International Journal of Advanced Technology and Engineering Exploration, Vol 11(121)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2024.111100562

Research Article

A novel phishing detection model using PolyScore feature evaluator and enhanced soft voting ensemble

Swetha C.V.[1*], Sibi Shaji[2] and B. Meenakshi Sundaram[3]
Research Scholar, School of Computational Sciences & IT, Garden City University, Bangalore, India[1]
Assistant Professor, School of Computational Sciences & IT, Garden City University, Bangalore, India[2]
Professor, Department of Computer Science & Engineering, New Horizon College of Engineering, Bangalore, India[3]

Received: 15-April-2024; Revised: 13-December-2024; Accepted: 14-December-2024

Abstract
Phishing remains a significant cybersecurity threat, particularly on social media platforms widely accessible online. With the increasing prevalence of social media usage and phishing incidents, there is an urgent need to enhance detection accuracy. This research addresses the challenge by leveraging advanced feature selection and classification techniques to propose a novel phishing detection model. The proposed model integrates a PolyScore feature evaluator and an enhanced soft voting ensemble method with geometric mean weighting. The methodology comprises three phases: (1) Data preprocessing, ensuring data cleanliness and readiness for classification; (2) Feature selection using the PolyScore evaluator, which employs six filter techniques with normalized rank percentiles to identify relevant features; and (3) Classification, combining predictions from seven classifiers using a geometric mean weighting-based soft voting ensemble. Ensemble pruning is applied during testing to remove classifiers with lower weights, optimizing the final results. Extensive experiments were conducted on three publicly available datasets, demonstrating superior performance in phishing detection. The proposed model achieved accuracy rates of 97.36% on dataset 1, 87.18% on dataset 2, and 98.27% on dataset 3. When compared to fifteen advanced classifiers, the model outperformed all others in phishing detection on dataset 1 and ranked as the second-best on dataset 3. This research introduces an innovative approach to phishing detection, combining weighted soft voting and ensemble pruning techniques to improve accuracy and robustness. The results highlight the model's effectiveness in addressing the growing challenge of phishing on online social networks, offering a more reliable solution compared to existing methods.

Keywords
Phishing detection, Social media security, Feature selection, Ensemble learning, Soft voting, Cybersecurity.

1.Introduction
Social networking has become an integral part of daily life. People worldwide use platforms such as Facebook, Instagram, and Twitter to communicate, stay informed, and explore the world [1]. Both individuals and businesses leverage social media to promote products, share updates about promotions and events, and attract new customers [2]. This makes social media an attractive platform for phishers to carry out their attacks. Social media phishing (SMP) occurs when attackers target users through platforms such as Instagram, LinkedIn, Facebook, or Twitter [3]. With over 1.3 billion users engaging on various social media platforms each month, SMP has become an increasingly appealing and effective tactic for fraudsters.

The availability of tools like Hidden Eye and ShellPhish further simplifies the process of executing these attacks [4]. Theft of personal information through such methods can result in fraud and other criminal activities.

While the majority of phishing attacks are still carried out via email, the trend is rapidly shifting as SMP attacks prove to be more effective. Research by Blackhat revealed that phishing attempts on social media have a success rate of up to 66%, compared to only 13.7% for email phishing, according to Google's analysis [5]. Additionally, the federal trade commission (FTC) reported a 30% rise in phishing-related losses in 2022, totaling $8.8 billion [6]. More specifically, scammers exploit LinkedIn, Instagram, Facebook, and Twitter to create fraudulent profiles, impersonate individuals, and spread cryptocurrency

*Author for correspondence

investment or giveaway scams through fake celebrity profiles. Social media providers account for 10.4% of phishing attacks, indicating inadequate cybersecurity measures and persistent threats posing significant challenges to detection and prevention efforts [7]. This highlights the urgent need to detect such persistent threat on online social networks (OSN).

The traditional methods often rely on simplistic rule-based analysis and blacklisting to provide immediate protection against known threats [8]. However, they struggle to detect newly emerging phishing threats and adapt to the dynamic nature of the evolving attacks [3]. This challenge has led to the adoption of machine learning (ML) classifiers in phishing detection, as they offer the capability to detect and adapt to the dynamic nature of phishing scams more effectively [9]. Traditional ML models for phishing detection often rely on rule-based classifiers and individual learning techniques such as k-nearest neighbour (KNN), random forest (RF), support vector machine (SVM), naïve Bayes (NB), and decision trees (DT) that lack robustness in detecting modern phishing attacks [10]. Thus, these models end up with high false-positive rates and low detection accuracy [4]. As a result of these limitations, researchers have been focusing more of their attention on ensemble learning approaches that combine numerous classifiers to improve detection accuracy [11]. Furthermore, even though these models have demonstrated promising results in phishing detection, there is a gap in the research about the optimal combination of feature selection approaches and classification models. Thus, existing methods frequently fail to examine the significance of feature selection and the full potential of ensemble models in the process of phishing detection in social media networks [6, 8]. The limitations of traditional detection methods and individual ML classifiers indicate the need for advanced approaches.

To address this research gap, this research seeks to develop a novel phishing detection model for detecting phishing uniform resource locators (URLs) in OSNs. The primary objective of the study is to enhance detection accuracy using multiple feature selection and classification methods within a unified framework. This proposed method utilises a PolyScore feature evaluator that makes use of various filtering feature evaluators for computing the feature scores and applies a normalised rank percentile to find the final scores. After selecting the important features, the model employs a multiplicative average weighting-based soft voting ensemble with multiple classifiers. The weights are determined using the geometric mean, and the results are combined through weighted soft voting. During this process, classifiers with the lowest weights are eliminated to optimize performance.

Phishing detection on OSNs presents unique challenges due to the rapid evolution of attack techniques and the vast volume of user-generated content. To address these issues, the proposed approach leverages advanced feature extraction with the PolyScore feature evaluator and a multiplicative average weighting-based soft voting ensemble. This model significantly enhances detection accuracy, minimizes false positives (FPs), and adapts more effectively to the dynamic nature of phishing attacks on OSNs compared to existing methods.

This study contributes to the development of more effective strategies for detecting phishing threats. Thus, the key contributions of the research are:

1. Development of a novel feature selection model to identify the most relevant features for classification analysis.
2. Proposed a novel classification model that employs enhanced soft voting ensembles with multiplicative average weighting and ensemble pruning.
3. Assessment of the proposed feature selection and classification models across various datasets to detect phishing attacks.
4. Comparison of the proposed model with existing conventional and state-of-the-art phishing detection models.
5. Identification of the limitations of the proposed model and suggestions for future enhancements.

This paper is structured as follows: Section 2 explores the literature associated with the proposed field. Section 3 explains the proposed phishing detection model, which incorporates a PolyScore feature evaluator and a multiplicative average weighting-based soft voting ensemble. Section 4 discusses the experimental and result analyses, including the dataset description, implementation details, and results analysis. Section 5 analyses the effectiveness of the proposed model by examining its strengths, implications, and limitations. Finally, section 6 concludes the work by discussing potential enhancements for future research.

2.Literature review

The field of phishing detection has been the subject of several proposed studies. A systematic literature review by [6] provides a comprehensive classification of phishing, its types, and its prevention, highlighting the contributions of leading organizations in anti-phishing research and development. Many researchers have used ML classifiers to analyze the performance and efficiency of phishing website detection [12]. Vaitkevicius and Marcinkevicius (2020) [13] evaluated 8 classifiers and reported that multilayer perceptron (MLP) and gradient tree boosting (GTB) classification methods offered improved performance in phishing detection. Researchers proposed and validated a new detection approach using gradient learning and sigmoidal threshold levels, utilizing NB, DT, and SVM classifiers for accuracy and efficiency [9]. A hybrid rule induction algorithm, combining repeated incremental pruning to produce error reduction (RIPPER) and partial decision tree (PART) algorithms, was proposed for separating phishing websites from genuine ones. Experiments show promising results, with an accuracy of 0.9453 and 0.9908, outperforming RIPPER and PART [10]. A study by [14] proposed a malicious URL detection model using three supervised ML classifiers: SVM, LR, and NB, with the NB algorithm showing the best results in detecting malicious URLs.

Boukhalfa et al. (2022) developed a new phishing attack detection system for Twitter using 23 features and the MLP, an artificial neural network (ANN) algorithm, achieving a 96% success rate in recent data [15]. A study proposed an ML detection model for phishing emails using three different datasets [16]. The model's accuracy and efficiency were achieved using a different number of features, with the best results achieved at 0.88, 1.00, and 0.97 consecutively using a boosted DT. Seven different ML algorithms to identify phishing URLs using nine different features were suggested [17], with the RF model achieving the highest accuracy of 95.2%. Furthermore, RF offered an improved accuracy of 87% over DT (82.4%) in detecting phishing attacks [18]. Researchers proposed a novel hybrid ensemble feature selection (HEFS) framework for ML-based phishing detection systems, and found that the RF classifiers effectively detected 94.6% of phishing and legitimate websites using only 20.8% of the original features [19]. Numerous studies have demonstrated the effectiveness of RF in various feature selection methods, including raker-based feature selection and principal component optimisation [20], the term frequency-inverse document frequency (TF-IDF) value of webpages [21], and feature selection using principal component analysis (PCA) [22]. Numerous studies confirm that RF is a superior model for detecting phishing from legitimate websites [23–25].

However, [26] claimed that NB acquired 98% of the highest accuracy in phishing URL detection. Both RF and extreme gradient boosting (XGB) demonstrated improved phishing detection, achieving accuracy of 98.5% and 98.33%, respectively, on the phishing detection task [27]. A comprehensive analysis of the performance of ML algorithms on multiple datasets by [28] revealed that RF and ANN outperform other classification methods, achieving over 97% accuracy. A study by [29] evaluated eight classifiers and reported that neural networks (NN), specifically RF, stacking ensemble models (SEM), AdaBoost (AB), and GTB, are highly effective in detecting phishing, while Bayesian and instance similarity-based classifiers perform poorly. The study supports previous research suggesting ensemble classification schemes, NN, and DT yield the best classification results, while classifiers like RF, SVM, MLP, and classification and regression trees (CART) do not.

Abdul et al. (2023) [30] experimented with three tuning factors, including data balancing, hyper-parameter optimization, and feature selection, on ML methods and datasets from University of California, Irvine (UCI) and Mendeley repositories. The results indicate that data balance marginally improves accuracy, while hyperparameter adjustment and feature selection significantly enhance performance, outperforming existing research. This method was similar to the one proposed by [31], who employed K-fold feature selection and the hyperparameter tuning method and reported that RF offered improved performance. Researchers proposed an ML-based aggregation analysis mechanism for automated page layout-based phishing detection, evaluating the accuracy of four popular classifiers and the factors influencing their results [32]. The research used a DT and wrapper for feature selection, achieving a detection accuracy rate of 96.32%. It then used two meta-heuristic algorithms, harmony search (HS) and SVM, to predict and detect fraudulent websites [33]. Researchers reported that HS outperformed SVM with accuracy rates of 94.13% and 92.80%.

