

A Generalized Wine Quality Prediction Framework by Evolutionary Algorithms

Terry Hui-Ye Chiu, Chienwen Wu, Chun-Hao Chen*

Department of Information and Finance Management, National Taipei University Technology, Taipei, 106 (Taiwan)

Received 30 October 2020 | Accepted 13 March 2021 | Published 21 April 2021



ABSTRACT

Wine is an exciting and complex product with distinctive qualities that makes it different from other manufactured products. Therefore, the testing approach to determine the quality of wine is complex and diverse. Several elements influence wine quality, but the views of experts can cause the most considerable influence on how people view the quality of wine. The views of experts on quality is very subjective, and may not match the taste of consumer. In addition, the experts may not always be available for the wine testing. To overcome this issue, many approaches based on machine learning techniques that get the attention of the wine industry have been proposed to solve it. However, they focused only on using a particular classifier with a specific set of wine dataset. In this paper, we thus firstly propose the generalized wine quality prediction framework to provide a mechanism for finding a useful hybrid model for wine quality prediction. Secondly, based on the framework, the generalized wine quality prediction algorithm using the genetic algorithms is proposed. It first encodes the classifiers as well as their hyperparameters into a chromosome. The fitness of a chromosome is then evaluated by the average accuracy of the employed classifiers. The genetic operations are performed to generate new offspring. The evolution process is continuing until reaching the stop criteria. As a result, the proposed approach can automatically find an appropriate hybrid set of classifiers and their hyperparameters for optimizing the prediction result and independent on the dataset. At last, experiments on the wine datasets were made to show the merits and effectiveness of the proposed approach.

KEYWORDS

Decision Tree, Genetic Algorithm, Machine Learning, Random Forest, Support Vector Machine, Wine Quality Prediction.

DOI: 10.9781/ijimai.2021.04.006

I. INTRODUCTION

WINE has always been an essential part of the dining culture in western countries. With the booming economy in Asia countries in recent decades, wine consumption has increased even more. From the manufacturer point of view, understanding the wine's quality and creating a steady production is an important goal for the industry. However, testing the quality of the wine is complex and diverse. The wine quality is evaluated in terms of subtlety and complexity [1], ageing potential, stylistic purity, varietal expression, ranking by experts, or consumer acceptance, etc. By excluding the controllable object measures, experts' views are very subjective because it can cause the most considerable influence on both winemakers and how consumers think of the wine's quality [2]. Instead of focusing on how experts qualify the wine, focusing on consumer satisfaction based on collectable scientific data is more useful for the majority of wine producers because understanding the desires of the majority of consumers is essential in the production and sales of wine.

Recording the steps of the wine production procedure is to preserve the quality and knowledge of the whole winemaking process. The collected information is the best tool to guarantee the wine quality.

The wine industry has currently established the protected designation of origin (PDO) system [3] with the support of analytical chemistry and chemometric tools to obtain information related to a specific wine. With the improvement of technology both in software and hardware, winemakers started to use the collected data to improve the winemaking technique. Due to the high cost and lack of technological resources, it was difficult for most wine industries to classify the wines based on the chemical components. Many algorithms based on machine learning to assess the quality of wine have gained much attention for the wine industry using another approach to determine what attributes make a "good" wine that the consumers can satisfy with them. For instance, Yeo et al. focused on predicting the wine price using a machine learning technique by using past historical wine price data [4]. For wine production, Ribeiro et al. utilized the linear regression, neuron network and decision tree for predicting the wine vilification [5]. Study in [6] collected the wine dataset on the Cabernet Sauvignon characteristics for the cost-efficient prediction.

In 2009, Cortez et al. collected a wine quality dataset which consists of significant larger instances [7]. Then, three machine learning models, including multiple regression, support vector machine (SVM) and neuron network (NN), are trained using the collected wine dataset. It shows that SVM outperforms the other two methods, and indicates the importance of the correct setting of hyperparameters. Over the years, the wine dataset has been adopted in several studies with various methods such as SVM [8], [9], [10], [11], random forest (RF) [11], [12],

* Corresponding author.

