International Journal of Safety and Security Engineering
Vol. 14, No. 5, October, 2024, pp. 1635-1645
Journal homepage: http://iieta.org/journals/ijsse

Enhancing Multi-Class Password Strength Prediction Through Machine Learning and Ensemble Techniques

Enas F. Aziz[1], Mohammed Rashad Baker[2*]

[1] Department of Electronic and Communications Techniques, Kirkuk Technical Institute, Northern Technical University, Kirkuk 36013, Iraq
[2] Department of Software, College of Computer Science and Information Technology, University of Kirkuk, Kirkuk 36013, Iraq

Corresponding Author Email: mohammed.rashad@uokirkuk.edu.iq

https://doi.org/10.18280/ijsse.140530

Received: 9 August 2024
Revised: 3 October 2024
Accepted: 15 October 2024
Available online: 31 October 2024

Keywords:
password strength detection, multi-class classification, password security, ML, ensemble learning

ABSTRACT

Password strength prediction plays an important role in improving protection against cyber threats as their frequency increases. Typically, rules are used more specifically, but not all evaluate passwords effectively. This research aims to explore a more advanced approach to password strength prediction that solves some of the existing shortcomings through a machine learning (ML) and ensemble model for multi-class classification. Here, in this research, we have employed Random Forest (RF), Decision Tree (DT), Stochastic Gradient Descent (SGD), and Logistic Regression (LR) algorithms with Bagging and Stacking ensembling techniques. We used the Sber Dataset from Kaggle, which includes 100,000 passwords for the experiment. In the data preprocessing, the main procedures applied were missing value handling and shuffling. Text preprocessing included tokenization using common stop words and Term Frequency-Inverse Document Frequency (TF-IDF). The dataset was balanced using Synthetic Minority Over-sampling Technique (SMOTE) to address the class imbalance. The results for the Bagging and Stacking ensembles of combining multiple ML models showed that our approach outperformed the individual models in classifying password strength into three categories: weak, medium, and strong. Stacking outperformed the other algorithms in the sense that more than one model was used to improve results and minimize errors. Thus, the proposed approach provides a more accurate and versatile measure for password validation eradicating the problems encountered with the original method. The results proved the high efficiency of the used methods and showed more efficiency in prediction performance in comparison with the baseline models.

1. INTRODUCTION

Passwords serve to defend against intrusions, hacks and unauthorized access and keep personal information secure once the password itself is secure enough. Even though several secure approaches to authentication exist (e.g., biometrics [1] and smart cards, etc.), password authentication is the most common technique for guaranteeing the protection of any system. However, passwords are exposed to different types of attacks. The most common form of attack is password guessing due to the predictable patterns that people often choose for setting passwords probably their name, telephone number, place names, keyboard paradigm, birth date, common phrases, family members' name or friends, domestic animal, etc. [2-4]. This will pave the route for attackers either online or offline to penetrate the system by assuming or guessing the passwords. Therefore, implementing a strong password by the user is the most effective way to protect the system against these online and offline attacks. Password security means creating strong and effective passwords that are less likely to guess and resistant to general attacks [5]. The strength of passwords is completely reliant on the user. Passwords are assessed using password-strength meters (PSMs) which are spirited tools that help users to make further secure passwords [6, 7]. The assessment is based on many factors such as password complexity, length, or randomness [8]. PSMs assist with weak user-chosen passwords [9].

Traditional methods of password validation often depend on some requirements and rules for creating and managing passwords (e.g., minimum length, inclusion of special characters, etc.). However, these methods can be insufficient in determining the strength of a password.

Nowadays, two different ways are used to estimate password strength: rule-based and ML models [10]. Due to the unique properties of ML such as adaptability, scalability, and improved performance, ML methods have been applied in many areas of science.

Several benefits motivate the use of the ML approach over rule-based methods when implementing password strength assessment [11]. Unlike traditional methods, discovered ML algorithms can learn new password patterns and develop new solutions to crack passwords, as they draw knowledge from a large amount of data to define the complex correlations between password characteristics and security. Static rule-based systems can only take into account one or two factors at a time such as the composition of character, and length, and can only guess the meaning of the string [12]. In addition, these models can always be retrained on new data or be periodically trained on fresh data when a new trend of password generation or password cracking is discovered. Other forms of ML, including ensemble learning, enable the approaches to gather various angles regarding password strength, as well as enable them to give more secure and accurate evaluations than could be provided independently [13].

This study investigates the use of ML and ensemble learning that combine multiple ML models for better performance to solve password-strength detection problems. In addition, ensemble methods offer several advantages over individual models including improved accuracy and performance [14]. Also, they can decrease overfitting and underfitting by using different features of the data to create a more general and accurate prediction [15] and balance between variance and bias.

This study is structured as follows: In the following section, we introduce related work. Section 3 describes the proposed multi-class classification prediction model for password strength based on ML algorithms in detail. In section 4, we discuss the experimental investigation and validation of the proposed model. In section 5, the results of the proposed approach are presented. Section 6 is the conclusion.

2. RELATED WORKS

In this section, we provide a short review about predicting password strength, and studies about password security. Different password strength meters are existed and used publicly. Many of them integrate mathematical-based approaches in their algorithm, such as the password's length, digits, lowercase and uppercase characters, number of special symbols and dictionary matching. Several studies and articles on assessing the strength of passwords have been published previously. Song et al. [14] used the multi-model ensemble learning model to evaluate the password strength of different complexity passwords. A real set of user passwords that leaked on the network was used as the experimental dataset. The dataset was used to train multiple existing password evaluation models as the sub-models. Then, multiple trained evaluation sub-models were used as the base learners for ensemble learning, and the ensemble learning strategy which was designed to be partial to weakness, was used to get all the advantages of sub-models. The experimental results show that the multi-model has a high accuracy and is universal. Also, it has good applicability in the evaluation of passwords. He et al. [15] presented Hybritus which utilises different website approaches into a comprehensive model of the attackers, with MLP neural networks. Their dataset includes more than 3.3 million passwords directed across 10 website checkers, to obtain feedback on the strength of the passwords ordered as weak, medium and strong. Features of passwords were then utilized to train and test Hybritus. The experimental data proved that password strength accuracy checking can be as high as 97.7% and over 94% even when only trained with just ten thousand passwords.

Many articles have been presented on the use of ML algorithms in password strength estimation in the last few years. For example, Darbutaitė et al. [16] proposed a machine-learning approach that supports a more realistic model, to estimate Lithuanian user password strength. Thus, a new dataset of complied password strength was produced. The proposed solution estimates the strength of five classes of passwords with a 77% accuracy.

Farooq [17] proposed a model that provides an efficient and common way to defend against attacks including online and offline by forcing the users to choose a strong password through the implementation of multiple ML algorithms like DT, Naïve Bayes (NB), RF, LR and Neural Network (NN) on a web application over real-time. After testing the models, the best results were recorded by DT with an accuracy of 99% and the lowest by NB at 87%.

Vijaya et al. [18] modeled a classification task by using password strength prediction and employed ML methods namely C 4.5 decision tree classifier, NB classifier, MLP, and support vector machine for learning the model. The results indicated that SVM performs well. The findings showed that the ML approach has a significant capability to categorize the cases: weak and strong passwords.

