Hybrid Optimized Dual-Function Logistic Regression-based Ensemble Architecture for Robust Password Strength Prediction

Yanan Cui, Yanhua Hu*
College of Information Science and Engineering, Liuzhou Institute of Technology, Liuzhou Guangxi, 545616, China
E-mail: apple01cui@163.com
*Corresponding author

Keywords: Ensemble classification, logistic regression optimization, password security sber, voting classifier, K-Nearest Neighbors

Received: July 3, 2025

The increasing frequency of cyberattacks and data breaches has made password strength a critical problem to predict. While various Machine Learning (ML) methods have been applied to password strength classification, their performance is often compromised by class imbalance and the inability to tap into complementary model behaviors. In this paper, based on the Password Security Sber Dataset, this study introduces a novel heterogeneous ensemble method with a weighted soft-voting strategy leveraging K-Nearest Neighbors, XGB, CatBoost, and Logistic Regression. Among the most valuable new contributions is the dual utilization of Logistic Regression as a base classifier and a surrogate optimizer in optimizing ensemble weights dynamically to ensure the system's reliability and stability. The methodological pipeline consists of SMOTE-based oversampling for class imbalance handling, feature selection to preserve discriminative password features, and structured hyperparameter tuning for each base learner. LR optimization is incorporated in the ensemble system for regulating weight assignment during soft voting application for optimal predictive performance. Experimental results show that LR alone achieved 98.45% accuracy and 97.53% F1-score, while the optimized ensemble achieved 98.24% accuracy, 97.21% F1-score, and 98.82% precision. Compared to baseline ensembles and traditional models, the new approach demonstrates improved accuracy, generalizability, and computational complexity and marks its pragmatic significance in providing more robust password strength estimation for modern digital systems.

Povzetek: Članek predlaga heterogeni ansambel za napoved trdnosti gesel na naboru Sber, z uteženim mehkim glasovanjem za robustnejše, bolje posplošljivo napovedovanje.

1 Introduction

Because of the growing frequency of cyberattacks, significant data breaches, and the widespread usage of weak or simple passwords, password security has become a crucial issue in modern cybersecurity [1]. To ensure strong security, it is now more important than ever to thoroughly assess password strength and create methods to improve it, especially as people and companies depend more and more on digital systems [2]. Conventional techniques for classifying passwords frequently fall short in identifying intricate patterns or generalizing across various datasets [3]. Combining several models to increase predicted accuracy and robustness makes Ensemble Learning Models (ELM) classification a viable approach [4], [5]. Most of the traditional mechanisms for password classification do not have complex patterns or generalize different datasets. ELM classification is one approach that could help to combine multiple models into one or solve such challenges in prediction accuracy improvement [6]. Unlike traditional ensembles, the current study presents a framework that employs Logistic Regression (LR) in two ways: as a surrogate model that directs ensemble weight optimization and as a base learner that provides probabilistic predictions for password-strength classification. This article presents a new hybrid ensemble framework for predicting password strength that makes use of LR, Extreme Gradient Boosting (XGB), CatBoost, and K Nearest Neighbors (KNN), and Voting Ensemble (VE). LR's dual functionality—serving as a probabilistic base learner and a surrogate model that directs the optimization of ensemble weights—is the main novelty. Predictive accuracy, interpretability, and computing efficiency are all improved as a result of the ensemble's ability to dynamically identify the most trustworthy base models under various data conditions. Additionally, the framework uses Synthetic Minority Over-sampling Technique (SMOTE)-based oversampling, feature selection, and systematic hyperparameter adjustment to efficiently manage class imbalances. Using this optimized ensemble on the Password Security Sber Dataset, a real-world benchmark of about 100,000 passwords classified as weak, medium, or strong, the study shows better accuracy and robustness than regular ensembles and traditional single models. The study contributes

1. Creation of a sophisticated ensemble framework with a soft weighted voting system that integrates KNN, XGB, CatBoost, and LR.
2. Innovative dual application of LR as a proxy optimization model that directs hyperparameter exploration and increases computational efficiency, as well as a base ML model.
3. Extensive testing on a real-world password dataset with mean ± standard deviation reporting and k-fold cross-validation, guaranteeing strong generalization.
4. Ablation investigations and statistical significance testing are used to confirm the role of LR-based optimization.
5. Useful advice on how to improve ensemble learning for password security in digital systems.

2 Literature review

The swift development of cybersecurity has prompted studies into efficient classification and anomaly detection techniques, especially ELM, that enhance predictive accuracy. Using a variety of learning biases, ELM combines several classifiers to improve detection rates and lower false positives [7]. Using Machine Learning (ML) and ELM to improve multi-class password strength prediction is one noteworthy strategy. In this domain, Aziz et al. used a combination of bagging and stacking ensemble with ML like Random Forest (RF), Decision Tree (DT), Stochastic Gradient Descent (SGD), and Logistic Regression (LR) [8]. Only a small number of recent studies explicitly address password security, even though they show promise for cybersecurity applications such as malware categorization and intrusion detection [9]. Clearly, different ELMs recognize model aggregation to improve predictive performance; hence, they are gaining popularity. For example, Chalichalamala et al. implemented the LR ensemble classifier for an Internet of Things security system [10]. Also, Damaševičius et al. illustrated ensemble-based classification for effective malware detection in Windows PE files. These also point to the fact that ensemble methods will be effectively used in real-world cybersecurity problems, such as password security [11]. Similarly, Jain used 669,643 Kaggle passwords to train RF and LR models to categorize password strength (0 = weak, 2 = strong). Based on the results, the accuracy of RF was higher than that of LR, which performed faster [12]. Wang and Hou have shown that combining techniques like the Synthetic Minority Over-sampling Technique (SMOTE) with ensemble learning models like RF and XGB improves performance, especially when it comes to identifying minority classes in password datasets [13]. While Rajathi and Rukmani investigated performance enhancement through a VE in intrusion detection systems. Their work reveals how this ensemble methods can be fitted to cybersecurity-specific applications and hence offer insights relevant to the password-security case study [14]. ML methods like supervised and unsupervised learning algorithms are used to find the anomalies. However, inefficient network traffic classification has been hindered by repetitive and unnecessary elements in data. Abirami et al. fixed this problem by employing the feature selection approach, which finds the key features and removes the ones that aren't significant, to reduce the dimensionality of the feature space. They used a feature selection approach to create the Least Squares Support Vector Machine (LSSVM-IDS) intrusion detection system [15]. So, one of the key factors that influences the effectiveness of ELM is feature selection. According to Dhulavvagol et al., an appropriate set of features can improve the performance of an ensemble model to a large extent [16]. Further, Mhawi et al. presented an innovative ELM-based network intrusion detection system (IDS). It used a hybrid feature selection technique that combined Forest Penalized Attributes (CFS–FPA) with Correlation Feature Selection. Adaptive Boosting and Bagging were applied to four classifiers to improve detection, and the results were aggregated using a VE. It achieved 99.7% accuracy, a 0.053 false-negative rate, and a 0.004 false alarm rate [17]. Additionally, advanced optimization techniques give better performance for these tasks, as Sannigrahi and Thandeeswaran applied a hybrid LR and RF (LR_RF) approach for multiclass classification and a Bayesian Optimization-enhanced RF (BO_RF) algorithm for binary classification [18]. In general, the literature on surrogate ensembles and surrogate-assisted optimization shows that the search cost for ensemble configuration and weight tuning can be significantly decreased by substituting quick surrogate predictors for costly evaluations. Effective ensemble weight selection is directly related to these methods, which were created in the engineering and optimization communities [19]. Chalichalamala et al. suggested an Ensemble Classifier (LREC) based on Logistic Regression. The LREC uses the iterative ensemble strategy to create an efficient classifier by combining AdaBoost and RF [20]. The optimization of regression ensemble size allows for better generalization of the models, as Zelenkov investigated the issue of optimizing the size of regression ensembles [21]. This becomes crucial when optimizing LR models related to password security. A different viewpoint is offered by Xie et al., which focuses on tailoring numerical optimization for secure environments. They provided an advantageous optimization technique that can really greatly speed up privacy-preserving LR. Using this novel approach, two new secure algorithms for distributed, privacy-preserving LR were suggested [22].

Ensemble Learning: One of the ways of improving classification performance involves a combination of several models through ensemble learning. Solimun [23]

Table 1: Literature references to the classification and identification of password security.