E-mail address: chchen@ntut.edu.tw

[13], [14], [15], decision-tree-based algorithms [13], [15], and NN [5], [8], [9] to predict the quality of the wine based on physiochemical characteristics in the wine. In addition, several pieces of research used feature selection to improve the accuracy of wine quality prediction such as recursive feature elimination, principal component analysis (PCA) [11], [15], the statistic-based approaches [6], [10], and the synthetic minority oversampling technique (SMOTE) [14].

Based on the literature, two phenomenons can be found: (1) The SVM-based and RF-based algorithms have been proven to provide good results [6], [7], [8], [9], [10], [11], [12], [13], [14], [16]; (2) Tree-based approaches are also popularly used for wine prediction [5], [17]. However, the past literature mostly focused on using or comparing different machine learning models to find the one that can provide the best prediction result for the specific dataset. In other words, when the wine datasets are changed, the obtained model may not provide the same quality of performance. To solve this problem, in this paper, we firstly propose a generalized wine quality prediction framework which consists of the hybrid model acquisition and online prediction phases. Secondly, based on the framework, the generalized wine quality prediction algorithm based on the genetic algorithms is proposed. It first encodes the classifiers as well as their hyperparameters into a chromosome. The fitness of a chromosome is evaluated by the average accuracy of the employed classifiers. The genetic operations are then performed to generate new offspring. The evolution process is continued until reaching the stop criteria. As a result, the proposed approach can automatically find an appropriate hybrid set of classifiers and their hyperparameters for optimizing the prediction result and is independent on the dataset. Experiments were conducted on the wine datasets to show the merits and effectiveness of the proposed approach. The main contributions of the proposed framework and approach are listed as follows:

1. The proposed framework can use all types of classifiers with their hyperparameters as input in the hybrid model acquisition phase to find the suitable hybrid model and its hyperparameters for wine quality prediction.
2. The proposed framework overcomes the problem of data dependency, which means it provides a mechanism that can automatically obtain not only the appropriate hybrid model but also the hyperparameters for the given dataset no matter where the data is collected, from which areas and countries.
3. Based on the proposed framework, the GA-based generalized wine quality prediction algorithm has been proposed, and the obtained hybrid model and hyperparameters are better than existing approaches in terms of accuracy.
4. Experiments also indicate that when using macro F1-score as a fitness function for the proposed approach, the hybrid model can not only reach a better macro-F1 score but also has similar accuracy when comparing to the existing approaches.

The paper is organized as follows. Section II reviews the past works of predicting the wine quality as well as the basic knowledge used in the proposed approach. In Section III, the detailed components used in the proposed approach are described. In Section IV, the generalized wine quality prediction framework and the proposed approach are stated. The obtained results are analyzed and explained in Section V. Finally, conclusions and future work are drawn in Section VI.

II. LITERATURE REVIEW AND BACKGROUND KNOWLEDGE

A. Literature Review

Over the years, several studies used the machine learning techniques to predict wine quality, including utilizing SVM, k-NN, decision tree (DT), random forest, neuron network, regression and

others. Before describing them, the recent related studies and methods are we summarized and shown in Table I.

TABLE I. SUMMARY OF RECENT STUDIES

	SVM	K-NN	Naïve Bayes	Decision Tree	Random Forest	Neural Network	Regression	Others
Cortez et al. [7] (2009)	x					x		x
Ribeiro et al. [5] (2009)				x		x	x	
Appalasamy et al. [17] (2012)			x	x				
Bhattacharjee et al. [10] (2016)	x		x					
Andonie et al. [6] (2016)		x				x		x
Er et al. [11] (2016)	x	x			x			
Hu et al. [14] (2016)				x	x			x
Petropoulos et al. [18] (2017)								x
Zhang et al. [9] (2017)	x					x	x	
Agrawal et al. [19] (2018)						x		
Gupta [8] (2018)	x					x	x	
Trivedi et al. [12] (2018)					x		x	
Sowmya et al. [20] (2019)				x	x			x
Kumar et al. [21] (2020)	x		x		x			
Mahima et al. [16] (2020)		x			x			
Ozalp et al. [22] (2020)					x			x
Shaw et al. [13] (2020)	x				x	x		

From Table I, according to the used techniques, four types of approaches, including studies using SVM, RF, DT and others, are reviewed as follows.