Sarkar and Nandan [10] proposed a prediction model of password strength classification with the aid of several supervised ML algorithms to classify passwords into multiple categories: Weak, Medium, and Strong passwords. XGB and MLP algorithms were implemented, to prove the strength and the corresponding category of a password. The findings display that XGB outperformed the other ML classifiers with an accuracy of 99%.

Divya et al. [19] presented a model by employing multiple ML methods, including DT, LR, RF, and K Nearest Neighbor, pushing users to select a strong password as protection against online and offline attacks. The Random Forest Classifier had the best results during the testing of the models over the test set, with an accuracy of 98%, and Logistic Regression achieved the lowest accuracy of 82%.

Rathi et al. [20] suggested a comparative analysis of soft computing techniques like BPN, HNN, BSB, CNN, LR and Bidirectional Associative Memory (BAM) for the classification of a strong password. The dataset only contains a singular attribute of passwords categorised into 3 classes weak, medium, and strong. For such a dataset, first, the experimental analysis of the results demonstrates that the simple logistic regression model produces better output when compared with CNN, BPN, HNN, LR, BSB and BAM. Second, LR has adapted for such an application, as well as playing a good prediction system.

Kuriakose et al. [21] suggested a prototype that implements numerous ML techniques such as DT, LR, NB and RF on a web application in real-time, by doing so it forces the users to choose a secure password and to perform analysis of efficiency for password strength analysis to produce the best results for the user.

Jamuna et al. [22] employed ML techniques to analyse password strength as a way to enable organisations launch of a multi-faceted defense against password breaches, whilst providing a highly secure environment. Support Vector Machine is used as a supervised learning algorithm for the categorisation of passwords. The features taken from the password dataset were used to train the linear and nonlinear SVM classification models. The trained models demonstrated a prediction accuracy of around 98% for 10-fold cross-validation.

Kim and Lee [23] suggested a multi-class classification prediction model for the strength of a password that is based on deep learning, which takes into account leaked frequency that evaluates password strength, whilst solving the problem of degraded evaluation reliability of existing indexes in the case of a password leak. The proposed model was able to correctly assess 99% of the 345 leaked passwords.

Mienye and Sun [24] presented an overview of the three main ensemble learning techniques: bagging, boosting and stacking. As well as their early progression to the latest algorithms. They focused on the commonly used ensemble algorithms like AdaBoost, RF, gradient boosting, XGBoost, LightGBM and CatBoost. They tried to briefly cover their algorithmic and mathematical representations.

Suganya et al. [25] proposed a framework to analyse password strength proactively. Support vector machines and filters are employed, as this framework can be used as a submodule of the access control scheme.

This study proposes a Multi-class Classification Prediction Model for password strength based on ML algorithms, including RF, DT, SGD, and LR. The model utilizes Bagging and Stacking ensemble techniques to enhance the classification accuracy of passwords based on their strength.

The novelty of this study is that the above algorithms in particular have not been implemented together in similar other works.

3. METHODOLOGY

This section outlines the comprehensive methodology utilized for classifying password strength using various ML techniques, including ensemble models. As depicted in Figure 1, the proposed workflow encompasses multiple stages: loading the dataset, preprocessing, feature extraction, dataset splitting, dataset balancing, model training, and evaluation. Each stage is crucial to ensuring the robustness and reliability of the final model. The subsequent subsections provide a detailed explanation of each step in the workflow.

3.1 Loading dataset

The dataset used in this work is associated with Sber and is available on Kaggle. Initially used in the “Beauty Contest of the Code from Sber,” this dataset has to be divided into three groups based on password complexity. The dataset is organized in terms of two columns: The first one includes plaintext passwords as strings while the second one portrays the difficulty class which are 0, 1 and 2. At this time, 0 has the lowest password scale, which means the password is not very strong, while 2 represents a high password scale meaning the password is very strong. The total size of the dataset used in the model is 100,000 records.

3.2 Preprocessing

Data preprocessing is crucial in the process of data preparation where certain complexities in the data are eliminated to facilitate a proper analysis. This process includes two main stages:

Handling missing values: This stage in the dataset is handled by imputation or deletion depending on the pattern and proportion of the missing data. This stage is important to demarcate the variables and factors to be used in the study and eliminate any possibility of bias in the research.

Data shuffling: This stage is important to shuffle the data to break any order that might exist in the dataset to ensure that the model developed is capable of generalizing the outcomes as it will not be influenced by specific sequences that exist in data.

3.3 Feature extraction

Feature extraction is performed on the raw data to transform it into a usable format for the ML models employed in this research. In this study, TF-IDF methods are employed [26]. The Tokenising process entails transforming the passwords into tokens which may include n-grams and/or characters. Thus, it contributes to text segmentation, that is, the division of the text into segments that can be analysed further [27]. TF-IDF on the other hand helps in transforming the textual data into numerical features. The level of importance of the token in the given document in comparison with the size of the total set is provided by the TF-IDF. This method brings emphasis to informative tokens while diminishing less informative tokens [28].

Table 1. Class distribution summary

Class	Training Set		Test Set
	Before SMOTE	After SMOTE	
0	10,744	59,466	2,684
1	59,466	59,466	14,812
2	9,790	59,466	2,504

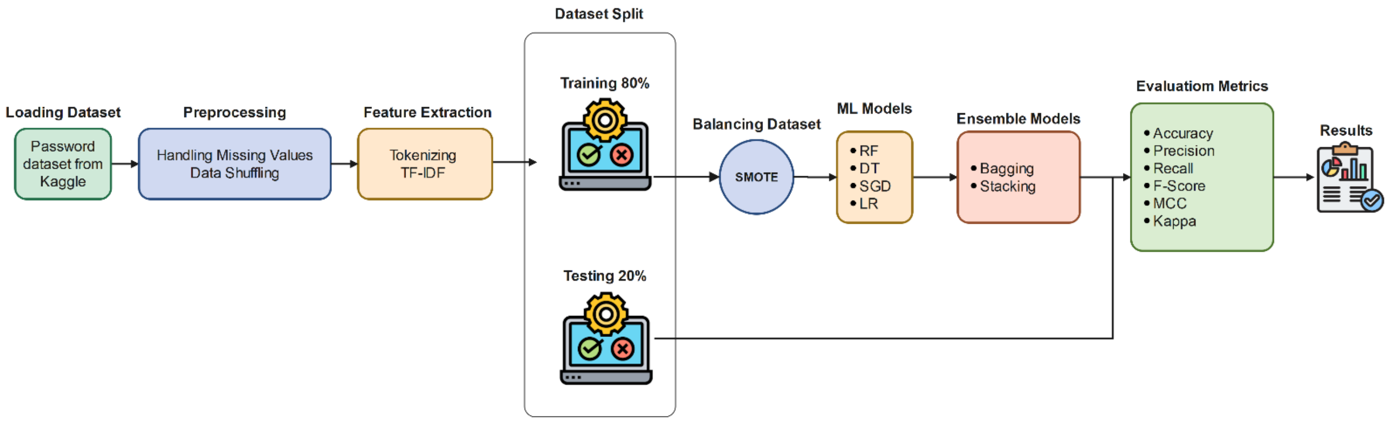


Figure 1. The proposed workflow

3.4 Balancing dataset

The dataset was split next into two sets with the ratio 80: 20 to test the efficiency of the ML models. The training set constitutes 80000 records while there are 20000 records in the testing set. The distribution of the training set in Table 1 is presented, which is unbalanced, which greatly affects the work of the ML models. To overcome this, a technique called SMOTE is used [29]. The concept of SMOTE is important to create synthetic samples from the minority class to balance the classes for improved learning from under-represented classes. The size of the training set by class distribution is presented in creating, it is seen that the distribution of records is highly imbalanced, with Class-1 having 59,466 records while Class-0 and Class-2 have 10,744 and 9,790, respectively. Distributions such as these can result in skewed model performance where the model could be best suited to the majority class while forgetting the minority classes. Other methods like SMOTE also come in handy in addressing this problem as they act to balance the dataset. The result demonstrated that after the application of SMOTE, all classes of the training set are balanced and the number of records per class is 59,466. On this balanced distribution, the generalization capability of the model is favored across all the classes as it ensures that equal opportunities for each class are given to the model during its training. However, the test set retains the original distribution of the class to ensure that the actual evaluation metrics infringe on an imbalanced real-world dataset. This method provides a more accurate evaluation of the model's ability to deal with class imbalance than the previous approach.