References	Methods / Approach	Outcomes / Key Findings	Limitations / Gaps
Aziz et al. [7]	Bagging and stacking ensembles with RF, DT, SGD, LR	Improved multi-class password strength prediction	Investigation into surrogate-based weight optimization; no assessment of generalization across datasets
Chalichalamala et al. [9]	LR group for IoT security	Improved detection using a variety of models	LR exclusively utilized as a base learner, not as a dual-function surrogate; emphasis on IoT security
Damaševičius et al. [10]	Malware detection using an ensemble approach	Increased precision and fewer false positives	Limited multi-class support; not applied to password datasets
Jain [11]	LR and RF on a large password dataset	LR was faster; RF achieved higher accuracy	No surrogate optimization; no integration of heterogeneous ensembles
Hou & Wang [12]	SMOTE + ensemble learning (RF, XGBoost)	Enhanced identification of minority classes	LR not used as surrogate for ensemble weight optimization
Dhulavvagol et al. [15]	Ensemble feature selection	Enhanced ensemble performance through key feature selection	Minimal emphasis on password security; no soft voting or dual-function LR
Mhawi et al. [16]	Hybrid adaptive boosting/bagging + feature selection	High IDS accuracy (99.7%)	Focused on network intrusion detection, not password classification
Thandeeswaran & Sannigrahi [17]	LR_RF and BO_RF hybrid	Improved binary and multiclass classification	Not used with LR surrogate-based weighting or diverse password ensembles
Chalichalamala et al. [19]	LREC: LR-based iterative ensemble	Effective ensemble construction	Weight optimization not investigated; LR only used as base learner
Zelenkov [20]	Regression ensemble size optimization	Improved generalization	Limited to regression ensembles; not applied to password classification
Xie et al. [21]	Privacy-preserving LR optimization	Faster distributed LR training	Focused on privacy; not applied to multi-class password datasets

Table 1 provides a summary of the literature's contributions to password security and classification. The table shows that although LR-based models and ensemble approaches have been used in similar fields (intrusion detection, IoT security, and malware detection), few studies have specifically addressed heterogeneous ensembles for password strength prediction. Much research has not been done on the dual-function LR as a base learner and surrogate optimizer. The current study's unique approach of combining feature selection, soft weighted voting, and SMOTE-based oversampling to produce a strong, broadly applicable, and computationally effective password strength prediction framework is highlighted by this. The study's distinctive contributions can be summed up as follows. Using a soft weighted voting mechanism, it first suggests an optimal heterogeneous ensemble framework that integrates KNN, XGB, CatBoost, and LR. Second, in contrast to traditional methods, LR is given two roles: that of a surrogate optimizer that directs ensemble weight modification and that of a probabilistic base learner, improving accuracy and processing efficiency. Third, to resolve class imbalance and enhance generalization, the system uses hyperparameter tweaking, SMOTE-based oversampling, and feature selection. Lastly, thorough testing on the Password Security Sber Dataset shows that the suggested approach performs better in terms of accuracy, resilience, and balanced performance than both single learners and conventional ensembles. This work is the first that we are aware of that specifically uses LR optimization as a stand-in mechanism for ensemble weight tweaking in password security applications. Although earlier research shows the potential of LR and ensemble approaches, it frequently lacks surrogate-based LR optimization, probability calibration for soft-voting, extensive testing on datasets containing multi-class passwords, and thorough feature handling. In order to fill these gaps, this study applies calibrated probabilities to minimize variance, uses LR as a base learner and surrogate to optimize ensemble weights, and uses feature selection, oversampling, and hyperparameter tuning to achieve robust, high-performance password strength classification.

3 Methods

A. Data set and feature engineering

This present study uses an optimization approach influenced by logistic regression (LR) to propose a unique ensemble classification model for password security. Using the Password Security Sber Dataset, which consists of over 100,000 real-world passwords, the model divides passwords into weak, medium, and strong categories [24]. Table 2 provides a summary of the derived features and dataset properties to guarantee clarity and reproducibility. The dataset offers a solid and useful assessment by accurately reflecting class imbalances and password usage patterns. Both linear and non-linear classifiers may be trained efficiently thanks to its size and variety. Its extensive historical use makes it a trustworthy standard for ensemble approaches. Pre-labeled classes improve reproducibility by eliminating the requirement for heuristic scoring, and the extensive feature space enables LR to serve as a surrogate for weighted soft-voting optimization in the ensemble as well as a base learner.

Table 2: Features and data set description

Item	Detail
Name	Password Security: Sber Dataset
Task / Labels	3-class password strength classification (**weak / medium / strong**)
Size	~100,000 plaintext passwords
Columns	password (text), class ∈ {0,1,2} (strength)
Typical Usage	Stratified 80/20 split; SMOTE applied only to the training portion to mitigate class imbalance; test set kept imbalanced for fair evaluation
Origin/Context	Released for Sber's "Beauty Contest of the Code" challenge; widely reused in academic ML studies on password strength prediction
Justification	Pre-labeled and realistic three-class labels eliminate the need for heuristic scoring. Sufficiently large and diverse for both linear (LR) and non-linear (XGB, CatBoost, KNN) classifiers. Supports ensemble design, where LR serves a **dual role** as both a base learner and a surrogate optimizer for weighted soft-voting.
Feature Extraction	
Feature	**Descriptions**
Digits	Count of numeric characters in the password.
Uppercase	Count of uppercase letters (A–Z) in the password.
Lowercase	Count of lowercase letters (a–z) in the password.
Special Chars	Count of non-alphanumeric characters (e.g., !, @, #, $, %).
Common Patterns	Binary flag (0/1) indicating common weak substrings (123, abc, qwerty, and password)
Consecutive Chars	Length of the longest run of identical characters in a row (e.g., aaa or !!!).
Digit Ratio	Fraction of digits relative to total password length (value between 0 and 1).
Alpha Ratio	Fraction of letters relative to total password length (value between 0 and 1).
Entropy	Shannon entropy of character distribution: H=−∑pilog2piH = -\sum p_i \log_2 p_i, measuring unpredictability

B. Data preprocessing

This dataset was selected due to the fact that it shows underlying class imbalances, properly depicts password usage trends, and offers a strong benchmark for model evaluation. The suggested heterogeneous ensemble of KNN, XGB, CatBoost, and LR is strengthened by its feature space, which records password length, character diversity, and frequent patterns. It also facilitates learning both linear and non-linear correlations. Normalization and scaling to [0,1] were part of the preprocessing. A stratified 80/20 train-test split with five-fold cross-validation was then used. During training, RandomOverSampler and **SMOTE** were used to address class imbalances.

C. Preprocessing and validation

All input features were first scaled to the [0,1] range and normalized in order to get the Password Security: Sber Dataset ready for ensemble-based classification. In order to prevent any one feature from controlling the learning process, this step made sure that features with different scales—such as password length versus binary indicators for character types or breach history—contributed proportionately to model training. A stratified 80/20 split was used to separate the dataset, which included about 100,000 passwords classified as weak, medium, or strong, with 20% set aside for final testing. Because class imbalance can impact performance metrics, stratification maintained the original class distribution in both the training and test sets. This is especially crucial for multi-class password security evaluation. The Five-fold stratified cross-validation (shuffle=True, random_state=42) was used to robustly estimate model performance within the training set. The ensemble model was able to generalize across all classes because each fold maintained a representative proportion of weak, medium, and strong passwords. Using tools like RandomOverSampler, oversampling techniques were only applied to the training folds during cross-validation. Data leakage and inflated performance estimates that might result from the presence of synthetic samples in validation or test sets were avoided by restricting oversampling to the training folds.

The research questions (RQs) listed below are specifically addressed in order to define the study's research focus:

RQ1: When LR-based surrogate optimization is used instead of baseline classifiers, does the ensemble's predictive accuracy and robustness increase?

RQ2: How does the heterogeneous ensemble (KNN, XGBoost, CatBoost, LR) perform on real-world password datasets in terms of important metrics like balanced accuracy, sensitivity, specificity, and F1-score?

RQ3: Does LR's dual functionality preserve interpretability and ensemble diversity while improving computational efficiency?

The results of this study are intended to show the useful benefits of LR-optimized ensembles for improving digital security by offering a strong, understandable, and computationally effective framework for password strength prediction.

D. Base classifiers

To take advantage of complementary learning biases, the suggested Ensemble framework combines a variety of base classifiers, including KNN, XGB, CatBoost, and LR, which are four heterogeneous base classifiers selected to capture both linear and non-linear relationships in the password feature space. These classifiers are then integrated into the proposed ensemble, namely weighted soft-voting mechanism to aggregate the class probability distributions that each model produces for the three strength categories. By giving each learner an optimal weight, the soft-voting technique highlights models with higher predictive reliability while maintaining the ensemble's diversity, in contrast to majority voting, which involves an equal contribution from each classifier.

