

Intelligent Phishing Website Detection before and after Multiple Informative Feature Selection Techniques: Machine Learning Approach

Kibreab Adane

Ph.D. Student, Faculty of Computing & Software Engineering, Arba Minch University, Institute of Technology, Arba Minch, Ethiopia.

Corresponding Author:

kibreab.adane@amu.edu.et

ORCID iD: <https://orcid.org/0000-0002-3021-5059>

Berhanu Beyene

Associate Prof., Ethiopian Cybersecurity Association, Addis Ababa, Ethiopia.

berhanebeyene@gmail.com

ORCID iD: <https://orcid.org/0000-0003-1398-0880>

Mohammed Abebe

Assistant Prof., Faculty of Computing & Software Engineering, Arba Minch University, Institute of Technology, Arba Minch, Ethiopia.

moshethio@gmail.com

ORCID iD: <https://orcid.org/0000-0003-0622-4841>

Received: 29 October 2022

Accepted: 01 January 2023

Abstract

Individuals and Organizations that rely on the Internet for communication, collaboration, and daily tasks regularly encounter security and privacy issues unless interventions of intelligent Cybersecurity defense systems have been made to counter them. The existing pieces of evidence reveal that phishing website attacks have drastically increased despite the scientific communities' best efforts to combat them. Based on the key research gaps explored, the study has made significant attempts to answer the following research questions: RQ#1: Which cross-validation techniques and model optimization parameters are appropriate for given datasets and classifiers? RQ#2: Which Classifier(s) yielded a superior Accuracy, F1-Score, AUC-ROC, and MCC value with acceptable train-test computational time before and after applying the Informative Feature Selection Techniques? RQ#3: What are the strengths and weaknesses of each Classifier after being applied with multiple Informative Feature Selection Techniques? RQ#4: Could the results of the top-performed Classifier and Informative Feature Selection Technique on Dataset one (DS-1) be consistent on Dataset two (DS-2)? The study used a Google Co-Lab environment and Python Code to conduct rigorous experiments. Our experimental findings reveal that the CAT-B Classifier demonstrated a superior phishing website detection performance in terms of (Accuracy, F1-Score, AUC-ROC, and MCC value with acceptable train-test computational time both before and after applying the UFS Feature Selection Technique by scoring 0.9764 accuracies, 0.9762 F1-Score, 0.996 AUC-ROC, and 0.9528 MCC Value with 6 Seconds train-test computational time. The study practically demonstrated implementing the CAT-B-UFS technique using a Python Code so that upcoming researchers can easily replicate their results and learn more. In future work, the study proposed implementing deep learning algorithms with proper feature selection techniques on Individual and Hybrid approaches to obtain more promising results.

Keywords: Machine Learning, Feature Selection Technique, Cat-Boost Classifier, Phishing

Website Detection, Uni-Variate Feature Selection, Information Network Security Agency (INSA)

Introduction

Individuals and organizations that rely on the Internet for communication, collaboration, and daily tasks regularly encounter security and privacy issues unless interventions of intelligent Cybersecurity defensive systems have been made. Among the security and privacy issues encountered in the cyberspace environment, phishing websites are well-known since the attackers use the replica of benign websites to harvest sensitive data and transfer malware by exploiting the existing technical defense strategies and by taking advantage of human weakness or behavior (Abdelhamid, Ayesh & Thabtah, 2014). Despite websites having HTTPS, which are supposed to be benign websites, to mimic benign websites, 74% of all phishing websites today incorporate HTTPS (APWG, 2023; Hannousse & Yahiouche, 2021).

Nearly 50% -80% of illegal websites were blocked following some form of financial loss (Jain & Gupta, 2019). Even though the blacklisting approach is found to be insufficient in detecting fresh phishing website attacks, it is now utilized by most widely used Internet applications such as Chrome, Firefox, Gmail, Google Search, Safari, Internet Explorer, and several web browser extensions to detect and alert warning messages when online users visiting them (Odeh, Alarbi, Keshta & Abdelfettah, 2020; Tang & Mahmoud, 2021). URL redirects online users anywhere on the Internet, but checking the legitimacy of URLs is usually unnoticed by them. Only looking at URL structures cannot guarantee safe website security unless the discovery of hidden malicious patterns from website contents/source codes is made. The challenge is: Do online users have access to and Know-how of website source code? That is where interventions in the emerging paradigm of technologies like machine learning are needed. A 2021 APWG (Anti-Phishing Working Group) report shows a dramatic increase in unique phishing website attacks, as shown in Figure 1.

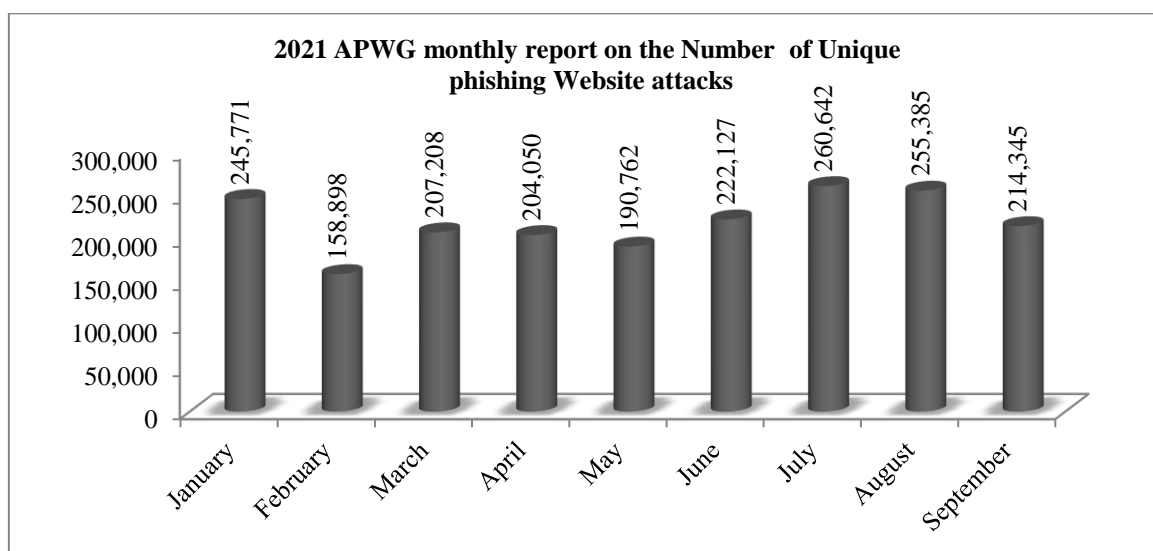


Figure 1: APWG 2021 Monthly report on fresh phishing website attacks (Adane & Beyene, 2022)

One of Ethiopia's recent breaking news stories was the hacking of Ethiopian institutions' websites, which can be considered core indicators of the current security defense system

defects. It also exposes a new dimension that machine-learning intervention can help solve. For example, in June 2020, Egypt-based Hackers took responsibility for cyber-attack attempts on several Ethiopian institutional websites (INSA, 2020). According to Information Network Security Agency (INSA) Chief Shumet Gizaw (Ph.D.), cyber-attack reports dated February 14, 2022, Ethiopia encountered more than 3,400 cyber-attack attempts in six months, which was recorded as the highest number in cyber-attack reporting history. Website attacks were the most frequent cyberattacks, accounting for 25% of incidents. Financial, educational, and service-provider institutions, government ministries, regional offices, and broadcasting media were among the main targets of the attacks (INSA, 2022a; INSA, 2022b). Hacking of Ethiopian Broadcasting Corporate (EBC) and Walta Info Facebook websites was recent (INSA, 2022b). The abovementioned challenges cannot be alleviated without an intelligent anti-phishing website attack.

Most problems in the different sectors today are solved using deep learning algorithms despite requiring High-Performance Computing (HPC) machines to conduct rigorous experiments and deployment for use. Hence, re-validating the significance of Machine Learning approaches in the perspectives of phishing website detection is vital to account for resource-constrained devices in developing continents like Africa in general and countries like Ethiopia in particular. Since it is impossible to ignore the promising performance of Deep Learning algorithms in phishing website detection, a future study will use Deep Learning algorithms to compare their performance to that of Machine-learning algorithms.

Due to their promising accuracy in predicting new attacks by discovering hidden patterns from complex datasets, Machine Learning and Deep Learning techniques are now widely used in cyber security, particularly in detecting phishing websites. Despite the scientific communities have made a great deal of effort to tackle the problem of phishing website detection using the techniques mentioned above, problems persist due to attackers regularly following novel strategies to exploit the existing anti-phishing strategies (Adane & Beyene, 2022).

Even though the scientific communities use numerous machine learning and deep learning algorithms to tackle issues associated with phishing website detection, these algorithms did not perform equally well in identifying phishing websites (ibid). Similarly, even though there are many website attributes used for phishing website detection, not each attribute has equal relevance for phishing website detection unless appropriate feature selection techniques are applied for better model accuracy, speeding up the model train-test computational time, and addressing over-fitting issues (Hannousse & Yahiouche, 2021; Masoudi-Sobhanzadeh, Motieghader & Masoudi-Nejad, 2019). This was the primary driving force for the study's decision to undertake a performance analysis of the top-performed classifiers (Random Forest, Gradient Boost, and Logistic Regression) and appropriate feature selection techniques (Uni-Variate Feature Selection, Recursive Feature Elimination, Pearson Correlation Coefficient, and Mutual Information) identified in the preliminary study (Adane & Beyene, 2022) to conduct rigorous experiments for appropriate evaluation and proposal of the superior phishing website detection model in terms of Accuracy, F1-Score, AUC-ROC, and MCC with acceptable train-test computational time.

The Cat-Boost Classifier was introduced in the study to detect phishing websites because it is the most recent version of the Boosting Machine-Learning algorithm, which incorporates new advancements, such as automatic encoding of a categorical variable for both classification and regression tasks, using permutation-driven random dataset sample selection strategy,

address prediction fluctuation issues caused by target leakage, balanced, fast and less prone to issues associated with over-fitting (Ibrahim, Ewusi & Ahenkorah, 2022; Hancock & Khoshgoftaar, 2020), but hasn't been used in any of the 30 most recent reviews of studies (Adane & Beyene, 2022) to compare performance.

Literature Review

the study thoroughly analyzed some recent, pertinent, and reliable related research works to identify the major gaps and propose suitable remedies,

In a preliminary investigation (ibid), 30 current research works on detecting phishing websites using ML and DL techniques were systematically examined to pinpoint glaring research gaps and find workable remedies. Unbalanced dataset usage, arbitrary selection of certain train-test dataset split ratios, scientific disagreements over the inclusion and exclusion of website features for phishing detection, failure to run-time analysis of the model, and the use of significant feature techniques either on a standalone basis or in hybrid approaches are a few of the key gaps that have been identified. Few research works included URL-based, web content-based, domain-based, and page-based website features for phishing Website detection were found in the examined studies (ibid).

The study by Hannousse and Yahiouche (2021) implemented Random Forest, Logistic Regression, Decision Tree, SVM, and Naïve-Bayes for phishing website detection. Each Classifier was applied with feature selection techniques such as Chi-square, Information-Gain (IG), Pearson Correlation Coefficient (PCC), and Relief-Rank. The study used balanced datasets containing 11,430 instances of phish-legitimate websites and 87 attributes collected from multiple sources such as (Phish-tank, Open-Phish, Alexa, and Yandex). This dataset was used in our study as a dataset (DS-1). The study used the Accuracy and F1-Score as core model evaluation metrics. The study's primary goal was to examine the importance of website features for each classifier. According to the authors' findings, the Random Forest and Chi-square combinations demonstrated a superior accuracy of 96.83%. The authors stated that classifiers such as Random Forest, SVM, and Decision trees are quite sensitive to the order of attributes in the datasets. The study fell to mention which train-test dataset split was used, and the top-performed model was tested on a single Dataset. Despite utilizing the same dataset (DS-1), the accuracy (97.64%) attained in our proposed study by the CAT-B-UFS was found to be superior to the accuracy (96.83%) attained by the combinations of Random Forest Classifier and Chi-square in the study (Hannousse & Yahiouche, 2021).

The study by (Gupta, Yadav, Razzak, Psannis, Castiglione & Chang, 2021) implemented K-NN, Logistic-Regression, Random-Forest, and SVM for phishing website detection. Each Classifier was applied with informative feature selection techniques such as Feature Correlation, Random-Forest score, and K best score. The study used nearly balanced datasets containing 11964 instances of phish-legitimate websites and 9 URL (Lexicon) attributes; the dataset used was named "ISCXURL-2016". The study used 80%-20% train-test dataset splits and numerous model evaluation metrics. According to the authors, the Random Forest Classifier demonstrated a superior accuracy of 99.57%. Each Classifier experimented on limited website features, the study fell to consider the domain and web-content-based features, and the top-performed model was tested on a single Dataset.

Abedin, Bawm, Sarwar, Saifuddin, Rahman and Hossain (2020) study implemented Random-Forest, K-NN, and Logistic-Regression for phishing website detection. The study used

a nearly balanced dataset containing 11,504 instances of phish-legitimate websites and 31 predictor attributes, collected from the Kaggle repository. The study used 80%-20% train-test dataset splits, precision, recall, ROC curves, and F1 scores as model evaluation metrics. According to the authors' findings, the Random Forest Classifier demonstrated a superior F1-Score and Precision of 97%. The study fell to mention the relevant feature selection techniques used, not conducting the train-test computational time of each Classifier, and the top-performed model was tested on a single Dataset. Despite using the same dataset (DS-2) as our study, the F1-Score (97.48%) attained in our research by the CAT-B-UFS was found to be superior to the F1-Score (97%) achieved by Random Forest Classifier in the study (ibid).

Hossain, Sarma and Chakma (2020) study implemented K-NN, SVM, Decision-Tree, Random-Forest, and Logistic-Regression for phishing website detection. The study used Principal Component Analysis (PCA) as a dimension reduction technique. The study used balanced datasets containing 10,000 instances of phish-legitimate websites and 48 attributes collected from the Mendeley repository. The study used precision, recall, ROC curves, and F1 Scores as model evaluation metrics. According to the Authors' findings, the Random Forest Classifier demonstrated a superior F1-Score of 99%. The study fell to consider the domain and page-based features; fell to conduct the train-test computational time of each Classifier, and the top-performed model was tested on a single Dataset.