Looking at the raw distribution of the training set, it can be observed that the data is not balanced and that the majority class contains 59,466 records while classes 0 and 2 have 10,744 and 9,790 records, respectively. This can cause problems in model performance that favors the majority class while it disregards the performance of the minority classes .

To address this issue, SMOTE is used to balance the data, as shown below. After applying SMOTE, the distribution of the records of each class in the training set is balanced, with 59,466 records per class. It also ensures that each class is evenly represented during the training process, which in turn aids the model in possibly giving it a chance to generalize to any class or category .

The test set preserves the imbalances of the sources' distribution to avoid distorting the metrics by including balanced data. This approach offers a better solution for evaluating the effectiveness of the given model in cases of class imbalance.

3.5 ML models

The ML models have a significant position in the categorization of password strength because of their learning capability how to infer complex patterns from data. Therefore, for this study, several models were selected to classify the passwords' strengths appropriately depending on the model's strength.

For instance, RF was chosen for its high-dimensional response capability and capacity to capture non-linear data association and it was found to be less sensitive to overfitting than other models [30]. DT was selected as the model for the interpretation of the results and its capability to model hierarchical decision rules which are helpful for password structure identification [31]. SGD was included for its fast computation regarding the large dataset and for its flexibility in detecting new structures in the data [32]. LR was chosen because of its probability estimation and better performance on linearly separable data to act as a linear model compared to the non-linear models [33].

These algorithms were further applied with the help of Bagging and Stacking ensemble learning approaches to get benefits of all the particular methods and to avoid their drawbacks. The purpose of bagging is to reduce variance and overfitting while Stacking's outcome allows a meta-learner to decide on the right number of base classifiers to combine or integrate. The use of this diverse set of algorithms as a way of ensembling helps offer a comprehensive way of capturing different aspects of password strength when dealing with different types of passwords.

3.5.1 RF

RF is one of the models of ensemble learning that is specifically designed for use in classification problems. The major strength of RF comes from reducing overfitting, which is a common issue with decision trees, by averaging the scores of many such trees [34]. It also improves the accuracy of the model, and its ability to counter noise or other problematic factors in the data, making it the better method for classification of the password strength.

3.5.2 DT

DT is one of the supervised learning algorithms which is used for classifying and also for regression purposes. It operates to classification by breaking the dataset into subsets according to the values of the input variables to generate a tree of decisions [35]. Nodes in the tree represent decision rules, while each leaf node represents an easily understandable and simple outcome for the application of classification rules. However, it is sensitive to overfitting, which can be controlled by techniques like pruning or by including several decision trees in the model, such as in the case of RF.

3.5.3 SGD

SGD is an efficient iterative algorithm widely used in large-scale machine learning tasks. It optimizes a variety of objectives by incrementally minimizing the loss function using the gradient descent method [36]. SGD is particularly effective for classification problems and supports various loss functions, including hinge loss for support vector machines and log loss for logistic regression.

3.5.4 LR

LR is a statistical technique implemented for binary classification issues, and it can be extended to multiple-class classification via methods like one-vs-rest. It models the odds of a certain class or event existing, such as password strength, based on one or more independent variables. The model is suitable for predicting the odds, as it utilizes the logistic function to squeeze the output of a linear equation between 0 and 1 [37]. LR is effective and simple, offering insights into the significance of different features in the classification task.

3.6 Ensemble models

Ensemble models improve the prediction quality as they integrate features of numerous base models. This process is done to get a better overall generalization and accuracy as compared to the individual models.

3.6.1 Bagging

Bootstrap aggregating, or bagging, is an ensemble learning technique that trains models independently on different subsamples of the dataset obtained by using the training with production with replacement. Every predictor or model, which is generally a decision tree, provides its prediction on its own, and the final prediction is arrived at by either aggregating the outputs in the case of regression or by taking a mean vote in the case of classification. Bagging makes predictions less variant and enables us to avoid over-fitting resulting in better accuracy [38]. Bagging is applied in the context of password strength classification to ensure that the model thus derived is stable and performs just as well or even better than in other parts of the datasets.

3.6.2 Stacking

Stacking is an ensemble technique that improves model accuracy by training a meta-learner to combine predictions from multiple base learners. Each base learner is trained independently on the same dataset, and their predictions serve as input features for the meta-learner. This approach allows the meta-learner to leverage the strengths and mitigate the weaknesses of each base model, enhancing overall performance. In contexts like password strength classification, stacking helps achieve higher accuracy by integrating diverse insights from various models [39].

3.7 Evaluation metrics

Evaluation metrics are essential for assessing the performance of classification models by measuring the accuracy and reliability of their predictions. Accuracy specifically calculates the proportion of correctly classified instances among the total tested, providing a basic measure of model performance. However, in cases of unbalanced classes, relying solely on accuracy might be misleading, which necessitates additional metrics like Precision and Recall. Precision is crucial in fields where false positives are costly, such as in medical diagnostics or fraud detection, and it measures the ratio of true positives to all predicted positives.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

On the other hand, Recall or sensitivity determines the proportion of actual positives that were correctly identified by the model with a high value on false negativities which might prove fatal in applications like cancer detection or surveillance measures. It is given by:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

F-Score harmoniously blends Precision and Recall into a single metric, balancing their respective influences on model performance. It is calculated as:

$$F - Score = 2 * \frac{P * R}{P + R} \quad (3)$$

Precision and Recall are two metrics that can be combined using the F-Score, which is the harmonic mean of the two. The F-Score gives a mid-level evaluation of the model's performance, which is vital when comparing the performance of one's models.

The metrics of High Precision and High Recall are as relevant as the other. The F-Score variation reaches from 0 to 1 with an improved score indicating better efficiency.

MCC is also known as "Matthews Correlation Coefficient", it expresses the capacity of the binary classification, and accounts for all four quadrants of the confusion matrix: true positive (TP), true negative (TN), false positive (FP,), true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). It ranges from -1 to +1, where +1 represents a perfect prediction telling us that the influence reaches its upper bound. While 0 represents the worst possible prediction or random guessing, -1 represents complete disagreement between prediction and observation. It is calculated as:

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (4)$$

Lastly, Kappa statistic assesses inter-rater agreement for categorical items, evaluating the agreement beyond chance. It is calculated as:

$$Kappa = \frac{P_o - P_e}{1 - P_e} \quad (5)$$

where, P_o is the observed agreement, and P_e is the expected agreement. This metric is valuable in contexts involving subjective annotations or multiple raters.