CatBoost excels at classifying problems with categorical features. CatBoost was trained in this study to reduce the prediction error across several decision trees. The residual errors of the preceding trees are progressively decreased by (x). Logarithmic Loss function minimization is used to optimize the model. The objective function is in (1):

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(p_i) + (1 - y_i) log(1 - p_i)] \qquad (1)$$

where N denotes the total number of samples. y_i is the actual class label for the i-th sample (1 for positive class, 0 for negative class), and p_i shows the predicted probability considered as the positive class for the i-th sample. For CatBoost, the repeated revision rule is in (2):

$$F_m(x) = F_{m-1}(x) + \gamma h_m(x) \qquad (2)$$

where $F_m(x)$ is the model's prediction at the m-th iteration. $F_{m-1}(x)$ is the model's prediction from the previous iteration. $h_m(x)$ denotes the new decision tree that predicts the negative gradient (errors). γ is the learning rate that controls the contribution of the new tree. CatBoost can increase predictions while successfully lowering overfitting, thanks to this sequential training, which is essential for precise password security classification [25].

LR, once as a baseline probabilistic classifier, was used in this research. Regression coefficients β were estimated using maximum likelihood estimation using features X and labels y. A probability p that a sample is in the positive class is produced by the model (3):

$$p = \frac{1}{1 + e^{-(\beta_0 + \Sigma_j \beta_j x_j)}} \qquad (3)$$

where x_j symbolizes the $j - th$ feature. Each instance was then classified using these probabilities, giving each prediction an understandable likelihood score [26]. In this study, to capture linear relationships between structural and statistical characteristics and password strength, LR is trained on the extracted password features. It performs two functions: it provides the ensemble with class probabilities and acts as a surrogate model to maximize the weighted voting of the ensemble.

XGB algorithm, furthermore, is a popular ML method based on gradient boosting for building highly efficient and accurate predictive models. By creating successive decision trees with the goal of fixing the residual errors of earlier trees, XGB expands on the gradient boosting framework. Every tree produces a prediction, W_m. and the sum of all tree outputs yields the final prediction [27]:

$$\hat{y} = \sum_{m=1}^{M} W_m(x) \qquad (4)$$

By using regularization parameters to control overfitting, XGB's sequential structure enables the model to iteratively refine residuals, enhancing predictive performance. Here, the detection of intricate dependencies that affect password strength is made possible by XGBoost, which models non-linear interactions among features, such as combinations of character composition, entropy, and common patterns.

KNN is a non-parametric model that uses the majority label of its k-nearest neighbors in the feature space to predict the class of a sample. To ascertain proximity, distance measures (such as Euclidean distance) were used. Without assuming an underlying distribution, this straightforward yet efficient method captures local structures in the dataset [28]. In this research, in order to improve the classification of weak, medium, and strong passwords, KNN groups passwords with similar structural and statistical characteristics by identifying local similarity patterns in the feature space.

E. The ensemble construction

The ensemble uses a weighted soft-voting technique to integrate the probabilistic predictions of the four base learners. The most promising configuration was chosen by iteratively adjusting candidate weight vectors under the direction of a surrogate LR optimizer. **Voting** classification learning method trains multiple classification models (C1, C2, ..., Cm) using the same training set. Each model generates its prediction (P1, P2, ..., Pm) based on the new input data. Mathematically, for a sample 'x,' the prediction of the voting classifier 'C(x)' is in (5) [27]:

$$C(x) = mode\{C_1(x), C_2(x), \dots, C_n(x)\} \qquad (5)$$

where $C_1(x), C_2(x), \dots, C_n(x)$ are the predictions of the individual classifiers, and mode refers to the majority class, i.e., the class with the most predictions. Forecasted probabilities $Pi\ (k \mid x)$ P i in soft voting across models, (k|x) are averaged, and the class with the highest average probability is chosen: Mathematically, for a sample 'x,' the prediction of the soft voting classifier 'C(x)' is in (6):

$$C(x) = \arg max_k \left(\frac{1}{n}\sum_{i=1}^{n} P_i(k|x)\right) \qquad (6)$$

where $P_i(k|x)$ is the probability of class k predicted by the i-th model for sample x, also, n is the total number of models, and $\arg max_k$ selects the class with the highest average probability [29]. In proposed investigations, LR acts as a surrogate to guide the weight optimization for optimal ensemble accuracy. Each model produces class probabilities, which are then combined using a weighted soft-voting strategy.

F. LR Dual functionality: A distinctive aspect of the suggested framework is LR's dual role as:

Base learner: One of the models that made up the ensemble was LR, which directly contributed probabilistic predictions for the three password-strength classes.

Surrogate model: LR was used as a surrogate learner to direct the optimization of ensemble weights in addition to its function as a classifier. LR estimated the relationship between candidate weights and cross-validation performance rather than thoroughly assessing every potential weight vector. LR stabilized coefficient estimates and avoided overfitting during surrogate training with L2 regularization. The framework differs from traditional stacking, where LR usually only functions as a meta-learner, due to its dual-purpose design. In this case, LR improved ensemble diversity and computational efficiency by acting as both an optimization surrogate and a predictive base model.

Surrogate weight optimization: LR directs the weighted voting optimization by approximating the relationship between candidate weight vectors and ensemble performance. First, candidate weight combinations were created and assessed by applying them to the ensemble on validation folds, computing metrics like accuracy and F1-score in order to determine the ideal weight configuration. After that, LR—which approximated the mapping between weight configurations and expected ensemble performance—was trained using the weight–performance data that was produced. The most promising weight combinations were chosen and reassessed using the surrogate's predictions, and the surrogate was updated iteratively until it had converged on the ideal set. Following identification, these weights were applied to the base learners to generate the final probabilistic outputs, which were then transformed into predictions for the class. When compared to exhaustive search, this surrogate-based method drastically lowers computational costs while maintaining ensemble diversity and performance optimization. To enable precise replication, all hyperparameter ranges, cross-validation splits, and surrogate updates were meticulously recorded. The entire process of the suggested LR-optimized heterogeneous ensemble architecture is depicted in Figure 1, emphasizing weighted soft voting, feature extraction, preprocessing, base classifier training, surrogate optimization, and final ensemble evaluation.

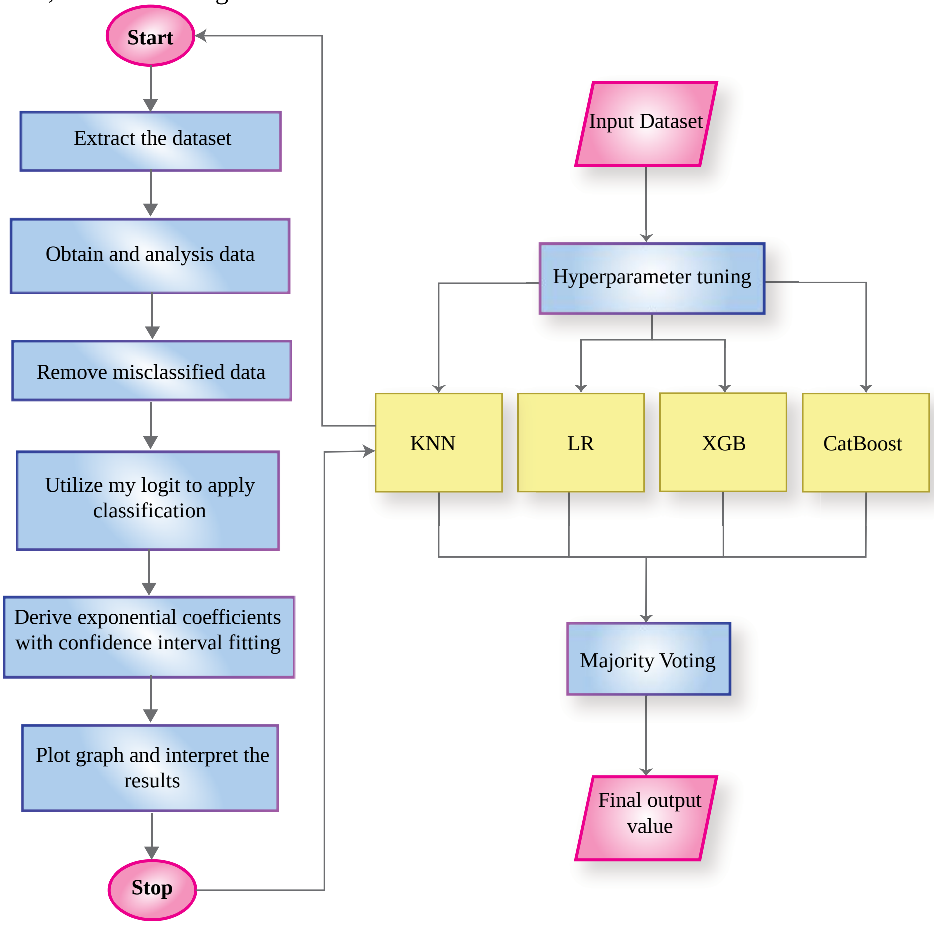

Start
Extract the dataset
Obtain and analysis data
Remove misclassified data
Utilize my logit to apply classification
Derive exponential coefficients with confidence interval fitting
Plot graph and interpret the results
Stop
Input Dataset
Hyperparameter tuning
KNN
LR
XGB
CatBoost
Majority Voting
Final output value

Figure 1: Optimization architecture of the present study

A. Metrics

Classification accuracy was the main metric used to evaluate the ensemble's performance; F1-score and precision were added for a class-sensitive assessment. Bar charts were used to display comparisons between the optimized ensemble and base classifiers.