Singhal, Chawla and Shorey (2020) implemented Random-Forest, Neural-Network, and Gradient-Boost Classifiers for phishing website detection. The study used balanced datasets containing 80,000 instances of phish-legitimate websites and 14 attributes collected from Majestic and Phish-tank repositories. The study used Accuracy, Precision, and Recall as core model evaluation metrics. According to the authors' findings, the Gradient Boost Classifier demonstrated a superior accuracy of 96.4%. The study fell to consider the domain and page-based features, to conduct the train-test computational time of each Classifier, and to mention feature selection techniques used, and the top-performed model was tested on a single Dataset.

Chiew, Tan, Wong, Yong and Tiong (2019) implemented Random-Forest, JRiP, PART, and C4.5 for phishing website detection. The study applied the Hybrid-Ensemble Feature Selection technique. The study used balanced datasets containing 10,000 instances of Phish-legitimate websites and 48 attributes, collected from multiple sources such as (Alexa, Phish-tank, Common-Crawl, and Open-Phish). The study used a 70%-30% train-test dataset split and used accuracy as a core model evaluation metric. According to the authors' findings, the Random Classifier demonstrated a superior accuracy of 94.6%. The study fell to consider the domain and page-based features, and the top-performed model was tested on a single Dataset.

Jain and Gupta (2019) used SVM, Logistic-Regression, Random-Forest, C4.5 Sequential Minimal Optimization, Adaboost, Neural Network, and Naïve-Bayes for phishing website detection. The study used nearly balanced datasets containing 2,544 instances of phish-legitimate websites, collected from multiple sources such as (Alexa, Phish-tank, and Stuffgate). The study used a 90%-10% train-test dataset split and numerous model evaluation metrics such as (Precision, Accuracy, F1-Score, and AUC-Roc). According to the authors' findings, the Logistic Regression demonstrated a superior accuracy of 98.4%. The study fell to consider the domain and URL-based features fell to mention relevant feature selection techniques used, fell to conduct the train-test computational time of each Classifier, and the top-performed model was tested on a single Dataset.

The following research questions are expected to be answered in this study to address the

aforementioned critical research gaps:

RQ#1: Which cross-validation techniques and model optimization parameters are appropriate for Datasets and Classifiers?

RQ#2: Which Classifier(s) yielded a superior Accuracy, F1-Score, AUC-ROC, and MCC value with acceptable train-test computational time before and after applying the Informative Feature Selection Techniques?

RQ#3: What are the strengths and weaknesses of each Classifier after being applied with multiple Informative Feature Selection Techniques?

RQ#4: Could the results of the top-performed Classifier and Informative Feature Selection Technique on Dataset one (DS-1) be found consistent on another Dataset (DS-2)?

RQ#5: How could the top-performing phishing website detection model implementations be shown practically?

Materials and Methods

To successfully attain the study’s core objective, this study followed significant steps, as demonstrated in Figure 2, to carry out in-depth experiments.

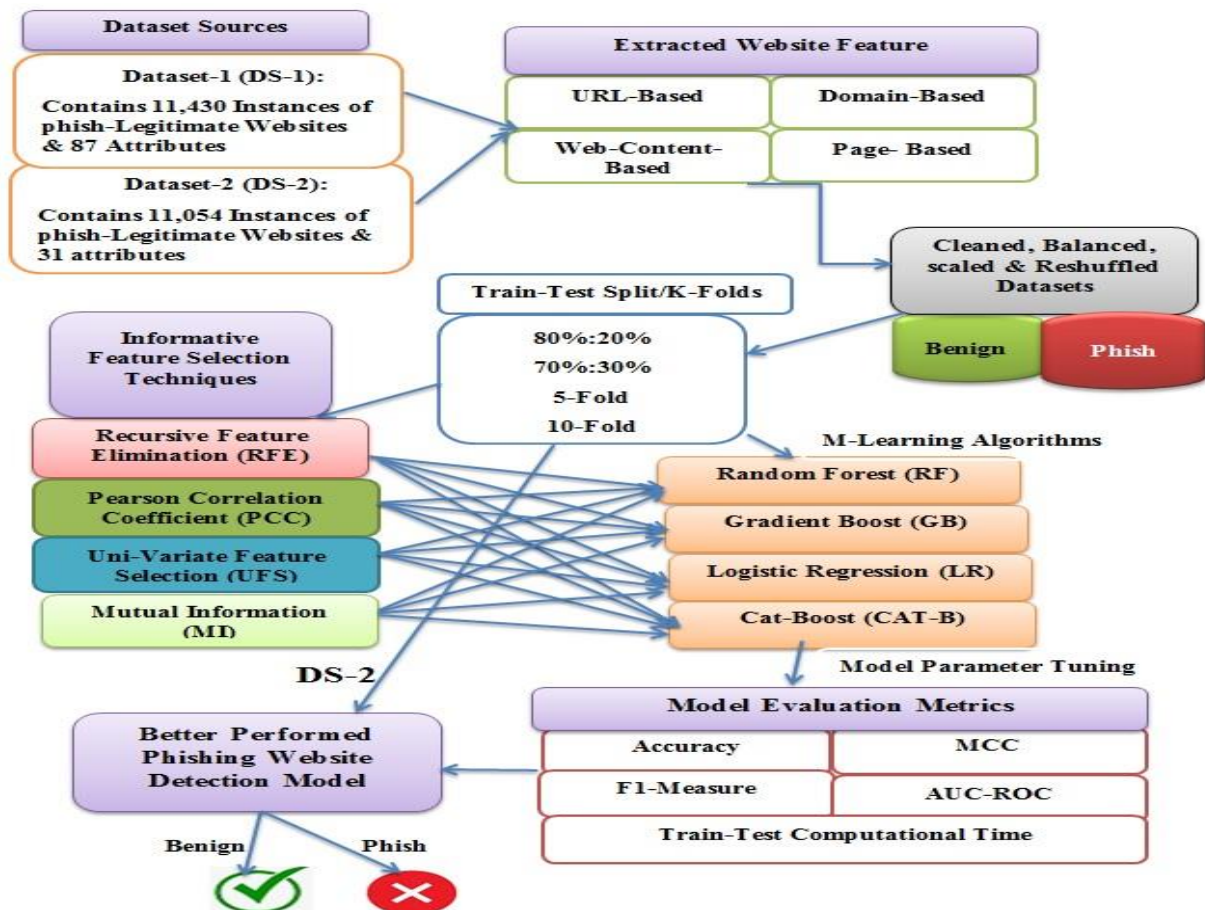


Figure 2: Proposed phishing website detection architecture

Brief explanations of each step illustrated in Figure 2 are presented as follows.

Dataset Descriptions

One of the common challenges M-learning researchers encounter is finding reputable datasets that incorporate the required features. Regardless of dataset size, using a well-cleaned representative dataset is more important than choosing a particular M-learning algorithm (Althnian, et al., 2021). The study used two reputable datasets covering various website feature categories, including URL, web content, domain, and page rank, as shown in Figure 2.

DS-1 was the recent benchmark dataset (Hannousse & Yahiouche, 2021) constructed and used to train and test the different Machine Learning Algorithms such as Random Forest, Logistic Regression, Decision Tree, SVM, and Naïve-Bayes. DS-1 contained 11,430 instances and 87 attributes, balanced and collected from reputable sources, such as Phish-tank, Open-Phish, Alexa, and Yandex (ibid). In DS-1, zero (0) was used to represent benign websites, while one (1) was used to describe phishing websites. DS-1 contains a mix of binary and non-binary numerical values for predictor attributes. In our study, DS-1 experimented on each Classifier and Feature Selection Technique presented in Figure 2.

The source of DS-2 was the Kaggle repository. As was stated in the study (Abedin, et al., 2020), Kaggle is one of the well-known public dataset repositories that contain a considerable amount of dataset collection to be used by the scientific communities to train Machine Learning Algorithms. DS-2 contained 11,054 instances and 31 attributes and was nearly balanced (56%: 44% phish-legitimate website ratios). The DS-2 was already used and tested by Abedin et al. (2020) on Classifiers such as K-NN, and Random-Forest Logistic-Regression. In DS-2, a negative one (-1) was used to represent benign websites, while a positive one (1) was used to represent phishing websites. DS-2 contains only binary values for both predictors and target variables. In our study, DS-2 experimented on the performed Classifier and Feature Selection Technique among Classifiers presented in Figure 2. DS-2 was accessed on 23 November 2022 from: <https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector>.

Cross-validation methods and/or Train-Test dataset split ratios

To choose appropriate train-test splits for the given Dataset and Classifiers, the study applied a variety of cross-validation techniques and/or train-test dataset splits, including 5-fold, 10-fold, 80%/20% split, and 70%/30% split as shown in Figure 2.

Implemented Informative Feature Selection Techniques and Machine Learning Algorithms

To experimentally test their effect on each classifier's performance, the study utilized well-known informative feature selection techniques, including UFS, MI, RFE, and PCC, as shown in Figure 2. The study purposefully selected three top candidate classifiers identified in the reviewed study (Adane & Beyene, 2022) as Random Forest, Gradient-Boost, and Logistic Regression, as well as introduced the Cat-Boost Classifier for phishing website identification because it is the most recent version of the Boosting Machine-Learning algorithm, but not utilized in the 30 recent reviewed studies (ibid) to conduct comparative performance analysis. The study applied different model optimization parameters for each Classifier.

Implementation Tools

The study utilized a Google Co-Lab environment to conduct rigorous experiments on each Classifier and Informative feature selection techniques to benefit from high-speed computing during training and testing each Supervised Machine Learning algorithm. The study utilized

Python code as the implementation language because it is easy to understand and has a rich ecosystem of libraries or packages for machine learning-oriented studies. The study practically demonstrated the implementations of the top-performed phishing website detection model and proper feature selection technique using Python code so that upcoming researchers can easily replicate their results and learn more.

Model Performance Evaluation Metrics

As was stated by Chicco and Jurman (2020), Accuracy and F1 score were still among the widely utilized model performance evaluation metrics in Machine Learning for binary or multiclass classification tasks. They were calculated on a contingency table or confusion matrix. However, these metrics were not reliable measures, especially on imbalanced datasets, due to yielding overoptimistic exaggerated results (Chicco & Jurman, 2020). To overcome issues associated with class imbalance, an evaluation metric like Matthews's Correlation Coefficient (MCC) is vital to obtaining balanced results of classifiers on data with different class sizes (Ibrahim, et al., 2022; Chicco & Jurman, 2020). However, a critical problem with the MCC metric is that the MCC is undefined when the entire row or column values become zero (Chicco & Jurman, 2020). The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) is a well-known metric to show how well the negative class's probabilities are detached from the positive class (Ibrahim et al., 2022).

In this study, the performance of classifiers like Cat-Boost (CAT-B), Gradient Boost (GB), Random Forest (RF), and Logistic Regression (LR) would be assessed using Accuracy, F1-measure, MCC, and AUC-ROC as standard model performance evaluation metrics, with and without using the four pertinent feature selection techniques shown in Figure 2. This is because the metrics mentioned above make it easier to interpret the performance of the classifiers' classification across all classes (Ibrahim, et al., 2022).

A contingency table or confusion matrix containing TPR, TNR, FPR, and FNR is used to demonstrate the outcome of the classification activities (Chicco & Jurman, 2020). The confusion matrix could be presented as follows:

Prediction	Phishing Website	Legitimate Website
Labeled as Phishing	True +Ve Rate (TPR)	False +Ve Rate (FPR)
Labeled as Legitimate	False -Ve Rate (FNR)	True -Ve Rate (TNR)

Accuracy metric incorporates the sum of correct predictions such as the True +Ve Rate (TPR) and True -Ve Rate (TNR) divided by the sum of all correct and incorrect predictions such as True +Ve Rate (TPR), False +Ve Rate (FPR), True -Ve Rate (TNR), and False -Ve Rate (FNR). In short, the accuracy metric formula could be written as:

$$\text{Sum (TPR+TNR) divided by Sum (TPR+TNR+FPR+FNR)}$$

Where TPR represents the number of phishing websites correctly labeled as Phishing, TNR represents the number of legitimate websites correctly tagged as legitimate. FPR represents the number of legitimate websites incorrectly marked as phishing websites, and in this case, the

FPR denies Internet users from accessing authentic websites. FNR represents the number of phishing websites wrongly labeled as legitimate; in this case, the FNR allows Internet users to visit phishing websites, which is dangerous (Ali & Malebary, 2020). FPR can be computed by dividing the FPR by the sum of FPR and TNR. FNR can be computed by dividing the FNR by the sum of FNR and TPR.

The F1-Score metric measures the harmonic mean between the Recall and Precision (Ali & Malebary, 2020; Chicco & Jurman, 2020). Recall (Sensitivity) can be computed by dividing the TPR by the sum of TPR and FNR. Precision can be computed by dividing the TPR by the sum of TPR and FPR. In short, the F1-measure formula can be written as:

$$F1\text{-Score} = 2(Precision \times Recall) \text{ divided by sum } (Precision + Recall)$$

MCC metric is a well-known and balanced metric for evaluating the performance of classifiers on data with different class sizes (Ibrahim et al., 2022), unaffected by the class-imbalance issue, is a confusion matrix method of computing the Pearson product-moment correlation coefficient (PPMCC) between the actual (observed) and Predicted class values and is the only metric that yields a high result only if the binary predictor was able to correctly predict the majority of both positive and negative data instances. MCC formula can be presented as (Chicco & Jurman, 2020):

$$MCC = \frac{(TPR \times TNR) - (FPR \times FNR)}{\sqrt{((TPR + FPR) \times (TPR + FNR)) \times (TNR + FPR) \times (TNR + FNR)}}$$

AUC-ROC metric the critical idea of the AUC-ROC metric is that True and False Positive Rates explain the model performance independently of the class distribution and are found to be statistically sufficient for characterizing a classifier performance in any target context (Flach, 2003). If all models were assessed on a test set equal to the expected class ratio, all AUC-ROC Curve points would be in the same horizontal line (ibid).

The computational time metric in the study context means the amount of time each classifier takes to complete training and test tasks. This is because prediction algorithms are supposed to provide a swift prediction time and the highest accuracy level before internet visitors hand over their confidential data to fraudulent websites.

Results

The study classified the experimental results according to the research questions of the proposed study to make the discussion of the essential findings easier to understand.