These metrics collectively form a rigorous framework for evaluating classification models, guiding researchers and practitioners in selecting appropriate models and optimizing their performance in diverse application domains.

4. RESULTS AND DISCUSSION

This section evaluates the performance of our suggested ML models: RF, DT, SGD, and LR, including Bagging, and Stacking ensembles. There are several measures to make a sophisticated comparison of models' performance. Some of the measures used are the Receiver Operating Characteristic (ROC) curves, confusion matrices and other general performance measures shown in Table 2 below. The table showcases key metrics for each model: In this case, the performances were evaluated by using six measures: accuracy, precision, recall, F-score, MCC, and Cohen's Kappa to give a comprehensive evaluation of their classification performances.

According to Table 2, the Stacking model achieves the highest score of 0.93, followed closely by RF at 0.91. These models' high accuracy indicates their strong overall performance in correctly classifying instances. However, it's important to note that accuracy alone can be misleading, especially in cases of imbalanced datasets, which is why we consider other metrics as well.

Precision, which measures the proportion of true positive predictions among all positive predictions, is highest for the Stacking model at 0.90 and lowest for SGD at 0.60.

Recall, also known as sensitivity, measures the proportion of actual positives that were correctly identified. Interestingly, LR shows the highest recall by 0.79, slightly outperforming even the Stacking model by 0.88 in this aspect. This indicates that LR is particularly good at identifying true positives, although its lower precision suggests it may achieve this at the cost of more false positives.

The F-Score, which is the harmonic mean of precision and recall, provides a balanced measure of a model's performance. The Stacking model leads with an F-Score of 0.89, followed by RF with 0.86 and Bagging with 0.83. This metric reinforces the overall superiority of the ensemble methods.

The Matthews Correlation Coefficient (MCC) and Cohen's Kappa are both measures that take into account true and false positives and negatives and are particularly useful for imbalanced datasets. Both metrics show very similar patterns across the models, with Stacking leading (0.83 for both), followed by RF and Bagging. The consistency between these metrics and the others lends credibility to the overall performance ranking of the models.

It is important that across all metrics, the Stacking model consistently outperforms the others, often by a significant margin. This suggests that the combination of multiple models in the Stacking approach is particularly effective for this classification task. The tree-based models (RF and DT) and the Bagging ensemble method show strong performance across all metrics, consistently ranking in the top half of the models. This indicates that these approaches are well-suited to the underlying structure of the data.

To quantitatively compare the performance differences between algorithms, we analyze the relative performance decreases using the Stacking model as a baseline. The RF model shows only a modest decrease in performance, with 2.15% lower accuracy, 3.33% lower precision, and 3.37% lower F-Score compared to Stacking. The Bagging model similarly demonstrates strong performance, with decreases of 4.30% in accuracy and 7.78% in precision. However, there is a more substantial performance drop when moving from ensemble methods to individual classifiers. The DT model shows an 8.60% decrease in accuracy and a 15.56% decrease in precision compared to Stacking, while LR and SGD show much larger decreases of 21.51% and 27.96% in accuracy, respectively.

When comparing ensemble methods (Stacking, Bagging) to individual methods (RF, DT, LR and SGD), we observe a clear performance advantage for ensemble approaches. Ensemble methods achieve an average accuracy of 0.91, while individual methods average only 0.79. This pattern is consistent across other metrics, with ensemble methods indicating more robust and reliable performance.

Table 3 shows the correlation matrix of performance metrics used in this study. A correlation analysis between different metrics reveals strong relationships that validate the robustness of our evaluation. There is a very strong positive correlation between accuracy and MCC 0.96, as well as between accuracy and Kappa 0.96. Similarly, precision and F-Score show a correlation of 0.97, while recall and F-Score correlate at 0.95. These strong correlations suggest that the metrics are largely in agreement about model performance rankings, providing additional confidence in our evaluation methodology.

When considering the relationship between model complexity and performance, we observe diminishing returns as complexity increases. Moving from simple linear models (LR and SGD) with an average accuracy of 0.70 to a single decision tree increases accuracy by 21.43%, while the further step to ensemble methods only yields an additional 7.06% improvement. This relationship suggests that while ensemble methods do provide the best performance, the additional computational complexity may not always justify the incremental improvement, particularly in resource-constrained environments.

Based on this quantitative analysis, we can make several evidence-based recommendations. For optimal performance across all metrics, the Stacking model is superior. However, if computational resources are limited, the RF model offers an excellent performance-complexity trade-off, with only a 2.14% reduction in accuracy compared to Stacking. The DT model could be considered when minimal complexity is required, as it provides decent performance despite an 8.67% accuracy reduction. Linear models should be avoided for this specific problem, as they underperform significantly, with accuracy reductions of over 20% compared to the best performer.

The effectiveness of the models is evaluated using Receiver Operating Characteristic (ROC) curves and confusion matrices, which provide visual and quantitative insights into their performance. The ROC curves clearly show this performance pattern.

Table 2. Comparison of evaluation metrics for the proposed models

Model	Accuracy	Precision	Recall	F-Score	MCC	Kappa
RF	0.91	0.87	0.87	0.86	0.79	0.79
DT	0.85	0.76	0.80	0.77	0.66	0.66
SGD	0.67	0.60	0.77	0.63	0.48	0.43
LR	0.73	0.65	0.79	0.68	0.54	0.50
Bagging	0.89	0.83	0.85	0.83	0.75	0.75
Stacking	0.93	0.90	0.88	0.89	0.83	0.83

Table 3. Correlation matrix of performance metrics

	Accuracy	Precision	Recall	F-Score	MCC	Kappa
Accuracy	1.00	0.99	0.91	0.96	0.96	0.96
Precision	0.99	1.00	0.89	0.97	0.96	0.95
Recall	0.91	0.89	1.00	0.95	0.95	0.93
F-Score	0.96	0.97	0.95	1.00	0.99	0.97
MCC	0.96	0.96	0.95	0.99	1.00	0.97
Kappa	0.96	0.95	0.93	0.97	0.97	1.00

For instance, Figure 2 displays the ROC curve for the RF model. The curve demonstrates a strong performance, with a substantial area under the curve (AUC). The RF model's curve closely follows the top-left corner of the plot, indicating a high true positive rate and a low false positive rate across various classification thresholds. This suggests that the RF model has a robust ability to distinguish between classes.