Several classification algorithms are combined to form an ensemble model for improving the accuracy of detection. A study compared the effectiveness of ensemble techniques, including sequential and

parallel methods, in identifying spear phishing and found that they outperformed traditional ML algorithms [4]. A study proposed two hybrid ensemble learning phishing email detection (HELPHED) methods: stacking ensemble learning and soft voting ensemble learning [11]. ML algorithms were also used to handle hybrid features, improving performance and minimising complexity. Experimental tests show ensemble learning, specifically soft voting ensemble learning, outperforms other algorithms on a rich dataset. A study presented an optimised stacking ensemble method for detecting phishing websites, utilizing genetic algorithms to tune ML methods such as RF, AB, XGB, Bagging, gradient boost (GB), and light gradient boosting machine (GBM) [34]. The best three models were chosen as base classifiers for this optimised stacking ensemble method, providing improved detection accuracy. Taha (2021) [35] introduced an intelligent ensemble learning method for detecting phishing websites using weighted soft voting, which uses a base classifier with four ML algorithms. The results indicate that the intelligent approach outperformed base classifiers and soft voting methods, achieving 95% accuracy. This method is the basis for the proposed phishing detection model. Apart from ML and ensemble learning methods, deep learning (DL) methods were also widely applied in detecting phishing attacks [36]. A study presented a hybrid DL model for phishing URL detection that uses deep NN methods and long short-term memory (LSTM) [37]. The experimental analysis indicates that this model outperformed other phishing detection models in terms of accuracy metrics. PhishingNet, a DL method for detecting phishing URLs, was proposed by [38]. It uses convolutional neural networks (CNN) to extract spatial and temporal features, fused with an attention-based hierarchical recurrent neural network (RNN), and trains a phishing URL classifier. Automatic feature representations improve generalization ability on new URLs, achieving competitive performance against other state-of-the-art approaches, according to experiments. Ashour et al. (2024) [39] published a study that showed an anti-phishing strategy for IoT systems in fog networks that used ML and DL had a 99.37% success rate with real-world datasets, showing its efficiency in real-time phishing detection. *Table 1* summarizes the significant recent studies related to the proposed work.

Table 1 Summary of notable related recent work

Study	Objective	Methods	Results	Advantages	Limitations
Boukhalfa et al. (2022) [15]	To propose a detection system for phishing attacks on Twitter.	ML algorithms for social media data.	Successfully identified phishing attempts on Twitter.	Addresses a niche area of phishing detection.	Limited to Twitter; may not be applicable to other platforms.
Mughaid et al. (2022) [16]	To develop an intelligent phishing detection system using DL.	Utilizes DL techniques for detection.	High accuracy and efficiency in detecting phishing attempts.	Leverages advanced DL methods for better performance.	Computationally intensive; requires significant resources.
Bountakas and Xenakis (2023) [11]	To develop a hybrid ensemble learning model for phishing email detection.	Combines ensemble learning techniques with hybrid features.	Achieved better detection performance compared to traditional methods.	Effective in reducing FPs and improving accuracy.	May require extensive feature engineering; limited to specific datasets.
Alnemari and Alshammari (2023) [25]	To detect phishing domains using ML techniques.	Utilizes various ML algorithms for classification.	High detection accuracy for phishing domains.	Demonstrates versatility across different datasets.	Performance may vary based on feature selection.
Joshi et al. (2023) [27]	To predict phishing attacks in blockchain networks.	Evaluation of multiple algorithms including SVM, RF, and NN.	Identified effective models for phishing prediction in blockchain contexts.	Provides insights into phishing detection in emerging technologies.	Limited to blockchain; may not generalize to traditional phishing scenarios.
Abdul et al. (2023) [30]	To analyze the performance impact of fine-tuned ML models for phishing URL detection.	Focus on fine-tuning various ML models for improved detection.	Fine-tuned models showed significant accuracy improvements.	Highlights the importance of model optimization.	Results may depend on the specific datasets used for training and testing.

Study	Objective	Methods	Results	Advantages	Limitations
Choudhary et al. (2023) [31]	To propose a ML approach for phishing attack detection.	Implements various ML algorithms for detection.	Achieved high detection rates with minimal FP.	Simple implementation with effective results.	May not address all types of phishing attacks; limited dataset scope.
Ashour et al. (2024) [39]	To develop an anti-phishing approach for IoT systems using ML.	Utilizes ML algorithms tailored for IoT environments.	Effective detection of phishing attempts in IoT settings.	Addresses a growing area of concern in cybersecurity.	Limited to IoT; may not be applicable to other domains.
Varshney et al. (2024) [6]	To provide a comprehensive perspective on anti-phishing strategies.	Review of various anti-phishing techniques and their effectiveness.	Identified key trends and challenges in phishing detection.	Comprehensive overview aids in understanding the field.	Lacks empirical data; primarily a review article.

An analysis of these significant existing studies reveals that most models employing ML techniques face certain limitations. Many focus on evaluating ML classifiers without incorporating all relevant features necessary for effective phishing detection. Additionally, numerous studies rely on a single dataset to assess their models, while some perform comparative analysis only on conventional classifiers. A prevalent use of synthetic datasets further complicates comparative evaluation. These models often suffer from low detection rates and high false alarm rates, highlighting the need for a novel phishing detection model to address these challenges effectively.

3.Methods

This section presented a novel phishing detection model designed to predict phishing links from OSNs. The model incorporates a PolyScore feature evaluator and an enhanced average weighting-based soft voting ensemble utilizing a geometric mean. Instead of relying on a single model, it combines decisions from multiple ML algorithms for feature selection and classification. This approach leverages the collective strengths of these algorithms, offering the potential to achieve better results than individual techniques.

The model comprises three primary phases for effectively detecting phishing incidents. The first phase is the collection and preprocessing of data; the second is the PolyScore feature evaluator, which is a feature selection phase; and the third is the average-weighing soft voting ensemble classifier, which is a classification phase. The data collection and preprocessing phase cleans the dataset and prepares it for the classification process. The features are selected in the feature selection phase after six filter techniques have been applied and evaluated using normalized rank percentiles for the computed final scores. In the classification phase, the results obtained from seven ML classifiers are blended by combining their prediction probabilities with weights calculated using geometric means that incorporate several evaluation measures. Moreover, ensemble pruning is performed by removing classifiers with the lowest weights during the testing phase. During the feature selection and classification phases, this ensemble of models complements the existing strengths and weaknesses among these models. Additionally, to differentiate between genuine and phishing websites, a weighted soft voting method is employed. This approach combines the outcomes of classifiers by assigning higher weights to base learners with greater influence and lower weights to those with less impact, enhancing the overall detection accuracy. To ensure effective weight computation, an array of metrics is used to compute the classifiers' weights instead of single performance measures. *Figure 1* depicts the framework of the proposed method for detecting phishing websites.

3.1Data collection and preprocessing

This section outlines the data collection and preprocessing steps involved in distinguishing phishing websites from legitimate ones. While this study evaluates three publicly available datasets, the data preprocessing methodology presented here is broadly applicable to phishing detection. Details of the specific datasets used in this study, which were sourced from publicly accessible repositories, are provided in Section 4.1.

For phishing URL detection, training datasets are typically created by gathering URLs from various sources, including legitimate websites and phishing URLs obtained from repositories such as PhishTank, OpenPhish, and similar platforms [15]. The URLs collected are classified as either authentic or phishing, depending on the established criteria. Additionally, URLs from social media sites are

collected using web crawlers or scrapers. However, it is not possible to use the URLs as a feature; instead, they are analyzed to extract a wide variety of features, including the protocol used, length of the URL, number of characters, number of special characters ("//", ".", "@") [40], host features such as name, host location, IP address, primary domain, its age, registration length, path, and more [14]. Features extracted from the address bar, hypertext markup language (HTML) or JavaScript, and domain can also be included to predict phishing from legitimate URLs. Features such as URL length, number of special characters, and protocol used are crucial for phishing detection as they often indicate suspicious or deceptive URL structures. Special characters and URL length can reveal attempts to obfuscate the true nature of a phishing site. Host features like IP address, domain age, and registration length provide additional context to help identify potentially fraudulent sites. Additionally, features extracted from the address bar, HTML, JavaScript, and domain-related attributes enhance the detection process by providing more granular details about the URL's legitimacy and potential risk.

After extracting all the possible features for predicting phishing websites, the dataset undergoes preprocessing to ensure it is clean, as it includes duplicated data and samples with missing values. This step enhances the effectiveness of the ML classifiers, as raw data can affect the model's learning capacity. The preprocessing step removes duplicate samples, fills in or removes the missing values with the mean values for the missing entries, and rescales the values for the larger feature values in the dataset. Regardless of the ML models being used, this phase is important for the classification process.

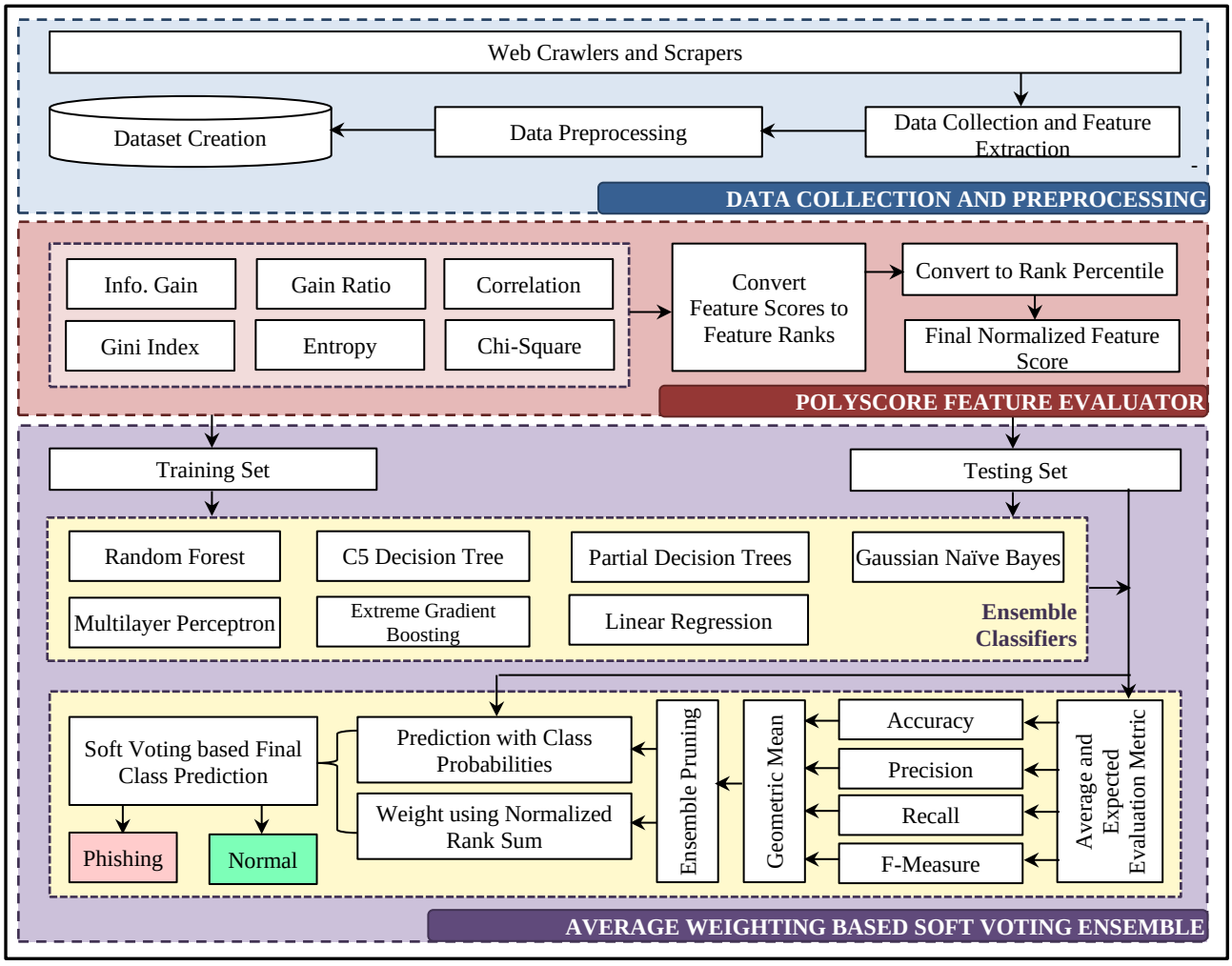


Figure 1 Overall framework of the proposed method

3.2PolyScore feature evaluator

This section presents the proposed PolyScore feature evaluator for selecting features for the classification process. The feature selection process helps identify the most informative features for classifying phishing from legitimate samples. Generally, the importance of the features is estimated using statistical analysis, domain knowledge, or ML models. The feature selection techniques can be filter-based, wrapper-based, or embedded. By evaluating features individually, filtering methods are explicit in their ranking of features according to feature characteristics, reduce overfitting, and do not require an iterative process. This makes them efficient and scalable. In the proposed model, a set of filtering approaches such as information gain (IG), gain ratio (GR), correlation (COR), entropy (ENT), Gini index (GI), and chi-square (CS) are used. These methods are discussed below.