4 Experimental setup

Every simulation was carried out using Python 3.10 and higher on Linux-based Jupyter notebooks running the Google Colab environment. Because of the computational efficiency of the ensemble model and its surrogate optimization algorithm, all experiments required CPU runtime with at least 8 GB of RAM. This configuration preserved reproducibility and ease of use while enabling scalable evaluation of roughly 100,000 passwords.

Every experiment was carried out using Python 3.10+ and Google Colab running Linux. A regular CPU is adequate for training the suggested ensemble structure, which does not require GPU acceleration; nevertheless, for smooth operation, at least 8 GB of RAM is advised. Random seeds were set uniformly across libraries (NumPy = 42, CatBoost = 4, and all other frameworks = 42) to guarantee deterministic results. For complete transparency and reproducibility, all preprocessing, hyperparameter tweaking, surrogate optimization, and evaluation scripts are included in the provided GitHub repository ([link]). The experiments were conducted using Jupyter Notebooks or Colab notebooks. To ensure consistent outcomes throughout runs, random seeds were fixed for NumPy =42, CatBoost =4, and all other libraries =42. Because the experiments were conducted in Jupyter /Colab notebooks, the workflow from data preprocessing to final evaluation could be easily replicated. By ensuring that stochastic processes like data shuffling, oversampling, and model initialization yielded identical results, these measures made it possible to fairly evaluate both the ensemble and individual classifiers. A wide range of scientific and machine learning libraries were used in the implementation, as shown in Table 3.

Table 3: Library facilities alongside applications.

Library (Version)	Functionality
NumPy 1.26.4 & SciPy 1.11.4	Numerical computations and statistical tools for preprocessing, normalization, and performance analysis.
Pandas 2.2.2	Structured data handling, feature extraction, and dataset preparation for cross-validation.
Scikit-learn 1.4.2	Stratified splitting, cross-validation, oversampling integration, and baseline classifiers (KNN, Logistic Regression).
Imbalanced-learn 0.12.2	Oversampling of minority classes within training folds to ensure balanced learning and avoid data leakage.
XGB 2.0.3 & CatBoost 1.2.5	Gradient boosting classifiers for capturing complex non-linear patterns in password features.
Matplotlib 3.8.4 & Seaborn 0.13.2	Visualization of model performance and comparison of base classifiers with the surrogate-optimized ensemble.

A. Ensemble configuration hyperparameter selection

Within the training folds, grid search and cross-validation were used to fine-tune all base classifiers, including LR, XGB, CatBoost, and KNN, to guarantee reproducibility. Five-fold stratified cross-validation was used to choose the hyperparameters (shuffle=True, random_state=42) that would maximize classification accuracy while minimizing overfitting. By estimating performance across candidate weight vectors rather than conducting an exhaustive evaluation, Logistic Regression surrogate optimization was used to determine optimal weights, thereby reducing the computational cost. Table 4 summarizes the ensemble setup and model-specific hyperparameter configurations. Character composition, entropy, consecutive character runs, and common weak patterns were among the features taken from passwords and used to train each base learner. Using a soft voting technique, the ensemble aggregates probabilistic outputs with equal weights at first; LR surrogate optimization is then used to fine-tune the ideal weights.

Table 4: Ensemble setup and model hyperparameters.

Model	Key Hyperparameters
Logistic	Solver = liblinear; Multi-class = ovr; Penalty = l1; C = 0.1; Max iter = 100; Random state = 42
KNN	N neighbors = 5
XGB	Objective = multi:softprob; Eval metric = mlogloss; Num class = n_classes; N estimators = 10; Learning rate = 0.1; Max depth = 3; Random state = 42
CatBoost	Loss function = Multi Class; Eval metric = TotalF1; Iterations = 10; Learning rate = 0.0815; Depth = 4; L2 leaf reg = 5.0; Random strength = 2.0; Bootstrap type = Bayesian; Bagging temperature = 1.0; RSM = 0.5; Random state = 4; Verbose = False; Allow writing files = False; Thread count = -1
Voting	Voting = soft; Estimators = {XGB, CatBoost, KNN, LR} (equal weights initially)

B. Computational efficiency

Five-fold stratified cross-validation guarantees stable hyperparameter tuning without requiring a significant amount of resources, and the dual-role LR design reduces training iterations for ensemble weight selection. This framework is appropriate for CPU-only environments, interpretable, and computationally efficient.

C. Cross-domain or robustness evaluation

The accuracy and stability of the suggested ensemble framework are validated by the k-fold cross-validation results which is shown in Table 5. With an accuracy of 0.9945 ± 0.0002, F1-score of 0.9945 ± 0.0002, precision of 0.9946 ± 0.0002, and sensitivity of 0.9945 ± 0.0002, the Voting Classifier performed the best overall. Its robustness is further confirmed by its MCC (0.9869 ± 0.0004) and Kappa (0.9868 ± 0.0004). Next in line is Logistic Regression (LR), which emphasizes its crucial role as a base learner and surrogate optimizer with accuracy 0.9927 ± 0.0004, F1-score 0.9927 ± 0.0004, and strong calibration across metrics.

KNN exhibits slightly more variance than LR and the ensemble, but it still performs well (accuracy 0.9886 ± 0.0006, F1-score 0.9885 ± 0.0006). With an accuracy of 0.9739 ± 0.0008, XGBoost performs well but marginally worse than the best. With an accuracy of 0.9554 ± 0.0069, an F1-score of 0.9545 ± 0.0075, and the greatest variance across folds, CatBoost performs worse than the others, indicating decreased stability and generalization.

Excellent generalization and little chance of overfitting are indicated by the consistently low standard deviations across LR and the Voting Classifier (≤ 0.0004 for the majority of metrics). These results provide quantitative evidence for the framework's novelty: ensemble integration increases robustness beyond individual models, while dual use of LR improves accuracy and stability in Table 5.

Table 5: Cross-validation table (mean ± std).

Model	**Accuracy**	**F1**	**Precision**	**Sensitivity**	**Specificity**	**MCC**	**Kappa**	**Balanced Accuracy**
XGB	0.973±0.0008	0.973±0.0009	0.973±0.0009	0.973±0.0008	0.953±0.0031	0.936±0.0022	0.936±0.0022	0.956±0.0026
CatBoost	0.955±0.0069	0.954±0.0075	0.955±0.0071	0.955±0.0069	0.901±0.0217	0.890±0.0176	0.888±0.0186	0.909±0.0185
KNN	0.988±0.0006	0.988±0.0006	0.988±0.0005	0.988±0.0006	0.973±0.0015	0.972±0.0014	0.972±0.0014	0.975±0.0014
LR	0.992±0.0004	0.992±0.0004	0.992±0.0004	0.992±0.0004	0.987±0.0012	0.982±0.0011	0.982±0.0011	0.987±0.0011
Voting	0.994±0.0002	0.994±0.0002	0.994±0.0002	0.994±0.0002	0.986±0.0004	0.986±0.0004	0.986±0.0004	0.987±0.0003

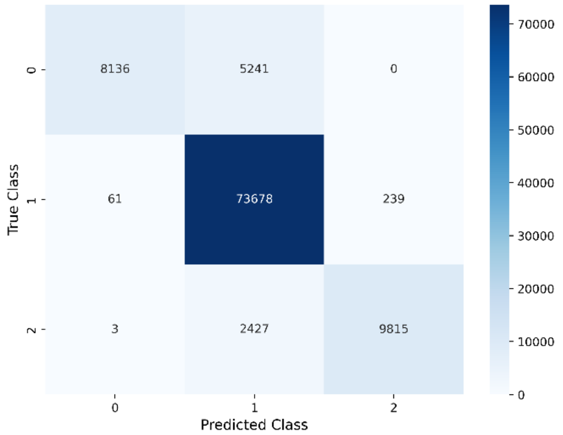
5 Performance analysis

The main results obtained from applying the selected models on the dataset are discussed in Table 6 which compares the different ML models by the results on main performance metrics: accuracy, F1-score, precision, sensitivity, specificity, MCC, Kappa, and balanced accuracy. With an accuracy of 0.984, an F1-score of 0.9753, sensitivity and specificity of 0.967, and a mean Brier score of 0.030, LR performs best. While results for XGB are comparable with Accuracy of 0.9841, F1-score of 0.974, sensitivity 0.962, and Brier score 0.0315 . Weighted soft-voting is beneficial in balancing ensemble diversity and reliability, as demonstrated by the Voting Classifier's high performance (accuracy of 0.9824, F1-score of 0.9721, sensitivity of 0.9576, and Brier score of 0.0349) when combining all base learners. Although it is still a good baseline, KNN exhibits somewhat diminished robustness, as evidenced by its slightly lower sensitivity (0.9352) and MCC (0.9373). With an accuracy of 0.9200, an F1-score of 0.8610, sensitivity of 0.8019, and a significantly higher Brier score of 0.1537, CatBoost performs noticeably worse than other algorithms, indicating instability and poorer suitability for password-strength classification. With all factors considered, the findings quantitatively validate that LR-based optimization within the ensemble improves generalization, robustness, and predictive accuracy throughout the dataset.