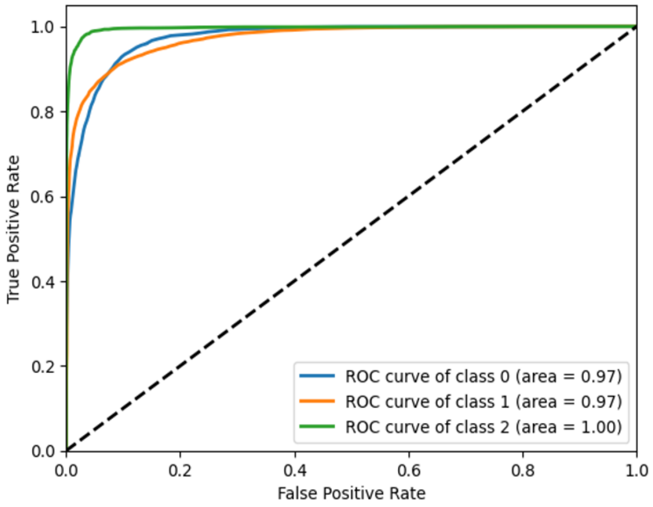


Figure 2. ROC curve for RF model

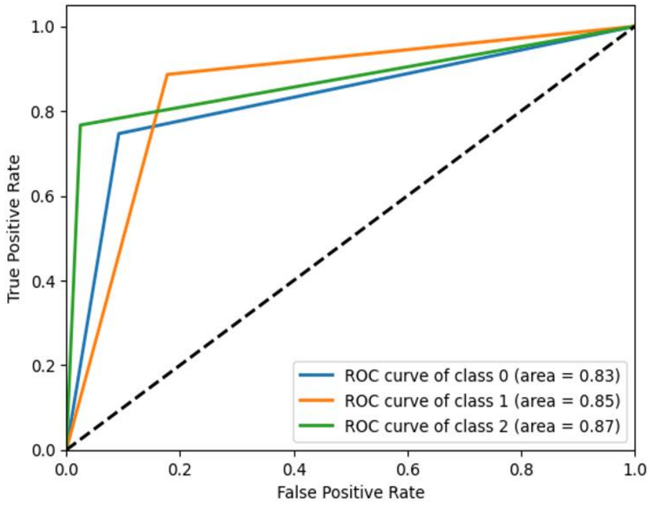


Figure 3. ROC curve for DT model

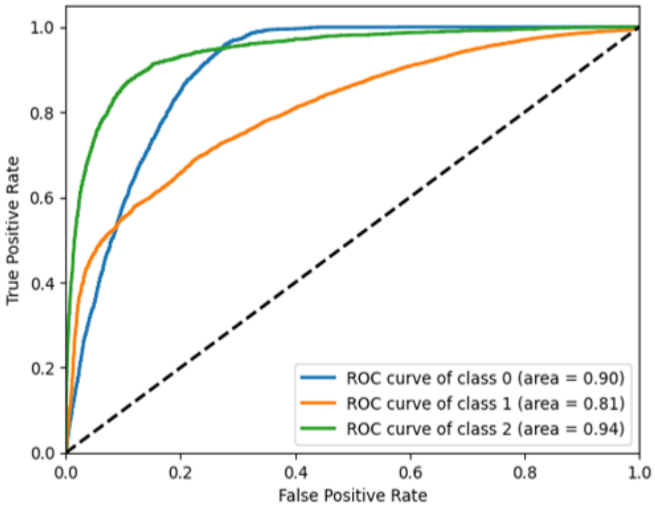


Figure 4. ROC curve for SGD model

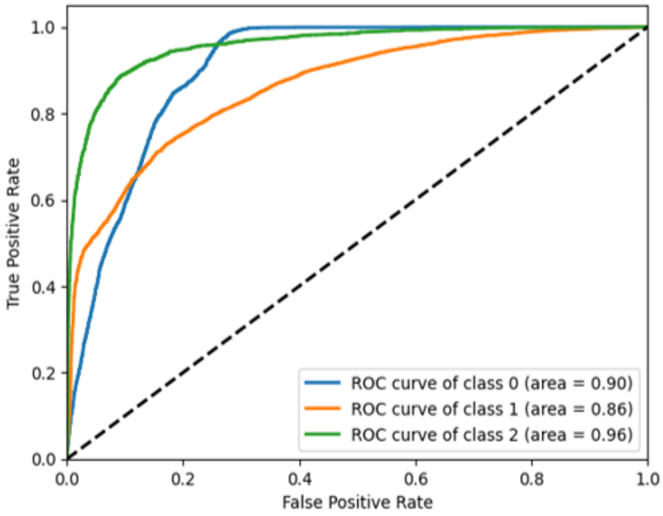


Figure 5. ROC curve for LR model

Figure 3 shows the ROC curve for the DT model. While the DT model exhibits good performance, its curve is slightly less optimal compared to the RF model. The AUC is smaller, suggesting that the DT model may have a slightly lower discriminative power than RF. This difference could be attributed to the ensemble nature of RF, which often leads to improved performance over single decision trees.

Figure 4 presents the ROC curve for the SGD model. Among all the models, the SGD curve appears closest to the diagonal line, indicating the weakest performance. The AUC for this model is the smallest, suggesting that SGD struggles to effectively separate the classes in this particular problem.

Figure 5 illustrates the ROC curve for the LR model. The LR model shows moderate performance, with its curve positioned between those of SGD and DT. While it outperforms SGD, it doesn't reach the levels of discriminative ability demonstrated by the tree-based models.

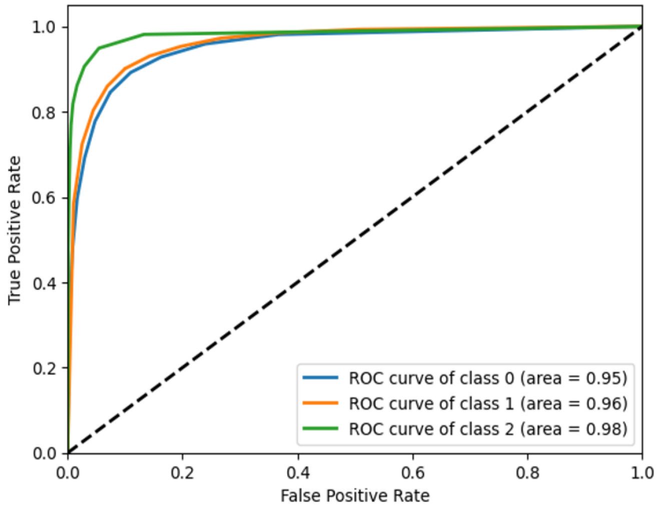


Figure 6. ROC curve for Bagging model

Figure 6 displays the ROC curve for the Bagging model. This ensemble method shows strong performance, with a curve that closely resembles that of the RF model. The similarity in performance between Bagging and RF is not surprising, as both are ensemble methods based on decision trees.

Figure 7 presents the ROC curve for the Stacking model.

Notably, this model exhibits the best performance among all six models. Its ROC curve hugs the top-left corner most closely, suggesting superior discriminative ability. The large AUC indicates that the Stacking model consistently achieves high true positive rates while maintaining low false positive rates across various classification thresholds.