(1) For studies using SVM, for instance, Cortez et al. [7] produced a large dataset for red and white vinho verde wines, a unique product from the Minho region of Portugal with the most common physicochemical tests selected as features. They selected optimal parameters associated with models by sensitivity analysis. The model selection was guided by parsimony search to find the best model. The results indicated that the SVM outperforms multiple regression and NN. In work presented by Gupta [8], it preprocessed the dataset using linear transformation to remove the inconsistent instances. Then, three models were trained using full features and the selected features by regression. Gupta summarized that SVM was the best model for wine quality prediction based on validating error rate. Also, precisions of SVM and NN using selected features were higher than that using all features. Zhang et al. [9] analyzed the Helan mountain wine dataset by using the SVM, logistic regression and NN as prediction models. The result indicated that classification algorithms were feasible for assessing wine quality, and it also showed the SVM performed better compared to other algorithms. Kumar et al. divided the red wine dataset into 70% and 30% for training and testing for evaluating the performances of SVM, RF and Naïve Bayes (NB) [21]. They used accuracy, recall, precision, F1 score and error rate as performance measurements. Based on the results, they suggested that combining and tuning models can provide better performance.

(2) For studies using RF, for instance, Shaw et al. focused on quality prediction performance analysis for the red wine out of three models, including the SVM, RF and NN [13]. They also indicated that the RF outperformed other models. Trivedi et al. firstly normalized the data and removed the outlier from the dataset, and then reduced the classifying labels of a used dataset from 10 classes (1-10) to 2 classes (bad and good). They discovered the accuracies of RF and logistic regression (LR) could achieve 84% and 76% [12]. Hu et al. focused

on handling data imbalance in white wine, by classifying labels to 3 classes that are low quality (3-4), normal (5-7) and high quality (8-9) [14]. They used synthetic minority over-sampling technique (SMOTE) to preprocess imbalance data and apply the processed data into RF, decision tree (DT) and AdaBoost. The experiments showed RF produces the best results in terms of error rates and receiver operating characteristic (ROC) values. Besides, Mahima et al. transformed the labels of the used wine dataset from 10 classes (1-10) to 3 classes, including bad (1-4), average (5-6) and good (7-10), for evaluating the k-NN and RF by the root-mean-square error (RMSE) [16]. They found that employing the most relevant features on both models provided better performance, and observed that the extreme instances could not be classified appropriately. Ozalp et al. applied a fuzzy logic and the random forest to predict the red wine quality using the instances with three labels that are low, medium and high [22]. Sowmya et al. classified the labels into three groups and used both RF and DT for wine quality prediction [20]. The study also used descriptive statistics to explain the association between each wine characteristics and wine quality.

(3) For studies using DT, for instance, Ribeiro et al. used the dataset with 326 samples with the chemical characteristic attributes of wine and subjective attributes from wine taster during the production phase for wine quality prediction by the DT, NN and linear regression [5]. The labels were divided into two classes: medium and good. The results showed that the DT and NN could reach exceedingly high accuracies from 86% to 99%. Appalasamy et al. indicated that the DT performances better than NB [17]. Furthermore, it drilled down the results and found that the accuracy of white wine was affected by a higher number of physicochemistry attributes when comparing to the red wine.