ENT: It is a method for determining the degree of unpredictability or uncertainty in a dataset. DT frequently utilise it to determine which attribute is suitable for splitting. The formula is given in Equation 1, in which p_i is the likelihood of a sample falling into a specific class.

$$ENT = \sum_{i=1}^{n} -p_i \log_2(p_i) \quad (1)$$

IG: It measures the reduction in ENT achieved by partitioning the data by an attribute. This can be defined as the difference between the ENT of the dataset and the ENT of the feature subset. The formula is given in Equation 2, where $H(S)$ is the ENT and $|S|$ is the number of instances in the dataset S, S_i is the subset of instances for the i^{th} value of the feature, and $H(S_i)$ is the ENT of the subset S_i.

$$IG = H(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} H(S_i) \quad (2)$$

GR: It is an improvement over IG in that it applies normalisation to the information gained by the ENT of the attribute. The formula to compute the GR is given in Equation 3.

$$GR = \frac{Information\ Gain}{Entropy} \quad (3)$$

COR: It assesses the relationship between two variables using Pearson's formula. For computing the feature score, it measures the COR of the feature with all the other features and class variables. The formula is given in Equation 4, where $Cov(X,Y)$ and σ are the covariances between two variables and their standard deviations.

$$COR = \frac{Cov(X,Y)}{\sigma_x \sigma_y} \quad (4)$$

GI: It evaluates the impurity of the dataset. The formula is given in Equation 5, where p_i is the likelihood of a sample falling into a specific class.

$$GI = 1 - \sum_{i=1}^{n} p_i \quad (5)$$

CS: It evaluates the independence of two variables. In the case of feature scoring, it measures the independence between the feature and the class variable. The formula is given in Equation 6, where O_i and E_i are the observed and expected frequencies of class *i*.

$$CS = \sum_{i=1}^{n} \frac{(O_i - E_i)}{E_i} \quad (6)$$

The selected filtering-based feature selection techniques (ENT, IG, GI, GR, COR, CS) are chosen for their efficiency, scalability, and ability to reduce overfitting. These methods rank features based on intrinsic characteristics, making them ideal for quickly identifying the most informative features in large, dynamic datasets.

Upon computing the feature scores using these statistical methods, the features are ranked based on the computed feature scores for all the individual methods. Here, N is the number of features in the dataset, and M is the number of filtering methods used. The feature rank for each method is then converted to percentiles as given in Equation 7, where R_{ij} represents the rank of the feature i of method j. Equation 7 ensures that percentiles increase with lower ranks. Then the final score of the feature i can be calculated by averaging the rankpercentile$_j$ for all the methods j as in Equation 8.

$$RankPercentile_{ij} = \left(1 - \frac{R_{ij}-1}{N}\right) \times 100 \quad (7)$$

$$FeatureScore_i = \frac{\sum_{j=1}^{m} RankPercentile_{ij}}{M} \quad (8)$$

Based on the final scores, a threshold can be set, and features with scores greater than the defined threshold are selected for further analysis. Generally, the threshold value can be set at 50. However, for the dataset with fewer features, the threshold can be set as low as 40, while for the dataset with a greater number of features, the threshold can be set as high as 55 or 60. The algorithm for the PolyScore feature evaluator is shown in Algorithm 1.

Algorithm 1. PolyScore feature evaluator

Input: Input dataset with N feature, M feature selection methods, threshold for feature selection
Output: List of selected features
Procedure PolyScoreFE()
Begin
Extract features and target class from input dataset

```
    Set the threshold value
    Encode categorical variables in feature & target class
    FS = [GI', 'IG', 'GR', 'COR', 'CS', 'ENT']
    Define functions for FS methods to calculate feature
     scores
    For each method in FS:
        Calculate feature scores
        If scores are calculated successfully then
            Calculate ranks & percentiles for scores
            Sort and store the results
        End If
        Merge the current results in an array
    End For
    For each feature in the dataset
      Calculate the average_percentiles
      If average_percentiles > threshold then
            Store feature in the selected_feature_list
      End If
    End For
  Return selected_feature_list
End Procedure
```

The illustration for the proposed PolyScore feature evaluator has been conducted using a sample weather dataset, commonly known from the Kaggle Repository (https://www.kaggle.com/datasets/), to determine whether to play football or not. This dataset includes four features: Outlook, Temperature, Humidity, and Windy, along with 15 instances. All the statistical methods, such as GI, IG, GR, COR, ENT, and CS, are applied, and the feature scores are calculated. The feature scores are then converted to feature ranks, where higher scores correspond to lower ranks and vice versa. Upon computing the ranks, percentiles are assigned as outlined in Equation 7 for each statistical method. The rank percentiles of a feature are averaged across all methods to obtain its final score.

In this case, with the threshold set at 50, features such as Outlook, Humidity, and Windy, which have higher scores, are selected for further analysis. The computed values using the proposed feature evaluator are presented in *Table 2*. This evaluation is significant as it demonstrates the functionality and accuracy of the PolyScore feature evaluator in selecting relevant features, which plays a crucial role in optimizing the analysis process in this study. Moreover, using a small dataset with fewer features helps to understand the nuances of the method and its underlying processes. By systematically applying and comparing statistical methods, the evaluation ensures robustness in feature selection, a critical step in achieving reliable results for the proposed methodology.

Table 2 Illustrative summary of PolyScore feature evaluator

Evaluators	Features	GI	IG	GR	COR	ENT	CS	Final Feature Score
Feature Scores	Outlook	0.4466	0.0955	0.0333	0.092	0.0285	2.0281	-
	Temperature	0.1297	0.0209	0	0.0894	0.0000	0.0222	-
	Humidity	0.2282	0.1833	0.0089	0.0775	0.0858	1.4	-
	Windy	0.1956	0	0.035	0.4933	0.0750	0.4	-
Feature Ranks	Outlook	1	2	2	2	3	1	-
	Temperature	4	3	4	3	4	4	-
	Humidity	2	1	3	4	1	2	-
	Windy	3	4	1	1	3	3	-
Rank Percentiles	Outlook	100	75	75	75	50	100	79.167
	Temperature	25	50	25	50	25	25	33.333
	Humidity	75	100	50	25	100	75	70.833
	Windy	50	25	100	100	75	50	66.667

3.3Multiplicative average weighting based soft voting ensemble

This section presents the enhanced average weighting-based soft voting ensemble for classifying the phishing samples. The ensemble classifiers in the proposed model make use of seven strong classifiers, such as RF, C5.0, PART, Gaussian NB (GNB), MLP, XGB, and logistic regression (LR), to make a single decision. These classifiers are chosen based on the computational resources and the tradeoff between model complexity and performance. Though SVM is also a popular choice for classification, its absence in the selected list is due to empirical evidence from existing studies, as the other classifiers were proven to be effective in classifying phishing and legitimate data. The description of these classifiers is discussed below.

- *RF:* It is an ensemble model that integrates multiple DTs to improve prediction performance. It can handle a large number of features and instances. It is more accurate as it can capture complex non-linear relationships in data [41].

- *LR:* It is a linear model for binary classification and computes probabilities for all the classes. It offers improved performance when there is a linear relationship between the features and the target class [17].
- *XGB:* It aggregates the predictions of multiple weak learners, typically DT, to create a strong predictive model. It is scalable and can handle a wide range of datasets with excellent efficiency [16].
- *MLP:* It is a NN model that can learn complex patterns and is capable of capturing non-linear relationships between features. It also can handle high-dimensional datasets [13].
- *C5.0 DT:* This method uses ENT and IG, respectively, to quantify the disorder in the attribute collection and the efficacy of each attribute. It can identify outliers and is suitable for imbalanced datasets by generating clear decision rules [9].
- *GNB:* It is a probabilistic classifier based on the Bayes theorem, considering the features are independent. It is suitable for high-dimensional data and categorical features [15].
- *PART:* With a focus on identifying precise rules, this rule-based classifier creates partial DT. It is best suited for imbalanced datasets [41].

Moreover, instead of hard voting, this model makes use of soft voting, as the method is more effective when individual classifiers have varying levels of confidence in their predictions or the classes are not equally balanced. Typically, each classifier in the soft voting ensemble predicts the probability of each class, and the final predictions are averaged across all classifiers for all the classes [35]. Finally, the class with the highest average probability is selected as the final decision for the given input data.

Furthermore, to improve the prediction accuracy, instead of using the class probabilities alone, the weights are computed using a geometric mean that incorporates various evaluation metrics such as accuracy, precision, recall, and F-measure. After computing the weights of the classifiers during training, ensemble pruning is performed, in which instead of using all seven classifiers, the top five with higher weights are used in the testing phase. This step is essential since the performance of the model differs with the different datasets. Ensemble pruning reduces complexity, improves generalization, and avoids performance degradation. The description of these performance metrics and their respective formulas is given in *Table 3* [42].