RQ#1: which cross-validation techniques and model optimization parameters are appropriate for given Datasets and Classifiers?

The study implemented multiple supervised Machine Learning Algorithms such as Cat-Boost (CAT-B), Gradient-Boost (GB), Random Forest (RF), and Logistic Regression (LR) for phishing website detection. Each Classifier's performance was evaluated before and after applying the informative feature selection techniques such as Uni-Variate Feature Selection (UFS), Recursive Feature Elimination (RFE), Mutual Information (MI), and Pearson Correlation Coefficient (PCC).

The study used two different datasets collected from reliable sources. Dataset one (DS-1)

contains balanced 11,430 instances of phish-Legitimate websites and 87 attributes; it includes a mix of binary and non-binary values for predictor variables. DS-1 experimented on each Classifier and informative feature selection technique in this study. Dataset two (DS-2) contains a nearly balanced 11,054 instances of phish-Legitimate websites and 31 attributes and has only binary values for predictors and target attributes. In this study, DS-2 was used to test the performance consistency of the top performed Classifier on DS-1.

Metrics including Accuracy, F1-Score, AUC-ROC, MCC, and train-test computational time were employed in this study to objectively assess the quality of models on the testing dataset before and after employing appropriate feature selection approaches. To choose the best one, the study used various Cross-Validation techniques and train-test dataset splits, including 5-fold, 10-fold, 80%/20% split, and 70%/30% split. The Cat-Boost, Gradient-Boost, and Random Forest Classifiers were tuned for model performance using N maximum tree depths and N estimators/iterations. Meanwhile, the Logistic Regression model was tuned using solvers such as 'liblinear' and 'newton-cg.' Table 1 demonstrates the preferred parameter values of each classifier before and after the application of each informative feature selection technique.

Table 1

Appropriate Model Parameter Values after Applying Informative Feature Selection Techniques on DS-1 and DS-2

Classifiers	Preferred Parameter Values on DS-1-Before	Selected Parameter Values on DS-1-After
CAT-B	<p>=>CAT-B: 6 Max-Tree Depth and 200 iterations were used before applying each informative feature selection technique.</p> <p>=> CAT-B performed better when (80%-20%) train-test dataset split was employed before and after applying each informative feature selection technique.</p>	<p>=> CAT-B-UFS: 6 Max-Tree Depth, 200 iterations, and top 62 website features were preferred.</p> <p>=> CAT-B-PCC: 6 Max-Tree Depth, 250 iterations, and top 62 website features were selected.</p> <p>=> CAT-B-RFE: 6 Max-Tree Depth, 200 iterations, and top 46 website features were preferred.</p> <p>=> CAT-B- MI: 6 Max-Tree Depth, 200 iterations, and top 68 website features were selected.</p>
GB	<p>=> GB: 6 Max-Tree Depth and 400 estimators were used before applying each informative feature selection technique.</p> <p>=> GB performed better when (10-Fold) Cross Validation was used before and after applying each informative feature selection technique.</p>	<p>=> GB –UFS: 6 Max-Tree Depth, 400 estimators, and top 67 website features were preferred.</p> <p>=> GB-PCC: 6 Max-Tree Depth, 400 estimators, and top 62 website features were preferred.</p> <p>=> GB- RFE: 6 Max-Tree Depth, 200 estimators, and top 74 website features were preferred.</p> <p>=> GB-MI: 6 Max-Tree Depth, 400 estimators, and top 72 website features were preferred.</p>
RF	<p>=> RF: 15 Max-Tree Depth and 200 estimators were used before applying each informative feature selection technique.</p> <p>=> RF performed better when (10-Fold) Cross Validation was used before and after applying each informative feature selection technique.</p>	<p>=> RF-UFS: 15 Max-Tree Depth, 200 estimators, and top 61 website features were preferred.</p> <p>=> RF-PCC: 15 Max-Tree Depth, 200 estimators, and top 62 website features were preferred.</p> <p>=> RF-RFE: 15 Max-Tree Depth, 100 estimators, and top 72 website features were preferred.</p> <p>=> RF- MI: 15 Max-Tree Depth, 200 estimators, and top 73 website features were preferred.</p>

Classifiers	Preferred Parameter Values on DS-1- Before	Selected Parameter Values on DS-1-After
LR	<p>=> LR: ‘newton-cg’ solver and 100 iterations were used before applying each informative feature selection technique.</p> <p>=> LR performed better when (80%:20%) train-test split was used before and after applying each informative feature selection technique.</p>	<p>=> LR-UFS: ‘newton-cg’ solver, 100 iterations, and top 76 website features were preferred.</p> <p>=> LR- PCC: ‘newton-cg’ solver, 100 iterations, and top 62 website features were preferred.</p> <p>=> LR- RFE: ‘newton-cg’ solver, 100 iterations, and top 74 website features were preferred.</p> <p>=> LR- MI: ‘newton-cg’ solver, 100 iterations, and top 68 website features were preferred for.</p>

RQ#2: Which Classifier(s) yielded a superior Accuracy, F1-Score, AUC-ROC, and MCC value with acceptable train-test computational time before and after applying the Informative Feature Selection Techniques?

Classifiers Performance Analysis on DS-1 before and after the UFS Technique

As can be seen in Table 2, before applying the UFS technique, the CAT-B classifier exhibited a superior accuracy (0.9738) and F1-Score (0.9735) compared to the Accuracy and F1-scores of the remaining classifiers, such as GB, RF, and LR. The GB Classifier attained a superior MCC value (0.9484), followed closely by the CAT-B MCC value (0.9475). Before applying the UFS technique, the GB and RF classifiers attained a superior AUC-ROC value (0.996), followed closely by the CAT-B AUC-ROC (0.995) and the LR AUC-ROC (0.986) values.

Although the GB-Classifier outperformed the RF (Accuracy, F1-Score, and MCC) and LR (Accuracy, F1-Score, AUC-ROC, and MCC), it took longer to compute the train-test results (20 Minutes and 34 Seconds) than any other Classifier in Table 2. The finding of our study was consistent with another study (Ibrahim et al., 2022) that stated the GB classifier was computationally expensive despite its advantages of reducing bias, lowering over-fitting, being insensitive to missing values, being scalable, and being flexible (ibid).

Table 2

Classifiers Performance Analysis on DS-1 before and after the UFS Technique

Classifiers	Accuracy on DS-1	F1-Score on DS-1	AUC-ROC on DS-1	MCC on DS-1	Confusion Metrics	Train-Test Compute Time
CATB-Model	0.9738	0.9735	0.995	0.9475	[1125, 30] [30,1101]	8 Sec.
GB-Model	0.969	0.969	0.996	0.9484	[1121, 34] [25,1106]	20 Min: 34 Sec.
RF-Model	0.965	0.965	0.996	0.944	[1122, 33] [31,1100]	1 Min:44 Sec.
LR-Model	0.951	0.9501	0.986	0.9021	[1107, 48] [64,1067]	2 Sec.

Results Before the UFS
Results After the UFS

Classifiers after UFS	Accuracy on DS-1	F1-Score on DS-1	AUC-ROC on DS-1	MCC on DS-1	Confusion Metrics	Train-Test Compute Time
CATB-UFS	0.9764	0.9762	0.996	0.9528	[1126, 29] [25,1106]	6 Sec.
GB-UFS	0.969	0.969	0.996	0.9545	[1126, 29] [23,1108]	20Min:25 Sec.
RF-UFS	0.965	0.965	0.996	0.9379	[1121, 34] [37,1094]	1 Min:42 Sec.
LG-UFS	0.9501	0.9493	0.987	0.9003	[1104, 51] [63,1068]	1 Sec.

Our experimental findings demonstrate the suitability of the UFS technique for the CAT-B Classifier in terms of (Accuracy, F1-Score, AUC-ROC, MCC, and train-test computational time) because following the application of the UFS technique, the CAT-B Classifier's overall performance in terms of Accuracy, F1-Score, AUC-ROC, and MCC were found to increase from 0.9738, 0.9735, 0.995, and 0.9475 respectively to 0.9764, 0.9762, 0.996, and 0.9528. Before and after applying the UFS technique, the CAT-B-UFS train-test computational time was faster than the GB-UFS and RF-UFS train-test computational time. The CAT-B-UFS Accuracy (0.9764) and F1-Score (0.9762) were superior to the GB-UFS, RF-UFS, and LR-UFS Accuracy and F1-Score, as demonstrated in Table 2.

The GB Classifier attained the same Accuracy (0.969), F1-Score (0.969), and AUC-ROC (0.996) both before and after employing the UFS technique. Contrarily, following the application of the UFS technique, the GB classifier's MCC value was found to increase from 0.9484 to 0.9545, making it the study's highest MCC score. Both before and after employing the UFS technique, the RF Classifier attained the same Accuracy (0.965), F1-Score (0.965), and AUC-ROC (0.996). On the other hand, adopting the UFS technique resulted in a drop in the RF classifier MCC value from 0.944 to 0.9379.

Our experimental findings demonstrate the UFS technique's unsuitability for the LG Classifier in terms of (Accuracy, F1-Score, and MCC score) because following the application of the UFS technique, the LR Classifier Accuracy, F1-Score, and MCC Value were found to decrease from 0.951, 0.9501, and 0.9021 respectively to 0.9501, 0.9493, and 0.9003. On the other hand, the UFS technique increased the LG Classifier AUC-ROC value by 0.001 (from 0.986 to 0.987) and decreased the LR Classifier train-test computational time by 1 second (from 2 Seconds to 1 Second). The accuracy attained by the LR Classifier before and after applying the UFS technique outperformed that of another study (Hannousse & Yahiouche, 2021) that used DS-1, LR Classifier, and obtained 0.948 Accuracy.

Following the application of the UFS technique, the train-test computational time of the

CAT-B, GB, RF, and LR was found to decrease from (8 Seconds, 20 Minutes: 34 Seconds, 1 Minute: 44 Seconds, and 2 Seconds) respectively to (6 Seconds, 20 Minutes: 25 Seconds, 1 Minute: 42 Seconds, and 1 Second), respectively. Despite having the fastest train-test computational time both before and after using the UFS technique, the LR-Classifier performed poorly when compared to the other Classifiers, such as CAT-B, GB, and RF, in terms of accuracy, F1-score, AUC-ROC, and MCC.

The CAT-B, GB, and RF Classifiers all achieved a higher AUC-ROC value of 0.996 following the application of the UFS technique, while the LR Classifier attained an AUC-ROC value of 0.987. Each Classifier achieved >0.98 AUC-ROC values, as demonstrated in Table 2, Figure 3, and Figure 4. These experimental findings indicate that each classifier is more likely to differentiate between the Positive and Negative classes due to attaining an AUC-ROC value closest to 1 (Ibrahim et al., 2022).

The MCC value range is between -1 and 1 (Ibrahim et al., 2022; Chicco & Jurman, 2020). A 1 indicates flawless categorization, while a -1 indicates flawless misclassification. MCC values closer to 1 show a strong correlation between the predicted and observed classes, whereas a weak correlation is when the MCC value is close to 0 (Ibrahim, et al., 2022; Chicco & Jurman, 2020). Before using the UFS technique, the GB-Classifier showed a superior correlation, or MCC, of 0.9484, which was followed by CAT-B, RF, and LR correlation values of 0.9475, 0.944, and 0.9021, respectively, as shown in Figure 3 and Table 2. After applying the UFS technique, the GB-Classifier showed a superior correlation of 0.9545, followed by CAT-B, RF, and LR, with correlation values of 0.9528, 0.9379, and 0.9003, respectively. These experimental findings demonstrate a strong connection between the predicted and observed classes due to the attained MCC value by each Classifier being closer to 1 (Ibrahim, et al., 2022; Chicco & Jurman, 2020).

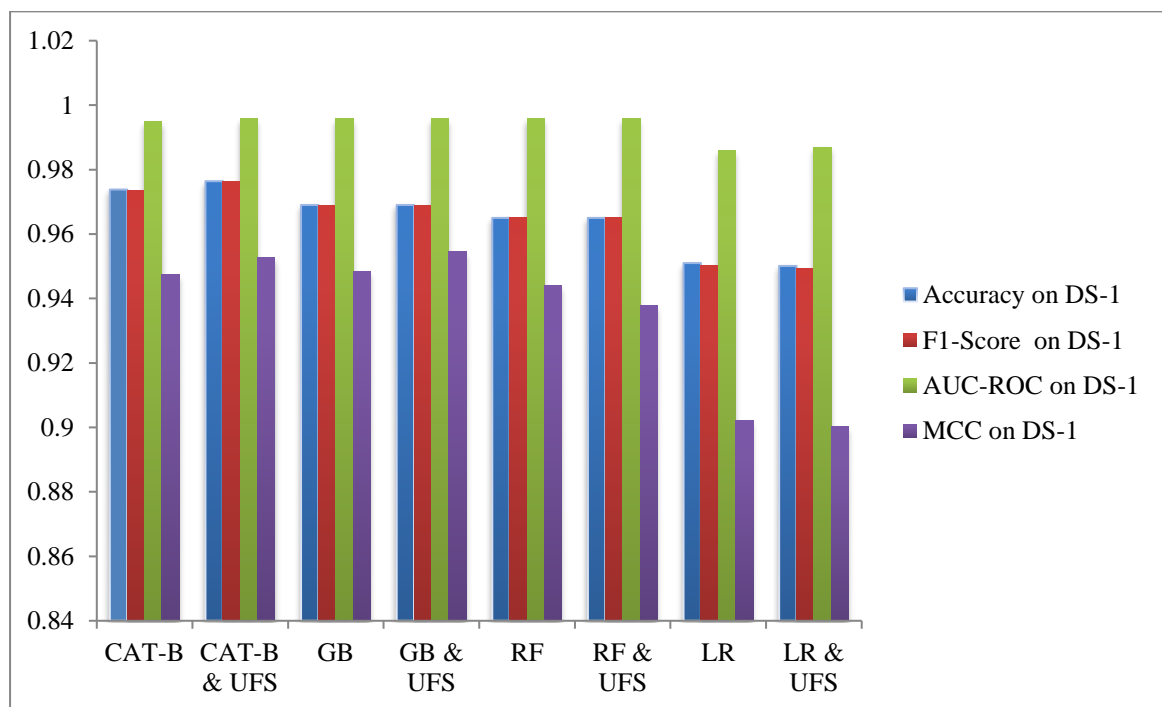


Figure 3: Classifiers Performance Analysis on DS-1 before and after the UFS Technique