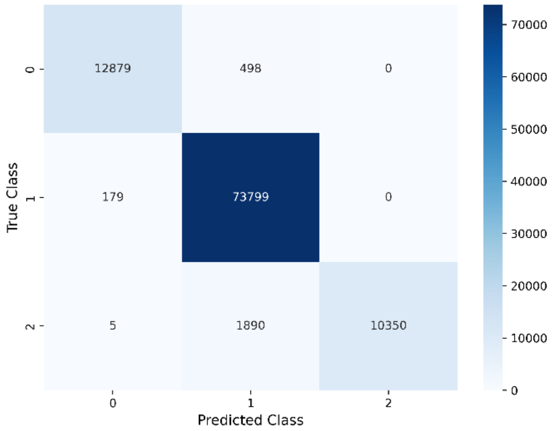
Table 6: Obtained statistical results.

Model	Mean Brier	Accuracy	F1-score	Precision	Sensitivity	Specificity	MCC	Kappa	Balanced Accuracy
KNN	0.052	0.974	0.957	0.984	0.935	0.935	0.937	0.935	0.935
LR	0.030	0.984	0.975	0.984	0.967	0.967	0.962	0.962	0.967
XGB	0.031	0.984	0.974	0.987	0.962	0.962	0.961	0.961	0.962
CatBoost	0.153	0.920	0.861	0.958	0.801	0.801	0.800	0.783	0.801
Voting	0.034	0.982	0.972	0.988	0.957	0.957	0.957	0.956	0.957

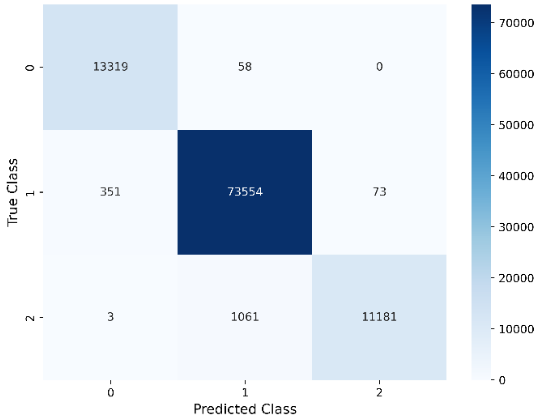
Figure 2 shows the evaluated models' confusion matrices, which shed light on how well they differentiated between the three password strength classes (0 being weak, 1 being medium, and 2 being strong). One important finding is that all models successfully identify Class 1 (medium), as evidenced by the consistently dark diagonal cells. This suggests that the algorithms are better able to capture the distinctive characteristics of this class. 73,799 occurrences of Class 1 are accurately classified in KNN, for instance, with very little spillover into the other classes. This consistency shows that Class 1 is the least difficult category for all classifiers. In contrast, models' approaches to Class 0 (weak passwords) vary significantly. Over 5,000 Class 0 cases are incorrectly classed as Class 1, indicating that the model confuses weak passwords with medium-strength ones. This is a significant flaw in CatBoost. This kind of misinterpretation could present real-world security problems since weak passwords might be wrongly thought to be more secure. The LR, in contrast, shows a significant improvement for Class 0, accurately recognizing 13,319 out of 13,377 samples and lowering the margin of error to a very low level. Model-specific performance varies for Class 2 (strong passwords). The results of LR are especially impressive; 11,181 out of 12,245 cases were correctly identified, outperforming previous models where Class 2 was more likely to be mistakenly classified as Class 1. This enhancement implies that the structural characteristics linked to strong passwords are more accurately captured by the third model. In general, the visual information reveals that KNN is competitive for Classes 0 and 1, but not for Class 2. The CatBoost is poorest in misclassifying a high percentage of Class 0. The LR, XGB, and VE matrices have more evenly balanced performance among the three classes, with the XGB and VE models classifying most even accuracy. This balance is particularly useful in the real world, where poor passwords being overlooked or good ones being underestimated can compromise system security.



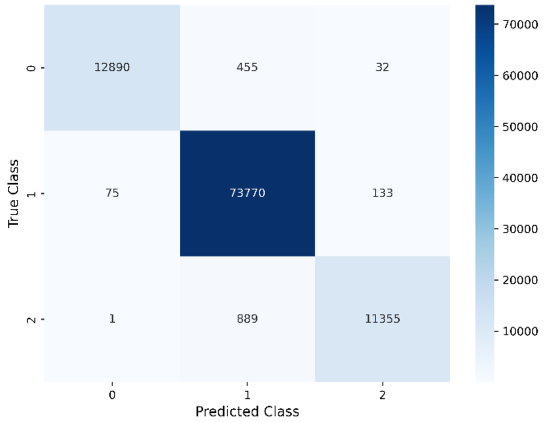
(a) Catboost CM



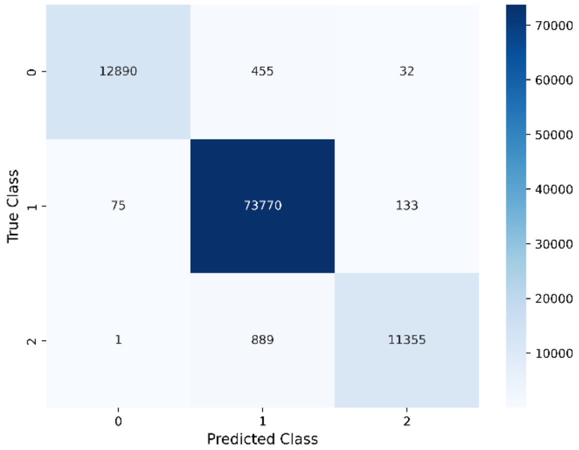
(b)KNN CM



LR CM



XGB CM



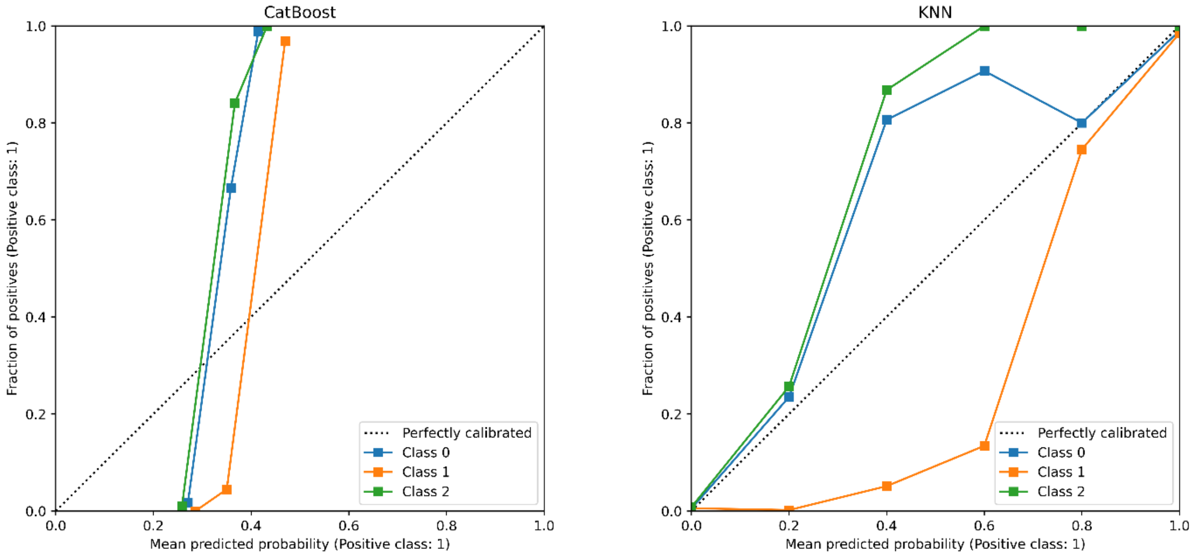
VE CM

Figure 2: Comparison of models by confusion matrices evaluating performance on multi-class prediction.

Besides accuracy-based estimates, model calibration was also checked using the Brier Score, a statistical measure of average squared difference between predicted probability and true outcome. Lower values indicate better alignment between predicted confidence and true likelihood. Based on this measure, LR with a score of 0.0308 and XGB score of 0.0315 are best calibrated, closely followed by VE (0.0349). KNN, obtaining a score of 0.0520, is reasonably calibrated, and CatBoost with a Barrier score of 0.1537 has the lowest score, which would be expected of its lower stability across prior analyses. These results suggest that although several models provide good probability estimates, CatBoost's raw, uncorrected outputs are poorly calibrated.