Table 3 Description of performance metrics

Metrics	Description	Formula
True positive (TP)	The number of correctly classified phishing instances	
True negative (TN)	The number of correctly classified non-phishing instances	
FP	The number of non-phishing instances classified as phishing instances	
False negative (FN)	The number of phishing instances classified as non-phishing instances	
Accuracy	The ratio of correctly predicted instances to the total number of instances	$\frac{\mathrm{TP}+\mathrm{TN}}{N}$
Precision (P)	The ratio of TP predictions to the all-positive predictions made by the classifiers	$\frac{\mathrm{TP}}{TP+FP}$
Recall (R)	The ratio of TP predictions to the all-actual positive samples	$\frac{\mathrm{TP}}{TP+FN}$
F-measure	It is the harmonic mean of the precision and recall metrics	$\frac{2\times \mathrm{P}\times \mathrm{R}}{P+R}$

The proposed weighting technique emphasizes the aggregation of performance across multiple evaluation dimensions. Geometric mean, also known as multiplicative average, is a significant statistic that computes the n^{th} root of n values. The geometric mean was chosen over the arithmetic or harmonic means for weighting the classifiers' predictions because it handles probabilistic outputs more effectively, reduces the impact of extreme values, and provides a more balanced aggregation of predictions. This choice ensures greater robustness and reliability in the ensemble model's performance. The values always lie between 0 and 1. Moreover, since the probability of each class is computed by the ensemble models, the weights are also computed for each class. Thus, the final prediction can be predicted as in Equation 9, where P_{ij} is the probability of the j^{th} class from the i^{th} classifier and W_{ij} is the class weights for the j^{th} class in the i^{th} classifier. Upon computing the weights, the top five classifiers with the highest weights are considered for the final prediction by eliminating the two classifiers with the lowest weights.

$$final\ prediction = argmax_j \left(\frac{1}{N}\sum_{i=1}^{N} W_{ij}P_{ij}\right) \quad (9)$$

The weights for the soft voting are computed as in Equation 10, where A, P, R and F denote the accuracy, precision, recall, and f measure for the j^{th} class and i^{th} classifier.

$$W_{ij} = \sqrt[4]{A_{ij} \times P_{ij} \times R_{ij} \times F_{ij}} \quad (10)$$

The detailed pseudocode for the proposed multiplicative average weighting-based soft voting ensemble is shown in Algorithm 2.

Algorithm 2. Multiplicative average weighting based soft voting ensemble

Input: Input dataset with N feature, test instances, M classification methods
Output: classification of test instances
Procedure MAW_SVE()
  Begin
    Extract the features and target class from dataset
    Encode categorical variables in feature and target class
    CLF = ['RF', 'C5.0', 'PART', 'GNB', 'MLP', 'XGB', 'LR']
    Define all classifiers in CLF
    For each classifier in CLF:
      Train the classifiers
      Compute the class-wise performance metrics
      Compute weights for each class for the classifier
    End For
    Sort the classifiers based on their weights
    Select top 5 classifiers CLF_S5 with higher weights
    For the given test instance
      For each classifier in CLF_S5
        For each class label in the dataset
          Predict the probabilities
          clfr_pred = probabilities * weights
          fnl_cls_pred = fnl_cls_pred + clfr_pred
        End For
      End For
      Classify test instances as the class having the highest final_class_prediction
    End For
  Return the predictions for the test instance
End Procedure

Thus, the proposed multiplicative average weighting-based soft voting ensemble employs seven strong classifiers to make a single decision by computing class-wise weights using a geometric mean of accuracy, precision, recall, and F-measure. During testing, ensemble pruning is applied to select the top five classifiers with the highest weights, ensuring reduced complexity and better generalization. For each test instance, the weighted probabilities of each class are computed and aggregated across the selected classifiers. Finally, the class with the highest aggregated probability is chosen as the final prediction, ensuring improved accuracy and robustness in classifying phishing samples.

4.Results

The experiment and analysis of the results for the proposed framework, including implementation details, dataset description, and result analysis, are presented in this section. The minimum requirements for the experimentation include an Intel Core i5 processor, 8 GB of RAM, and 50 GB of storage. The software setup includes Python 3.x and libraries such as Scikit-learn, NumPy, Pandas, Matplotlib, and Seaborn, along with TensorFlow or PyTorch for ML tasks. Additionally, the critdd package from GitHub is utilized for generating TikZ code for vector graphics. Moreover, it is essential to ensure that all software and libraries are up-to-date to facilitate the implementation and analysis processes effectively.

4.1Dataset description

For assessing the efficiency and performance of the proposed model, three publicly accessible phishing datasets have been used in this study.

1. The UCI Phishing 2015 dataset [43] contains data collected from various legitimate and phishing websites, including sources such as PhishTank and MillerSmiles. It comprises 11,055 samples, with 30 features encompassing URLs, domain names, and HTML attributes. While it provides a diverse representation of phishing attacks, the dataset may face challenges related to class imbalance and feature variability.
2. The UCI Phishing 2016 dataset [44] consists of 1,353 samples with 10 features, categorized into legitimate, phishing, and suspicious websites. The dataset is compiled from sources such as Yahoo directories and Starting Point directories, developed using a PHP web script, along with data from the PhishTank archive. Despite its utility, the dataset presents challenges due to the diversity of sources and potential inconsistencies in representing phishing attack scenarios.
3. The MDP Phishing 2018 dataset [45] contains 10,000 samples, evenly divided between phishing and legitimate websites, with 48 features. The data is collected from sources such as PhishTank, OpenPhish, Alexa, and Common Crawl. While it effectively addresses various phishing types, the dataset may encounter challenges due to variations in feature sets and potential issues related to class diversity.

The sample datasets used in the analysis are summarised in *Table 4.*

Table 4 Details of datasets used in the study

ID	Dataset name	#Instances	#Features
Dataset 1	UCI Phishing 2015	11055 (Legitimate: 6157; Phishing:4898)	30
Dataset 2	UCI Phishing 2016	1353 (Legitimate: 548; Suspicious:103; Phishing: 702)	10
Dataset 3	MDP Phishing 2018	10000 (Legitimate: 5000; Phishing:5000)	49

4.2Implementation details

The implementation is carried out using Python 3 on Google Colab, which is a Jupyter Notebook environment that does not require any setup and is completely hosted in the cloud. The preprocessing, feature selection methods, and traditional classification models are implemented using various libraries, including Sklearn. Moreover, critdd is from GitHub, a Python package that generates Tikz code for vector graphics (https://mirkobunse.github.io/critdd/). Initially, the datasets were preprocessed, and the proposed PolyScore feature evaluator and the multiplicative average weighting-based soft voting ensemble models were applied to these datasets. The three datasets with varied numbers of features, like medium (30), small (10), and large (49) are used for the evaluation of the proposed feature selection. Here, the feature scores are computed based on the average percentiles, and 15 out of 30 features from Dataset 1 with scores greater than 50, 6 out of 9 features from Dataset 2 with scores greater than 40, and 20 out of 49 features from dataset 3 with scores greater than 55 are selected. The overview of the features in the three datasets and their scores are shown in *Table 5*, in which the selected features using the proposed feature selection model are given in bold. The proposed feature selection model is compared with other models such as IG, GR, COR, GI, ENT, and CS for analysis. Moreover, the proposed multiplicative average weighting-based soft voting ensemble model is compared with various other existing classifiers such as XGB, RF, MLP, C5.0, PART, GNB, and LR using two test options. In the first test option, the dataset is divided into training and testing sets, where the training set contains 70% and the testing set contains 30% of the data. *Table 6* provides a summary of the percentage split of samples in each class. In the second evaluation method, 10-fold cross-validation is employed. This method involves splitting the dataset into 10 subsets. In each iteration, 9 subsets are used for training, and the remaining subset is used for testing. This process is repeated 10 times, ensuring that each subset serves as the test set exactly once. This approach provides a robust evaluation of the model's performance by averaging results across all folds. Moreover, hyperparameters for the ML models were selected using grid search cross-validation. This method systematically evaluates hyperparameter combinations to identify optimal settings, ensuring models were fine-tuned for optimal performance through an exhaustive search a predefined grid. The models are assessed using various performance metrics as listed in *Table 3*.

Table 5 Feature scores using proposed feature selection method

UCI Phishing 2015 Dataset	
Feature	**Score**
web_traffic	77.222
having_Sub_Domain	74.444
Prefix_Suffix	72.778
SSLfinal_State	68.333
Request_URL	68.333
Google_Index	67.222
URL_of_Anchor	66.111
Domain_registeration_length	65.556
having_At_Symbol	65.000
SFH	63.333
Statistical_report	60.556
Links_in_tags	58.333
URL_Length	56.111
Shortining_Service	56.111
RightClick	53.333
age_of_domain	48.333
Favicon	47.222
Redirect	46.667
having_IP_Address	45.000
Page_Rank	44.167
Abnormal_URL	43.056
port	38.333
Links_pointing_to_page	36.667
Submitting_to_email	36.389
HTTPS_token	35.278
popUpWidnow	33.611
DNSRecord	32.778
double_slash_redirecting	32.500
on_mouseover	29.722
Iframe	27.500
UCI Phishing 2016 Dataset	
Feature	**Score**
SFH	81.481
popUpWidnow	75.926
Request_URL	68.519
SSLfinal_State	66.667
URL_of_Anchor	48.148
web_traffic	42.593
age_of_domain	40.741
having_IP_Address	40.741
URL_Length	35.185

MDP Phishing 2018 Dataset	
Feature	**Score**
PctExtResourceUrls	83.854
PctNullSelfRedirectHyperlinks	83.854
ExtMetaScriptLinkRT	82.292
PathLength	81.250
PctExtHyperlinks	78.646
ExtNullSelfRedirectHyperlinksRT	78.125
NumDots	77.604
PathLevel	77.604
UrlLength	76.042
NumNumericChars	73.438
NumQueryComponents	70.833
NumDash	69.271
NumAmpersand	67.708
NumSensitiveWords	62.500
SubmitInfoToEmail	62.500
PctExtResourceUrlsRT	60.938
InsecureForms	60.417
ImagesOnlyInForm	59.375
QueryLength	57.813
IpAddress	57.292
FrequentDomainNameMismatch	54.167
SubdomainLevel	53.125
DoubleSlashInPath	52.604
RandomString	52.083
IframeOrFrame	48.958
NumPercent	47.917
AbnormalExtFormActionR	47.917
EmbeddedBrandName	47.396
NumUnderscore	46.354
UrlLengthRT	45.833
NoHttps	45.313
HostnameLength	41.667
NumDashInHostname	40.625
FakeLinkInStatusBar	36.458
RelativeFormAction	32.813
SubdomainLevelRT	32.813
HttpsInHostname	31.944
ExtFavicon	31.250
DomainInPaths	30.556
AtSymbol	28.646
ExtFormAction	27.604
NumHash	27.083
DomainInSubdomains	26.042
MissingTitle	26.042
AbnormalFormAction	24.479
PopUpWindow	24.479
TildeSymbol	23.958
RightClickDisabled	15.104

Table 6 Summary of samples in percentage split

Category	**Class**	**Dataset Name**		
		Dataset 1	**Dataset 2**	**Dataset 3**
Training	Legitimate	4310	384	3500
	Phishing	3429	72	3500
	Suspicious	-	491	-
Testing	Legitimate	1847	164	1500
	Phishing	1469	31	1500
	Suspicious	-	211	-

4.3Results analysis

This section discusses the results of the experimental analysis conducted. The performance analysis begins with an evaluation of the proposed feature selection algorithm, followed by an assessment of the proposed ensemble classifiers. Finally, the output of the proposed model is compared against various state-of-the-art phishing detection models to highlight its effectiveness.