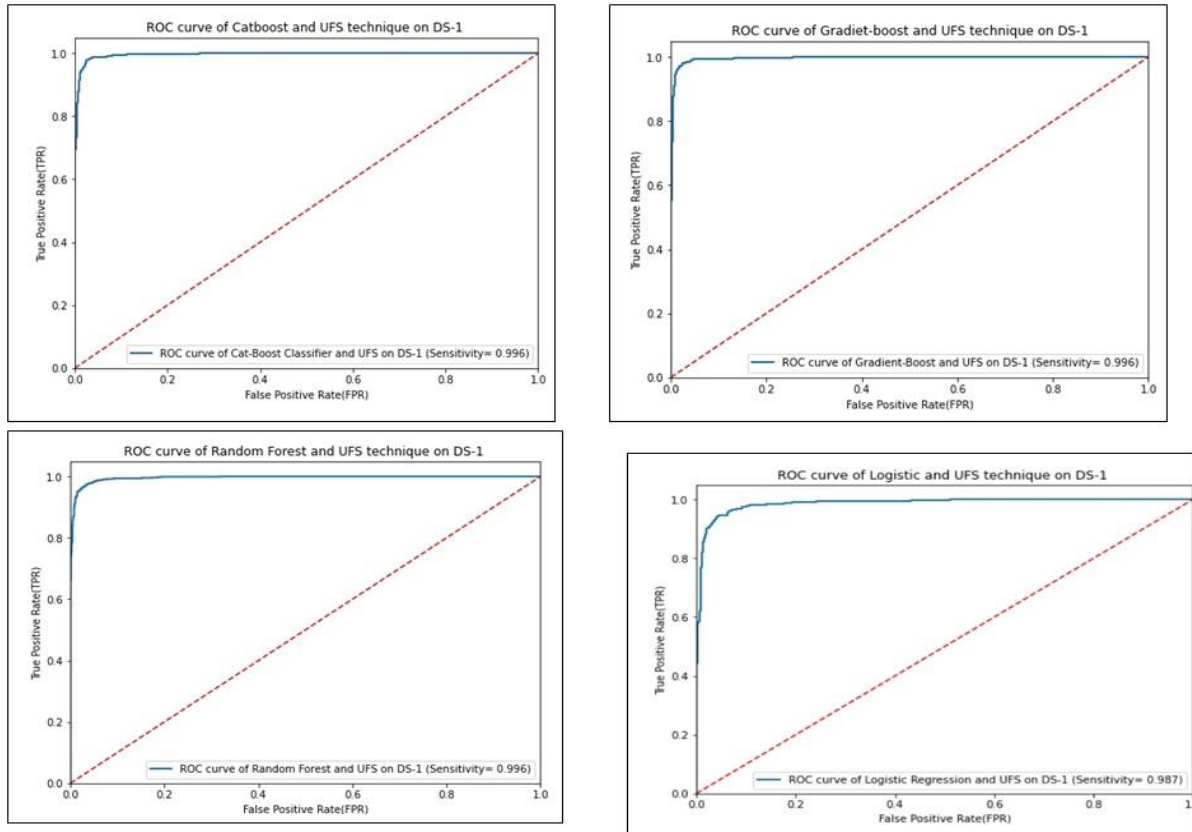


Figure 4: Classifiers AUC-ROC Curve on DS-1 after the UFS Technique

Classifiers Performance Analysis on DS-1 Before and after the PCC Technique

As demonstrated in Table 3, following the application of the PCC technique on DS-1, the CAT-B Classifier's overall performance in terms of Accuracy, F1-Score, and MCC were found to increase from 0.9738, 0.9735, and 0.9475 respectively to 0.9764, 0.9761, and 0.9528. In this study, the Accuracy and MCC values attained by the CAT-B-UFS and CAT-B –PCC techniques were the same (0.9764 and 0.9528, respectively). In this study, the Accuracy (0.9764) attained by the CAT-B-UFS and CAT-B –PCC, and the F1-score (0.9762) attained by the CAT-B-UFS were found to be superior to the other classifiers' Accuracy and F1-score. Before and after applying the PCC technique, the CAT-B Classifier attained the same AUC-ROC value of 0.995.

The CAT-B-PCC MCC value (0.9528) was superior to those of the other Classifiers employed with the PCC technique. On the other hand, the MCC value (0.9528) obtained by the CAT-B-PCC technique was lower by 0.0017 compared to the GB-UFS MCC value (0.9545). After using the PCC technique, the CAT-B Classifier train-test computational time decreased from 8 seconds to 7 seconds.

As demonstrated in Table 3, after applying the PCC technique on DS-1, the GB Classifier MCC value increased by 0.0026 (from 0.9484 to 0.9510). Contrarily, the GB-PCC technique's MCC value (0.9510) was lower by 0.0035 than the GB-UFS MCC value (0.9545). Before and after applying the PCC technique, the GB Classifier attained a superior AUC-ROC score (0.996) compared to the CAT-B-PCC AUC-ROC score (0.995), the RF-PCC AUC-ROC score (0.995), and the LR-PCC AUC-ROC score (0.8985). After applying the PCC technique, the GB Classifier train-test computational time they decreased from 20 Minutes and 34 Seconds to 18 Minutes and 37 Seconds. Contrarily, after using the PCC technique, the GB Classifier Accuracy



and F1-Score were found to decrease by 0.001 (from 0.969 to 0.968).

Before and after applying the PCC technique, the RF Classifier attained the same Accuracy and F1-Score of 0.965. This result was consistent with another study (Hannousse & Yahiouche, 2021) that employed the DS-1, RF Classifier, and PCC technique and attained an accuracy of (0.965). Our experimental findings demonstrate the PCC technique's unsuitability for the RF Classifier in terms of (AUC-ROC, MCC, and train-test computational time) because following the application of the PCC technique, the RF Classifier AUC-ROC, and MCC Value was found to decrease from 0.996 and 0.944 respectively to 0.995 and 0.9379 and the RF Classifiers' train-test computational time increased from 1 Minute and 44 Seconds to 1 minute and 48 Seconds.

Before and after applying the PCC technique, the LR Classifier attained the same AUC-ROC value of 0.986 and the same train-test computational time of 2 seconds. In this study, as compared to other classifiers, scoring the fastest train-test computational time was the significant achievement of the LR Classifier. Contrarily, our experimental findings demonstrate the PCC technique's unsuitability for the LR Classifier in terms of (Accuracy, F1-Score, and MCC value) because after applying the PCC technique, the LR Classifier Accuracy, F1-Score, and MCC values were found to decrease from 0.951, 0.9501, and 0.9021 respectively to 0.9493, 0.9484 and 0.8985, as shown in Table 3.

Table 3
Classifiers Performance Analysis on DS-1 before and after the PCC Technique

Classifiers	Accuracy on DS-1	F1-Score on DS-1	AUC-ROC on DS-1	MCC on DS-1	Confusion Metrics	Train-Test Compute Time
CATB-Model	0.9738	0.9735	0.995	0.9475	[1125, 30] [30,1101]	8 Sec.
GB-Model	0.969	0.969	0.996	0.9484	[1121, 34] [25,1106]	20 Min: 34 Sec.
RF-Model	0.965	0.965	0.996	0.944	[1122, 33] [31,1100]	1 Min:44 Sec.
LR-Model	0.951	0.9501	0.986	0.9021	[1107, 48] [64,1067]	2 Sec.

Results Before the PCC 			Results After the PCC 			
Classifiers after PCC	Accuracy on DS-1	F1-Score on DS-1	AUC-ROC on DS-1	MCC on DS-1	Confusion Metrics	Train-Test Compute Time
CATB-PCC	0.9764	0.9761	0.995	0.9528	[1127, 28] [26,1105]	7 Sec.
GB-PCC	0.968	0.968	0.996	0.9510	[1125, 30] [26,1105]	18Min:37 Sec.
RF-PCC	0.965	0.965	0.995	0.9379	[1121, 34] [37,1094]	1 Min:48 Sec.
LG-PCC	0.9493	0.9484	0.986	0.8985	[1103, 52] [64,1067]	2 Sec.

Following the application of the PCC technique, the CAT-B Classifier demonstrated superior correlation or MCC value of (0.9528), followed by the GB-PCC, RF-PCC, and LR-PCC with correlation values of 0.9510, 0.9379, and 0.8985, respectively. These experimental findings indicate the presence of a strong connection between the predicted and observed classes due to the MCC value attained by CAT-B-PCC, GB-PCC, and RF-PCC were found to be closer to 1, as per (Ibrahim, et al., 2022; Chicco & Jurman, 2020).

Following the application of the PCC technique, each Classifier achieved >0.98 AUC-ROC values, as demonstrated in Table 3, Figure 5, and Figure 6. These experimental findings indicate

that each classifier has a greater probability of differentiating between the Positive Class and the Negative Class (Ibrahim, et al., 2022) as a result of attaining an AUC-ROC value closest to 1.

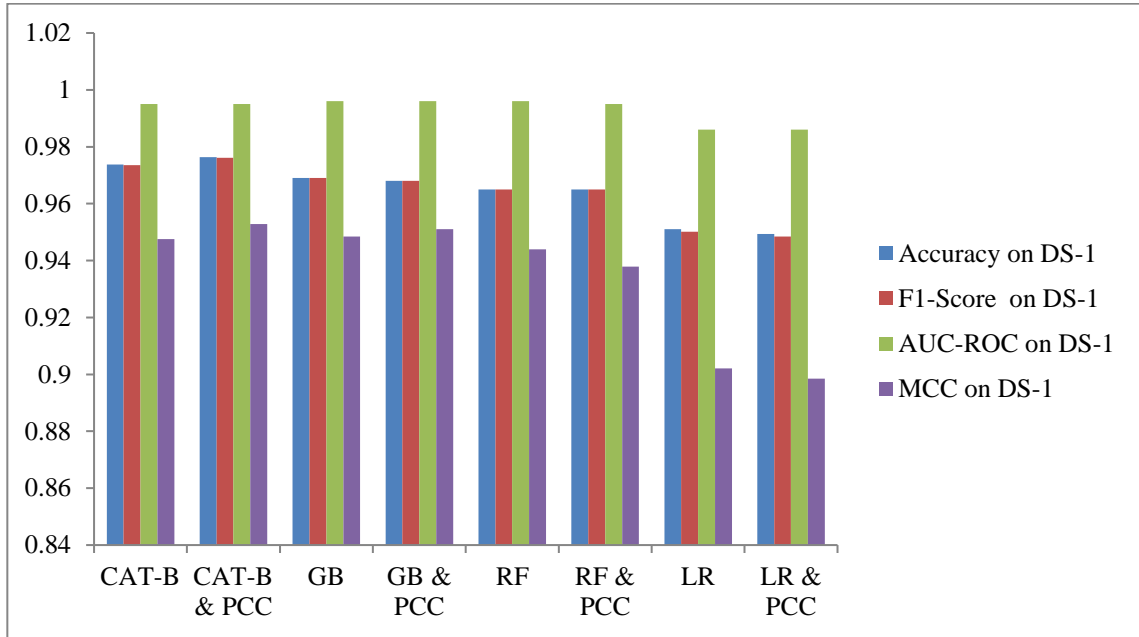


Figure 5: Classifiers Performance Analysis on DS-1 before and after the PCC Technique

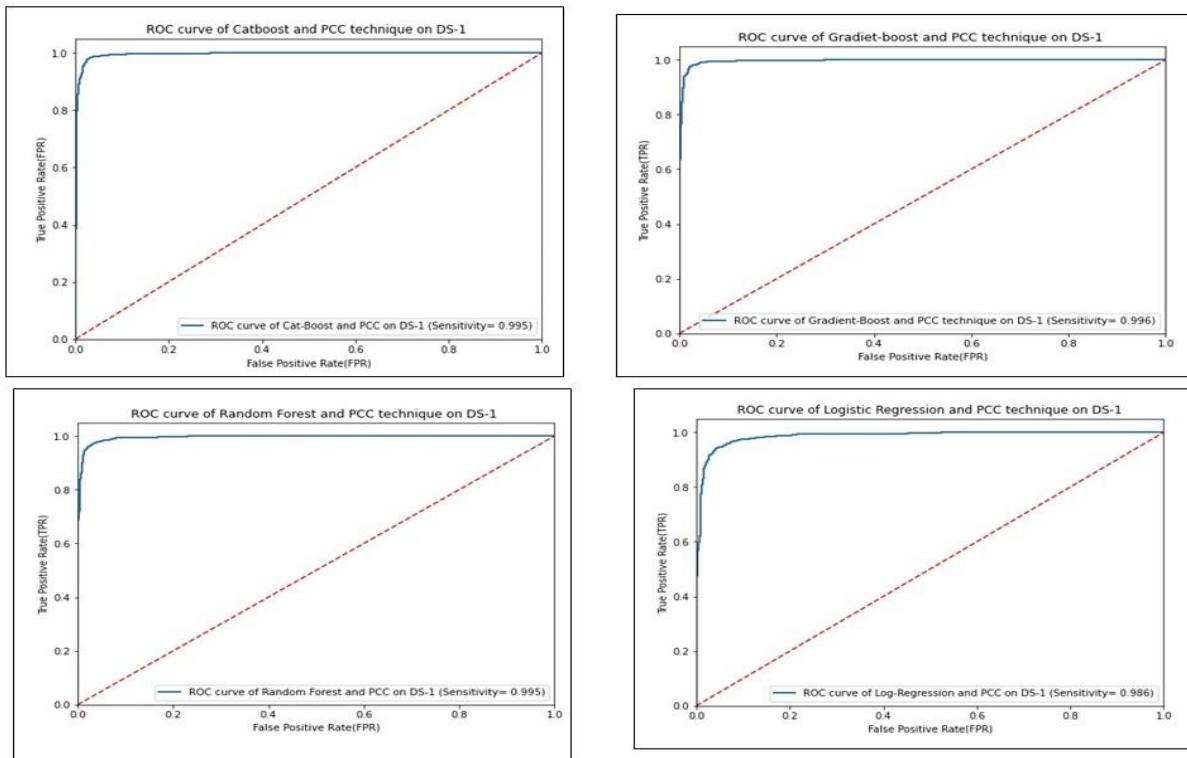


Figure 6: Classifiers AUC-ROC Curve on DS-1 after the PCC Technique

Classifiers Performance Analysis on DS-1 before and after the RFE Technique

Our experimental findings demonstrate the suitability of the RFE technique for the CAT-B Classifier in terms of (Accuracy, F1-Score, AUC-ROC, and MCC score) because following the

application of the RFE technique, the CAT-B Classifier's overall performance in terms of Accuracy, F1-Score, AUC-ROC, and MCC were found to increase from 0.9738, 0.9735, 0.995, and 0.9475 respectively to 0.9746, 0.9744, 0.996, and 0.9493. Contrarily, after applying the RFE technique, the CAT-B Classifier train-test computational time increased from 8 Seconds to 4 Minutes and 42 Seconds. In this study, the Accuracy, F1-Score, MCC, and train-test Computational time of the CAT-B-UFS and CAT-B-PCC were found to be better than that of the CAT-B-RFEs' Accuracy, F1-Score, MCC, and train-test computational time. Contrarily, the CAT-B-RFEs' Accuracy, F1-Score, and MCC values were superior to the GB-RFE, RF-RFE, and LR-RFE (Accuracy, F1-Score, and MCC). The CAT-B-RFE train-test computational time was faster than the GB-RFE and RF-RFE train-test computational time, as shown in Table 4.