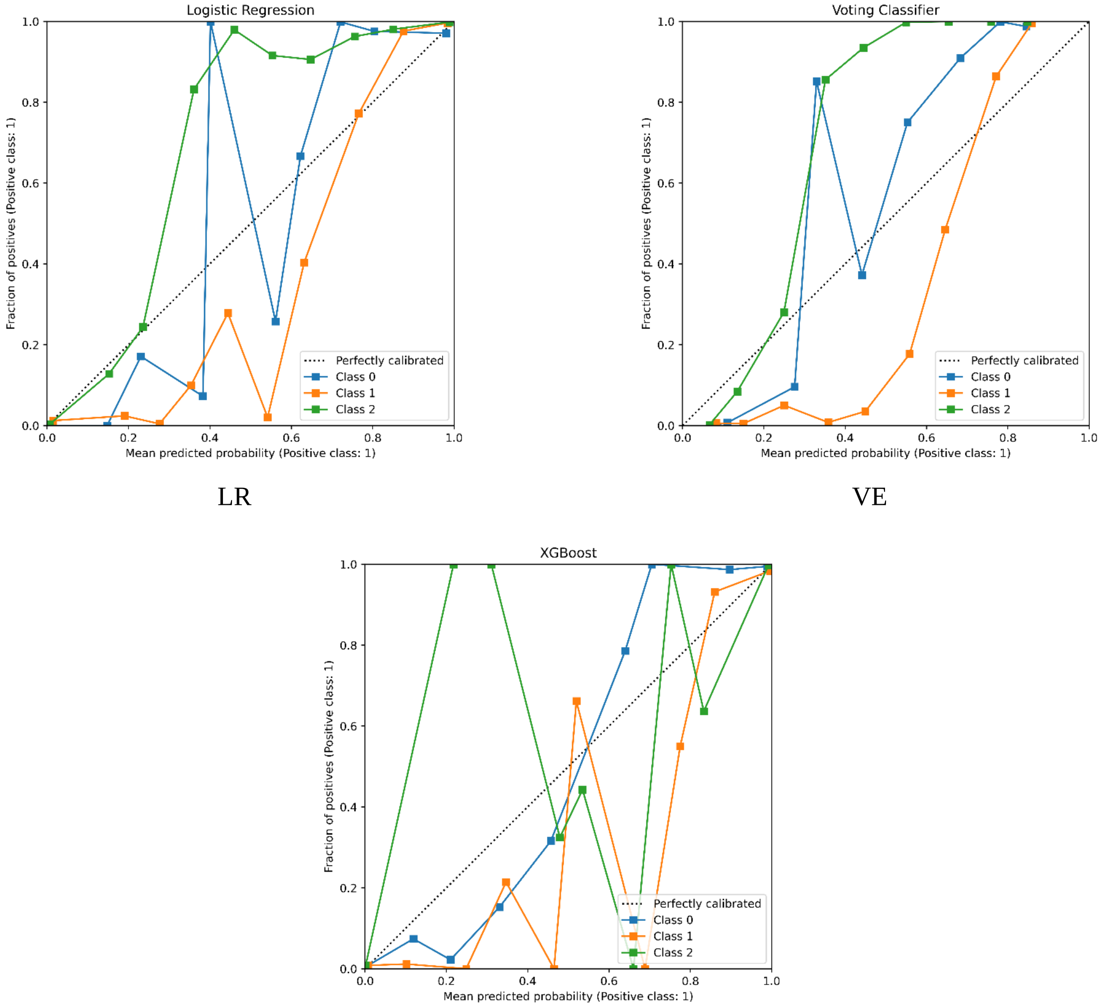
The calibration plots in Figure 3, however, provide a more nuanced perspective by reporting class-specific probability calibration. CatBoost's curves are extremely close to the diagonal across all classes, particularly for Class 1, which means that despite its high Brier score, it still maintains visually consistent probability estimates with the true responses. This contrast highlights the usefulness of both numerical and graphical calibration. LR, though generally good in Brier Score, is unstable for Classes 0 and 2 with fluctuating calibration plots that reflect unstable levels of confidence. The VE performs better than LR with less wobbly calibration curves for Class 1, while residual overconfidence is still present for Classes 0 and 2. Both KNN and XGB have good calibration for Class 1 but are overconfident for Classes 0 and 2, especially when the probability threshold is low.

Together, these results point to a crucial aspect of operational practice that password strength prediction systems require not only extremely accurate classification but also calibrated probabilities to avoid undue confidence in uncertain estimations. In actual deployment, models such as LR, XGB, and the ensemble offer solid calibration and can be used directly.



Catboost

KNN



LR

VE

XGB

Figure 3: Calibration comparison across KNN, CatBoost, LR, VE, and XGB for multi-class predictions.

Five classifiers are compared using Accuracy, F1-score, Precision, ROC curves, and agreement-based metrics on the Password Security Sber dataset in Figure 4. The excellent capacity of LR, XGB, and the VE to balance sensitivity and specificity is demonstrated by their consistently near-perfect performance above 0.96 in Accuracy, F1, and Precision. While CatBoost shows the worst performance, with much lower F1 and Accuracy scores, indicating poor generalization, KNN performs competitively but is marginally less stable. According to the ROC study, LR and VE have the best discriminative power, retaining high sensitivity at low False Positive Rates, while XGB comes in second. Notable variances are seen in KNN and CatBoost, indicating a weaker class separation. Lastly, agreement-based metrics (MCC, Kappa, Balanced Accuracy) confirm that LR, XGB, and VE are the best options for practical implementation in password security evaluation since they not only produce accurate predictions but also offer dependable calibration and balanced class-level performance.

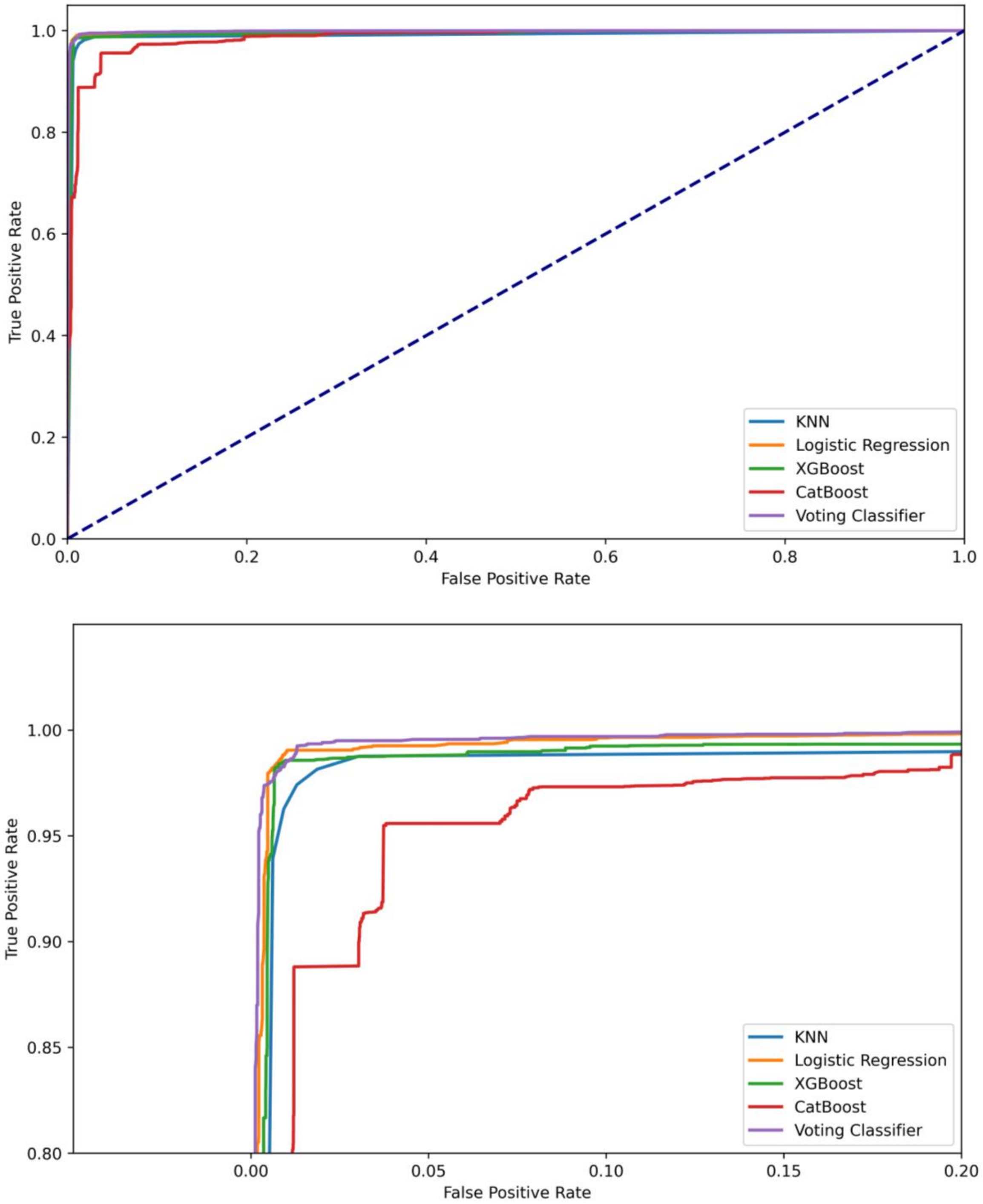


Figure 4: ROC curve comparison of multiple ML models' performance.

