClassifierOptimizer	36.191	0.386	0.362	0.344	0.286	0.799	0.306
	LogitBoost	36.191	0.386	0.362	0.344	0.286	0.799	0.306
	MultiClassClassifier	27.500	0.047	0.275	0.009	−0.004	0.731	0.244
	MultiClassClassifierUpdateable	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	RandomCommittee	93.746	0.937	0.937	0.937	0.903	0.988	0.968
	RandomizableFilteredClassifier	91.231	0.912	0.912	0.912	0.902	0.955	0.878
	RandomSubSpace	80.456	0.806	0.805	0.803	0.781	0.981	0.878
	Stacking	13.857	0.139	0.139	0.243	0.243	0.105	0.105
	WeightedInstancesHandlerWrapper	13.857	0.139	0.139	0.243	0.243	0.105	0.105
	vot	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	CVParameterSelection	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		45.677	0.443	0.456	0.473	0.447	0.713	0.468
Rules	DecisionTable	65.394	0.668	0.654	0.645	0.614	0.967	0.755
	JRip	48.375	0.657	0.484	0.494	0.476	0.826	0.492
	OneR	63.89	0.646	0.639	0.625	0.594	0.798	0.448
	PART	67.381	0.673	0.674	0.673	0.636	0.957	0.783

(Continued)

TABLE 12 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	ZeroR	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		51.781	0.556	0.518	0.536	0.5126	0.809	0.516
Misc	InputMappedClassifier	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		13.857	0.139	0.139	0.243	0.243	0.499	0.105

TABLE 13 Comparative analysis of 40 classifiers utilizing feature selection via PC, on dataset DS2.

Learning S	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	69.436	0.699	0.694	0.694	0.661	0.960	0.810
	Random forest	69.397	0.699	0.694	0.694	0.661	0.965	0.812
	REPTree	58.103	0.588	0.581	0.582	0.535	0.941	0.679
	DecisionStump	17.341	0.203	0.173	0.309	0.205	0.587	0.130
	HoeffdingTree	29.536	0.352	0.295	0.274	0.203	0.765	0.293
	LMT	61.391	0.619	0.614	0.614	0.572	0.953	0.736
	J4B	59.023	0.597	0.59	0.591	0.545	0.946	0.706
Avg		52.032	0.536	0.520	0.536	0.487	0.873	0.595
Bayes	BayesNet	30.133	0.332	0.301	0.295	0.233	0.779	0.313
	NaiveBayes	27.353	0.374	0.274	0.258	0.219	0.713	0.282
	NaiveBayesUpdateable	27.353	0.374	0.274	0.258	0.219	0.713	0.282
Avg		28.280	0.306	0.283	0.273	0.223	0.735	0.292
Functions	Logistic	26.629	0.308	0.266	0.245	0.185	0.748	0.292
	SimpleLogistic	26.688	0.311	0.267	0.243	0.186	0.745	0.285
	SMO	30.309	0.345	0.303	0.291	0.231	0.746	0.255
	MultilayerPerceptron	42.464	0.447	0.425	0.429	0.367	0.840	0.469
Avg		88.665	0.888	0.886	0.886	0.771	0.910	0.887
Lazy	IBK	69.436	0.699	0.694	0.693	0.661	0.96	0.809
	Kstar	59.630	0.617	0.596	0.602	0.559	0.941	0.678
	LWL	24.016	0.261	0.24	0.372	0.286	0.748	0.293
Avg		88.652	0.891	0.886	0.885	0.773	0.922	0.913
Meta	AdaBoostM1	17.341	0.203	0.173	0.309	0.205	0.587	0.13
	AttributeSelectedClassifier	48.990	0.500	0.490	0.488	0.433	0.918	0.597
	Bagging	61.489	0.622	0.615	0.617	0.574	0.952	0.727
	ClassificationViaRegression	55.744	0.597	0.557	0.559	0.502	0.937	0.659
	FilteredClassifier	53.131	0.545	0.531	0.534	0.482	0.932	0.643
	IterativeClassifierOptimizer	68.751	0.316	0.312	0.294	0.231	0.807	0.361
	LogitBoost	31.248	0.316	0.312	0.294	0.231	0.807	0.361
	MultiClassClassifier	26.668	0.125	0.267	0.182	0.121	0.704	0.273
	MultiClassClassifierUpdateable	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	RandomCommittee	69.358	0.698	0.694	0.693	0.606	0.965	0.825
	RandomizableFilteredClassifier	68.408	0.688	0.684	0.683	0.649	0.958	0.825

(Continued)

TABLE 13 (Continued)

Learning S	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	RandomSubSpace	51.712	0.537	0.517	0.517	0.468	0.903	0.563
	Stacking	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	WeightedInstancesHandlerWrapper	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	vot	13.857	0.139	0.139	0.243	0.243	0.499	0.105
	CVParameterSelection	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		38.883	0.364	0.365	0.399	0.361	0.750	0.404
Rules	DecisionTable	50.166	0.531	0.502	0.507	0.455	0.913	0.568
	JRip	46.604	0.637	0.466	0.493	0.406	0.857	0.52
	OneR	21.364	0.191	0.214	0.199	0.110	0.558	0.132
	PART	58.025	0.590	0.580	0.581	0.534	0.944	0.691
	ZeroR	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		38.003	0.417	0.380	0.404	0.360	0.754	0.402
Misc	InputMappedClassifier	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		55.694	0.557	0.557	0.715	0.500	0.500	0.506

TABLE 14 Comparative analysis of 40 classifiers utilizing feature selection via CAE, on dataset DS3.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	87.785	0.878	0.878	0.878	0.745	0.874	0.835
	Random forest	92.219	0.922	0.922	0.922	0.837	0.972	0.972
	REPTree	89.350	0.893	0.894	0.893	0.776	0.942	0.930
	DecisionStump	75.527	0.753	0.755	0.75	0.476	0.747	0.729
	HoeffdingTree	71.897	0.720	0.719	0.72	0.414	0.753	0.754
	J4B	90.023	0.900	0.900	0.900	0.790	0.927	0.903
Avg		84.467	0.844	0.844	0.843	0.673	0.869	0.853
Bayes	BayesNet	86.589	0.866	0.866	0.865	0.717	0.938	0.942
	NaiveBayes	72.397	0.820	0.724	0.721	0.547	0.924	0.913
	NaiveBayesUpdateable	72.397	0.820	0.724	0.721	0.547	0.924	0.913
Avg		77.128	0.835	0.771	0.769	0.603	0.928	0.667
Functions	Logistic	88.133	0.881	0.881	0.880	0.750	0.945	0.944
	MultilayerPerceptron	88.111	0.881	0.881	0.881	0.750	0.938	0.939
	SGD	87.589	0.877	0.876	0.874	0.738	0.860	0.825
	SimpleLogistic	87.980	0.880	0.880	0.879	0.746	0.944	0.944
	SMO	85.807	0.861	0.858	0.855	0.701	0.837	0.801
Avg		87.524	0.876	0.875	0.873	0.737	0.904	0.890
Lazy	IBK	87.893	0.879	0.879	0.879	0.746	0.886	0.855
	Kstar	89.284	0.895	0.893	0.891	0.775	0.950	0.952
	LWL	79.069	0.794	0.791	0.783	0.555	0.869	0.859
Avg		85.416	0.856	0.854	0.851	0.692	0.901	8,667
Meta	AdaBoostM1	86.046	0.860	0.860	0.860	0.707	0.929	0.929