Table 4
Classifiers Performance Analysis on DS-1 before and after the RFE Technique

Classifiers	Accuracy on DS-1	F1-Score on DS-1	AUC-ROC on DS-1	MCC on DS-1	Confusion Metrics	Train-Test Compute Time
CATB-Model	0.9738	0.9735	0.995	0.9475	[1125, 30] [30,1101]	8 Sec.
GB-Model	0.969	0.969	0.996	0.9484	[1121, 34] [25,1106]	20 Min: 34 Sec.
RF-Model	0.965	0.965	0.996	0.944	[1122, 33] [31,1100]	1 Min:44 Sec.
LR-Model	0.951	0.9501	0.986	0.9021	[1107, 48] [64,1067]	2 Sec.

↑
↓

Classifiers after RFE	Accuracy on DS-1	F1-Score on DS-1	AUC-ROC on DS-1	MCC on DS-1	Confusion Metrics	Train-Test Compute Time
CATB-RFE	0.9746	0.9744	0.996	0.9493	[1125, 30] [28,1103]	4 Min:42 Sec.
GB-RFE	0.968	0.968	0.996	0.9475	[1122, 33] [27,1104]	2 Hr:26Min:26 Sec.
RF-RFE	0.965	0.965	0.996	0.9335	[1117, 38] [38,1093]	14 Min:52 Sec.
LG-RFE	0.9431	0.9420	0.986	0.8864	[1100, 55] [75,1056]	33 Sec.

Our experimental findings demonstrate the RFE technique's unsuitability for the GB Classifier in terms of (Accuracy, F1-Score, MCC, and train-test computational time) because following the application of the RFE technique, the GB Classifier Accuracy, F1-Score, and MCC Value were found to decrease from 0.969, 0.969, and 0.944 respectively to 0.968, 0.968, and 0.9475. The GB Classifier train-test computational time they were increased from 1 Minute and 44 Seconds to (2 Hours: 26 Minutes: 26 Seconds).

In this study, the RF-UFS, RF-PCC, and RF-RFE all attained the same Accuracy (0.965), F1-Score (0.965), and AUC-ROC score (0.996). Contrarily, after applying the RFE technique, the RF Classifier MCC value was found to decrease from 0.944 to 0.9335, while the RF Classifier train-test computational time was found to increase from 1 Minute and 44 Seconds to 14 Minutes and 52 Seconds.

Our experimental findings demonstrate the RFE technique's unsuitability for the LR Classifier in terms of (Accuracy, F1-Score, MCC value, and train-test computational time) because following the application of the RFE technique, the LR Classifier Accuracy, F1-Score, and MCC Value were found to decrease from 0.951, 0.9501, and 0.9021 respectively to 0.9431,

0.9420, and 0.8864 and the LR Classifier train-test computational time increased from 2 Seconds to 14 Seconds.

Regardless of the Accuracy, F1-score, AUC-ROC, and MCC values obtained after the RFE technique, the train-test computational time for each Classifier was found to increase due to using the RFE technique according to our experimental findings. This might be because RFE uses a Wrapper-based dimension reduction strategy, as opposed to filter-based approaches like UFS, PCC, and MI, which choose pertinent features either at the data preprocessing stage and/or independently of the learning algorithm (Hannousse & Yahiouche, 2021).

Following the application of the RFE technique, the CAT-B Classifier demonstrated superior correlation or MCC value of (0.9493), followed by GB-RFE, RF-RFE, and LR -RFE with correlation values of 0.9475, 0.9335, and 0.8864, respectively. These experimental findings demonstrate a strong connection between the predicted and observed classes due to CAT-B, GB, and RF Classifiers attaining MCC values closer to 1 (Ibrahim et al., 2022; Chicco & Jurman, 2020). After employing the RFE technique, each Classifier achieved >0.98 AUC-ROC values, as demonstrated in Table 4, Figure 7, and Figure 8. These experimental findings indicate that each classifier is more likely to differentiate between the Positive and Negative classes due to attaining an AUC-ROC value closest to 1, as per (ibid).

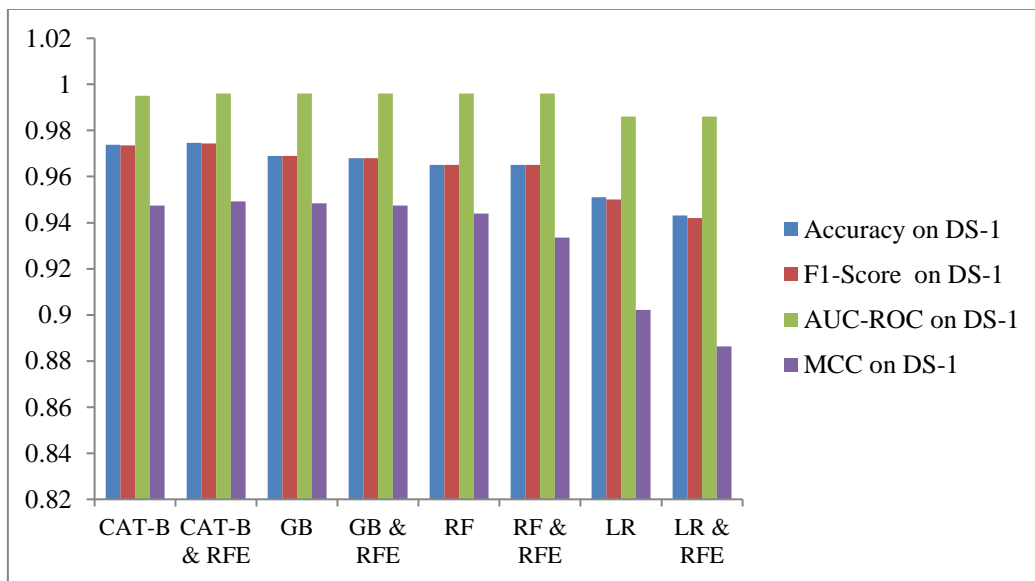


Figure 7: Classifiers Performance Analysis on DS-1 before and after the RFE Technique

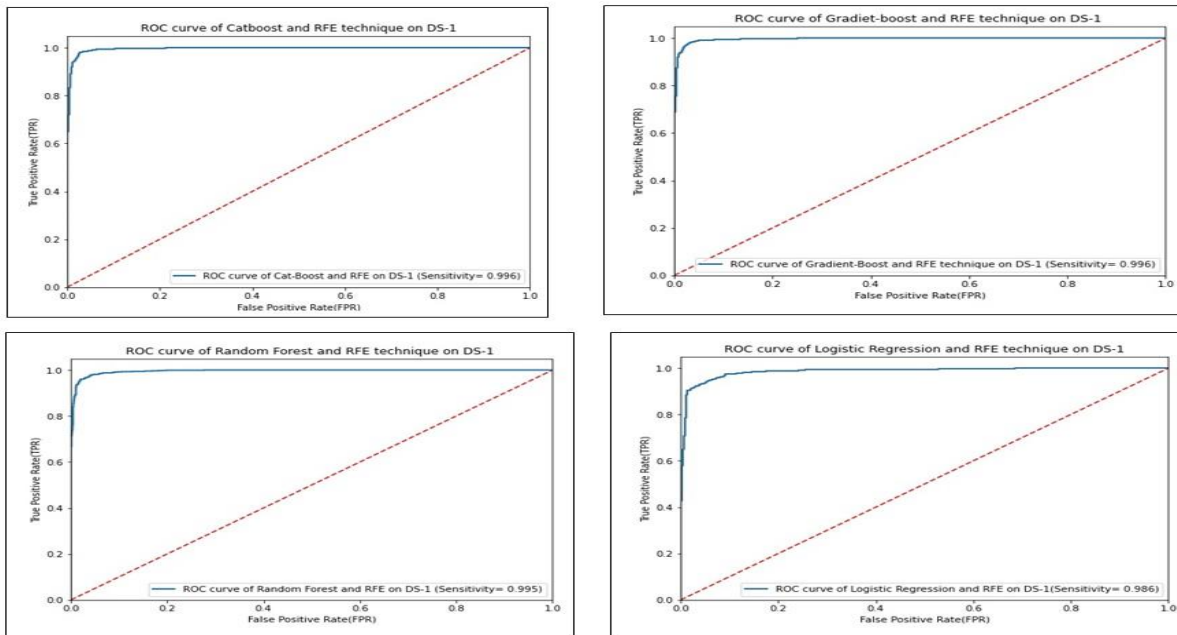


Figure 8: Classifiers AUC-ROC Curve on DS-1 after the RFE Technique

Classifiers Performance Analysis on DS-1 before and after the MI Technique

Our experimental findings demonstrate the suitability of the MI technique for the CAT-B Classifier in terms of (Accuracy, F1-Score, AUC-ROC, and MCC score) because following the application of the MI technique, the CAT-B Classifier's overall performance in terms of Accuracy, F1-Score, AUC-ROC, and MCC were found to increase from 0.9738, 0.9735, 0.995, and 0.9475 respectively to 0.9742, 0.9739, 0.996, and 0.9442. Contrarily, the train-test computational time of CAT-B-MI increased by 2 seconds (from 8 to 10 seconds), as shown in Table 5.

The accuracy, F1-Score, and MCC scores of the CAT-B-UFS, CAT-B-PCC, and CAT-B-RFE were higher than the CAT-B-MI Accuracy, F1-Score, and MCC scores. The CAT-B-MI scored the same AUC-ROC (0.996) as the CAT-B-UFS and CAT-B-RFE. The CAT-B-MI Accuracy (0.9742) and F1-Score (0.9739) were superior to the Accuracy and F1-Score of the GB-MI, RF-MI, and LR-MI. The AUC-ROC scores for CAT-B-MI, GB-MI, and RF-MI were all the same (0.996), as indicated in Table 5.

Even though the GB Classifier attained the same Accuracy (0.969), F1-Score (0.969), and AUC-ROC (0.996) both before and after using the MI technique, the GB Classifier AUC-ROC score was found to increase from 0.9484 to 0.9510 following the applications of the MI technique and after using the MI technique, the GB Classifier train-test computational time increased from 20 Minutes and 34 Seconds to 20 Minutes and 58 Seconds. As indicated in Table 5, the GB-MI MCC score (0.9510) was found to be superior to the MCC scores of the CAT-B-MI (0.9484), RF-MI (0.9388), and LR-MI (0.9003).

Our experimental findings demonstrate the MI technique's unsuitability for the RF Classifier in terms of (Accuracy, F1-Score, MCC, and train-test computational time) because following the application of the MI technique, the RF Classifier Accuracy, F1-Score, and MCC Value were found to decrease from 0.965, 0.965, and 0.944 respectively to 0.963, 0.963, and 0.9388. The RF Classifier train-test computational time they were increased from 1 Minute and 44 Seconds to (2 Hours: 26 Minutes: 26 Seconds).

Our experimental findings demonstrate the MI technique's unsuitability for the LR Classifier in terms of (Accuracy, F1-Score, and MCC) because following the application of the MI technique, the LR Classifier Accuracy, F1-Score, and MCC Value were found to decrease from 0.951, 0.9501, and 0.9021 respectively to 0.9501, 0.9489, and 0.9003 respectively. The LR Classifier train-test computational time was the same (2 Minutes) before and after applying the MI technique. The GB and RF Classifiers achieved the same AUC-ROC score of (0.996) before and after using the MI technique. Following the application of the MI technique, the CAT-B Classifier's AUC-ROC score increased from 0.995 to 0.996 and the LR Classifier's AUC score risen from 0.986 to 0.987. The LR Classifier attained the same Accuracy (0.9501), AUC-ROC (0.987), and MCC Score (0.9003) after the application of both the UFS and MI techniques as shown (in Table 2 and Table 5), respectively.

Table 4

Classifiers Performance Analysis on DS-1 before and after the MI Technique

Classifiers	Accuracy on DS-1	F1-Score on DS-1	AUC-ROC on DS-1	MCC on DS-1	Confusion Metrics	Train-Test Compute Time
CATB-Model	0.9738	0.9735	0.995	0.9475	[1125, 30] [30,1101]	8 Sec.
GB-Model	0.969	0.969	0.996	0.9484	[1121, 34] [25,1106]	20 Min: 34 Sec.
RF-Model	0.965	0.965	0.996	0.944	[1122, 33] [31,1100]	1 Min:44 Sec.
LR-Model	0.951	0.9501	0.986	0.9021	[1107, 48] [64,1067]	2 Sec.

Results Before MI ↑
Results After MI ↓

Classifiers after MI	Accuracy on DS-1	F1-Score on DS-1	AUC-ROC on DS-1	MCC on DS-1	Confusion Metrics	Train-Test Compute Time
CATB-MI	0.9742	0.9739	0.996	0.9484	[1125, 30] [29,1102]	10 Sec.
GB-MI	0.969	0.969	0.996	0.9510	[1125, 30] [26,1105]	20 Min:58 Sec.
RF-MI	0.963	0.963	0.996	0.9388	[1119, 36] [34,1097]	1 Min:59 Sec.
LG-MI	0.9501	0.9489	0.987	0.9003	[1104, 51] [63,1068]	2 Sec.