(4) For other studies, for instance, Petropoulos et al. used geographical information to predict the quality of wine grow on different sections in the wine region of Nemea, Greece, using fuzzy logic multi-criteria decision-making system [18]. Andonie et al. used data collected from Cabernet Sauvignon in Washington state with 180 samples for wine quality prediction via classifiers in Weka, including the RF, IBk, multilayer perceptron, KStar, etc [6]. The dataset consists of 32 features, and the six labels. Comparing to other works, it not only focused on finding the best model but also aimed to find a trade-off between the number of used features and accuracy. Bhagyalaxmi et al. proposed a framework by gathering the characteristic of red wine and judging the quality of red wine based on client inclinations [10]. Agrawal et al. used multilayer perceptron model with rectified linear unit for building prediction model, and the best accuracy is 53% for both red and white wines [19].

To summarise, most recent wine quality prediction works used the dataset acquired by Cortez et al. [7], but not all works used both red and white wine dataset for the experimental evaluations. Works in [12] and [21] only used red wine dataset for the experiments. In [12], they discovered RF has a better performance. In [21], they revealed that SVM performs better than RF. Works in [13] and [14] only used white wine dataset for experiments. Both works indicated RF provides better performance on white wine dataset. For works [8], [10], [11], [16], [17], they used both red and white wine datasets for experimental analysis. In [8], [10], they obtained the SVM performs better than other models. [37] [18] indicated that RF is the best among models. In [17], the DT was reported as the most suitable model for wine quality prediction.

B. Classifier

This section briefly describes the classifiers used in this work, including the SVM, random forest, and decision tree.

1. The SVM Classifier

The support vector machine (SVM) is a supervised machine learning model for solving a classification problem [23]. The main concept of SVM is utilized the kernel function to find the hyperplane that can separate instances into categories. As mentioned earlier, SVM [8], [9], [10], [11] have proven to be an effective classifier for wine quality prediction.

There are three hyperparameters in SVM that are the penalty factor C , parameter gamma γ and kernel function *kernel*. The C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the training error. A small value of C tends to emphasize the margin while ignoring the outliers in the training data. A large value of C tends to obtain the best fit for the training data that may cause the overfitting problem. The γ defines the influence degree of a single training. With a small value of γ , the model may not be easy to capture the character of the data. With a large value of γ , the influence area of the support vectors is limited to itself. The final one is *kernel*. There are three types of kernels, including the linear, poly and rbf, can be employed to find the best fit model for the given dataset. Hyperparameter tuning relies more on experimental results than theory, and therefore the best method to determine the optimal settings is by trial and error. By auto fine-tuning the hyperparameters, the SVM can achieve better performance.

2. Random Forest

Random forest (RF) is a supervised learning algorithm, and several studies have shown that using RF can provide a good prediction accuracy [15], [24]. In general, the RF algorithm creates different decision trees using randomly sampled instances. Then, in the prediction phase, based on the prediction results of the trees, a voting technique is used to determine the best solution. Due to using multiple decision trees for prediction, the advantage of RF over other methods is that it can reduce the overfitting. The RF has six hyperparameters: (1) "number of estimators" means the number of trees in the forest, (2) "maximum features" refers to the max number of features considered for splitting a node, (3) "maximum depth" is the maximum number of levels in each decision tree, (4) "minimum samples split" indicates the minimum number of instances placed in a node before splitting the node. (5) "minimum samples leaf" is the minimum number of instances allowed in a leaf node, and (6) "bootstrap" represents a method for sampling instances with or without replacement.

3. Decision Tree

The decision tree (DT) belongs to the supervised learning algorithm. The DT is a tree structure in which each internal node represents a feature, and each leaf node represents a label. The branches represent conjunctions of features that lead to those labels, also known as the decision rules. The main concept behind the DT is to find features which contain the most information. Once the feature is found using the selected criteria, the instances will be split by the feature. The process of finding the feature and split instances is continued until reaching the stopping criterion.