The performance of the proposed PolyScore feature evaluator is assessed using accuracy as the evaluation metric. The feature selection model is evaluated using ten classifiers, including RF, MLP, XGB, C5.0, PART, GNB, LR, SVM, KNN, and AB, for all three datasets. Further, the values are compared with the several existing feature selection models, such as the IG, GR, COR, GI, ENT, and CS models. The best accuracy scores are highlighted in bold in *Table 7*. The results for dataset 1 indicate that IG feature selection offers better performance with SVM, GR has the highest accuracy using AB, COR has improved accuracy with KNN, GI and ET show improved results with PART and RF respectively, and MLP, CS has better accuracy with LR. On the other hand, the proposed PolyScore has better performance with XGB, C5.0, and GNB classifiers. Moreover, the proposed model offers better accuracy for XGB, PART, and GNB classifiers with dataset 2 and RF, LR, and SVM classifiers with dataset 3. Though the proposed model did not outperform all the other models for all the classifiers in terms of accuracy, it still exhibits consistent, strong performance across all classifiers.

The feature selection methods are ranked based on their performance using a critical difference (CD) diagram generated with the critdd package. This package applies the Friedman test and the Wilcoxon signed-rank test, combined with Holm's method, to test hypotheses regarding significant differences in performance among the methods. The CD diagram for various feature selection methods for the three datasets is given in *Figure 2*. It is a single 2-dimensional CD diagram containing 3 CD diagrams for each dataset. The ranks in the CD diagram represent the relative performance of different feature selection methods across classifiers and datasets. In practical terms, a lower rank signifies that a method consistently outperforms others in terms of accuracy or other relevant metrics. The diagram visually summarizes the mean ranks, illustrating how often a method performs better than its competitors.

For instance, if a feature selection method consistently ranks lower across multiple datasets, it indicates its superior ability to identify the most relevant features, leading to better classification performance. The CD diagram in *Figure 2* confirms that the proposed method outperforms others by maintaining a consistently lower rank across all datasets. This highlights its effectiveness in selecting features that improve the overall accuracy and reliability of the phishing detection model. This visual representation confirms the superiority of the proposed method in a clear and interpretable manner.

Table 7 Accuracy evaluation proposed feature selection methods

Feature Selection Methods	Classifiers									
	RF	**MLP**	**XGB**	**C5.0**	**PART**	**GNB**	**LR**	**SVM**	**KNN**	**AB**
UCI Phishing 2015 Dataset										
IG	96.01	90.66	95.26	95.14	95.20	92.82	93.38	**93.43**	95.80	92.86
GR	95.87	95.63	95.41	95.17	95.11	92.16	93.55	92.97	95.45	**92.88**
COR	96.05	95.16	95.75	94.90	94.96	94.51	93.24	93.15	**92.85**	92.85
GI	96.12	95.72	94.63	95.57	**95.48**	93.00	93.40	93.06	95.81	92.61
ENT	**96.15**	**95.81**	95.74	94.89	94.96	92.85	93.30	93.16	96.17	92.83
CS	95.87	95.48	94.93	95.24	95.32	92.84	**93.61**	93.03	95.66	92.70
PolyScore	96.11	95.78	**95.88**	**95.85**	95.34	**92.83**	93.57	93.39	95.73	92.81
UCI Phishing 2016 Dataset										
IG	86.21	86.70	84.24	87.44	86.95	82.02	85.71	86.95	86.70	84.24
GR	85.96	86.45	85.22	86.45	87.44	82.19	**87.19**	86.94	87.93	85.47
COR	85.97	86.50	85.23	86.44	87.43	82.18	87.18	86.93	87.92	85.45
GI	**89.15**	**88.41**	84.42	**87.92**	87.41	82.02	85.47	85.22	**88.67**	**85.71**
ENT	86.21	86.69	84.24	87.43	86.95	82.01	85.71	**87.00**	86.70	84.24
CS	83.74	83.74	83.50	85.96	83.25	82.20	85.47	83.00	85.96	83.99
PolyScore	88.44	87.36	**86.25**	87.80	**87.94**	**82.74**	85.44	86.51	88.34	85.23
MDP Phishing 2018 Dataset										
IG	98.17	96.12	**96.90**	96.78	96.91	**84.69**	92.68	92.45	**96.87**	**94.79**
GR	97.93	95.97	96.20	96.63	96.90	83.73	92.80	92.07	95.47	94.47
COR	97.88	95.72	96.58	**97.19**	96.97	85.43	91.63	91.09	96.75	94.46
GI	97.57	95.50	96.57	96.37	96.77	82.80	92.30	91.30	94.40	93.70
ENT	98.07	95.77	96.40	96.70	**97.30**	83.53	92.10	91.90	96.50	94.27
CS	98.00	**96.20**	96.10	96.60	96.89	83.49	92.76	92.18	95.93	94.47
PolyScore	**98.20**	95.87	96.09	96.73	96.47	83.90	**93.07**	**92.60**	96.00	94.45

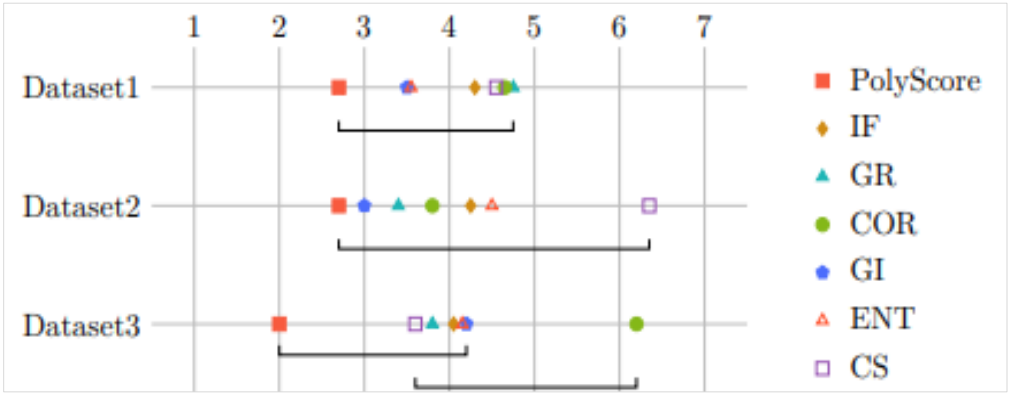


Figure 2 CD diagram for various feature selection methods

Upon selecting the features using the proposed PolyScore feature evaluator, the selected features are evaluated with the proposed multiplicative average weighting-based soft voting ensemble model, which is evaluated using percentage split and 10-fold cross-validation. Furthermore, performance is assessed using various performance metrics such as accuracy, precision, recall, and F-measure. Further, the performance of the proposed model is also compared with other models such as XGB, RF, MLP, C5.0, PART, GNB, and LR used in this study. The results of the proposed phishing detection model for the

percentage split option for Datasets 1 and 2 are shown in *Table 8*. The result indicates that the proposed model has better performance with an accuracy of 95.47%, precision of 95.44%, recall of 95.47%, and F-measure of 95.44% for dataset 1, offers 87.99% accuracy, 87.89% precision, 87.99% recall, and 87.94% F-measure for dataset. Moreover, *Table 9* presents the detailed analysis for the dataset 3 including confusion matrix for better understanding. The results indicate that the proposed model offers better performance with 98.04% accuracy, 98.02% precision, 97.95% recall, and 97.99% F-measure for dataset 3, respectively. The area under the curve (AUC) curves in *Figures 3* and *4* for datasets 1 and 3, respectively, indicate that the proposed model has better performance than many other existing classifiers under comparison.

Table 8 Analysis with percentage split for proposed ensemble model

Performance Metrics	Various Classifiers							
	Proposed	**XGB**	**RF**	**MLP**	**C5.0**	**PART**	**GNB**	**LR**
UCI Phishing 2015 Dataset								
Accuracy	95.467	95.056	95.237	95.116	94.573	94.393	59.512	91.800
Precision	95.442	95.071	95.235	95.114	94.581	94.409	78.726	91.793
Recall	95.467	95.056	95.237	95.116	94.573	94.393	59.512	91.800
F-measure	95.442	95.060	95.236	95.114	94.576	94.397	54.898	91.788
UCI Phishing 2016 Dataset								
Accuracy	87.990	86.700	86.700	84.975	87.931	87.931	81.281	85.714
Precision	87.891	86.536	86.618	83.793	87.757	87.757	74.201	83.789
Recall	87.990	86.700	86.700	84.975	87.931	87.931	70.820	81.874
F-measure	87.940	86.594	86.633	83.707	87.762	87.762	79.870	80.878

Table 9 Analysis with percentage split for proposed ensemble model with MDP phishing 2018 dataset

Classifiers	TP	FP	FN	TN	Accuracy	Precision	Recall	F-measure
Proposed	1436	29	30	1507	98.035	98.02	97.954	97.987
XGB	1432	31	30	1507	95.867	95.352	96.141	95.745
Rf	1426	37	34	1503	97.967	97.881	97.948	97.915
C5.0	1395	68	56	1481	97.633	97.471	97.671	97.571
MLP	1389	74	90	1447	94.533	94.942	93.915	94.426
PART	1359	104	65	1472	94.367	92.891	95.435	94.146
GNB	1415	48	508	1029	81.467	96.719	73.583	83.579
LR	1312	151	171	1366	89.267	89.679	88.469	89.070