Following the application of the MI technique, the GB Classifier demonstrated superior correlation or MCC value of (0.9510), followed by CAT-B-MI, RF-MI, and LR -MI with correlation values of 0.9484, 0.9388, and 0.9003, respectively. These experimental findings demonstrate a strong connection between the predicted and observed classes due to the MCC values attained by CAT-B, GB, and RF Classifiers closer to 1 (Ibrahim et al., 2022; Chicco & Jurman, 2020). After employing the MI technique, each Classifier achieved >0.98 AUC-ROC values, as demonstrated in Table 5, Figure 9, and Figure 10. These experimental findings indicate that each classifier is more likely to differentiate between the Positive and Negative classes due to attaining an AUC-ROC value closest to 1, as per (ibid).

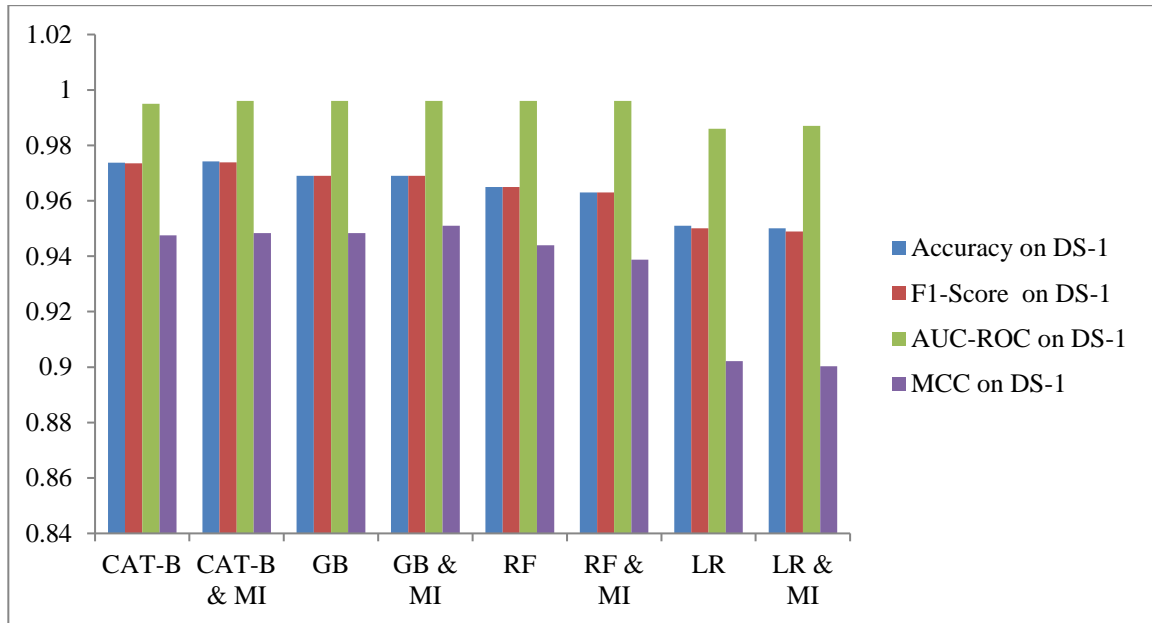


Figure 9: Classifiers Performance Analysis on DS-1 before and after the MI Technique

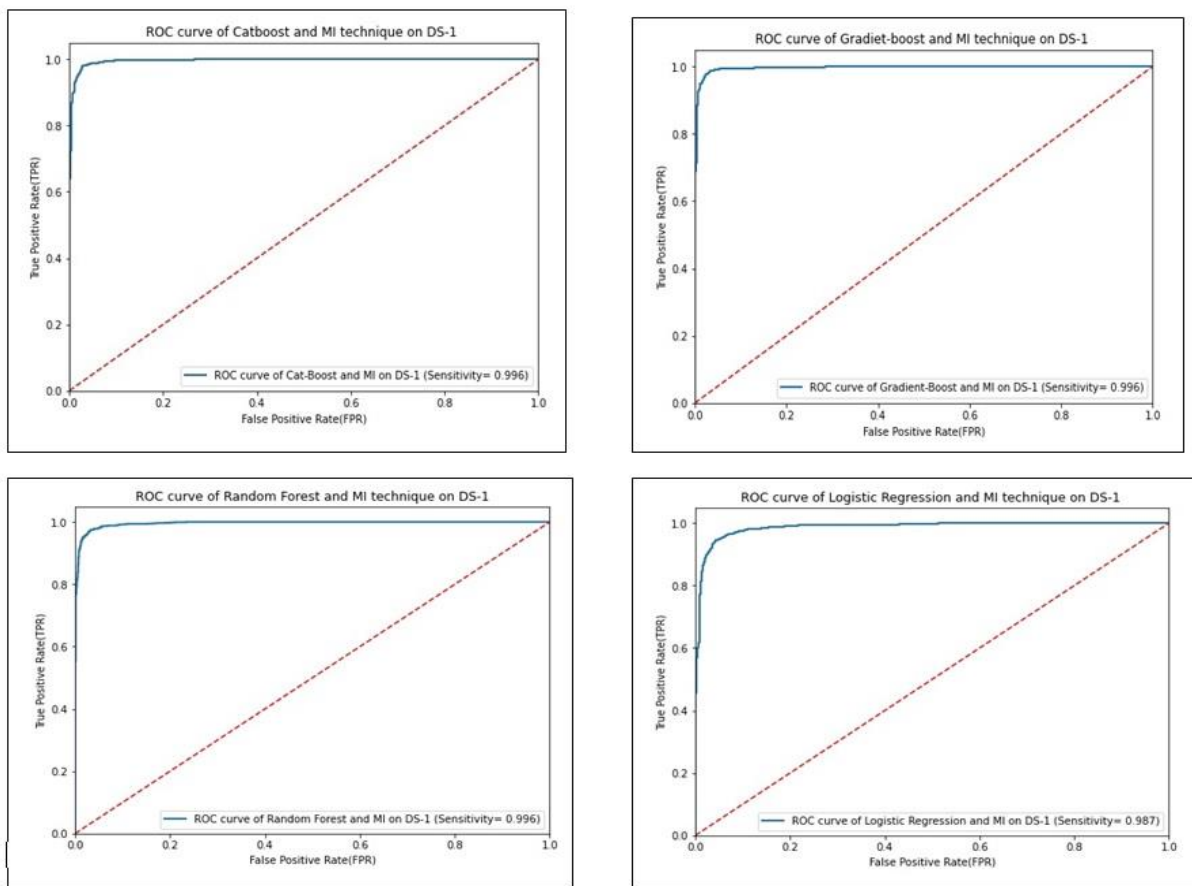


Figure 10: Classifiers AUC-ROC Curve on DS-1 after the MI Technique

RQ#3: What are the strengths and weaknesses of each Classifier after being applied with multiple Informative Feature Selection Techniques?

In this study, an attempt has been made to identify the strengths and weaknesses of each Classifier after being applied with multiple feature selection techniques such as the UFS, PCC, RFE, and MI, as shown in Table 6.

The **Green shaded color** indicates that the model's performance is boosted regarding Accuracy, F1-Score, AUC-ROC, and MCC. At the same time, the train-test computational time is reduced after using a specific informative feature selection technique. According to our experimental results, the CAT-B and UFS technique combination was determined to fully satisfy all model assessment metric criteria, followed by the CAT-B and PCC technique combination, as shown in Table 6. The CAT-B-RFE and CAT-B-MI met most model evaluation metric criteria except for the computational train-test computational time. The **Yellow shaded color** indicates that the model performance was the same before and after applying the informative feature selection technique in terms of Accuracy, F1-Score, AUC-ROC, MCC, and the train-test computational time. In our investigation, using a limited number of informative features to achieve the same accuracy that F1-Score, AUC-ROC, and MCC reached before informative selection is better than using all website features in Datasets. The GB-UFS, GB-MI, RF-UFS, and RF-RFE fall under this category regarding Accuracy, F1-Score, and AUC-ROC results. The **Red shaded color** indicates the model performed poorly in Accuracy, F1-Score, AUC-ROC, MCC, and the train-test computational time, following the application of a particular informative feature selection technique. The GB-RFE, RF-MI, LR-RFE, and LR-MI fell under this category regarding Accuracy, F1-Score, MCC, and Train-test computational time.

Table 5

Classifiers' Strength and Weakness after Being Applied with Informative Feature Selection Techniques

Classifiers	After	Accuracy	F1-Score	AUC-ROC	MCC	Time Taken
CAT-B	UFS	Increased	Increased	Increased	Increased	Decreased
	PCC	Increased	Increased	Equal	Increased	Decreased
	RFE	Increased	Increased	Increased	Increased	Increased
	MI	Increased	Increased	Increased	Increased	Increased
GB	UFS	Equal	Equal	Equal	Increased	Decreased
	PCC	Decreased	Decreased	Equal	Increased	Decreased
	RFE	Decreased	Decreased	Equal	Decreased	Increased
	MI	Equal	Equal	Equal	Increased	Increased
RF	UFS	Equal	Equal	Equal	Decreased	Decreased
	PCC	Equal	Equal	Decreased	Decreased	Decreased
	RFE	Equal	Equal	Equal	Decreased	Increased
	MI	Decreased	Decreased	Equal	Decreased	Increased
LR	UFS	Decreased	Decreased	Increased	Decreased	Decreased
	PCC	Decreased	Decreased	Equal	Decreased	Equal
	RFE	Decreased	Decreased	Equal	Decreased	Increased
	MI	Decreased	Decreased	Increased	Decreased	Increased

According to (Masoudi-Sobhanzadeh, et al., 2019), in any regression or classification activity, data preparation is a crucial step because specific data may have redundant and misleading effects, while other data may not affect the performance of Classifiers. Therefore, choosing the optimal and small size features from the bulky ones is found to be vital to boosting

model accuracy, train-test computation time reduction, and fighting model over-fitting issues (Hannousse & Yahiouche, 2021; Masoudi-Sobhanzadeh, et al., 2019). Our experimental findings demonstrate that no single feature selection technique is found to boost the Accuracy, F1-Score, AUC-ROC, MCC, and the train-test computational time of each classifier. This means that the performance of each Classifier depends on the type of feature selection technique(s) chosen and the nature and characteristics of the Datasets. This was the primary driving force for the study's decision to undertake a performance analysis of the best model and an informative feature selection method on two distinct, reliable datasets.

RQ#4: Could the results of the top-performed Classifier and Informative Feature Selection Technique on Dataset one (DS-1) be consistent on Dataset two (DS-2)?

According to our experimental findings (Table 5), the combinations of the CAT-B and UFS techniques was determined to fully satisfy all model assessment metric criteria by boosting the model's performance in terms of Accuracy, F1-Score, AUC-ROC, MCC, and the train-test computational time, as compared to the remaining Classifiers and informative feature selection techniques. The CAT-B and UFS technique combinations attained 0.9764 accuracies, 0.9762 F1-Score, 0.996 AUC-ROC, and 0.9528 MCC Value with 6 Seconds train-test computational time. Because CAT-B-UFS performed best in this study, the performance of the CAT-B and UFS techniques on DS-1 and DS-2 were compared to ensure that the results were consistent.

According to Hancock and Khoshgoftaar (2020), the Cat-Boost (CAT-B) Classifier is an open-source, enhanced version of Gradient Boosting (GB), and it is a family member of an ensemble machine learning technique. CAT-B can automatically handle categorical variables for classification and regression tasks (Ibrahim, et al., 2022; Hancock & Khoshgoftaar, 2020). CAT-B Classifier introduced new advancements to boosting algorithms. It contained novel techniques such as Ordered Target Statistics (OTS). It Ordered Boosting techniques as base predictors for automatic encoding of a categorical variable when building a decision tree, using permutation-driven random dataset sample selection strategy, address prediction fluctuation issues caused by target leakage, balanced, fast, and less prone to issues associated with over-fitting (Ibrahim, et al., 2022; Hancock & Khoshgoftaar, 2020). These could be the main reasons the CAT-B Classifier demonstrated a superior phishing website detection performance in this study.

Uni-variate Feature Selection (UFS) is a category of a filter-based feature selection technique used to assess the significance of each attribute that is found to be independent of the others (Mourtaji, Bouhorma, Alghazzawi, Aldabbagh & Alghamdi, 2021). The UFS conducts an Analysis of Variance (ANOVA) to identify the attributes that strongly correlate with or substantially impact the target variable. The P-value (0.05) is used as the cut-off value and the f-statistic value as the score to select relevant attributes. Since the UFS is a filter-based feature selection technique, it selects the informative features at the data preprocessing stage or independently of the Machine Learning Algorithms. This could be one of the main reasons the UFS technique demonstrated faster train-test computational time when applied with each Classifier used in this study.

Table 6

Preferred Parameter Values for CAT-B-UFS on DS-1 and DS-2

Classifiers and Feature Selection Technique	Preferred Parameter Values on DS-1	Preferred Parameter Values on DS-1
CAT-B-UFS	<p>=>6 Max-Tree Depth, 200 iterations, the top 62 website features, and 80%:20% train-test dataset split were preferred for CAT-B and UFS.</p> <p>=> DS-1 contained 11,430 instances of Phish-Legitimate websites and 87 attributes, and it was balanced (has 50%:50%) phish-legitimate website ratios.</p>	<p>=>9 Max-Tree Depth, 100 iterations, 27 top website features, and 70%:30% train-test dataset split were preferred for CAT-B and UFS.</p> <p>=>DS-2 contained 11,054 instances of Phish-Legitimate websites and 31 attributes, and it was nearly balanced (has 56%: 44%) Phish-Legitimate website dataset ratios.</p>

As can be seen in Table 8, following the applications of the UFS technique, the accuracy (0.9764) and F1-Score (0.9762) attained on DS-1, and the accuracy (0.9720) and F1-Score (0.9748) attained DS-2 by the CAT-B-UFS was found to be superior to each Classifier Accuracy and F1-Score attained on DS-1 and DS-2. The MCC value (0.9432) attained on DS-2 by the CAT-B-UFS was superior to each Classifier MCC value attained on DS-2. The train-test computational time attained on DS-1 and DS-2 by the CAT-B-UFS was faster than that of the GB-UFS and RF-UFS attained on DS-1 and DS-2. On the other hand, the testing set accuracy of 0.9764 (97.64%) attained by the CAT-B-UFS was considered to be the highest as compared to the results attained by another study (Hannousse & Yahiouche, 2021) that used DS-1 and obtained 96.83% accuracy by the Random Forest Classifier and the Chi-Square technique. The testing set Accuracy of 0.9720(97.20%) and F1-Score 0.9748(97.48%) attained by the CAT-B-UFS was considered to be the highest as compared to the results attained by another study by (Abedin et al., 2020) that used DS-2 and obtained 97% F1-score by the Random Forest Classifier.