(Continued)

TABLE 14 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	AttributeSelectedClassifier	88.089	0.881	0.881	0.880	0.749	0.930	0.910
	Bagging	90.415	0.904	0.904	0.904	0.799	0.963	0.964
	ClassificationViaRegression	88.459	0.884	0.885	0.884	0.757	0.949	0.948
	FilteredClassifier	89.828	0.898	0.898	0.898	0.786	0.938	0.919
	IterativeClassifierOptimizer	87.937	0.879	0.879	0.879	0.746	0.941	0.941
	LogitBoost	87.937	0.879	0.879	0.879	0.746	0.941	0.941
	MultiClassClassifier	88.133	0.881	0.881	0.880	0.750	0.945	0.944
	MultiClassClassifierUpdateable	87.589	0.877	0.876	0.874	0.738	0.806	0.825
	RandomCommittee	91.632	0.916	0.916	0.916	0.824	0.960	0.951
	RandomizableFilteredClassifier	85.068	0.851	0.851	0.851	0.688	0.858	0.822
	RandomSubSpace	90.110	0.902	0.901	0.900	0.792	0.961	0.962
	Stacking	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	WeightedInstancesHandlerWrapper	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	vot	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	CVParameterSelection	60.595	0.606	0.606	0.755	0.755	0.499	0.522
Avg		81.476	0.814	0.814	0.851	0.756	0.825	0.821
Rules	DecisionTable	65.551	0.667	0.656	0.648	0.616	0.966	0.748
	JRip	45.224	0.605	0.452	0.463	0.437	0.817	0.463
	OneR	63.897	0.646	0.639	0.625	0.594	0.798	0.448
	PART	59.776	0.597	0.598	0.597	0.551	0.945	0.708
	ZeroR	13.857	0.139	0.139	0.243	0.243	0.499	0.105
Avg		49.661	0.530	0.496	0.515	0.488	0.805	0.494
Misc	InputMappedClassifier	60.595	0.606	0.606	0.755	0.243	0.499	0.105
Avg		60.595	0.606	0.606	0.755	0.243	0.499	0.105

TABLE 15 Comparative analysis of 40 classifiers utilizing feature selection via CAE, on dataset DS3.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	90.263	0.903	0.903	0.903	0.797	0.899	0.864
	Random forest	94.262	0.943	0.943	0.942	0.879	0.981	0.981
	REPTree	91.045	0.911	0.911	0.911	0.812	0.945	0.927
	DecisionStump	79.091	0.794	0.791	0.792	0.568	0.778	0.744
	HoeffdingTree	82.764	0.827	0.828	0.827	0.638	0.865	0.848
	J4B	91.871	0.919	0.919	0.919	0.829	0.928	0.898
Avg		88.216	0.882	0.882	0.882	0.753	0.899	0.876
Bayes	BayesNet	88.350	0.884	0.884	0.882	0.754	0.946	0.95
	NaiveBayes	86.763	0.867	0.868	0.867	0.721	0.922	0.911
	NaiveBayesUpdateable	86.763	0.867	0.868	0.867	0.721	0.922	0.911
Avg		87.292	0.872	0.873	0.872	0.732	0.93	0.924
Functions	Logistic	89.632	0.897	0.896	0.895	0.782	0.952	0.949

(Continued)

TABLE 15 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	SGD	89.806	0.899	0.898	0.897	0.786	0.884	0.853
	MultilayerPerceptron	89.567	0.896	0.896	0.896	0.782	0.941	0.937
	SimpleLogistic	89.611	0.896	0.896	0.895	0.781	0.952	0.949
	SMO	87.980	0.882	0.88	0.878	0.747	0.861	0.828
Avg		89.319	0.894	0.893	0.892	0.775	0.918	0.903
Lazy	IBK	89.654	0.896	0.897	0.896	0.782	0.896	0.867
	Kstar	90.436	0.91	0.904	0.902	0.802	0.954	0.955
	LWL	79.091	0.794	0.791	0.792	0.568	0.893	0.892
Avg		86.394	0.866	0.864	0.863	0.717	0.914	0.904
Meta	AdaBoostM1	90.197	0.902	0.902	0.902	0.794	0.957	0.957
	AttributeSelectedClassifier	91.436	0.914	0.914	0.914	0.82	0.939	0.918
	Bagging	92.501	0.925	0.925	0.925	0.843	0.969	0.965
	ClassificationViaRegression	91.002	0.910	0.911	0.911	0.811	0.961	0.958
	FilteredClassifier	91.545	0.915	0.915	0.915	0.822	0.931	0.912
	IterativeClassifierOptimizer	90.197	0.902	0.902	0.901	0.794	0.959	0.959
	LogitBoost	90.197	0.902	0.902	0.901	0.794	0.959	0.959
	MultiClassClassifier	89.632	0.897	0.896	0.895	0.782	0.952	0.949
	MultiClassClassifierUpdateable	89.806	0.899	0.898	0.897	0.786	0.884	0.853
	RandomCommittee	93.284	0.933	0.933	0.933	0.859	0.971	0.965
	RandomizableFilteredClassifier	84.938	0.85	0.849	0.849	0.685	0.856	0.819
	RandomSubSpace	91.958	0.92	0.92	0.919	0.831	0.972	0.972
	Stacking	60.595	0.606	0.606	0.606	0.755	0.499	0.522
	WeightedInstancesHandlerWrapper	60.595	0.606	0.606	0.606	0.755	0.499	0.522
	vot	60.595	0.606	0.606	0.606	0.755	0.499	0.522
	CVParameterSelection	60.595	0.606	0.606	0.606	0.755	0.499	0.522
Avg		83.067	0.830	0.83	0.830	0.790	0.831	0.829
Rules	DecisionTable	89.611	0.897	0.896	0.895	0.781	0.945	0.945
	JRip	91.784	0.918	0.918	0.918	0.827	0.926	0.916
	OneR	78.330	0.781	0.783	0.783	0.541	0.766	0.721
	PART	91.871	0.919	0.919	0.919	0.829	0.943	0.924
	ZeroR	60.595	0.606	0.606	0.755	0.499	0.499	0.522
Avg		82.438	0.824	0.824	0.854	0.6954	0.8158	0.805
Misc	InputMappedClassifier	60.595	0.606	0.606	0.755	0.499	0.499	0.522
Avg		60.595	0.606	0.606	0.755	0.499	0.499	0.522

TABLE 16 Comparative analysis of 40 classifiers utilizing feature selection via GRAE, on dataset DS3.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	91.567	0.916	0.916	0.916	0.823	0.912	0.884
	Random forest	93.740	0.937	0.937	0.937	0.869	0.977	0.977

(Continued)