There are many hyperparameters that can be tuned for the DT. In this paper, we focused on six of them as following: (1) "Criterion" represents a function used to measure the quality of a split and could be "gini" for the gini impurity and "entropy" for the information gain, (2) "Splitter" is the strategy used to choose the split at each node. Two options are available. The first one is to choose the best split, and another is to random choose the best split, (3) "Minimum samples split" means the minimum number of samples required to split an internal node, (4) "Minimum samples leaf" is the minimum number of samples required for a leaf node, (5) "Maximum features" indicates the number of features is considered when looking for the best split, and (6) "Maximum depth" is the maximum depth of the tree.

C. The Genetic Algorithms

The basic concept of the genetic algorithms (GA) derived from Charles Darwin's theory of natural evolution and can be used in many fields [23]. For instance, Holland applied GA on adaptive and artificial systems [25]. In GA, each solution is encoded in a string called a chromosome, and could be represented in a binary or decimal form. Two main genetic operators that are crossover and mutation are utilized to generate offspring. The crossover and mutation produce offspring as new possible optimal solutions by swapping or mutating genes of the chromosomes. The fitness function in GA is used to evaluate the fitness of chromosomes in the population. The selection process is employed to generate the next population based on the fitness values of chromosomes. The evolutionary process is continued until reaching the stopping criterion, e.g., reaching a predefined number of generations, obtaining a chromosome with the qualified fitness value. In this study, the GA is utilized to search the suitable set of classifiers and the hyperparameters to form the hybrid model for the different dataset automatically. More detailed explanation of the proposed approach will be stated in the next section.

III. COMPONENTS OF GA-BASED HYBRID MODEL

This section describes the main components associated with the GA-based hybrid quality prediction algorithm. Those components include chromosome encoding, initialization of population, fitness function, and lastly crossover and mutation operations.

1. Chromosome Encoding

This paper aims to find an appropriate set of classifiers and their hyperparameters as the hybrid model for wine quality prediction to fit different wine dataset. Hence, the chromosome consists of two major parts, including the hyperparameter and model parts. The encoding schema for a chromosome C_i is shown in Fig. 1.

From Fig. 1, in the first part, it represents the hyperparameters for the k classifiers, which means k sections should be used. Thus, the length of the first part is the sum of the number of hyperparameters used for every classifier. The second part decides what algorithms are selected for the hybrid model, and every classifier is represented by a bit. If the value of $w^{\text{model},i}$ is 1, it means $w^{\text{model},i}$ is a part of the hybrid model. In the following, take three models, SVM, RF and DT, as an example. Assume the numbers of hyperparameters of the three models are 3, 6, and 6. Therefore, in this case, the chromosome consists of 18 genes. The first 15 genes in the chromosome are used to represent the hyperparameters of three models. The first 3 genes belong to SVM, the 4th to 9th genes belong to RF, and the last 6 genes belong to DT. The last 3 genes in the second part decide which model(s) should be activated. It can be represented as follows:

Chromosome C_i : $[i^{\text{svm}}, i^{\text{rf}}, i^{\text{dt}}, i^w]$, for $i \in P$ is the population.

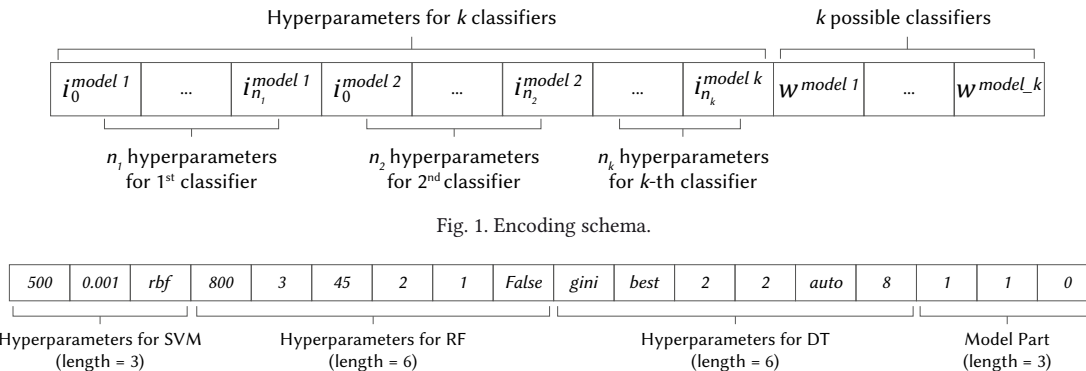


Fig. 2. A possible chromosome.