Table 8 shows that when the CAT-B-UFS technique was tested on DS-1 and DS-2, the Accuracy, F1-Scores, MCC, AUC-ROC, and Train-test Computational Time were better and closer. When tested on DS-1, the CAT-B-UFS achieved better accuracy (0.964), F1-score (0.9762), and MCC value (0.9528) than the CAT-B-UFS did when tested on DS-2 (accuracy (0.9720), F1-score (0.9748), and MCC value (0.9432)). However, when tested on DS-2, the CAT-B-AUC-ROC UFS's (0.997) and train-test computational time (4 Seconds) were found to be better than the CAT-B-UFS's AUC-ROC (0.996) and Train-test Computational time (6 Seconds) attained on DS-1.

In this study, the RF Classifier demonstrated improved performance in terms of Accuracy, F1-Score, AUC-ROC, and train-test computational time when experimenting on DS-2 as opposed to Accuracy, F1-Score, AUC-ROC, and train-test computational time attained by the RF Classifier both before and after applying each pertinent feature selection technique on DS-1. As was stated in the study (Hannousse & Yahiouche, 2021), classifiers such as Random Forest, SVM, and Decision trees are quite sensitive to the order of attributes in the datasets. According to (Chiew, et al., 2019), the Random Forest Classifier could attain better accuracy when implemented with the Hybrid feature selection technique than when implemented with single feature selection techniques. From the pieces above of evidence, it is possible to conclude

that the Random Forest Classifier was a Dataset (DS) and Feature Selection Technique dependent. The MCC values obtained by CAT-B-UFS on DS-1 and DS-2 values closer to 1 or >0.94 show a good correlation between the predicted and observed classes. As shown in Table 8, our experimental results show that the CAT-B-UFS classifier is more likely to differentiate between the Positive Class and the Negative Class due to the AUC-ROC value reached on DS-1 and DS-2, closest to 1 or >0.995.

Table 7

CAT-B-UFS Performance Comparisons on DS-1 and DS-2 against the GB-UFS, RF-UFS, and LR-UFS

Classifier and FST	Dataset	Accuracy	F1-Score	AUC-ROC	MCC	Confusion Matrix	Time Taken
CAT-B-UFS	DS-1	0.9764	0.9762	0.996	0.9528	[1126, 29] [25,1106]	6 Seconds
	DS-2	0.9720	0.9748	0.997	0.9432	[1428, 51] [42,1796]	4 Seconds

Performance Comparisons on DS-1&DS-2

Classifier and FST	Dataset	Accuracy	F1-Score	AUC-ROC	MCC	Confusion Matrix	Time Taken
GB-UFS	DS-1	0.969	0.969	0.996	0.9545	[1126, 29] [23,1108]	20 Minutes & 25 Seconds
	DS-2	0.969	0.968	0.997	0.9414	[1432, 47] [49,1789]	2 Minute &13 Seconds
RF-UFS	DS-1	0.965	0.965	0.996	0.9379	[1121, 34] [37,1094]	1 Minute &42 Seconds
	DS-2	0.968	0.967	0.997	0.9377	[1421, 58] [44,1794]	21 Seconds
LR-UFS	DS-1	0.9501	0.9493	0.987	0.9003	[1104, 51] [63,1068]	1 Second
	DS-2	0.927	0.926	0.977	0.8468	[1352,127] [124,1714]	2 Seconds
=>6 Max-Tree Depth, 400 estimators, 67 top website features, and 10-Fold Cross Validations were better parameter values for the GB-UFS on DS-1				=>7 Max-Tree Depth, 200 estimators, 25 top website features were better parameter values for the GB-UFS on DS-2.			
=>15 Max-Tree Depth, 200 estimators, 61 top website features, and 10-Fold Cross Validations were better parameter values for the RF-UFS on DS-1.				=>15 Max-Tree Depth, 100 estimators, 26 top website features were better parameter values for the RF-UFS on DS-2.			
=>Solver: 'newton-cg', 100 iterations, 76 top website features, and 80%-20% dataset splits were better parameter values for the LR -UFS on DS-1.				=>Solver: 'liblinear', 100 iterations, 27 top website features, 5-Fold Cross Validations were better parameter values for the LR-UFS on DS-2.			

RQ#5: How can to demonstrate the superior implementation of the phishing website detection model in practice?

In this study, the CAT-B-UFS was determined to fully satisfy all model assessment metric criteria by boosting the model's performance in terms of Accuracy, F1-Score, AUC-ROC, MCC, and the train-test computational time, as compared to the remaining Classifiers and informative feature selection techniques. Therefore, the study tried to exhibit the practical implementations of the CAT-B-UFS using Python code as follows:

```
!pip install catboost

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting catboost
  Downloading catboost-1.1.1-cp38-none-manylinux1_x86_64.whl (76.6 MB)
    76.6 MB 1.3 MB/s
Requirement already satisfied: plotly in /usr/local/lib/python3.8/dist-packages (from catboost) (5.5.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.8/dist-packages (from catboost) (3.2.2)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages (from catboost) (1.15.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from catboost) (1.7.3)
Requirement already satisfied: pandas>=0.24.0 in /usr/local/lib/python3.8/dist-packages (from catboost) (1.3.5)
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.8/dist-packages (from catboost) (1.21.6)
Requirement already satisfied: graphviz in /usr/local/lib/python3.8/dist-packages (from catboost) (0.10.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=0.24.0->catboost) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=0.24.0->catboost) (2022.6)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->catboost) (1.4.4)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib->catboost) (0.11.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->catboost) (3.0.1)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.8/dist-packages (from plotly->catboost) (8.1.0)
Installing collected packages: catboost
Successfully installed catboost-1.1.1
```

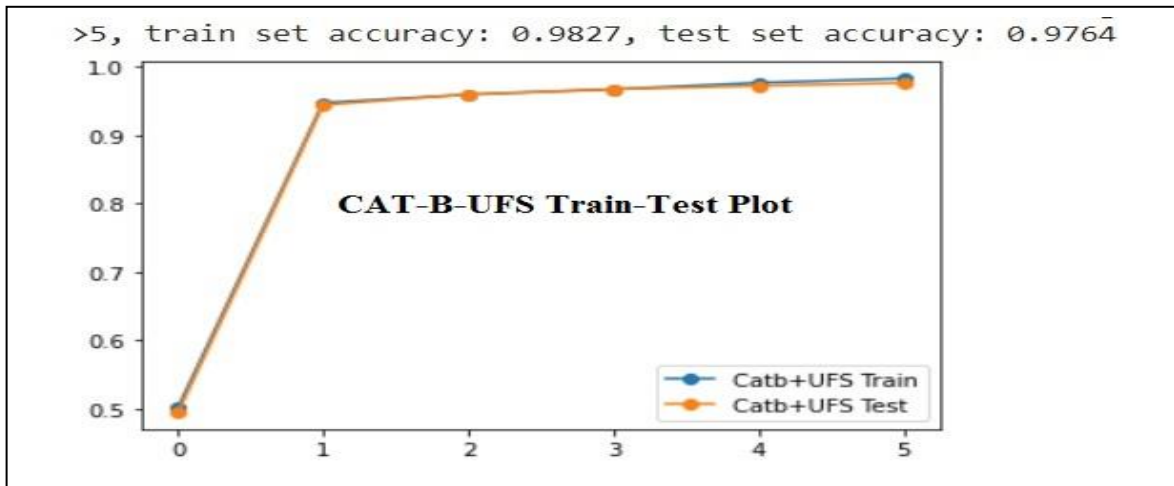
CAT-B Classifier Installation

```
from sklearn.metrics import accuracy_score
from sklearn import metrics
from numpy.core.fromnumeric import trace
from sklearn.feature_selection import f_classif, f_regression
from sklearn.feature_selection import SelectKBest, SelectPercentile
from matplotlib import pyplot
#How to select the top 62 features using Univariate Feature Selection (UFS)
ufs = SelectKBest(f_classif, k=62).fit(X_train.fillna(0), y_train)
#X_train.columns[ufs.get_support()] # to display the lists of the top features selected by the UFS
y_train_ufs=ufs.transform(X_train)
y_test_ufs=ufs.transform(X_test)
train_result, test_result = list(), list() # lists to hold tree depth results
treedepthresults = [treedepth for treedepth in range(6)]# Define maximum range of tree depth
for treedepth in treedepthresults:
    # configure the parameters for the cat-boost classifier and UFS
    CAT_B = CatBoostClassifier(max_depth=treedepth,iterations=200,learning_rate = 0.1)
    CAT_B_ufs=CAT_B.fit(y_train_ufs,y_train)# fitting sample data and the selected features to CAT-B-Model
    train_valholder = CAT_B_ufs.predict(y_train_ufs)
    train_accuracy = accuracy_score(y_train, train_valholder) # define train set-accuracy holder
    train_result.append(train_accuracy)
    test_yufspred = CAT_B_ufs.predict(y_test_ufs)
    test_accuracy = accuracy_score(y_test, test_yufspred)# Define testing set-accuracy holder
    test_result.append(test_accuracy)
    # Display Train-test accuracy of the Cat-Boost Classifier and UFS as per defined maximum tree depth
    print('>%d, train set accuracy: %.4f, test set accuracy: %.4f' % (treedepth, train_accuracy, test_accuracy))
# train-test plot
pyplot.plot(treedepthresults, train_result, '-o', label='Catb+UFS Train')
pyplot.plot(treedepthresults, test_result, '-o', label='Catb+UFS Test')
pyplot.legend()
pyplot.show()
```

CAT-B-UFS Implementation code

```
# Visualizing the Confusion Matrix
from sklearn.metrics import confusion_matrix
CBUFScm = confusion_matrix(y_test, test_yufspred)
print("Confusion Matrix of Cat-Boost and UFS on DS-1 is:\n", CBUFScm)
print("*****")
print("Model Evaluation Cat-Boost and UFS on DS-1 is:")
from sklearn.metrics import matthews_corrcoef
#the performance of the Cat-Boost Classifier +UFS interms of (Accuracy, F1-Score, MCC, Precision, Recall, FPR, FNR)
#Accuracy=the Sum of (TNR+TPR) divided by the Sum of (TNR+FPR+FNR+TPR)
print("Test-set Accuracy after UFS is : %.4f" % ((CBUFScm [0,0] + CBUFScm [1,1])/(CBUFScm [0,0] + CBUFScm [0,1] + CBUFScm [1,0] + CBUFScm [1,1])))
recall = CBUFScm[1,1]/(CBUFScm [1,0] + CBUFScm [1,1])#Recall=TPR divided by the sum of (FNR+TPR)
print("Recall value after UFS is : %.4f" % (recall))
precision = CBUFScm [1,1]/(CBUFScm [0,1] + CBUFScm [1,1]) #Precision=TPR divided by the sum of (FPR+TPR)
print("Precision value is: %.4f" %(precision))
print("F-measure is: %.4f" % (2*((precision*recall)/(precision + recall))))
print("False Negative rate(FNR): %.4f" %(CBUFScm[1,0]/(CBUFScm [1,1] + CBUFScm[1,0]))) #FNR= FNR divided by the sum of (TPR+FNR)
print("False Positive rate(FPR): %.4f" %(CBUFScm[0,1]/(CBUFScm [0,1] + CBUFScm [0,0])))#FPR=FPR divided by the sum of (FPR+TNR)
print("MCC value of Catb-UFS is: %.4f" % matthews_corrcoef(y_test,test_yufspred))
#MCC= (TPR x TNR) divided by Squire Root ((TPR + FPR) x (TPR+FNR) x (TNR+FPR) x (TNR+FNR))
print("*****")
```

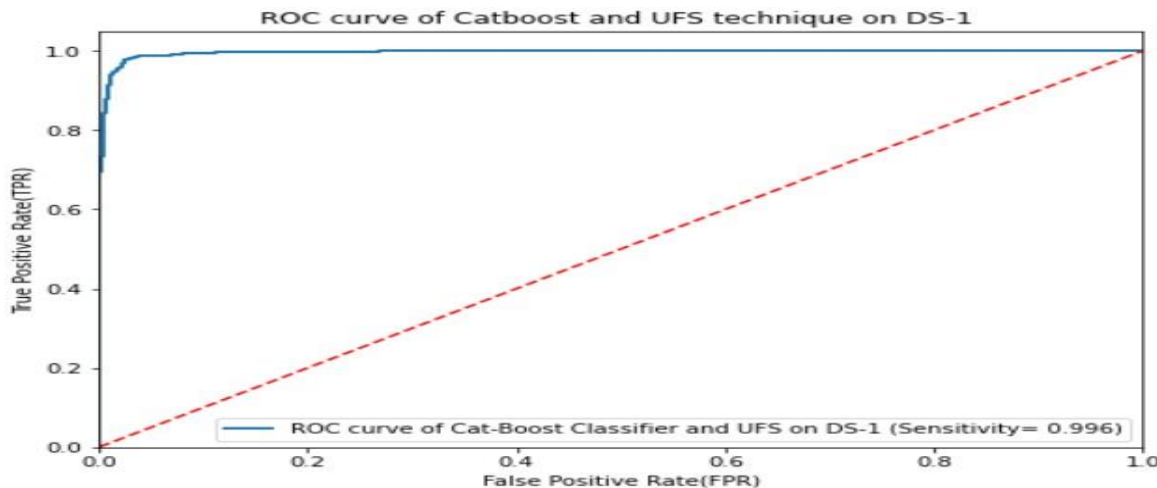
How to computing the Performance of CAT-B-UFS using Multiple Model Evaluation Metric



```

Confusion Matrix of Cat-Boost and UFS on DS-1 is:
[[1126  29]
 [ 25 1106]]
*****
Model Evaluation Cat-Boost and UFS on DS-1 is:
Test-set Accuracy after UFS is : 0.9764
Recall value after UFS is : 0.9779
Precision value is: 0.9744
F-measure is: 0.9762
False Negative rate(FNR): 0.0221
False Positive rate(FPR): 0.0251
MCC value of Catb-UFS is: 0.9528
*****
    
```

CAT-B-UFS Model Evaluation Results on Dataset(DS-1)



Discussion

Online activities in today's society include interacting with a particular website or webpage. Phishing websites, on the other hand, rank among the most common forms of cybercrime and are designed to get sensitive data, including SSNs, credit card numbers, ATM passcodes, login credentials, and significant barriers to online activity. Phishing website attacks are dynamic since they rely on the skills and analytical abilities of the attacker. The misjudgment of online

users could result in the loss of money, valuable data like (health records, military secrets, and bank customer records), credibility, and reputation, negatively impacting national security. Addressing such types of attacks requires the intervention of cutting-edge techniques like artificial intelligence in general and machine learning in particular. However, the performance of these techniques relies on the nature of datasets or requires the use of cleaned and representative datasets along with the optimal model parameters and informative feature selection techniques to speed up model prediction time, enhance accuracy, and address over-fitting issues.