TABLE 16 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	REPTree	91.567	0.916	0.916	0.916	0.823	0.912	0.884
	DecisionStump	78.048	0.789	0.778	0.775	0.532	0.773	0.758
	HoeffdingTree	77.483	0.773	0.775	0.773	0.523	0.798	0.794
	LMT	92.805	0.928	0.928	0.928	0.849	0.962	0.955
	J4B	92.110	0.921	0.921	0.921	0.834	0.949	0.935
Avg		88.188	0.881	0.881	0.880	0.750	0.897	0.857
Bayes	BayesNet	91.523	0.916	0.915	0.915	0.822	0.971	0.971
	NaiveBayes	76.505	0.838	0.765	0.765	0.603	0.949	0.940
	NaiveBayesUpdateable	76.505	0.838	0.765	0.765	0.603	0.949	0.940
Avg		81.511	0.864	0.815	0.815	0.676	0.956	0.950
Functions	Logistic	91.436	0.914	0.914	0.914	0.821	0.967	0.964
	SimpleLogistic	91.371	0.914	0.914	0.913	0.818	0.967	0.963
	SMO	88.154	0.885	0.882	0.879	0.752	0.862	0.83
	MultilayerPerceptron	92.545	0.926	0.925	0.925	0.844	0.963	0.959
Avg		88.665	0.888	0.886	0.886	0.771	0.910	0.887
Lazy	IBK	90.915	0.909	0.909	0.909	0.909	0.909	0.885
	Kstar	91.588	0.921	0.916	0.914	0.827	0.971	0.972
	LWL	77.874	0.777	0.779	0.775	0.528	0.855	0.858
Avg		88.652	0.891	0.886	0.885	0.773	0.927	0.913
Meta	AdaBoostM1	77.874	0.777	0.779	0.775	0.528	0.855	0.858
	AttributeSelectedClassifier	92.219	0.922	0.922	0.922	0.837	0.951	0.938
	Bagging	93.066	0.931	0.931	0.930	0.854	0.974	0.973
	ClassificationViaRegression	90.893	0.909	0.909	0.908	0.808	0.965	0.962
	FilteredClassifier	92.916	0.929	0.929	0.929	0.851	0.942	0.928
	IterativeClassifierOptimizer	90.806	0.908	0.908	0.907	0.807	0.963	0.963
	LogitBoost	90.806	0.908	0.908	0.907	0.807	0.963	0.963
	MultiClassClassifier	91.436	0.914	0.914	0.914	0.802	0.967	0.964
	MultiClassClassifierUpdateable	90.697	0.908	0.907	0.906	0.804	0.895	0.866
	RandomCommittee	93.544	0.935	0.935	0.935	0.864	0.961	0.951
	RandomizableFilteredClassifier	90.197	0.902	0.902	0.902	0.794	0.904	0.880
	RandomSubSpace	92.653	0.927	0.927	0.926	0.846	0.975	0.975
	Stacking	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	WeightedInstancesHandlerWrapper	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	vot	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	CVParameterSelection	60.595	0.606	0.606	0.755	0.755	0.499	0.522
Avg		83.093	0.830	0.830	0.867	0.752	0.831	0.831
Rules	DecisionTable	90.415	0.904	0.904	0.904	0.798	0.95	0.951
	JRip	91.154	0.911	0.912	0.911	0.814	0.925	0.917
	OneR	78.330	0.781	0.783	0.782	0.541	0.766	0.72
	PART	93.001	0.93	0.93	0.93	0.853	0.969	0.963

(Continued)

TABLE 16 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	ZeroR	60.595	0.606	0.606	0.755	0.755	0.499	0.522
Avg		81.649	0.822	0.816	0.846	0.713	0.829	0.821
Misc	InputMappedClassifier	60.595	0.606	0.606	0.755	0.755	0.499	0.522
		55.694	0.557	0.557	0.715	0.505	0.506	0.506

TABLE 17 Comparative analysis of 40 classifiers utilizing feature selection via IGAE, on dataset DS3.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	91.154	0.912	0.912	0.912	0.815	0.908	0.876
	Random forest	94.631	0.946	0.946	0.946	0.887	0.983	0.982
	REPTree	59.238	0.925	0.925	0.925	0.842	0.956	0.945
	DecisionStump	78.048	0.78	0.78	0.775	0.532	0.773	0.758
	HoeffdingTree	85.785	0.858	0.858	0.856	0.772	0.883	0.865
	LMT	92.719	0.927	0.927	0.927	0.847	0.965	0.962
	J4B	92.262	0.923	0.923	0.923	0.838	0.934	0.916
Avg		84.834	0.895	0.895	0.894	0.783	0.914	0.901
Bayes	BayesNet	88.502	0.885	0.885	0.884	0.758	0.953	0.955
	NaiveBayes	88.22	0.885	0.882	0.88	0.753	0.946	0.936
	NaiveBayesUpdateable	88.22	0.885	0.882	0.88	0.753	0.946	0.936
Avg		88.314	0.885	0.883	0.881	0.754	0.948	0.943
Functions	Logistic	90.697	0.907	0.907	0.906	0.804	0.963	0.961
	SimpleLogistic	90.632	0.906	0.906	0.906	0.803	0.963	0.961
	SMO	88.002	0.883	0.88	0.878	0.748	0.861	0.828
	MultilayerPerceptron	90.806	0.908	0.908	0.908	0.807	0.958	0.954
Avg		88.665	0.888	0.886	0.886	0.771	0.910	0.887
Lazy	IBK	90.241	0.902	0.902	0.902	0.795	0.9	0.872
	Kstar	89.980	0.905	0.9	0.898	0.793	0.951	0.952
	LWL	78.417	0.783	0.784	0.78	0.54	0.862	0.867
Avg		86.213	0.863	0.862	0.86	0.709	0.904	0.897
Meta	AdaBoostM1	89.611	0.896	0.896	0.896	0.781	0.957	0.957
	AttributeSelectedClassifier	92.240	0.922	0.922	0.922	0.837	0.947	0.93
	Bagging	93.631	0.936	0.936	0.936	0.866	0.974	0.973
	ClassificationViaRegression	92.501	0.925	0.925	0.925	0.843	0.967	0.963
	FilteredClassifier	92.479	0.925	0.925	0.925	0.842	0.935	0.919
	IterativeClassifierOptimizer	90.980	0.901	0.901	0.909	0.801	0.964	0.964
	LogitBoost	90.980	0.901	0.901	0.909	0.811	0.964	0.964
	MultiClassClassifier	90.697	0.907	0.907	0.906	0.804	0.963	0.961
	MultiClassClassifierUpdateable	90.349	0.904	0.903	0.903	0.797	0.891	0.891
	RandomCommittee	94.196	0.942	0.942	0.942	0.878	0.978	0.973
	RandomizableFilteredClassifier	83.438	0.834	0.834	0.834	0.652	0.839	0.802

(Continued)

TABLE 17 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	RandomSubSpace	92.979	0.934	0.934	0.934	0.852	0.975	0.975
	Stacking	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	WeightedInstancesHandlerWrapper	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	vot	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	CVParameterSelection	60.595	0.606	0.606	0.755	0.755	0.499	0.522
Avg		83.529	0.833	0.835	0.872	0.799	0.834	0.835
Rules	DecisionTable	89.676	0.898	0.897	0.895	0.783	0.946	0.946
	JRip	92.436	0.924	0.924	0.924	0.841	0.926	0.912
	OneR	78.330	0.781	0.783	0.782	0.541	0.766	0.72
	PART	92.349	0.923	0.923	0.923	0.839	0.96	0.951
	ZeroR	60.595	0.606	0.606	0.755	0.755	0.499	0.522
Avg		82.677	0.826	0.826	0.855	0.751	0.819	0.812
Misc	InputMappedClassifier	60.595	0.606	0.606	0.755	0.755	0.499	0.522
Avg		60.595	0.606	0.606	0.755	0.755	0.499	0.522

TABLE 18 Comparative analysis of 40 classifiers utilizing feature selection via PC, on dataset DS3.