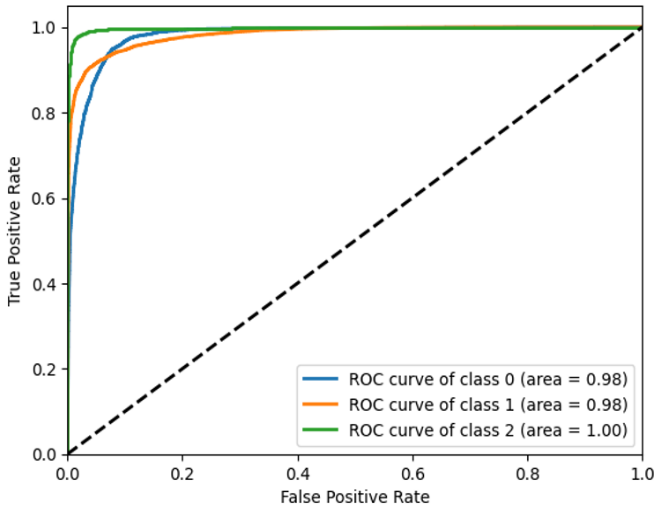
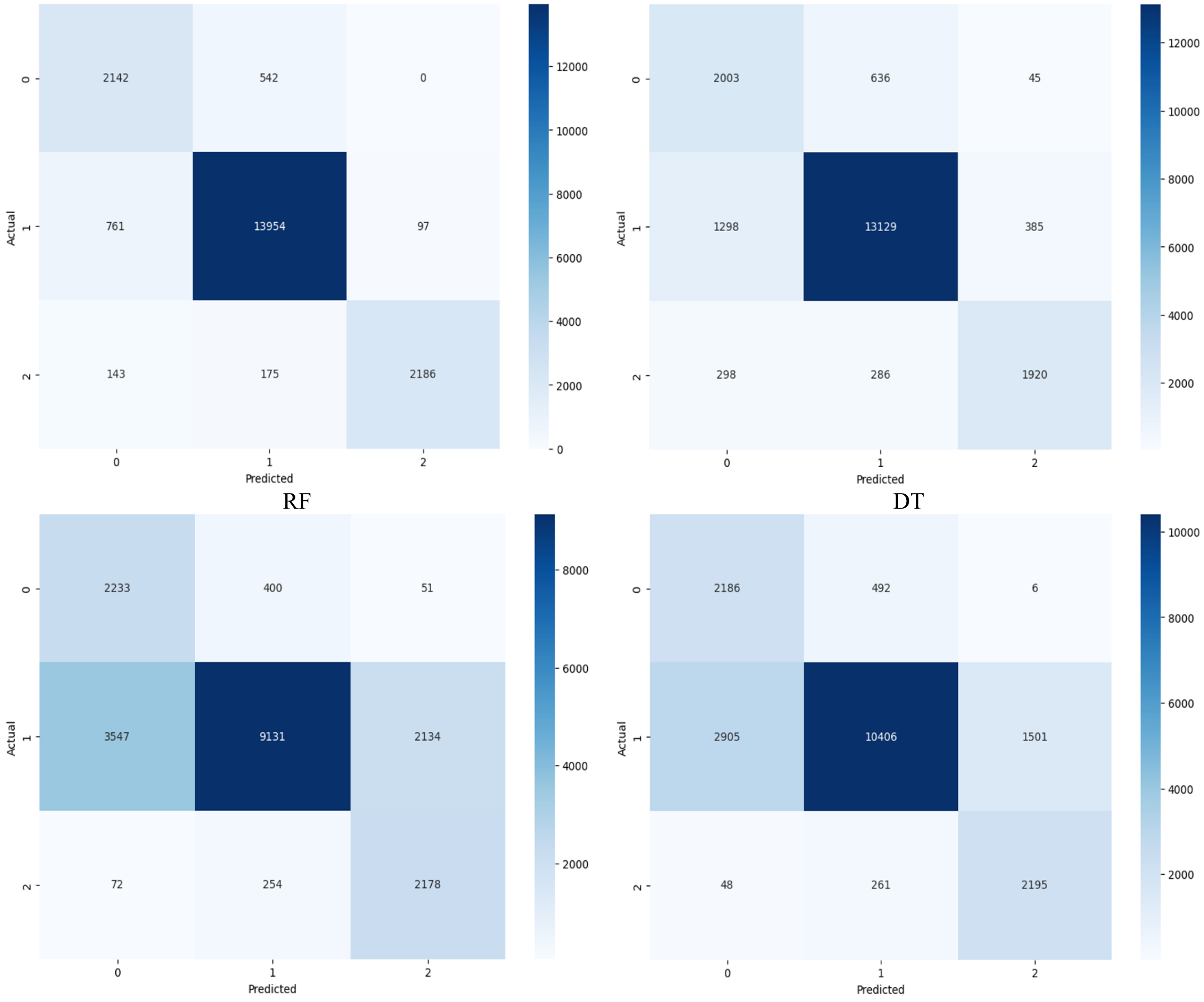


Figure 7. ROC curve for Stacking model

Figure 8 shows confusion matrices for six models, providing a basis for comparing their classification accuracy. These matrices confirm the ROC curve analysis. The Stacking model achieves the highest accuracy, as indicated by its number of true positives and true negatives, and has an excellent ROC curve. Both RF and Bagging models also perform well, displaying high accuracy. DT model lags slightly behind the ensemble methods. LR model falls below the 0.8 threshold, showing lower performance, while SGD model records the most misclassifications, which is consistent with its ROC curve results.

The AUC plot and confusion matrices demonstrate that ensemble methods outperform other models like Stacking, RF, and Bagging in this classification problem. Tree-based models, such as RF and DT, perform well with high-performance rates meeting the system’s threshold of 0.89. In contrast, linear models such as SGD and LR have slightly lower scores. These results suggest that the data structure is likely complex and non-linear, which benefits from the robust features of ensemble methods.

The conclusions made in the current study have important implications for the general area of cybersecurity and password protection. The better accuracy of the given ensemble model suggests that there are better and more complex ways to try and enhance password protection.

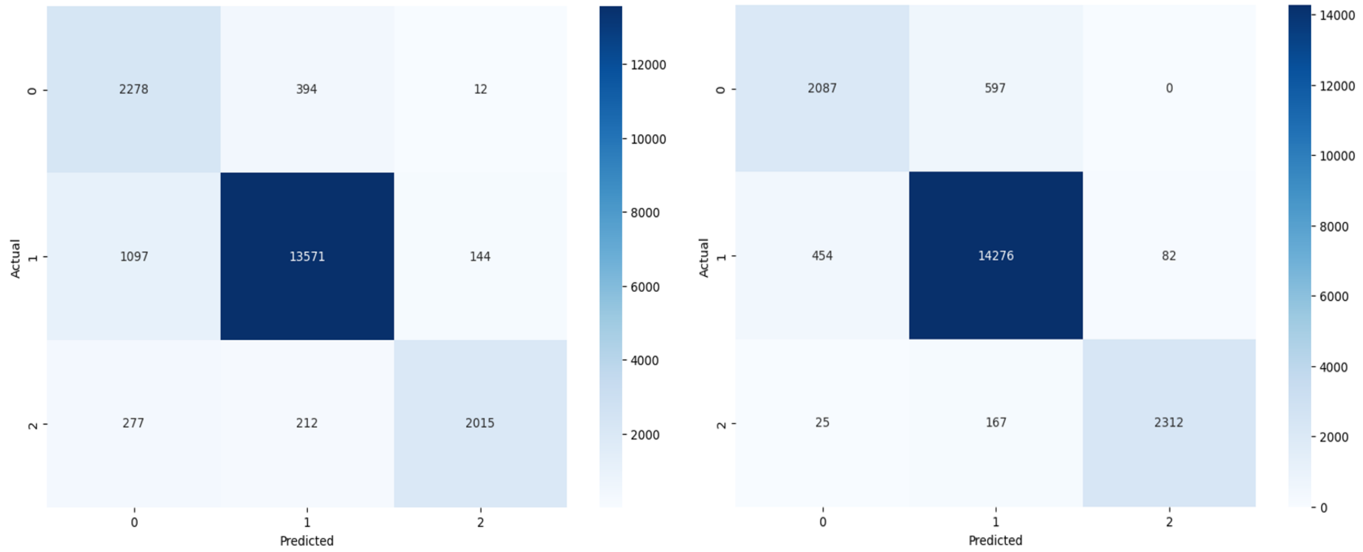


RF

DT

SGD

LR



Bagging

Stacking

Figure 8. Confusion matrix for proposed models

Our model could be implemented as a module in password management software that would improve the overall rating of password strength in the context of cybersecurity. This could result in enhanced guidance for users when creating complex passwords to enhance the security of a person or company's property. It is especially beneficial that our model can provide 'medium' strength passwords with a better classification, as this is an issue that most traditional password strength meters have faced.