In light of the abovementioned issues, some of the best classifiers for detecting phishing websites found in recent studies, such as RF, GB, and LR, were employed in this study. The most recent classifier, Cat-Boost (CAT-B), was also included in the experiments, which yielded encouraging results despite not being incorporated by similar studies.

Despite the scientific community using some relevant feature selection techniques, we could not identify a single technique that was universally appropriate for most classifiers. In light of this, we conducted rigorous experiments on each classifier with and without diverse, relevant feature selection techniques like UFS (ANOVA-F-test), MI, PCC, and RFE. Each classifier performed relatively better when combined with the UFS (ANOVA-F-test) technique than with the RFE, MI, and PCC techniques, as per the model evaluation findings shown in Table 5.

The first Dataset (DS-1) experimental findings show that greater accuracy is attained by combining UFS, PCC, MI, and RFE techniques with the CAT-B classifier despite the MI and RFE techniques increasing the computational time. As can be seen from Table 5, the combination of the UFS (ANOVA-F-test) technique with the CAT-B classifier enhanced (accuracy, F1-score, AUC-ROC, and MCC score) while cutting down computational time. Similarly, combining the PCC technique with the CAT-B classifier improved (accuracy, F1 score, and MCC score) while reducing computational time and scoring the same AUC-ROC result. Combining the UFS technique with the GB classifier enhanced the MCC score and attained the same (accuracy, F1 score, and AUC-ROC) while reducing computational time. Combining the UFS technique with the RF classifier reduced computational time while scoring the same (accuracy, F1 score, and AUC-ROC). Despite exhibiting the fastest computational time, applying each feature selection technique with the LR classifier demonstrates poor performance as per results indicated by most model evaluation metrics.

Each reviewed research work underscored a notable limitation in assessing the reliability of the phishing website detection model's capabilities: the utilization of only one dataset. (See literature review section). To fill the gaps mentioned above, in this study, each classifier experimented with two reputable public datasets (i.e., balanced and imbalanced datasets) to look for each classifier's performance consistency. It was done based on the insight that the classifier exhibited higher and consistent performance in the distinct nature of the dataset and can learn and adapt to the newly devised phishing websites. For instance, the RF classifier performed best in 17 of 30 recently reviewed research works, with a mini-max accuracy of 94.6% and 99.57% per systematic review findings (Adane & Beyene, 2022). However, in this study, RF did not perform at the greatest level in terms of accuracy when compared to the performances of CAT-B and GB despite having faster processing speeds in both datasets. This exhibits the dataset-dependent nature of the RF classifier. As noted in a recent study (Hannousse and Yahiouche, 2021), Machine Learning algorithms like RF, Decision trees, and

SVM are quite sensitive to the order of attributes in the datasets. The experimental findings in Table 7 reveal that combining the CAT-B classifier and the UFS technique exhibits superior accuracy and acceptable train-test computational time in both datasets.

Despite using the same dataset (DS-1), the accuracy (97.64%) attained by our proposed approach, i.e., the CAT-B-UFS combination, demonstrates an improvement of 0.81% over the accuracy achieved by Hannousse and Yahiouche (2021). Despite using the same dataset (DS-2), the accuracy (97.2%) attained by our proposed approach, i.e., the CAT-B-UFS combination, demonstrates an improvement of 0.2% over the accuracy achieved by the study (Abedin et al., 2020).

Another significant gap noted in the reviewed research works (Abedin et al., 2020; Hossain et al., 2020; Singhal et al., 2020) was focusing on model accuracy while overlooking the reporting of train-test computational time. It is crucial to analyze its accuracy and computing time to ensure a model is suitable for real-time implementation. Because of this, our research concentrated on reporting model accuracy and computing time.

The main contributions of our research work are threefold: i) identification of a single better feature selection technique for each classifier, ii) identification of a single best-performed classifier when applied to balanced and imbalanced datasets, identification of proper model evaluation metric when using imbalanced dataset like Matthews's Correlation Coefficient (MCC) and iii) exhibiting the implementation process of the top -performing model using Python code including its optimal parameters to facilitate the reproducibility of the research findings.

Conclusions

Most problems in the different sectors today are solved using deep learning algorithms despite requiring High-Performance Computing (HPC) machines to conduct rigorous experiments and deployment for use. Hence, re-validating the significance of Machine Learning approaches in the perspectives of phishing website detection is vital to account for resource-constrained devices in developing continents like Africa in general and countries like Ethiopia in particular.

In this study, a significant attempt has been made to overcome problems associated with phishing website detection. The study implemented Multiple Supervised M-Learning algorithms such as Cat-Boost, Gradient-Boost, Random Forest, and Logistic Regression. The study applied multiple Informative Feature selection techniques and multiple Cross-Validation techniques to validate their effects on the performance of each Classifier experimentally. The study used different model optimization parameters such as maximum tree depth, estimators/iterations, and Solvers. The study used two other reputable datasets named DS-1 and DS-2 to test the performance consistency of the proposed phishing website model. The study explored the feature selection techniques that have more, less, and no contributions to the Classifiers' performance in terms of Accuracy, F1-Score, AUC-ROC, MCC, and train-test Computational time. The study practically demonstrated the implementations of the top-performed Classifier and Feature Selection technique using Python code to allow upcoming researchers to replicate their results and learn more.

According to our experimental findings, no single feature selection technique is suited for all Classifiers to boost the Accuracy, F1-Score, AUC-ROC, MCC, and the train-test computational time. This means that the performance of each Classifier depends on the type of

informative feature selection technique(s) chosen and the nature and characteristics of the Datasets. This was the primary driving force for the study's decision to undertake a performance analysis of the best-performed model and an informative feature selection technique on two reliable datasets.

According to our experimental findings, the CAT-B and UFS combinations demonstrated higher and more consistency (Accuracy, F1-Score, AUC-ROC, and MCC) while decreasing train-test computational time when both DS-1 and DS-2 experimented. In contrast, the LR Classifier was the only Classifier that attained poor Accuracy, F1-Score, and MCC values following applications of each feature selection technique, such as UFS, PCC, RFE, and MI. The CAT-B-RFE and CAT-B-MI combinations exhibited increasing (Accuracy, F1-Score, AUC-ROC, and MCC) while increasing train-test computational time. In this study, each Classifier showed slow train-test computational time following the applications of the RFE and MI techniques.

Following the applications of the UFS and PCC techniques, each Classifier showed a shorter train-test computational time. In this study, the GB-RFE combination attained the longest train-test computing time (2 Hours, 26 Minutes, and 26 Seconds), while the LR-UFS combination attained the shortest train-test computational time (1 Second). The GB-UFS combination in this study achieved the most incredible MCC value (0.9545), closely followed by the CAT-B-UFS combination, which had an MCC value of 0.9528. Our experimental findings demonstrate a strong connection between the predicted and observed classes due to the MCC values attained by the CAT-B, GB, and RF Classifiers closer to 1. In this study, each Classifier achieved >0.98 AUC-ROC values. These experimental findings indicate that each classifier has a greater probability of differentiating between the Positive Class and the Negative Class due to the AUC-ROC value of each classifier being closer to 1

In future work, the study planned to implement deep learning algorithms with proper feature selection techniques on Individual and Hybrid approaches to undertake a rigorous comparative performance analysis to obtain more promising results so that it can be adopted by global institutions in general and Ethiopian Institutions, in particular, to boost their cyber security defense strategy from the phishing website detection perspective.

Acknowledgments

The authors thank all participating editors and anonymous reviewers for their valuable suggestions and comments. The authors would like to thank Arba Minch University for providing the funding necessary for our study, which has the project code GOV/AMU/Ph.D./TH02/AMiT/FCSE/02/15 and GOV/AMU/SMALLSCALE/ TH02/AMiT/FCSE/01/15 to be completed successfully.

References

- Abdelhamid, N., Ayesh, A. & Thabtah, F. (2014). Phishing detection based Associative Classification data mining. *Expert Systems with Applications*, 41(13), 5948–5959. <https://doi.org/10.1016/j.eswa.2014.03.019>
- Abedin, N. F., Bawm, R., Sarwar, T., Saifuddin, M., Rahman, M. A. & Hossain, S. (2020, December). Phishing attack detection using machine learning classification techniques. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)* (pp. 1125-1130). IEEE. <https://doi.org/10.1109/ICISS49785.2020.9315895>

- Adane, K. & Beyene, B. (2022). Machine learning and deep learning based phishing websites detection: the current gaps and next directions. *Review of Computer Engineering Research*, 9(1), 13–29. <https://doi.org/10.18488/76.v9i1.2983>
- Ali, W. & Malebary, S. (2020). Particle Swarm Optimization-Based Feature Weighting for Improving Intelligent Phishing Website Detection. *IEEE Access*, 8, 116766–116780. <https://doi.org/10.1109/ACCESS.2020.3003569>
- Althnian, A., AlSaeed, D., Al-Baity, H., Samha, A., Dris, A. Bin, Alzakari, N., Abou Elwafa, A. & Kurdi, H. (2021). Impact of dataset size on classification performance: An empirical evaluation in the medical domain. *Applied Sciences (Switzerland)*, 11(2), 796. <https://doi.org/10.3390/app11020796>
- APWG. (2023). *Phishing activity trends report, 2nd Quarter 2023*. Retrieved from https://docs.apwg.org/reports/apwg_trends_report_q2_2023.pdf?_gl=1*_4onbyz*_ga*MTI2NTYwMjQ1Ni4xNjk5Nzk5Njk4*_ga_55RF0RHXSr*MTY5OTc5OTY5OC4xLjAuMTY5OTc5OTY5OC4wLjAuMA..&_ga=2.86143135.64577318.1699799699-1265602456.1699799698
- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1),6. <https://doi.org/10.1186/s12864-019-6413-7>
- Chiew, K. L., Tan, C. L., Wong, K. S., Yong, K. S. C. & Tiong, W. K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, 484, 153–166. <https://doi.org/10.1016/j.ins.2019.01.064>
- Flach, P. A. (2003). The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 194-201).
- Gupta, B. B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A. & Chang, X. (2021). A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. *Computer Communications*, 175, 47–57. <https://doi.org/10.1016/j.comcom.2021.04.023>
- Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: an interdisciplinary review. *Journal of Big Data*, 7, 94. <https://doi.org/10.1186/s40537-020-00369-8>
- Hannousse, A. & Yahiouche, S. (2021). Towards benchmark datasets for machine learning based website phishing detection: An experimental study. *Engineering Applications of Artificial Intelligence*, 104, 104347. <https://doi.org/10.1016/j.engappai.2021.104347>
- Hossain, S., Sarma, D. & Chakma, R. J. (2020). Machine learning-based phishing attack detection. *International Journal of Advanced Computer Science and Applications*, 11(9), 378–388. <https://dx.doi.org/10.14569/IJACSA.2020.0110945>
- Ibrahim, B., Ewusi, A. & Ahenkorah, I. (2022). Assessing the suitability of boosting machine-learning algorithms for classifying arsenic-contaminated waters: A novel model-explainable approach using SHapley Additive exPlanations. *Water*, 14(21), 3509. <https://doi.org/10.3390/w14213509>
- INSA. (2020). *INSA foils cyber attacks from Egypt*. A 6th Months Cyber-attack Reports dated on June 23, 2020 Via Ethiopian News Agency. Retrieved from https://www.ena.et/web/eng/w/en_15454

- INSA. (2022a). *An increasing level of cyber-attacks in Ethiopia*. A 6th Months Cyber-attack Reports dated on February 14, 2022. Retrieve from <https://www.facebook.com/INSA.ETHIOPIA/posts/319500900216492>
- INSA. (2022b). *Causes of Walta-info Facebook website hacking*. A 6th Months Cyber-attack Reports dated on February 14, 2022. Retrieved from <https://www.facebook.com/INSA.ETHIOPIA/posts/319430846890164>
- Jain, A. K. & Gupta, B. B. (2019). A machine learning based approach for phishing detection using hyperlinks information. *Journal of Ambient Intelligence and Humanized Computing*, 10(5), 2015–2028. <https://doi.org/10.1007/s12652-018-0798-z>
- Masoudi-Sobhanzadeh, Y., Motieghader, H. & Masoudi-Nejad, A. (2019). FeatureSelect: A software for feature selection based on machine learning approaches. *BMC Bioinformatics*, 20(1), 170. <https://doi.org/10.1186/s12859-019-2754-0>
- Mourtaji, Y., Bouhorma, M., Alghazzawi, D., Aldabbagh, G. & Alghamdi, A. (2021). Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network. *Wireless Communications and Mobile Computing*, 2021, 8241104. <https://doi.org/10.1155/2021/8241104>
- Odeh, A., Alarbi, A., Keshta, I. & Abdelfettah, E. (2020). Efficient prediction of phishing websites using multilayer perceptron (mlp). *Journal of Theoretical and Applied Information Technology*, 98(16), 3353–3363. Retrieved from <http://www.jatit.org/volumes/Vol98No16/14Vol98No16.pdf>
- Singhal, S., Chawla, U. & Shorey, R. (2020, January). Machine learning & concept drift based approach for malicious website detection. In *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)* (pp. 582-585). IEEE. <https://doi.org/10.1109/COMSNETS48256.2020.9027485>
- Tang, L. & Mahmoud, Q. H. (2021). A survey of machine learning-based solutions for phishing website detection. *Machine Learning and Knowledge Extraction*, 3(3), 672–694. <https://doi.org/10.3390/make3030034>