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
Tree	Random tree	89.567	0.895	0.896	0.896	0.781	0.895	0.863
	Random forest	92.784	0.928	0.928	0.928	0.848	0.971	0.971
	REPTree	90.154	0.901	0.902	0.901	0.793	0.939	0.926
	DecisionStump	76.266	0.811	0.763	0.735	0.522	0.693	0.699
	HoeffdingTree	87.068	0.871	0.871	0.870	0.728	0.909	0.891
	LMT	61.391	0.619	0.614	0.614	0.572	0.953	0.736
	J4B	91.132	0.911	0.911	0.911	0.814	0.931	0.911
Avg		84.052	0.847	0.840	0.836	0.722	0.898	0.714
Bayes	BayesNet	87.198	0.873	0.872	0.877	0.713	0.935	0.939
	NaiveBayes	87.198	0.873	0.872	0.877	0.713	0.934	0.924
	NaiveBayesUpdateable	87.198	0.873	0.872	0.879	0.713	0.934	0.924
Avg		87.198	0.873	0.872	0.87	0.599	0.934	0.929
Functions	Logistic	88.958	0.889	0.89	0.889	0.767	0.952	0.952
	SimpleLogistic	88.567	0.886	0.886	0.884	0.759	0.952	0.949
	SMO	86.742	0.867	0.867	0.865	0.721	0.847	0.812
	MultilayerPerceptron	89.611	0.896	0.896	0.896	0.782	0.943	0.943
		88.665	0.888	0.886	0.886	0.771	0.910	0.888
Lazy	IBK	89.437	0.894	0.894	0.894	0.777	0.895	0.873
	Kstar	88.611	0.893	0.886	0.883	0.765	0.938	0.941
	LWL	76.266	0.811	0.763	0.735	0.522	0.894	0.893
Avg		88.652	0.891	0.886	0.885	0.773	0.927	0.912
Meta	AdaBoostM1	87.524	0.875	0.875	0.874	0.737	0.94	0.942

(Continued)

TABLE 18 (Continued)

Learning strategy	Classifier	Accuracy	Precision	Recall	F-measure	MCC	ROC area	PRC area
	AttributeSelectedClassifier	89.458	0.895	0.895	0.894	0.778	0.934	0.918
	Bagging	91.588	0.916	0.916	0.916	0.823	0.962	0.961
	ClassificationViaRegression	88.893	0.890	0.889	0.888	0.766	0.942	0.939
	FilteredClassifier	90.697	0.907	0.907	0.907	0.804	0.935	0.92
	IterativeClassifierOptimizer	89.067	0.891	0.891	0.89	0.707	0.949	0.951
	LogitBoost	89.067	0.891	0.891	0.89	0.707	0.949	0.951
	MultiClassClassifier	88.958	0.890	0.889	0.889	0.767	0.952	0.952
	MultiClassClassifierUpdateable	88.133	0.883	0.881	0.88	0.765	0.864	0.831
	RandomCommittee	92.110	0.921	0.921	0.921	0.834	0.959	0.95
	RandomizableFilteredClassifier	86.655	0.866	0.867	0.866	0.772	0.871	0.843
	RandomSubSpace	91.219	0.913	0.912	0.911	0.816	0.964	0.966
	Stacking	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	WeightedInstancesHandlerWrapper	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	vot	60.595	0.606	0.606	0.755	0.755	0.499	0.522
	CVParameterSelection	60.595	0.606	0.606	0.755	0.755	0.499	0.522
Avg		82.234	0.822	0.822	0.859	0.775	0.826	0.825
Rules	DecisionTable	89.241	0.892	0.892	0.892	0.773	0.935	0.937
	JRip	90.306	0.903	0.903	0.902	0.796	0.906	0.896
	OneR	74.570	0.743	0.746	0.742	0.458	0.723	0.679
	PART	89.915	0.899	0.899	0.898	0.788	0.944	0.934
	ZeroR	60.595	0.606	0.606	0.755	0.755	0.499	0.522
Avg		80.925	0.808	0.809	0.837	0.714	0.801	0.793
Misc	InputMappedClassifier	60.595	0.606	0.606	0.755	0.755	0.499	0.522
Avg		55.694	0.557	0.557	0.715	0.055	0.505	0.506

TABLE 19 Best classifier with respect to the evaluation metric and the dataset.

Dataset	Accuracy	Precision	Recall	MCC	F-measure	ROC area
DS1	IBK, RC	RC, RF	RF	RF	RT	RT
DS2	RT, RF	RT, RF	RT, RF	RT, RF	RT, RF	RF, RF
DS3	RF	RF	RF	RF	RF, RepTREE	RF, LOGISTIC

Besides, when focusing solely on optimizing the precision metric through a strategic approach perspective, adopting the tree learning strategy can be highly effective.

Table 11 presents the results of applying 40 classifiers to the phishing website dataset (DS2), focusing on recall and MCC.

According to Table 11, the Random Tree classifier performs exceptionally well on the Recall metric. At the same time, the Random Forest model achieves the best MCC among all considered classification models.

Furthermore, Trees prove themselves to be an exceptional learning strategy, producing superior output compared to seven alternative strategies from both recall and MCC perspectives.

The results obtained from the 40 classifiers applied to the phishing website dataset (DS2) for the recall and MCC metrics are presented in Table 12. Random tree classifier demonstrates superior recall, while the random forest and the random tree stand out with exceptional performance on MCC among the classification models considered. Also, compared to the seven learning strategies under review, Trees shows better results for both the Recall and MCC measures.

Additionally, these two classifiers have been most effective on this dataset, as indicated by their respective evaluation scores in Table 12.

The accuracy and precision metrics for the phishing dataset (DS2) were evaluated using 40 classifiers, and the results are presented in Table 13.

From the table, it is evident that the IBK model under the lazy learning strategy, along with the random tree model under the tree learning approach, achieved the highest accuracy and precision values.

Furthermore, based on the findings in Table 13 regarding optimizing the precision metric for the Learning Strategy factor, Tree Learning should be selected for its superior performance.

Table 14 displays the results of forty classifiers applied to a dataset (DS3) containing phishing websites. The evaluation metrics were F-measure and ROC area. Among these, the random forest classifier showed exceptional performance in both F-measure and ROC, compared with all seven learning strategies under scrutiny. Additionally, Trees displayed better outcomes than others on both measures.

From Table 14, the scores for each evaluation method indicate that, among the classifiers tested, they were most efficient on this dataset when compared with the other methods employed herein.

The results of running 40 classifiers on the phishing website dataset (DS3) are shown in Table 15, including F-measure and ROC metrics. According to the table, the random forest classifier outperforms other classification models on both F-measure and ROC for this dataset.