From the above analysis, there are several advantages of our approach to password management. First, it can be utilized to scan current password files for weaknesses that could go unnoticed and/or uncovered by a mechanized rule-based approach. This could be particularly useful for massive organisations, and therefore, may require reinforcement of their general security systems. Second, the flexible nature of the machine learning method allows the model to be refined periodically as the trend of password creation and new threats are identified.

Furthermore, the results presented in this study stress the need to go beyond the simplistic rules of password length and its complexity. Thus, our investigation shows that there is a practically feasible and considerably more efficient approach to the assessment of password strength, and this implies that the policies currently in place could be adjusted to help people develop genuinely strong passwords rather than those which conform to a number of fixed rules.

In the context of cybersecurity education and training, our model can be employed as an educational model. If the passwords are categorized in terms of the degree of weakness, medium, and strong, it might be useful to explain to the users, why some of them fall into such categories, then, it may lead to improvement of password creation among users, numerous platforms and services.

Finally, the results provide the members of the cybersecurity community with the data that can be used in the further discussion of the degree of security, which is necessary to achieve at the cost of usability. Our approach might lead to having more flexible password policies that offer high security even if they are not as rigid as those provided in current systems. Therefore, our results not only contribute to the particular subproblem of predicting passwords' strength but also have broader implications for the state and evolution of cybersecurity, ways of protecting accounts, and users' awareness of threats in a world, where digital presence continues to grow.

5. CONCLUSIONS

This research work presents the use of new models such as ML and ensemble algorithms for password strength prediction. Besides RF, DT, SGD and LR algorithms, bagging and stacking ensemble methods have also been used; thus, the final developed model was further refined to effectively classify the strength of a password. The outcomes demonstrated that ensemble methods, specifically for password strength prediction, boosted the performance of individual algorithms. Using SMOTE for balancing the TF-IDF and dataset for feature extraction further led to the model's effectiveness. The limitations of traditional rule-based password validation techniques were addressed using this approach, as well as providing a more comprehensive evaluation of password strength.

The model that we are proposing has a number of practical implications for cybersecurity, as it can be implemented into password creation systems, thus encouraging users to create stronger passwords. It can also be incorporated into organizations to audit existing password databases and identify vulnerabilities. Additionally, there are several areas for future research that for more improvements.

In the future, the development of the model could lead in the direction of adapting to the evolution of password creation and new types of cyber threats. The accuracy of strength prediction could be improved by enhancing the model's ability to consider user-specific or organization-specific techniques. The user's understanding and trust could also be increased by developing techniques that provide a clear explanation for the model's prediction. By incorporating multiple languages into the model, it can evaluate passwords in different languages which would expand and broaden its applicability. Combining the model's ability to predict password strength with other security measures, such as multi-factor authentication, could provide a more comprehensive security solution. Future research can focus on ways to optimize the model's performance for real-time uses, ensuring fast response times even with larger password databases. The model's effectiveness could further be improved and validated, by conducting rigorous testing of the model against different password-cracking methods.

This study aims to provide a significant step forward in the area of password strength prediction. We have come up with a more accurate and robust technique for evaluating the strength of a password, via incorporating ML and ensemble methods. With cyber threats continuing to further evolve with time, advanced models such as ours will have a crucial role in the enhancement of overall cybersecurity posture. Further future developments in this field will have the potential to refine and improve these methods, whilst addressing challenges that keep emerging in digital security.

Our findings imply the procedure of including ensemble-based machine learning models for refining the current approach for evaluating password strength along with 'medium' strength passwords. Appropriate dynamic passwords can be a solution to the conflict between security requirements and usability. The fundamental requirement of most models is retraining and updating, especially given the constantly emerging new threats. Also, giving advice when setting passwords enables the users to enhance the decisions made.

Since most models do not consider the use of passwords in other languages and cultures, future research should expand on the concept of making the signal better assess password strength. Temporal variations of password strength characteristics and modeling of corresponding changes should be further analyzed. Thorough testing against best-known attacks on passwords will demonstrate effectiveness. Therefore, it will be possible to increase the effectiveness of feedback when using password strength prediction along with analysis of user behavior. It is also recommended that researchers should also work on making the decision-making of the ensemble models more understandable and also on how the password strength prediction could be combined with biometric authentication for effective multiple-factor authentication.

The following research recommendations and directions can enable the enhancement of password security, minimisation of human error threats within an organisation, and security of organisational data within a dynamic technological landscape.

REFERENCES

[1] Sui, Y., Zou, X., Du, E.Y. (2011). Biometrics-based authentication: A new approach. In 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), Lahaina, HI, USA, pp. 1-6. https://doi.org/10.1109/ICCCN.2011.6005767

[2] Juozapavičius, A., Brilingaitė, A., Bukauskas, L., Lugo, R.G. (2022). Age and gender impact on password hygiene. Applied Sciences, 12(2): 894. https://doi.org/10.3390/app12020894

[3] Murali, P.S. (2022). Password strength analysis: A survey. International Journal of Scientific Research in Engineering and Management, 6(12): 1-6. https://doi.org/10.55041/ijsrem17298

[4] Chanda, K. (2016). Password security: An analysis of password strengths and vulnerabilities. International Journal of Computer Network and Information Security, 8(7): 23-30. https://doi.org/10.5815/ijcnis.2016.07.04

[5] Dell'Amico, M., Michiardi, P., Roudier, Y. (2010). Password strength: An empirical analysis. In 2010 Proceedings IEEE INFOCOM, San Diego, CA, USA, pp. 1-9. https://doi.org/10.1109/INFCOM.2010.5461951

[6] Guo, Y., Zhang, Z. (2018). LPSE: Lightweight password-strength estimation for password meters. Computers & Security, 73: 507-518. https://doi.org/10.1016/j.cose.2017.07.012

[7] de Carné de Carnavalet, X., Mannan, M. (2014). From very weak to very strong: Analyzing password-strength meters. In Network and Distributed System Security Symposium (NDSS 2014). Internet Society.

[8] Faster Capital. (2024). Password strength meters: Assessing the robustness of weaklongs. https://fastercapital.com/content/Password-Strength-Meters--Assessing-the-Robustness-of-Weaklongs.html.