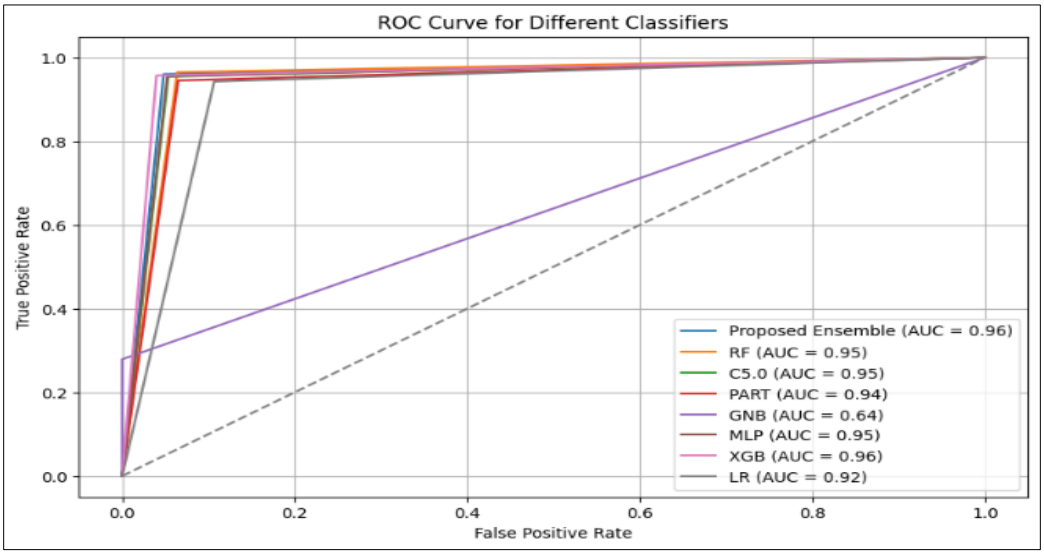


Figure 3 AUC curve for various classifiers on UCI phishing 2015 dataset

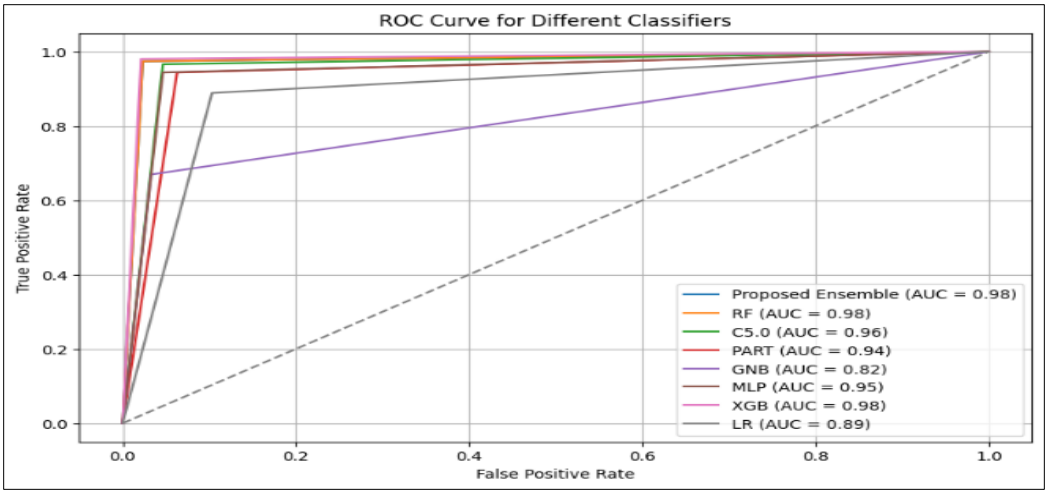


Figure 4 AUC curve for various classifiers on MDP phishing 2018 dataset

Furthermore, the performance of the proposed ensemble model is evaluated using 10-fold cross-validation, and the results are compared with those of various conventional classifiers used in the study. The outcomes of the proposed phishing detection model under the 10-fold cross-validation approach are presented in *Table 10*. The result indicates that the proposed model outperforms with an accuracy of 97.56%, precision of 97.16%, recall of 97.36%, and F-measure of 97.16% for dataset 1 and provides better results with 87.18% accuracy, 86.83% precision, 87.18% recall, and 86.66% F-measure for dataset 2. Furthermore, for better and detailed understanding, the confusion matrix is provided in *Table 11* along with the other performance metrics for dataset 3. The results indicate that the proposed model achieved improved accuracy of 98.27%, precision of 97.62%, recall of 98.18%, and F-measure of 93.35% for dataset 3. The proposed multiplicative average weighting-based soft voting ensemble model is assessed by comparing the results with other existing models and computing the CD between these models. The CD diagram for the various classification models across the three datasets is presented in *Figure 5*, where *(a)* represents the percentage split, a method where the dataset is divided into a fixed ratio (e.g., 70% for training and 30% for testing), and *(b)* represents the 10-fold cross-validation. The figure indicates that the proposed ensemble classification model for phishing detection achieved better performance across all datasets with percentage split, where RF and PART acquired the second position. Similarly, in the case of cross-validation, XGB and RF acquired the second position, while the proposed model consistently outperformed other classifiers across all datasets.

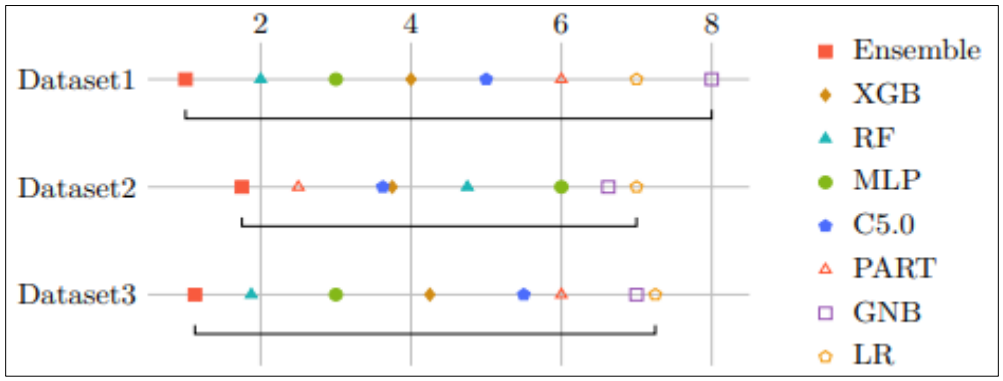
Table 10 Analysis with 10-fold cross validation for proposed ensemble model

Performance metrics	Various classifiers							
	Proposed	**XGB**	**RF**	**MLP**	**C5.0**	**PART**	**GNB**	**LR**
UCI Phishing 2015 dataset								
Accuracy	97.561	95.079	95.305	94.202	94.744	94.573	60.353	91.841
Precision	97.160	95.078	95.304	94.207	94.744	94.570	78.799	91.857
Recall	97.361	95.079	95.305	94.202	94.744	94.573	60.353	91.841
F-measure	97.157	95.078	95.303	94.194	94.744	94.570	55.578	91.823
UCI Phishing 2016 dataset								
Accuracy	87.179	86.179	86.475	85.514	85.070	85.218	80.488	81.966
Precision	86.830	85.833	86.007	83.289	84.782	84.848	77.263	77.461
Recall	87.179	86.179	86.475	85.514	85.070	85.218	80.488	81.966
F-measure	86.658	85.932	86.081	83.279	84.827	84.943	78.623	79.330

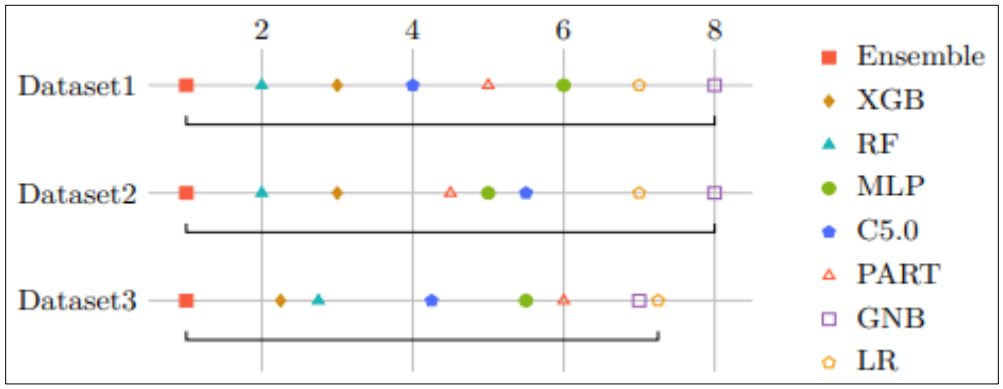
Table 11 Analysis with 10-fold cross validation for proposed ensemble model with MDP phishing 2018 dataset

Classifiers	**TP**	**FP**	**FN**	**TN**	**Accuracy**	**Precision**	**Recall**	**F-measure**
Proposed	4881	119	154	4846	98.27	97.62	98.182	98.353
XGB	4874	126	156	4844	97.11	97.6	96.653	97.124

Classifiers	TP	FP	FN	TN	Accuracy	Precision	Recall	F-measure
Rf	4880	120	169	4831	97.18	97.48	96.899	97.188
C5.0	4747	253	253	4747	93.29	92.24	94.219	93.219
MLP	4612	388	283	4717	94.94	94.94	94.94	94.94
PART	4618	382	360	4640	92.58	92.36	92.768	92.564
GNB	4840	160	1668	3332	81.72	96.8	74.37	84.115
LR	4290	710	546	4454	87.44	85.8	88.71	87.231



(a)



(b)

Figure 5 CD diagram for various classification models (a) Evaluation using percentage split (b) Evaluation using cross validation

While the above analysis demonstrates that the proposed model outperforms traditional classifiers, its reliability is further validated by comparing the results with existing state-of-the-art classifiers used for phishing attack detection. To this end, an extensive comparative analysis was conducted with state-of-the-art approaches from the literature, as detailed in *Table 12*.

Using the UCI Phishing 2015 datasets, the best accuracy results from 22 recent studies were retrieved and compared with those of the proposed study. The results show that the proposed model ranks 8th in accuracy. Although models such as fusion with DL [39], RF [31], Light GBM [34], Stacking Ensemble [29], RF [27], RF [26], and CNN+RNN [38] acquired an accuracy of above 97.8%, the proposed model's accuracy is 97.6%. This also indicates that the proposed model outperforms 15 recent studies, highlighting its achievement in achieving high accuracy. On the other hand, only five studies have reported the results for the MDP phishing 2018 dataset. The comparison indicates that the Stacking Ensemble [29] acquires the highest performance with 98.76%, while the proposed model ranks second place with 98.3% accuracy. This indicates that the proposed model is above the five recent models proposed for phishing detection. Thus, the comparison of results indicates that the proposed model has improved performance than many of the traditional and state-of-the-art classifiers.

Table 12 Comparative performance analysis

Phishing detection methods	Dataset1	Dataset2
RF with feature selection [24]	96.5	-
RF with ranker + Principal component optimization [20]	97.33	-
CNN+RNN [38]	97.9	-
Hybrid ensemble method [19]	94.6	-
Page layout features [32]	93	-
Heuristic nonlinear regression [33]	92.8	-
RF and NN [28]	97	-
MLP and GTB [13]	97.22	97.42
RF [22]	97	-
NB [26]	98.03	-
Intelligent ensemble learning [35]	95	-
RF [21]	97	-
Light GBM [34]	98.65	-
Stacking ensemble [29]	98.58	98.76
RF [23]	88.92	97.5
RF [18]	86	-
Fine-tuned with the GB method [30]	97.47	98.27
RF [31]	98.8	97.87
RF [25]	97.3	-
RF [27]	98.5	-
Fusion with DL [39]	99.37	-
Proposed soft voting ensemble	97.6	98.3

4.4Computational complexity

The computational complexity of the proposed ensemble model, which integrates RF, LR, XGB, MLP, C5.0 DT, GNB, and PART, reflects the diverse nature of these algorithms. *RF and XGB* are ensemble methods with DT. RF builds multiple DT in parallel, leading to a training complexity of $O(t.n.f.(log\ n))$, where t is the number of trees, n is the number of instances, and f is the number of features. XGB uses boosting to sequentially improve predictions, with a similar complexity due to the iterative nature of its trees and gradient updates. LR is a linear model with a complexity of $O(n.f^2)$, for training, where n is the number of instances and f is the number of features. This model is efficient when features have a linear relationship with the target. GNB has a simpler complexity of $O(n.f)$, leveraging the independence assumption among features, making it suitable for high-dimensional data. MLP is a NN with a complexity of $O(n.f.l^2)$, where l is the number of layers. The non-linear nature of MLP requires substantial computational resources for training, especially with deep networks. C5.0 DT and PART classifiers focus on decision rules and have complexities that depend on the depth and number of rules generated. For feature selection, techniques like recursive feature elimination add $O(n.f^2)$, as they iteratively evaluate subsets of features. Thus, the ensemble's prediction complexity is $O(k.n)$, where k is the number of classifiers, combining the individual complexities. This results in significant computational demands in both time and memory, driven by the varied and complex nature of the constituent algorithms.

5.Discussion

The internet and World Wide Web are crucial for modern society, with advancements in technology attracting more beneficial web applications, but also increasing cyberattacks. Phishing attacks are the most common cyberattacks, using social engineering methods to deceive users into providing sensitive information or installing malware on their computers. Phishers persist in enticing victims despite the abundance of anti-phishing solutions. Thus, this work introduces an advanced phishing detection model designed to predict phishing links from OSNs. The model leverages a PolyScore feature evaluator and a refined soft voting ensemble, enhanced by a geometric mean-based weighting approach. Instead of relying on a single model for feature selection and classification, the model aggregates judgments from multiple ML algorithms. This multi-model strategy is preferred because combining different algorithms often yields better results than individual techniques.