Additionally, Trees is the most effective learning strategy for achieving high marks on both evaluation measures among the seven strategies considered here.

When 40 classifiers were applied to the phishing website dataset (DS3), Table 16 shows the results for both the F-measure and ROC metrics. According to this table, among the considered classification models, the random forest classifier achieves superior results in terms of F-measure and ROC on the same dataset. Besides, Trees, as a learning strategy, demonstrates top-notch performance across both evaluation criteria when juxtaposed with seven other strategies.

The results of applying 40 classifiers to the phishing website dataset, with respect to F-measure and ROC metrics, are shown in Table 17. The random forest classifier outperforms the other considered classification models on both measures for this dataset, as shown in Table 17.

Notably, Trees proves superior as a learning strategy, based on its performance across all evaluation criteria among the seven strategies compared here, particularly on F-measure and ROC metrics.

The results of using 40 classifiers on the phishing website detection dataset (DS3) are depicted in Table 18 and analyzed using accuracy and pre-session metrics. The data reveal that Random Forest achieves the best F-MEASURE and ROC scores, while other top-performing methods, Random Committee and J4B, also achieve outstanding F-MEASURE and ROC scores.

Evaluating the learning strategy indicates that Tree attains optimal results for F-MEASURE and ROC METRICS.

The summary of the comparative analysis of 40 classifiers across three datasets, as presented in Tables 4–18, is shown in Table 19. In this table, "RC" refers to a random committee, "RF" denotes random forest, and "RT" stands for random tree.

The study revealed that the classifier delivered superior results across the considered metrics and datasets. According to Table 11, the random tree classifier achieved superior results on 13 occasions, and the random forest classifiers were best seven times. Random committee classifiers besides IBK performed well twice.

This indicates that, for phishing datasets, random forest is the preferred option, compared with committee classifiers, which ranked second. The phishing website said this: the random forest classifier excelled on the phishing dataset, where all attributes were integer types, and it showcased excellent performance on the phishing website detection dataset, or even the phishing website dataset, which included integer types. Its exceptional ability to perform well regardless of the number/types of attributes makes it evident why random forest remains a preferred choice among classification techniques.

5 Best feature selection method to use with phishing website datasets

The objective of this section is to determine the optimal feature selection approach suitable for phishing datasets. To achieve this, five popular methods are assessed and compared: ClassifierAttributeEval (CAE), CorrelationAttributeEval (CAE), GainRatioAttributeEval (GRAI), InfoGainAttributeEval (IGAE), and principal components. The default settings and parameters in WEKA were utilized throughout all evaluations.

These feature selection methods were applied to a phishing dataset, with 40 classification models trained using only the top-performing 15 features, corresponding to 0.50% of the available attributes (30).

Moreover, the evaluation metrics used earlier, such as MCC, accuracy, and precision, will be analyzed again comprehensively within the same segment under Section Four's scope.

The phishing dataset underwent various feature selection methods, after which 40 classification models were trained using the top-performing 0.50% of features (15 out of 30). This section evaluates the accuracy, precision, and MCC metrics outlined in Section 4 using Table 20. Dissecting four feature selection techniques considered for this study, applied to the phishing dataset, and evaluated for accuracy, is showcased.

As shown in Table 20, the random forest and IBK classifiers achieved their highest accuracies with CAE as their selected method, also revealing the Functions strategy as superior across accuracy-based field analyses in this particular case.

In addition, Tree performed optimally when acclimated alongside CAE's specified attribute-selection methodology.

Table 20 clearly shows that using only 0.50% of the features generally improves accuracy.

For instance, random forest classifiers achieved the best accuracy result (96.2) when all features were used, while the random forest achieved (96.1). However, both classifiers attained their highest accuracy scores with the phishing dataset by utilizing GRAE, besides the IGAE feature selection method on just 0.50% of its features, resulting in an overall improvement based on Table 20 analysis evidence, which suggests that employing a preprocessing step, such as feature selection, may enhance predictive performance

TABLE 20 Evaluation of the considered feature selection methods on the phishing dataset-(DS1) using the accuracy metric.

Learning strategy	Classifier	CAE	CAE	GRAE	IGAE	PC
Tree	Random tree	90.502	95.911	95.649	95.640	94.219
	Random forest	90.664	96.436	96.255	96.191	94.473
	REPTree	89.561	94.898	94.850	94.744	93.523
	DecisionStump	84.730	88.891	88.891	88.891	88.891
	HoeffdingTree	88.801	94.002	93.930	93.903	93.062
	LMT	90.610	95.766	95.829	95.676	94.373
	J4B	90.031	95.450	95.630	95.106	93.794
Avg		89.271	0.236	94.434	94.307	93.191
Bayes	BayesNet	87.535	92.772	92.781	92.636	92.356
	NaiveBayes	87.535	92.772	92.781	92.645	92.365
	NaiveBayesUpd	55.694	92.772	55.694	55.694	92.365
Avg		76.921	0.173	80.419	80.325	92.362
Functions	Logistic	88.647	93.369	93.387	93.378	92.682
	SGD	88.738	93.306	93.351	93.514	91.705
	SimpleLogistic	88.629	93.306	93.351	93.432	92.645
	SMO	88.955	93.315	93.324	93.523	91.714
	VotedPerceptro	88.358	93.288	93.333	93.365	92.555
Avg		86.187	0.241	88.665	88.665	88.665
Lazy	IBK	90.755	96.119	95.829	95.730	94.237
	Kstar	90.393	96.128	95.983	95.649	94.165
	LWL	84.730	88.991	88.973	89.018	88.991
Avg		88.652	0.205	88.652	93.465	88.652
Meta	AdaBoostM1	87.435	92.582	92.582	92.582	92.166
	AttributeSelectedClassifier	87.363	94.400	94.400	94.310	92.935
	Bagging	89.977	95.486	95.404	95.386	93.830
	ClassificationViaRegression	89.036	94.536	94.436	94.635	93.188
	FilteredClassifier	90.031	95.450	95.630	95.106	93.794
	IterativeClassifierOptimizer	87.806	92.736	92.736	92.736	92.220
	LogitBoost	87.806	92.736	92.736	92.736	92.437
	MultiClassClassifier	88.647	93.369	93.387	93.378	92.682
	MultiClassClassifierUpdateable	88.738	93.306	93.351	93.514	93.830
	RandomCommittee	90.755	96.408	90.755	96.408	94.409
	RandomizableFilteredClassifier	90.230	94.292	90.230	94.771	90.230
	RandomSubSpace	89.027	93.414	93.984	93.450	92.691
	Stacking	55.694	55.694	55.694	55.694	55.694
	WeightedInstancesHandlerWrapper	55.694	55.694	55.694	55.694	55.694
	vot	55.694	55.694	55.694	55.694	55.694
	CVParameterSelection	55.694	55.694	55.694	55.694	55.694
Avg		80.602	79.996	87.202	84.487	86.661
Rules	DecisionTable	88.177	92.863	92.998	92.971	93.025

(Continued)

TABLE 20 (Continued)

Learning strategy	Classifier	CAE	CAE	GRAE	IGAE	PC
	JRip	89.271	94.753	94.527	94.563	93.306
	OneR	84.730	88.891	88.891	88.891	88.891
	PART	90.375	95.585	95.459	95.468	94.355
	ZeroR	55.694	55.694	55.694	55.694	55.694
Avg		81.649	85.557	81.649	85.517	85.054
Misc	InputMappedClassifier	55.694	55.694	55.694	55.694	55.694
Avg		55.694	55.694	55.694	55.694	55.694

TABLE 21 Evaluation of the considered feature selection methods on the phishing dataset-(DS1) using the precision metric.