[9] Golla, M., Dürmuth, M. (2018). On the accuracy of password strength meters. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, Canada, pp. 1567-1582. https://doi.org/10.1145/3243734.3243769

[10] Sarkar, S., Nandan, M. (2022). Password strength analysis and its classification by applying machine learning based techniques. In 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, pp. 1-5. https://doi.org/10.1109/ICCSEA54677.2022.9936117

[11] Galbally, J., Coisel, I., Sanchez, I. (2017). A new multimodal approach for password strength estimation—Part II: Experimental evaluation. IEEE Transactions on Information Forensics and Security, 12(12): 2845-2860. https://doi.org/10.1109/TIFS.2017.2730359

[12] Landauer, C. (1990). Correctness principles for rule-based expert systems. Expert Systems with Applications, 1(3): 291-316. https://doi.org/10.1016/0957-4174(90)90009-J

[13] Wang, C., Wang, Y., Chen, Y., Liu, H., Liu, J. (2020). User authentication on mobile devices: Approaches, threats and trends. Computer Networks, 170: 107118. https://doi.org/10.1016/j.comnet.2020.107118

[14] Song, C., Fang, Y., Huang, C., Liu, L. (2018). Password strength estimation model based on ensemble learning. Journal of Computer Applications, 38(5): 1383-1388. https://doi.org/10.11772/j.issn.1001-9081.2017102516

[15] He, Y., Alem, E.E., Wang, W. (2020). Hybritus: a password strength checker by ensemble learning from the query feedbacks of websites. Frontiers of Computer Science, 14: 1-14. https://doi.org/10.1007/s11704-019-7342-y

[16] Darbutaitė, E., Stefanovič, P., Ramanauskaitė, S. (2023). Machine-learning-based password-strength-estimation approach for passwords of Lithuanian context. Applied Sciences, 13(13): 7811. https://doi.org/10.3390/app13137811

[17] Farooq, U. (2020). Real time password strength analysis on a web application using multiple machine learning approaches. International Journal of Engineering Research and Technology, 9(12): 359-364.

[18] Vijaya, M.S., Jamuna, K.S., Karpagavalli, S. (2009). Password strength prediction using supervised machine

learning techniques. In 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, Bangalore, India, pp. 401-405. https://doi.org/10.1109/ACT.2009.105

[19] Divya, R., Devamane, S.B., Dharshini, V., Deepika, S. (2023). Performance analysis of machine learning algorithms for password strength check. In 2023 International Conference on Computational Intelligence for Information, Security and Communication Applications (CIISCA), Bengaluru, India, pp. 338-343. https://doi.org/10.1109/CIISCA59740.2023.00071

[20] Rathi, R., Visvanathan, P., Kanchana, R., Anand, R. (2020). A Comparative analysis of soft computing techniques for password strength classification. In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, pp. 1-3. https://doi.org/10.1109/ic-ETITE47903.2020.463

[21] Kuriakose, M.S., Teja, G.K., Srivatsava, A.H., Duggi, S., Jonnalagadda, V. (2022). Machine Learning Based Password Strength Analysis. International Journal of Innovative Technology and Exploring Engineering, 11(8): 5-8. https://doi.org/10.35940/ijitee.h9119.0711822

[22] Jamuna, K.S., Karpagavalli, S., Vijaya, M.S. (2009). A novel approach for password strength analysis through support vector machine. International Journal of Recent Trends in Engineering and Technology, 2(1): 79-82.

[23] Kim, S.J., Lee, B.M. (2023). Multi-class classification prediction model for password strength based on deep learning. Journal of Multimedia Information System, 10(1): 45-52. https://doi.org/10.33851/jmis.2023.10.1.45

[24] Mienye, I.D., Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. IEEE Access, 10: 99129-99149. https://doi.org/10.1109/ACCESS.2022.3207287

[25] Suganya, G., Karpgavalli, S., Christina, V. (2010). Proactive password strength analyzer using filters and machine learning techniques. International Journal of Computer Applications, 7(14): 1-5. https://doi.org/10.5120/1333-1788

[26] Jihad, K.H., Baker, M.R., Farhat, M., Frikha, M. (2022). Machine learning-based social media text analysis: Impact of the rising fuel prices on electric vehicles. In International Conference on Hybrid Intelligent Systems, pp. 625-635. https://doi.org/10.1007/978-3-031-27409-1_57

[27] Sherif, S.M., Alamoodi, A.H., Albahri, O.S., Garfan, S., Albahri, A.S., Deveci, M., Baker, M.R., Kou, G. (2023). Lexicon annotation in sentiment analysis for dialectal Arabic: Systematic review of current trends and future directions. Information Processing & Management, 60(5): 103449. https://doi.org/10.1016/j.ipm.2023.103449

[28] Rashad, M.R., Utku, A. (2023). Unraveling user perceptions and biases: A comparative study of ML and DL models for exploring twitter sentiments towards ChatGPT. Journal of Engineering Research. https://doi.org/10.1016/j.jer.2023.11.023

[29] Shaker, E., Baker, M.R., Mahmood, Z. (2022). The impact of image enhancement and transfer learning techniques on marine habitat mapping. Gazi University Journal of Science, 36(2): 592-606. https://doi.org/10.35378/gujs.973082

[30] Sattaru, N.C., Baker, M.R., Umrao, D., Pandey, U.K., Tiwari, M., Chakravarthi, M.K. (2022). Heart attack anxiety disorder using machine learning and artificial neural networks (ANN) approaches. In 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, pp. 680-683. https://doi.org/10.1109/ICACITE53722.2022.9823697

[31] Gupta, R., Tanwar, S., Tyagi, S., Kumar, N. (2020). Machine learning models for secure data analytics: A taxonomy and threat model. Computer Communications, 153: 406-440. https://doi.org/10.1016/j.comcom.2020.02.008

[32] Habib, G., Qureshi, S. (2022). Optimization and acceleration of convolutional neural networks: A survey. Journal of King Saud University-Computer and Information Sciences, 34(7): 4244-4268. https://doi.org/10.1016/j.jksuci.2020.10.004

[33] Kalantar, B., Pradhan, B., Naghibi, S.A., Motevalli, A., Mansor, S. (2018). Assessment of the effects of training data selection on the landslide susceptibility mapping: A comparison between support vector machine (SVM), logistic regression (LR) and artificial neural networks (ANN). Geomatics, Natural Hazards and Risk, 9(1): 49-69. https://doi.org/10.1080/19475705.2017.1407368

[34] Nooruldeen, O., Baker, M.R., Aleesa, A.M., Ghareeb, A., Shaker, E.H. (2023). Strategies for predictive power: Machine learning models in city-scale load forecasting. e-Prime-Advances in Electrical Engineering, Electronics and Energy, 6: 100392. https://doi.org/10.1016/J.PRIME.2023.100392

[35] Choudhary, R., Gianey, H.K. (2017). Comprehensive review on supervised machine learning algorithms. In 2017 International Conference on Machine Learning and Data Science (MLDS), Noida, India, pp. 37-43. https://doi.org/10.1109/MLDS.2017.11

[36] Peppes, N., Daskalakis, E., Alexakis, T., Adamopoulou, E., Demestichas, K. (2021). Performance of machine learning-based multi-model voting ensemble methods for network threat detection in agriculture 4.0. Sensors, 21(22): 7475. https://doi.org/10.3390/s21227475

[37] Baker, M.R., Taher, Y.N., Jihad, K.H. (2023). Prediction of people sentiments on twitter using machine learning classifiers during Russian aggression in Ukraine. Jordanian Journal of Computers and Information Technology, 9(3): 189-206. https://doi.org/10.5455/jjcit.71-1676205770

[38] Baker, M.R., Mahmood, Z.N., Shaker, E.H. (2022). Ensemble learning with supervised machine learning models to predict credit card fraud transactions. Revue d'Intelligence Artificielle, 36(4): 509-518. https://doi.org/10.18280/ria.360401

[39] Godahewa, R., Bergmeir, C., Webb, G.I., Montero-Manso, P. (2023). An accurate and fully-automated ensemble model for weekly time series forecasting. International Journal of Forecasting, 39(2): 641-658. https://doi.org/10.1016/j.ijforecast.2022.01.008