The experimental analysis performed can be divided into three sections: assessing the performance of the PolyScore feature evaluator, evaluating the proposed ensemble classifiers, and comparing the model's output with other state-of-the-art phishing detection models. The PolyScore feature evaluator's

performance is measured using accuracy across ten classifiers: RF, MLP, XGB, C5.0, PART, GNB, LR, SVM, KNN, and AB. Results are compared against existing feature selection models like IG, GR, COR, GI, ENT, and CS (*Table 7*). The proposed PolyScore model shows superior performance with certain classifiers across different datasets. For instance, it outperforms others with XGB, C5.0, and GNB classifiers on Dataset 1; with XGB, PART, and GNB on Dataset 2; and with RF, LR, and SVM on Dataset 3. Although PolyScore does not always achieve the highest accuracy with every classifier, it demonstrates consistently strong performance. The CD diagram ranking the feature selection methods across the three datasets highlights the superior rank of the PolyScore model compared to other methods (*Figure 2*).

After feature selection, the selected features are evaluated using the proposed soft voting ensemble model. Performance is assessed via percentage split and 10-fold cross-validation, with metrics like accuracy, precision, recall, and F-measure. The proposed model's results are compared with conventional classifiers (XGB, RF, MLP, C5.0, PART, GNB, and LR) (*Tables 8* and *9*). The model achieves notable accuracy: 95.47% for Dataset 1, 87.99% for Dataset 2, and 98.04% for Dataset 3, with corresponding precision, recall, and F-measure values for percentage split. The AUC curves for the proposed model also shows its superior performance. Additionally, the model's performance under 10-fold cross-validation with notable accuracy of 97.56% for Dataset 1, 87.18% for Dataset 2, and 98.27% for Dataset 3 is compared with other classifiers, confirming its effectiveness across all datasets (*Tables 10* and *11*). The summarized results in a CD diagram demonstrate that the proposed model consistently outperforms others, both in percentage split and cross-validation scenarios.

The superior performance of the proposed model on the MDP Phishing 2018 dataset compared to the UCI Phishing 2015 and 2016 datasets can be attributed to several factors. Firstly, the MDP Phishing 2018 dataset has a balanced distribution of phishing and legitimate samples, which likely contributes to more stable and accurate model predictions. Additionally, this dataset includes 49 features, many of which are highly informative in distinguishing between phishing and legitimate websites, as highlighted by the feature selection process. The diversity and richness of these features allow the proposed model to capture intricate patterns that may be less discernible in the UCI Phishing datasets, which have fewer features and a more imbalanced class distribution. In contrast, the UCI Phishing 2015 dataset, despite having more samples, may include features that are less discriminative, thereby affecting the model's overall performance. Similarly, the UCI Phishing 2016 dataset's smaller sample size and limited feature set might have contributed to the relatively lower performance. The proposed model's ability to effectively utilize the richer feature set and balanced class distribution in the MDP Phishing 2018 dataset highlights its robustness and adaptability in handling diverse datasets, leading to enhanced accuracy and reliability in phishing detection.

The proposed model shows improved performance over many traditional classifiers in Phishing attack detection. When compared with 22 recent studies using the UCI Phishing 2015 dataset, it ranks 8th with a 97.6% accuracy, just below models like RF [31], LightGBM [34], and CNN+RNN [38] (*Table 12*). It outperforms 15 other studies, highlighting its effectiveness. For Dataset 3, the proposed model achieves 98.3% accuracy, securing the second position among five studies, slightly behind the SEM [29]. Overall, the proposed model demonstrates notable accuracy and ranks competitively against state-of-the-art classifiers.

Thus, to demonstrate the efficiency of the proposed model, its performance is analyzed across various dimensions. Dataset variability is assessed by comparing results on both balanced and imbalanced datasets, considering different numbers of samples and features to highlight the model's robustness. Feature selection outcomes are evaluated using different methods, showing their impact on classification accuracy. A classifier comparison is conducted to contrast the ensemble model with individual classifiers like RF and XGB, demonstrating the ensemble's advantages. Scalability is assessed by varying test options, including percentage split and cross-validation methods. These analyses provide a thorough understanding of the model’s effectiveness and limitations in diverse scenarios.

The proposed approach stands out from previous models through its innovative integration of advanced feature selection techniques, incorporating URL, domain, and HTML-based features in a unique manner. It employs a hybrid classification algorithm using ensemble methods, which significantly enhances detection accuracy and performance

metrics. This model also distinguishes itself by leveraging a diverse range of datasets, including real-time data sources, to improve robustness and generalizability. Additionally, its scalable design ensures adaptability to emerging phishing threats, offering a more effective and future-proof solution compared to traditional models. These advancements collectively contribute to superior phishing detection and classification.

The proposed model showed high accuracy but underperformed in specific scenarios. It experienced failures due to limitations in feature representation, where some phishing techniques might not be fully captured by selected features. The dataset's characteristics, including class imbalance and the diversity of phishing tactics, may have introduced complexity that affected model performance. Additionally, the model's sensitivity to certain patterns could lead to misclassifications. Despite its strengths, the model struggled with FNs and positives, indicating a need for better feature engineering, data balancing, and model tuning to address these weaknesses and improve overall performance.

5.1Implications of the study

The findings have significant implications for both theoretical development and practical application in cybersecurity. The enhanced accuracy and reduced error rates observed in the proposed model contribute to theoretical advancements in phishing detection by validating the effectiveness of advanced feature selection and ensemble methods. Practically, the model provides a robust tool for cybersecurity professionals, enabling improved detection of Phishing attempts and a reduction in associated risks. Its ability to perform well across diverse datasets suggests its applicability in various operational environments, enhancing its utility for practitioners. Integrating such a model into existing cybersecurity frameworks can strengthen defenses against evolving Phishing threats, bridging theoretical insights with practical solutions for more secure online environments.

5.2Limitations of the study

While the proposed model demonstrates superior performance compared to both traditional and advanced classifiers, it does have certain limitations. The study utilizes a limited dataset with basic features and a small number of instances, which may not fully represent the diversity and complexity of real-world phishing attacks. The evaluation primarily focuses on classification accuracy without considering critical aspects such as time complexity, processor, and memory usage, which are essential for practical deployment. It is worth noting that while the C5.0 DT and PART classifiers are typically effective in ensemble classification for imbalanced datasets, the proposed method does not incorporate any specific data balancing techniques. This limitation could potentially impact the performance of the model on datasets with significant class imbalance, as these classifiers may not fully address the issue without explicit balancing measures. Additionally, the model could benefit from the integration of DL techniques, such as CNNs and RNN, to enhance feature selection and classification accuracy. Despite its high performance, the model still requires improvement in identifying phishing websites more accurately. Future research should address these limitations by employing larger, more diverse datasets and advanced techniques to enhance the model’s robustness and applicability in real-world scenarios.

A complete list of abbreviations is listed in *Appendix I.*

6.Conclusion and future work

An advanced phishing detection model was proposed, incorporating a novel PolyScore feature evaluator alongside an enhanced soft voting ensemble method. The primary goal of the study is to improve detection accuracy by combining multiple approaches, executed in three main phases: data preparation, feature selection, and classification. The first step is data preparation, ensuring the information is clean and ready for classification. In the second phase, the PolyScore feature evaluator selects significant features using six filtering approaches and applies normalized rank percentiles. During the classification stage, the predictions from seven classifiers are combined using geometric mean-weighting-based soft voting, and ensemble pruning removes classifiers with lower weights during testing. Extensive experiments conducted on three publicly available datasets show promising results, with higher accuracy and resilience compared to individual classifiers. The proposed model achieves high accuracy rates across three publicly available datasets: 97.36% for Dataset 1, 87.18% for Dataset 2, and 98.27% for Dataset 3. The model outperforms 15 state-of-the-art classifiers on Dataset 1 and 5 out of 6 models on Dataset 3, showing its robustness and effectiveness in detecting phishing websites. These results validate the model's potential to be applied in real-world phishing

detection tasks. The results confirm the robustness and accuracy of the proposed approach. By integrating feature evaluation and soft voting ensemble techniques, the model is able to offer higher detection accuracy while remaining resilient to various phishing techniques. The model shows superior performance across all datasets and proves effective in detecting phishing websites by evaluating URL, host, and HTML features.

While the model shows promising results, there are several areas for further improvement. First, beyond the features considered in this study, future research should investigate testing the model using larger datasets that include more advanced features. This will help assess the model's effectiveness across a broader range of phishing attempts and provide insights into its scalability. Second, although the model has demonstrated improved accuracy, future research should evaluate time complexity, processor usage, and memory consumption. These factors are crucial for understanding how the model will perform in practical, real-time systems. These considerations are especially important when implementing the model in large-scale systems or real-time applications. Third, when datasets contain inherent class imbalances, data balancing techniques should be employed to improve performance and ensure higher classification accuracy. Although the model performs well with balanced datasets, this step is necessary for datasets with imbalanced classes, which are often encountered in real-world scenarios. Fourth, future studies should assess the model's performance against a wider variety of phishing attack types, including novel and evolving phishing techniques, to maintain its relevance in an ever-changing environment. Phishing techniques are constantly evolving, and the model must be adaptable to these changes. Fifth, although the model performs well, DL techniques such as CNNs and RNNs could be explored to improve feature selection and classification. These methods have the potential to handle complex patterns more effectively and enhance detection accuracy. Finally, there is room for improvement in real-time phishing detection, particularly through integration into web browsers or security software, to block users from accessing phishing websites as they occur.

Acknowledgment
None.

Conflicts of interest
The authors have no conflicts of interest to declare.

Data availability
The data used in this study were obtained from publicly accessible repositories: (1) UCI Phishing 2015, available at https://archive.ics.uci.edu/dataset/327/phishing+websites; (2) UCI Phishing 2016, available at https://archive.ics.uci.edu/dataset/379/website+phishing; (3) MDP Phishing 2018, available at https://data.mendeley.com/datasets/h3cgnj8hft/1.

Author's contribution statement
Swetha C.V.: Conceptualization, literature review, design of the work, data collection, implementation, analysis and interpretation of results, and manuscript preparation. **Sibi Shaji:** Conceptualization, design of the work, supervision, review of the results and critical assessment of the manuscript. **B. Meenakshi Sundaram:** Conceptualization, design of the work, supervision, review of the results and critical assessment of the manuscript.