Learning strategy	Classifier	CAE	CAE	GRAS	IGAE	PC
Tree	random tree	0.880	0.919	0.937	0.937	0.810
	Random forest	0.879	0.922	0.938	0.937	0.812
	REPTree	0.550	0.571	0.611	0.591	0.679
	DecisionStump	0.203	0.140	0.203	0.140	0.13
	HoeffdingTree	0.341	0.232	0.353	0.249	0.293
	LMT	0.602	0.621	0.778	0.788	0.736
	J4B	0.732	0.911	0.672	0.671	0.706
Avg		0.295	0.236	0.641	0.616	0.595
Bayes	BayesNet	0.295	0.713	0.731	0.731	0.313
	NaiveBayes	0.291	0.230	0.307	0.723	0.282
	NaiveBayesUpdateable	0.358	0.230	0.307	0.251	0.282
Avg		0.294	0.520	0.448	0.568	0.292
Functions	Logistic	0.335	0.145	0.253	0.072	0.292
	SGD	0.877	0.351	0.251	0.078	0.285
	SimpleLogistic	0.642	0.146	0.401	0.157	0.255
	SMO	0.267	0.185	0.426	0.423	0.469
Avg		0.667	0.241	88.665	88.665	88.676
Lazy	IBK	0.642	0.917	0.936	0.932	0.809
	Kstar	0.563	0.543	0.67	0.957	0.678
	LWL	0.745	0.446	0.276	0.345	0.293
Avg		0.359	0.205	88.657	0.744	88.652
Meta	AdaBoostM1	0.359	0.14	0.203	0.140	0.130
	AttributeSelectedClassifier	0.272	0.667	0.667	0.667	0.597
	Bagging	0.139	0.677	0.704	0.710	0.727
	ClassificationViaRegression	0.878	0.583	0.621	0.634	0.659
	FilteredClassifier	0.873	0.730	0.754	0.771	0.643
	IterativeClassifierOptimizer	0.745	0.396	0.347	0.386	0.361

(Continued)

TABLE 21 (Continued)

Learning strategy	Classifier	CAE	CAE	GRAS	IGAE	PC
	LogitBoost	0.139	0.396	0.347	0.386	0.361
	MultiClassClassifier	0.139	0.142	0.286	0.047	0.273
	MultiClassClassifierUpdateable	0.139	0.139	0.139	0.139	0.105
	RandomCommittee	0.139	0.920	0.937	0.937	0.825
	RandomizableFilteredClassifier	0.667	0.908	0.906	0.912	0.800
	RandomSubSpace	0.605	0.770	0.774	0.806	0.563
	Stacking	0.646	0.139	0.139	0.139	0.105
	WeightedInstancesHandlerWrapper	0.597	0.557	0.139	0.139	0.105
	vot	0.139	0.139	0.139	0.139	0.105
	CVParameterSelection	0.139	0.139	0.139	0.139	0.105
Avg		80.602	0.455	7.510	0.443	7.474
Rules	DecisionTable	0.667	0.668	0.667	0.668	0.568
	JRip	0.605	0.625	0.641	0.657	0.52
	OneR	0.646	0.646	0.646	0.646	0.132
	PART	0.597	0.625	0.671	0.673	0.691
	ZeroR	0.139	0.139	0.139	0.139	0.105
Avg		81.649	0.540	81.649	0.556	0.403
Misc	InputMappedClassifier	55.694	0.139	0.139	0.139	0.105
Avg		55.694	0.139	55.693	0.139	55.693

TABLE 22 Evaluation of the considered feature selection methods on the phishing dataset-(DS1) using the MCC metric.

Learning strategy	Classifier	CAE	CAE	GRAS	IGAE	PC
Tree	Random tree	0.745	0.797	0.823	0.815	0.781
	Random forest	0.837	0.879	0.869	0.887	0.848
	REPTree	0.776	0.812	0.823	0.842	0.793
	DecisionStump	0.476	0.568	0.532	0.532	0.522
	HoeffdingTree	0.414	0.638	0.523	0.700	0.728
	J4B	0.790	0.829	0.834	0.838	0.814
Avg		0.673	0.753	0.734	0.769	0.747
Bayes	BayesNet	0.717	0.754	0.822	0.758	0.730
	NaiveBayes	0.547	0.721	0.603	0.753	0.730
	NaiveBayesUpdateable	0.547	0.721	0.603	0.753	0.730
Avg		0.603	0.732	0.676	0.754	0.730
Functions	Logistic	0.750	0.782	0.820	0.804	0.767
	SGD	0.738	0.782	0.818	0.803	0.759
	SimpleLogistic	0.746	0.781	0.752	0.748	0.721
	SMO	0.701	0.747	0.844	0.807	0.782
Avg		0.733	0.773	0.7712	0.771	0.771
Lazy	IBK	0.746	0.782	0.909	0.795	0.777
	Kstar	0.775	0.802	0.827	0.793	0.765

(Continued)

TABLE 22 (Continued)

Learning strategy	Classifier	CAE	CAE	GRAS	IGAE	PC
	LWL	0.555	0.568	0.528	0.54	0.522
Avg		0.692	0.717	0.773	0.709	0.773
Meta	AdaBoostM1	0.707	0.794	0.528	0.781	0.737
	AttributeSelectedClassifier	0.749	0.820	0.837	0.837	0.778
	Bagging	0.799	0.843	0.854	0.866	0.823
	ClassificationViaRegression	0.757	0.811	0.808	0.843	0.766
	FilteredClassifier	0.786	0.822	0.851	0.842	0.804
	IterativeClassifierOptimizer	0.746	0.794	0.807	0.810	0.770
	LogitBoost	0.746	0.794	0.807	0.810	0.770
	MultiClassClassifier	0.750	0.782	0.820	0.804	0.767
	MultiClassClassifierUpdateable	0.738	0.786	0.804	0.797	0.750
	RandomCommittee	0.824	0.859	0.864	0.878	0.834
	RandomizableFilteredClassifier	0.688	0.685	0.794	0.652	0.72
	RandomSubSpace	0.792	0.831	0.846	0.852	0.816
	Stacking	0.755	0.755	0.755	0.755	0.755
	WeightedInstancesHandlerWrapper	0.755	0.755	0.755	0.755	0.755
	vot	0.755	0.755	0.755	0.755	0.755
	CVParameterSelection	0.755	0.755	0.755	0.755	0.755
Avg		0.756	0.790	0.790	0.799	0.772
Rules	DecisionTable	0.616	0.781	0.798	0.783	0.773
	JRip	0.437	0.827	0.814	0.841	0.796
	OneR	0.594	0.541	0.541	0.541	0.458
	PART	0.551	0.829	0.853	0.839	0.788
	ZeroR	0.243	0.499	0.755	0.755	0.755
Avg		0.488	0.6954	0.713	0.751	0.714
Misc	InputMappedClassifier	0.243	0.499	0.755	0.755	0.755
Avg		0.243	0.499	0.500	0.755	0.500

for various classification models specifically through adoption of the CAE technique.