For i^w , it consists of three genes $[W_s, W_r, W_t]$, where W_s, W_r, W_t are used to represent the voting weight of SVM, RF, and DT. Each model has its own hyperparameters. i^{svm} represents the set of hyperparameters for SVM that are $[C, \gamma, \text{kernel}]$. i^{rf} indicates the set of hyperparameters for the RF that are $[R_o, R_1, R_2, R_3, R_4, R_5]$. i^{dt} represents the set of hyperparameters for DT that are $[D_o, D_1, D_2, D_3, D_4, D_5]$ as described in previous section. Hence, a possible chromosome is shown in Fig. 2.

In Fig. 2, the values of the model part are 1, 1 and 0 that means the SVM and the RF are used as the hybrid model. In accordance with the hyperparameter part, the first three genes, 500, 0.001 and *rbf*, represent values of the penalty factor, gamma and kernel function used for SVM. The 4th to 9th genes, 800, 3, 45, 2, 1 and False, represent values of the number of estimators, maximum features, maximum depth, minimum samples split, minimum samples leaf and bootstrap used for RF. The last six genes, *gini*, *best*, 2, 2, *auto* and 8, in the hyperparameter part represent criterion, splitter, minimum samples split, minimum samples leaf, maximum features and the maximum depth, used for DT.

2. Initialization of Population

Population initialization is the first step in the process of the GA. The population is a set of chromosomes, and the initial population $P(0)$ in this case, is randomly generated. In the previous section, we mentioned that each prediction algorithm has its own set of hyperparameters. For example, SVM consists of three hyperparameters $[C, \gamma, \text{kernel}]$.

Although the suitable setting for the three parameters are $\text{kernel} = [\text{rbf}]$, $C = [9]$, and $\gamma = [2, 0.5, 0.125]$ based on three different datasets (astroparticle, bioinformatic and vehicles) [22], it still cannot guarantee they are suitable for all datasets. Therefore, in order to tune the hyperparameters for every algorithm, based on the chromosome encoding scheme and the population size, the initial population can be generated randomly.

3. Fitness Function

To evaluate the population of genes in the chromosome, the GA requires a fitness function to rank the fitness values of chromosomes based on considering the factors. When designing the fitness function, it should be used to measure how close a chromosome is to the target solution. Designing a useful fitness function is essential to reduce the size of the population and to make the GA more likely to find the optimal solution in less time. In the proposed approach, the average value of the accuracies of active models is employed to calculate the fitness values for a chromosome. Thus, the formula of the fitness function to evaluate a chromosome C_i as $f(C_i)$ is defined as follows:

$$f(C_i) = \frac{\sum_{j=1}^k w^k \cdot \text{acc}(M_j(i^{m^j}))}{\sum_{j=1}^k w^k} \quad (1)$$

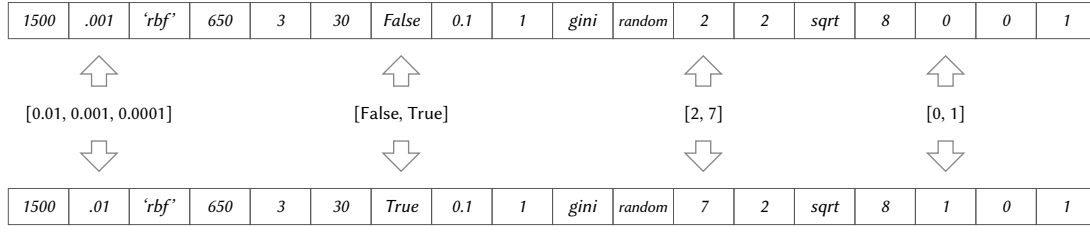


Fig. 3. Illustration of mutation operator.