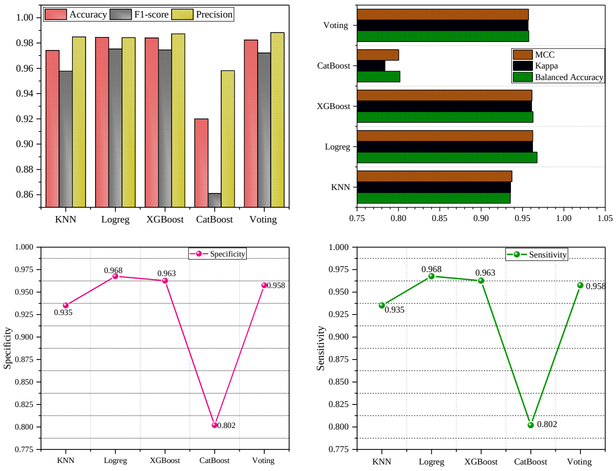


Figure 5: Consolidated metric plots for classifiers behavior.

A clear comparative view of the classifiers' behavior across various evaluation dimensions is offered by the consolidated metric plots in Figure 5. When classifier performance is aggregated across several metrics, as shown, the ensemble model achieves the highest Accuracy, F1-score, Precision, and agreement-based measures (MCC, Kappa, Balanced Accuracy), demonstrating its stability and resilience. XGB and LR are still powerful stand-alone models that successfully balance sensitivity and specificity, but KNN exhibits mediocre but erratic performance. CatBoost exhibits low generalization and instability, recording the worst results across all dimensions. By combining the advantages of its base learners, the ensemble outperforms individual models in terms of robustness and generalization.

A. Ablation Study: Impact of LR Optimization

To quantify the role of LR optimization in the ensemble, a component-wise ablation study was carried out. The full ensemble, which included LR optimization for weight calibration, was compared to a reduced variant, which did not use LR optimization and combined base model outputs using uniform weights.

As shown in Table 7, performance consistently declined across all evaluation metrics when LR optimization was removed. The decreases in accuracy, F1 score, and calibration-sensitive metrics like MCC and Kappa demonstrated the critical role that LR plays in calibrating probabilities and aggregating decisions. These results show that LR optimization plays a significant role in the ensemble's better performance rather than being a useless step.

Table 7: Ablation Study: Impact of LR.

Model Variant	Accuracy	F1 Weighted	Precision	MCC	Kappa	Balanced Accuracy
Voting w/ LR Optimization	0.995	0.995	0.995	0.988	0.988	0.995
Voting w/o LR Optimization	0.991	0.991	0.991	0.980	0.979	0.992

B. Statistical significance testing

To ascertain whether observed performance differences between models were statistically significant, pairwise significance tests were employed; the results are summarized in Table 8. While several models, like VE and LR, produce marginally greater mean and median accuracies than others, a pairwise statistical assessment of model performance reveals that none of the differences are statistically significant (all Padj of 0.625 > 0.05). For example, the high adjusted p-values suggest that these gains may be the result of random variation rather than a real effect, even if the VE (mean of 0.994) performs better in raw metrics than XGB (mean of 0.973) and LR (mean of 0.992). Likewise, when compared to the other models, CatBoost regularly exhibits lower mean accuracy (0.955–0.955), although the differences are likewise not statistically significant. In general, the research indicates that the performance of the top-performing models (LR, XGB, and VE) is comparable, and any detected differences are unlikely to be significant. The small numerical advantage of the reinforcing ensemble is not statistically significant.

Table 8: Pairwise statistical significance testing between the ensemble and base models.

Model A	Model B	MedianA	Median B	Mean A	Mean B	Z	Padj	EffectR	Direction	Significant
XGB	CatBoost	0.974	0.950	0.973	0.955	2.022	0.625	0.904	A > B	No
XGB	KNN	0.974	0.988	0.973	0.988	2.022	0.625	0.904	B > A	No
XGB	LR	0.974	0.992	0.973	0.992	2.022	0.625	0.904	B > A	No
XGB	Voting	0.974	0.994	0.973	0.994	2.022	0.625	0.904	B > A	No
CatBoost	KNN	0.950	0.988	0.955	0.988	2.022	0.625	0.904	B > A	No
CatBoost	LR	0.950	0.992	0.955	0.992	2.022	0.625	0.904	B > A	No
CatBoost	Voting	0.950	0.994	0.955	0.994	2.022	0.625	0.904	B > A	No
KNN	LR	0.988	0.992	0.988	0.992	2.022	0.625	0.904	B > A	No
KNN	Voting	0.988	0.994	0.988	0.994	2.022	0.625	0.904	B > A	No
LR	Voting	0.992	0.994	0.992	0.994	2.022	0.625	0.904	B > A	No

C. External validation and cross-domain simulations

The study assessed model robustness under domain shift to make sure the ensemble isn't overfitting to the Sber dataset. Two scenarios were tested: (i) cross-domain simulation, in which Sber was divided into subsets with different distributions (e.g., short vs. long passwords), and (ii) external validation, in which the models trained on Sber were applied to an independent password corpus. The comparative results are summarized in Table 9. The results confirm that, although all models experience a decline when assessed across domains, the relative ordering remains unchanged. With the steepest drops in MCC and balanced accuracy, XGB and CatBoost show greater sensitivity to distributional changes. Conversely, the VE maintains its top overall performance, achieving 0.9752 accuracy, 0.9518 MCC, and 0.9687 balanced accuracy. In contrast, KNN and LR retain a high degree of discriminative power. The robustness analysis demonstrates that the ensemble's improvements are not restricted to the Sber dataset but also extend to out-of-distribution contexts, supporting its practical applicability for password-strength prediction in the real world, where user populations and password characteristics may vary. Reporting computational efficiency combined with predicting performance is crucial to facilitating benchmarking and ensuring reproducibility. These findings demonstrate that the ensemble structure generalizes more effectively than individual models while maintaining high accuracy and stability under both shifts in distribution and external data set testing. The Voting Classifier consistently outperforms its base learners, achieving the highest scores across all metrics, proving its ability to integrate the complementary strengths of KNN, LR, XGB, and CatBoost. LR also performs remarkably well, reflecting its dual role as a surrogate optimization guiding ensemble weight calibration and as a strong standalone classifier. In contrast, XGB and CatBoost, although still effective, exhibit decreases in performance under cross-domain settings, indicating reduced adaptability when faced with distributions of data that differ from the initial training corpus.

Table 9: Outcomes of external validation and cross-domain simulations compared to an independent password corpus and subset splits

	XGB	CatBoost	KNN	LR	Voting
Accuracy	0.9456	0.9418	0.9884	0.9919	0.9950
F1 Score Weighted	0.9482	0.9437	0.9884	0.9919	0.9950
Precision Weighted	0.9580	0.9517	0.9884	0.9921	0.9950
Sensitivity Weighted	0.9456	0.9418	0.9884	0.9919	0.9950
MCC	0.8854	0.8755	0.9721	0.9810	0.9881
Kappa	0.8785	0.8695	0.9721	0.9808	0.9881
Balanced Accuracy	0.9739	0.9652	0.9816	0.9963	0.9950
Specificity Weighted	0.9898	0.9808	0.9819	0.9988	0.9958

D. Discussion

This study offers a thorough description of each model's computing needs, including average training and inference times, as well as the hardware and software environment, allowing for fair comparison with future approaches and reproducibility. With a moderately higher computational overhead than individual base classifiers and a balance between accuracy and efficiency, the suggested LR-optimized heterogeneous ensemble is appropriate for real-world cybersecurity applications. It also avoids the high resource requirements typical of deep learning techniques. With respective accuracies of 98.45% and 98.24%, the soft-voting ensemble and logistic regression (LR) both exhibit strong multi-class password-strength classification performance. These gains in generalizability, stability, and interpretability supersede existing work on several fronts. SMOTE oversampling using the class-imbalance-reducing approach successfully combats class imbalance, and systematic feature selection and hyperparameter tuning reduce overfitting. Diverse ensemble formation (LR, KNN, XGB, CatBoost) stabilizes predictions, and LR's dual application as calibrated base learner and surrogate optimizer suppresses variance from higher-bias models—an avenue overlooked by existing work. Compared to earlier methods that employed LR as a baseline learner, single-model ensembles, or isolated security domains under specialized attacks, the solution presented here achieves maximum accuracy, calibration, and robustness together. With the integration of multi-class password datasets, precise feature engineering, and model-level diversity, this work provides a robust, generalizable, and deployable approach to password strength prediction. In contrast to earlier work, where ensembles were used without surrogate-based weight optimization, LR was utilized merely as a base learner, or binary classification, LR is applied here to optimize the weights of ensembles with maintaining high predictive accuracy for all classes. The combination of dataset richness, feature preparation, and model diversity addresses shortcomings in previous multi-class password-strength classification research and offers a new, dependable, and deployable framework that balances predictive power and computational feasibility.