References
[1] Venkatesh SC, Shaji S, Sundaram BM. A fake profile detection model using multistage stacked ensemble classification. Proceedings of Engineering and Technology Innovation. 2024; 26:18-32.
[2] Infante A, Mardikaningsih R. The potential of social media as a means of online business promotion. Journal of Social Science Studies. 2022; 2(2):45-9.
[3] Alharbi A, Alotaibi A, Alghofaili L, Alsalamah M, Alwasil N, Elkhediri S. Security in social-media: awareness of phishing attacks techniques and countermeasures. In 2nd international conference on computing and information technology 2022 (pp. 10-6). IEEE.
[4] Sonowal G, Sharma A, Kharb L. Spear-phishing emails verification method based on verifiable secret sharing scheme. Journal of Information Assurance & Security. 2021; 16(3): 117-24.
[5] https://www.waterstons.com/insights/articles/what-social-media-phishing-and-how-can-it-affect-you-and-your-business. Accessed 30 November 2024.
[6] Varshney G, Kumawat R, Varadharajan V, Tupakula U, Gupta C. Anti-phishing: a comprehensive perspective. Expert Systems with Applications. 2024; 238:122199.
[7] Damaraju A. Mitigating phishing attacks: tools, techniques, and user education. Revista Espanola De Documentacion Cientifica. 2024; 18(02):356-85.
[8] Qabajeh I, Thabtah F, Chiclana F. A recent review of conventional vs. automated cybersecurity anti-phishing techniques. Computer Science Review. 2018; 29:44-55.
[9] Cui B, He S, Yao X, Shi P. Malicious URL detection with feature extraction based on machine learning. International Journal of High Performance Computing and Networking. 2018; 12(2):166-78.
[10] Adewole KS, Akintola AG, Salihu SA, Faruk N, Jimoh RG. Hybrid rule-based model for phishing URLs detection. In emerging technologies in computing: second international conference, iCETiC

2019, London, UK, 2019 (pp. 119-35). Springer International Publishing.
[11] Bountakas P, Xenakis C. Helphed: hybrid ensemble learning phishing email detection. Journal of Network and Computer Applications. 2023; 210:103545.
[12] Zamir A, Khan HU, Iqbal T, Yousaf N, Aslam F, Anjum A, et al. Phishing web site detection using diverse machine learning algorithms. The Electronic Library. 2020; 38(1):65-80.
[13] Vaitkevicius P, Marcinkevicius V. Comparison of classification algorithms for detection of phishing websites. Informatica. 2020; 31(1):143-60.
[14] Wejinya G, Bhatia S. Machine learning for malicious URL detection. In ICT systems and sustainability: proceedings of ICT4SD 2020 (pp. 463-72). Springer Singapore.
[15] Boukhalfa K, Guelmaoui MA, Saidani A, Ramdane Y. A proposal phishing attack detection system on twitter. International Journal of Information Security and Privacy. 2022; 16(1):1-27.
[16] Mughaid A, Alzu'bi S, Hnaif A, Taamneh S, Alnajjar A, Elsoud EA. An intelligent cyber security phishing detection system using deep learning techniques. Cluster Computing. 2022; 25(6):3819-28.
[17] Ali MS, Jain AK. Efficient feature selection approach for detection of phishing URL of Covid-19 era. In international conference on cyber security, privacy and networking 2021 (pp. 45-56). Cham: Springer International Publishing.
[18] Mandadi A, Boppana S, Ravella V, Kavitha R. Phishing website detection using machine learning. In 7th international conference for convergence in technology 2022 (pp. 1-4). IEEE.
[19] Chiew KL, Tan CL, Wong K, Yong KS, Tiong WK. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. Information Sciences. 2019; 484:153-66.
[20] Salihovic I, Serdarevic H, Kevric J. The role of feature selection in machine learning for detection of spam and phishing attacks. In advanced technologies, systems, and applications III: proceedings of the international symposium on innovative and interdisciplinary applications of advanced technologies 2019 (pp. 476-83). Springer International Publishing.
[21] Vishva ES, Aju D. Phisher fighter: website phishing detection system based on URL and term frequency-inverse document frequency values. Journal of Cyber Security and Mobility. 2022:83-104.
[22] Alam MN, Sarma D, Lima FF, Saha I, Hossain S. Phishing attacks detection using machine learning approach. In third international conference on smart systems and inventive technology 2020 (pp. 1173-9). IEEE.
[23] Sarasjati W, Rustad S, Santoso HA, Syukur A, Rafrastara FA. Comparative study of classification algorithms for website phishing detection on multiple datasets. In international seminar on application for technology of information and communication (iSemantic) 2022 (pp. 448-52). IEEE.
[24] Hutchinson S, Zhang Z, Liu Q. Detecting phishing websites with random forest. In machine learning and intelligent communications: third international conference, MLICOM 2018, Hangzhou, China 2018 (pp. 470-9). Springer International Publishing.
[25] Alnemari S, Alshammari M. Detecting phishing domains using machine learning. Applied Sciences. 2023; 13(8):1-16.
[26] Kumar J, Santhanavijayan A, Janet B, Rajendran B, Bindhumadhava BS. Phishing website classification and detection using machine learning. In international conference on computer communication and informatics 2020 (pp. 1-6). IEEE.
[27] Joshi K, Bhatt C, Shah K, Parmar D, Corchado JM, Bruno A, et al. Machine-learning techniques for predicting phishing attacks in blockchain networks: a comparative study. Algorithms. 2023; 16(8):1-12.
[28] Khan SA, Khan W, Hussain A. Phishing attacks and websites classification using machine learning and multiple datasets (a comparative analysis). In international conference on intelligent computing methodologies 2020 (pp. 301-13). Springer International Publishing.
[29] Mohammed BA, Al-mekhlafi ZG. Accuracy of phishing websites detection algorithms by using three ranking techniques. International Journal of Computer Science and Network Security. 2022; 22(2):272-82.
[30] Abdul SSR, Balasubaramanian S, Al-kaabi AS, Sharma B, Chowdhury S, Mehbodniya A, et al. Analysis of the performance impact of fine-tuned machine learning model for phishing URL detection. Electronics. 2023; 12(7):1-26.
[31] Choudhary T, Mhapankar S, Bhddha R, Kharuk A, Patil R. A machine learning approach for phishing attack detection. Journal of Artificial Intelligence and Technology. 2023; 3(3):108-13.
[32] Mao J, Bian J, Tian W, Zhu S, Wei T, Li A, et al. Phishing page detection via learning classifiers from page layout feature. EURASIP Journal on Wireless Communications and Networking. 2019; 2019:1-4.
[33] Babagoli M, Aghababa MP, Solouk V. Heuristic nonlinear regression strategy for detecting phishing websites. Soft Computing. 2019; 23(12):4315-27.
[34] Al-sarem M, Saeed F, Al-mekhlafi ZG, Mohammed BA, Al-hadhrami T, Alshammari MT, et al. An optimized stacking ensemble model for phishing websites detection. Electronics. 2021; 10(11):1-18.
[35] Taha A. Intelligent ensemble learning approach for phishing website detection based on weighted soft voting. Mathematics. 2021; 9(21):1-13.
[36] Karthikeya A, Sai YB, Hariharan S, Rao AC, Jignash D, Prasad AB. Prevention of cyber attacks using deep learning. In 9th international conference on advanced computing and communication systems 2023 (pp. 1332-6). IEEE.
[37] Ozcan A, Catal C, Donmez E, Senturk B. A hybrid DNN–LSTM model for detecting phishing URLs. Neural Computing and Applications. 2023; 35: 4957–73.

[38] Huang Y, Yang Q, Qin J, Wen W. Phishing URL detection via CNN and attention-based hierarchical RNN. In 8th international conference on trust, security and privacy in computing and communications/13th IEEE international conference on big data science and engineering 2019 (pp. 112-9). IEEE.
[39] Ashour MM, Marzouk ES, Abdelhalim E. Anti-phishing approach for IoT system in fog networks based on machine learning algorithms. Mansoura Engineering Journal. 2024; 49(3):1-22.
[40] Ferreira M. Malicious URL detection using machine learning algorithms. In proceedings of the digital privacy and security conference 2019 (pp. 114-22).
[41] Aljabri M, Mohammad RM. Click fraud detection for online advertising using machine learning. Egyptian Informatics Journal. 2023; 24(2):341-50.
[42] Bama SS, Ahmed MI, Saravanan A. A survey on performance evaluation measures for information retrieval system. International Research Journal of Engineering and Technology. 2015; 2(2):1015-20.
[43] Mohammad R, McCluskey L. Phishing websites. UCI Machine Learning Repository. 2015.
[44] https://archive.ics.uci.edu/dataset/379/website+phishing. Accessed 30 November 2024.
[45] Tan CL. Phishing dataset for machine learning: feature evaluation. Mendeley Data. 2018.

Swetha C.V. is a Research Scholar at the School of Computational Sciences and IT, Garden City University, Bangalore. She completed her B.E. (Computer Science and Engineering) and M.Tech. (Computer Science and Engineering) from Visvesvaraya Technological University, Belgaum. She has cleared the State Eligibility Test (KSET) in Computer Science and is currently pursuing a Ph.D. With over 10 years of teaching experience, her research interests include Cybersecurity, Online Social Networks, Data Mining, Machine Learning, and Deep Learning.
Email: cvswetha1987@gmail.com

Sibi Shaji has 23 years of experience in the educational and techno-managerial fields. She holds a Ph.D. specializing in Information Systems, along with postgraduate degrees in MCA and B.Ed. in Computer Science. Additionally, she specializes in Human Resource Management, holding an MBA and M.Phil. in HR. Proficient in both technical and managerial domains, she teaches MBA, MCA, and BCA students. She has presented numerous papers at international and national conferences in the fields of Computer Science and Management. Her areas of interest include Management Information Systems, Computer Architecture, and the Analysis and Design of Algorithms.
Email: sibishaji1@gmail.com

B. Meenakshi Sundaram has over 25 years of experience in academia, leading initiatives in curriculum development, academic programs, and industry partnerships in the field of Computer Science and Engineering. He holds a Ph.D. and multiple professional certifications, aligning educational programs with global standards. He has supervised research, published extensively, and contributed significantly to skill development and advancements in deep learning. His expertise fosters a Dynamic Academic Environment while nurturing future leaders in technology.
Email: bmsundram@gmail.com

Appendix I

S. No.	Abbreviation	Description
1	AB	AdaBoost
2	ANN	Artificial Neural Network
3	AUC	Area Under the Curve
4	CART	Classification and Regression Trees
5	CD	Critical Difference
6	CNN	Convolutional Neural Networks
7	COR	Correlation
8	CS	Chi-Square
9	DL	Deep Learning
10	DT	Decision Trees
11	ENT	Entropy
12	FN	False Negative
13	FP	False Positive
14	FTC	Federal Trade Commission
15	GB	Gradient Boost
16	GBM	Gradient Boosting Machine
17	GI	Gini Index
18	GNB	Gaussian NB
19	GR	Gain Ratio
20	GTB	Gradient Tree Boosting
21	HEFS	Hybrid Ensemble Feature Selection
22	HELPHED	Hybrid Ensemble Learning Phishing Email Detection
23	HS	Harmony Search
24	HTML	Hypertext Markup Language
25	IG	Information Gain
26	KNN	K-Nearest Neighbour
27	LR	Logistic Regression
28	LSTM	Long Short-Term Memory
29	ML	Machine Learning
30	MLP	Multilayer Perceptron
31	NB	Naïve Bayes
32	NN	Neural Network
33	OSN	Online Social Networks
34	PART	Partial Decision Tree
35	PCA	Principal Component Analysis
36	RF	Random Forest
37	RIPPER	Repeated Incremental Pruning to Produce Error Reduction
38	RNN	Recurrent Neural Network
39	SEM	Stacking Ensemble Model
40	SMP	Social Media Phishing
41	SVM	Support Vector Machine

42	TF-IDF	Term Frequency-Inverse Document Frequency
43	TN	True Negative
44	TP	True Positive
45	UCI	University of California, Irvine
46	URL	Uniform Resource Locator
47	XGB	Extreme Gradient Boosting