The evaluation results for the phishing dataset using five feature selection methods are shown in Table 21, with emphasis on the precision metric. The Random Forest classifier using the GRAE method achieved the highest precision, yielding remarkable results regardless of the feature selection method.

Moreover, function and tree strategies proved to be efficient learning approaches for the precision metrics in this dataset. GRAE achieved the trees' maximum precision.

Comparing Table 21 (utilizing all features) and using only 0.50% percentiles confirms that a general improvement in precision results can be seen when utilizing fewer attributes such as those demonstrated in Table 1's findings; for instance, while utilizing every attribute resulted in a top score reaching 0.938, lessened usage proved more beneficial overall performance-wise across varying methodologies examined previously through these tables mentioned above.

Hence, according to Table 21, using feature selection as a preprocessing step may improve the overall predictive performance of most classification models.

Hence, according to Table 21, using feature selection as a preprocessing step may improve the overall predictive performance of most classification models,

Table 22 displays the assessment results for five feature selection techniques applied to the phishing dataset, using MCC as the metric. The Random Forest classifier achieved the highest MCC values with the IGAE technique, and the Functions strategy produced optimal learning results on this data set. The tree method achieved favorable results by applying IGAE for feature selection.

Moreover, comparing all features vs. using only 0.50% showed an improvement in overall performance when examined against

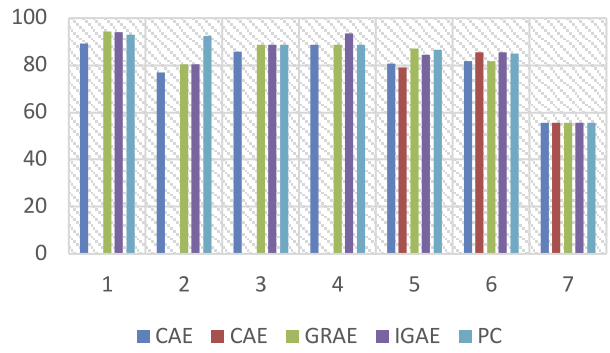

FIGURE 3
Evaluation of the best feature selection method to use with phishing website datasets-(DS1).

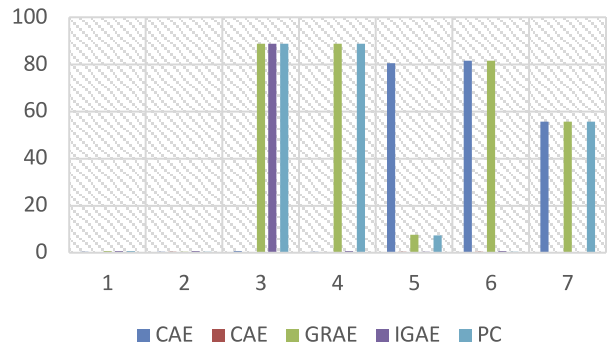

FIGURE 4
Evaluation of the best feature selection method to use with phishing website datasets-(DS2).

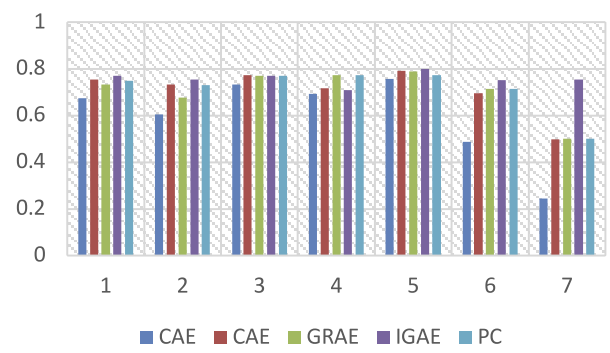

FIGURE 5
Evaluation of the best feature selection method to use with phishing website datasets-(DS3).

the MCC matrix, exemplified by the best-case scenario, in which using all available features yielded a score of 0.887 via Random Forest classification.

According to Table 15, considerable progress is expected in refined prediction accuracy across various classification models if appropriate feature selection is conducted during preprocessing, particularly when leveraging responsive methods such as those designated "IGEA."

The optimal approach to optimization is demonstrated in Figure 5, which shows that IGAE Feature selection reigns supreme.

Figures 3–5 reveal that the feature selection IGAE and GRAE, in addition to the tree learning strategy, exhibit superior performance compared to other strategies in terms of accuracy, precision, recall, MCC, F-measure, and ROC area across three datasets. Moreover, the rules and misc learning strategies demonstrate subpar results across almost all metrics for those same three datasets.

Consequently, it is strongly advised against using rules other than the misc learning strategy course of study for phishing detection.

6 Conclusion and future research

This research aimed to identify optimal characteristics for creating a stronger machine learning model for detecting phishing websites. Over the past three decades, machine learning has made significant strides and has been implemented in many practical applications, including identifying malicious web pages used in scams or identity theft.

The paper investigates the best classification model for detecting these site types. While exploring which classification method would best handle phishing website detection datasets, the author discovered that an ensemble approach combining Random Forest, Random Tree, and IBK classifiers proved most effective. In conclusion, after evaluating several feature selection methods for detecting fraudulent websites, InfoGainAttributeEval and GainRatioAttributeEval were deemed reliable options. However, further appraisals focusing on variables such as the additional classification styles mentioned above should continue to be considered alongside other metrics. Comparing their performance will provide additional insight into refining detection accuracy for tracing illicit online activity.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://data.mendeley.com/datasets/h3cgnj8hft/1.

Author contributions

RA: Writing – original draft, Writing – review & editing. MB: Writing – review & editing, Writing – original draft, Data curation. KA: Writing – original draft, Methodology, Writing – review & editing. AA: Writing – review & editing, Software, Writing – original draft. YH: Writing – review & editing, Writing – original draft, Project administration. FA: Writing – review & editing, Writing – original draft, Visualization. EQ: Writing – review & editing, Writing – original draft.

Funding

The author(s) declared that financial support was not received for this work and/or its publication.

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declared that generative AI was not used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Abdelhamid, N., Ayesh, A., and Thabtah, F. (2014). Phishing detection based associative classification data mining. *Expert Syst. Appl.* 41, 5948–5959. doi: 10.1016/j.eswa.2014.03.019

Abutaha, M., Ababneh, M., Mahmoud, K., and Baddar, S. A. H. (2021). "URL phishing detection using machine learning techniques based on URLs lexical analysis," in *2021 12th International Conference on Information and Communication Systems (ICICS)* (Valencia: IEEE), 147–152. doi: 10.1109/ICICS52457.2021.9464539

Alazaidah, R., Ahmad, F. K., Mohsen, M. F. M., and Junoh, A. K. (2018). Evaluating conditional and unconditional correlations capturing strategies in multi label classification. *J. Telecommun. Electr. Comput. Eng.* 10, 47–51.