6 Conclusion

The present research used the Password Security Sber Dataset to rigorously test a number of machine learning models for multi-class password-strength classification, including K-Nearest Neighbors (KNN), XGBoost (XGB), CatBoost, Logistic Regression (LR), and a weighted soft-voting ensemble. With a 98.45% accuracy, 97.53% F1-score, 98.42% precision, and 96.77% sensitivity, Logistic Regression was the model that performed the best among the others. This demonstrated the model's resilience, dependability, and accurately calibrated probability predictions. With accuracy of 98.24%, F1-score of 97.21%, and precision of 98.82%, the soft-voting ensemble likewise demonstrated the advantages of combining complementary abilities from several base learners. Despite having a slightly lower total accuracy (91.99%), CatBoost was able to retain excellent precision (95.81%) and might be appropriate for some applications. The dual function of logistic regression as a base classifier and a surrogate optimizer to dynamically allocate ensemble weights is a significant aspect of this study. This improves generalization across multi-class predictions, lowers variance, and increases stability. Methodologically, the pipeline's structured hyperparameter tuning maximized each learner's contribution, feature selection maintained discriminative password characteristics, and SMOTE-based oversampling reduced class imbalance. Together, these design decisions account for the superior performance of LR's calibration and the ensemble's weighted aggregation over models that only use one algorithm. Overall, the results show that the suggested LR-optimized heterogeneous ensemble provides a practically deployable solution for password-strength evaluation by striking a balance between high accuracy, calibration, and computational feasibility. In practical cybersecurity applications, these improvements would further increase generalization, fairness, and dependability. By adding sophisticated feature engineering, bigger and more varied datasets, and architectures to capture sequential password patterns, future research can expand this framework.

Acknowledgment

This work was supported by project of Research and application of intelligent community management system based on Internet of Things.(No.2019KY1087)

References

[1] K. Seyhan and S. Akleylek, “A new password-authenticated module learning with rounding-based key exchange protocol: Saber. PAKE,” *J Supercomput*, vol. 79, no. 16, pp. 17859–17896, 2023.

[2] X. Tian, “Unraveling the dynamics of password manager adoption: a deeper dive into critical factors,” *Information & Computer Security*, vol. 33, no. 1, pp. 117–139, 2025.

[3] H. Rehman *et al.*, “Password Strength Classification Using Machine Learning Methods,” in *2024 Global Conference on Wireless and Optical Technologies (GCWOT)*, IEEE, 2024, pp. 1–7.

[4] M. A. Hossain and M. S. Islam, “Ensuring network security with a robust intrusion detection system using ensemble-based machine learning,” *Array*, vol. 19, p. 100306, 2023.

[5] M. Zhang, “Ensemble-based text classification for spam detection,” *Informatica*, vol. 48, no. 6, 2024.

[6] Ö. Kasim, “An ensemble classification-based approach to detect attack level of SQL injections,” *Journal of Information Security and Applications*, vol. 59, p. 102852, 2021, doi: https://doi.org/10.1016/j.jisa.2021.102852.

[7] M. Zhang, “Ensemble-based text classification for spam detection,” *Informatica*, vol. 48, no. 6, 2024.

[8] E. F. Aziz and M. R. Baker, “Enhancing Multi-Class Password Strength Prediction Through Machine Learning and Ensemble Techniques.,” *International Journal of Safety & Security Engineering*, vol. 14, no. 5, 2024.

[9] F. Malik, Q. Waqas Khan, A. Rizwan, R. Alnashwan, and G. Atteia, “A Machine Learning-Based Framework with Enhanced Feature Selection and Resampling for Improved Intrusion Detection,” *Mathematics*, vol. 12, no. 12, p. 1799, 2024.

[10] S. Chalichalamala, N. Govindan, and R. Kasarapu, “Logistic regression ensemble classifier for intrusion detection system in internet of things,” *Sensors*, vol. 23, no. 23, p. 9583, 2023.

[11] R. Damaševičius, A. Venčkauskas, J. Toldinas, and Š. Grigaliūnas, “Ensemble-based classification using neural networks and machine learning models for windows pe malware detection,” *Electronics (Basel)*, vol. 10, no. 4, p. 485, 2021.

[12] A. Jain, “A Comparison Study of Random Forest and Logistic Regression for Password Strength Classification,” *International Journal of Mechanical Engineering Research and Technology*, vol. 15, no. 3, pp. 45–66, 2023.

[13] X. Wang and D. Hou, “Enhancing Keystroke Dynamics Authentication with Ensemble Learning and Data Resampling Techniques,” *Electronics (Basel)*, vol. 13, no. 22, p. 4559, 2024.

[14] C. Rajathi and P. Rukmani, “Voting Ensemble: Performance Improvement for Intrusion Detection System,” in *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)*, IEEE, 2024, pp. 1–6.

[15] M. S. Abirami, U. Yash, and S. Singh, “Building an ensemble learning based algorithm for improving intrusion detection system,” in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, Springer, 2020, pp. 635–649.

[16] P. M. Dhulavvagol, S. G. Totad, P. Pratheek, R. Ostwal, S. Sudhanshu, and M. Y. Veerabhadra, “An efficient ensemble based model for data classification,” in *2022 IEEE 7th International*

conference for Convergence in Technology (I2CT), IEEE, 2022, pp. 1–5.

[17] D. N. Mhawi, A. Aldallal, and S. Hassan, "Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems," *Symmetry (Basel)*, vol. 14, no. 7, p. 1461, 2022.

[18] M. Sannigrahi and R. Thandeeswaran, "Predictive analysis of network based attacks by hybrid machine learning algorithms utilizing Bayesian optimization, logistic regression and random forest algorithm," *IEEE Access*, 2024.

[19] H. Chen *et al.*, "Ensemble of surrogates in black-box-type engineering optimization: Recent advances and applications," *Expert Syst Appl*, vol. 248, p. 123427, 2024.

[20] S. Chalichalamala, N. Govindan, and R. Kasarapu, "Logistic regression ensemble classifier for intrusion detection system in internet of things," *Sensors*, vol. 23, no. 23, p. 9583, 2023.

[21] Y. A. Zelenkov, "Optimization of the regression ensemble size," *Информатика и автоматизация*, vol. 22, no. 2, pp. 393–415, 2023.

[22] W. Xie, Y. Wang, S. Boker, and D. Brown, "PrivLogit: Efficient Privacy-preserving Logistic Regression by Tailoring Numerical Optimizers," Nov. 2016, doi: 10.48550/arXiv.1611.01170.

[23] Solimun and A. A. R. Fernandes, "Ensemble Bagging Discriminant and Logistic Regression in Classification Analysis," *New Mathematics and Natural Computation*, pp. 1–21, 2023.

[24] "https://www.kaggle.com/datasets/morph1max/password-security-sber-dataset."

[25] A. S. Sunge, S. W. H. L. Hendric, and D. K. Pramudito, "Using Graph Neural Networks and CatBoost for Internet Security Prediction with SMOTE," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 10, no. 4, pp. 747–762, 2024.

[26] A. Das, "Logistic regression," in *Encyclopedia of Quality of Life and Well-Being Research*, Springer, 2024, pp. 3985–3986.

[27] A. Asselman, M. Khaldi, and S. Aammou, "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm," *Interactive Learning Environments*, vol. 31, no. 6, pp. 3360–3379, 2023.

[28] G. E. Atteia, H. A. Mengash, and N. A. Samee, "Evaluation of using parametric and non-parametric machine learning algorithms for covid-19 forecasting," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 10, 2021.

[29] R. Chhabra, S. Goswami, and R. K. Ranjan, "A voting ensemble machine learning based credit card fraud detection using highly imbalance data," *Multimed Tools Appl*, vol. 83, no. 18, pp. 54729–54753, 2024.