Alazaidah, R., Al-Shaikh, A., Al-Mousa M, R., Khafajah, H., Samara, G., Alzyoud, M., et al. (2024). Website phishing detection using machine learning techniques. *J. Stat. Applic. Probab.* 13, 119–129. doi: 10.18576/jsap/130108

Alazaidah, R., Alzyoud, M., Al-Shanableh, N., and Alzoubi, H. (2023b). "The significance of capturing the correlations among labels in multi-label classification: an investigative study," in *AIP Conference Proceedings*, Vol. 2979 (Jordan: AIP Publishing). doi: 10.1063/5.0177340

Alazaidah, R., Samara, G., Aljaidi, M., Haj Qasem, M., Alsarhan, A., and Alshammari, M. (2023a). Potential of machine learning for predicting sleep disorders: a comprehensive analysis of regression and classification models. *Diagnostics* 14:27. doi: 10.3390/diagnostics14010027

Al-Batah, M. S., Alzboon, M. S., and Alazaidah, R. (2023). Intelligent heart disease prediction system with applications in Jordanian hospitals. *Int. J. Adv. Comput. Sci. Applic.* 14, 1151–1159. doi: 10.14569/IJACSA.2023.0140954

Aljofey, A., Bello S, A., Lu, J., and Xu, C. (2025). Comprehensive phishing detection: a multi-channel approach with variants TCN fusion leveraging URL and HTML features. *J. Netw. Comput. Applic.* 238:104170. doi: 10.1016/j.jnca.2025.104170

Alluwaici, M., Junoh, A. K., AlZoubi, W. A., Alazaidah, R., and Al-luwaici, W. (2020). New features selection method for multi-label classification based on the positive dependencies among labels. *Solid State Technol.* 63.

Alluwaici, M. A., Junoh, K., and Alazaidah, R. (2020). New problem transformation method based on the local positive pairwise dependencies among labels. *J. Inform. Knowl. Manag.* 19:2040017. doi: 10.1142/S0219649220400171

Alzyoud, M., Alazaidah, R., Aljaidi, M., Samara, G., Qasem, M., Khalid, M., et al. (2024). Diagnosing diabetes mellitus using machine learning techniques. *Int. J. Data Netw. Sci.* 8, 179–188. doi: 10.5267/j.ijdns.2023.10.006

APWG (2021). *Phishing Activity Trends Reports, 4th Quarter 2020*. Anti-Phishing Working Group. Available online at: https://apwg.org/trendsreports/ (Accessed May 09, 2021).

Athulya, A. A., and Praveen, K. (2020). "Towards the detection of phishing attacks," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)* (Tirunelveli, India: IEEE), 337–343. doi: 10.1109/ICOEI48184.2020.9142967

Barik, K., Misra, S., and Mohan, R. (2025). Web-based phishing URL detection model using deep learning optimization techniques. *Int. J. Data Sci. Anal.* 20, 1–23. doi: 10.1007/s41060-025-00728-9

Chapla, H., Kotak, R., and Joiser, M. (2019). "A machine learning approach for URL based web phishing using fuzzy logic as classifier," in *2019 International Conference on Communication and Electronics Systems (ICCES)* (Coimbatore: IEEE), 383–388. doi: 10.1109/ICCES45898.2019.9002145

Chiew, K. L., Tan, C. L., Wong, K., Yong, K. S. C., and Tiong, W. K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Inform. Sci.* 484, 153–166. doi: 10.1016/j.ins.2019.01.064

Cui, Q. (2019). *Detection and Analysis of Phishing Attacks*. Diss. Université d'Ottawa/University of Ottawa.

Gandotra, E., and Gupta, D. (2021). An efficient approach for phishing detection using machine learning, multimedia security: algorithm development. *Anal. Applic.* 239–253. doi: 10.1007/978-981-15-8711-5_12

Ganjei, M. A., and Boostani, R. (2022). A hybrid feature selection scheme for high-dimensional data. *Eng. Appl. Artif. Intell.* 113:104894. doi: 10.1016/j.engappai.2022.104894

Gareth, J., Witten, D., Hastie, T., Tibshirani, R., and Taylor, J. (2023). "Statistical learning," in *An Introduction to Statistical Learning: With Applications in Python* (Cham: Springer International Publishing), 15–67.

Mohammad, R. M., Thabtah, F., and McCluskey, L. (2015). *Phishing Websites Features*. School of Computing and Engineering, University of Huddersfield.

Ni, J., Shen, K., Chen, Y., Cao, W., and Yang, S. X. (2022). An improved deep network-based scene classification method for self-driving cars. *IEEE Trans. Instrument. Measur.* 71, 1–14. doi: 10.1109/TIM.2022.3146923

Nti, I. N., Narko-Boateng, O., Adekoya, A. F., and Somanathan, A. R. (2022). Stacknet based decision fusion classifier for network intrusion detection. *Int. Arab J. Inform. Technol.* 19, 478–490. doi: 10.34028/iajit/19/3A/8

Pei, M., Feng, Y., Changlong, Z., and Minghua, J. (2022). Smoke detection algorithm based on negative sample mining. *Int. Arab J. Inform. Technol.* 19, 1–9. doi: 10.34028/iajit/19/4/15

Rao, R. S., Vaishnavi, T., and Pais, A. R. (2020). CatchPhish: detection of phishing Websites by inspecting URLs. *J. Ambient Intell. Hum. Comput.* 11, 813–825. doi: 10.1007/s12652-019-01311-4

Rashid, J., Mahmood, T. M., Nisar, W., and Nazir, T. (2020). "Phishing detection using machine learning technique," in *2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)* (Riyadh: IEEE), 43–46. doi: 10.1109/SMART-TECH49988.2020.00026

Sahingoz, O. K., Buber, E., Demir, O., and Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Syst. Applic.* 117, 345–357. doi: 10.1016/j.eswa.2018.09.029

Srivastava, S. (2014). Weka: a tool for data preprocessing, classification, ensemble, clustering and association rule mining. *Int. J. Comput. Applic.* 88:10. doi: 10.5120/15389-3809

Su, J.-M., Chang, J., Indrayani, N. L. D., and Wang, C. (2023). Machine learning approach to determine the decision rules in ergonomic assessment of working posture in sewing machine operators. *J. Saf. Res.* 87, 15–26. doi: 10.1016/j.jsr.2023.08.008

Tan, C. L. (2018). *Phishing Dataset for Machine Learning: Feature Evaluation*. Mendeley Data. Available online at: https://data.mendeley.com/datasets/h3cgnj8hft/1 (Accessed May 10, 2021).

Ubing, A. A., Kamilia, S., Abdullah, A., Jhanjhi, N., and Supramaniam, M. (2019). Phishing Website detection: an improved accuracy through feature selection and ensemble learning. *Int. J. Adv. Comput. Sci. Appl.* 10, 252–257. doi: 10.14569/IJACSA.2019.0100133

Vigneswari, T., Vijaya, N., and Kalaiselvi, N. (2021). Early prediction of cervical cancer using machine learning techniques. *Turkish J. Physiother. Rehabil.* 32, 262–269.

Warburton, D. (2020). *2020 Phishing and Fraud Report*. F5 Labs.

Zabihimayvan, M., and Doran, D. (2019). "Fuzzy rough set feature selection to enhance phishing attack detection," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (New Orleans, LA: IEEE). doi: 10.1109/FUZZ-IEEE.2019.8858884