were M_j is the j -th prediction model adopted in the chromosome C_i . Continue the previous example, when the three prediction models, SVM, RF and DT, are used, the fitness value of a chromosome C_i is calculated as:

$$f(C_i) = \frac{acc(SVM(r^{svm}))w_S + acc(RF(r^f))w_F + acc(DT(r^{dt}))w_T}{w_S + w_F + w_T} \quad (2)$$

where the W_S , W_F , W_T are weights of the three models, and the function $acc()$ is the accuracy function which is utilized to measure the accuracy of a model with an assigned set of parameters. The accuracy is calculated using the formula:

$$acc(M_i) = \frac{\text{Total number of correct prediction by } M_i}{\text{Total number of prediction by } M_i} \quad (3)$$

4. Crossover

In this section, the crossover operation performed in this study is described. Based on the crossover operator used in the steady-state GA (SSGA) [26], we made a slight modification and presented the modified crossover operator, steady-state crossover operator (SSCO), for the proposed approach. The difference between the SSCO and that used in SSGA is the way for the selection of parents for crossover. The pseudocode for SSCO is illustrated in Table II.

TABLE II. PSEUDOCODE FOR THE SSCO

Crossover Procedure: SSCO(P , c_rate , $pSize$)	
Input: population P	
Parameters: crossover rate c_rate and population size $pSize$	
Output: newly generated population P'	
1	Procedure modified_SSGA(){
2	Create new population P'
3	$P' \leftarrow$ add the "elite" chromosome from P
4	Crossover:
5	Select parent C_1 from P randomly
6	Select parent C_2 from P' randomly
7	Offspring $O \leftarrow$ uniformCrossover(C_1 , C_2 , c_rate)
8	If O is better than the worst chromosome from P then
9	Add O to P'
10	If $P' < pSize$
11	GoTo Crossover
12	Else
13	$P \leftarrow P'$
14	}

From Table II, the SSCO first creates the new population P' (line 2). Then, the elite chromosome is picked from the original population P and copied to the new population P' (line 3). After that, two chromosomes C_1 and C_2 are selected from P and P' (lines 5-6). To make the crossover more effective, the uniform crossover operator is employed for gene exchanging (line 7) [27]. It first generates a number of genes to be exchanged according to the given crossover rate, and the exchanging genes follow the randomly generated positions. At last, the new chromosome O is formed based on the exchanging positions and added to P' (line 8). Take C_1 as base chromosome and C_2 as an inserted chromosome as an example. The genes arrangement

for C_2 is $[g_1^2, g_2^2, \dots, g_5^2, \dots, g_8^2, \dots, g_{k-1}^2, g_k^2]$, where * indicates the genes that will be passed to C_1 to form the new offspring. Hence, the new offspring O is generated as: $[p_1^2, p_2^2, \dots, p_5^2, \dots, p_8^2, \dots, p_{k-1}^2, p_k^2]$. The process is continued until $pSize$ chromosomes are generated (lines 9-12). In other words, the benefit is that the best chromosome can be utilized as parents to produce the next generation. In addition, the reason to select only the best chromosome is to keep sufficient diversity and avoid premature convergence.

5. Mutation

In biological evolution, due to genes in chromosome may mutate, it provides offspring has the ability to survive when suffering environment changing. Hence, the aim for mutation is to keep the diversify of the population and to prevent the GA trapped in a local optimal [28]. There are several types of mutation. In this study, based on the uniform mutation [29], the modified uniform mutation is employed to mutate randomly selected gene(s) with a mutation probability p_m . In original uniform mutation, the operator mutates the value of the randomly selected gene with the uniform random value between a specified upper and lower bound. Instead of selecting a value between a specific range, the proposed approach only allows the mutation operator to select a value from the given specified list. Continue the previous example, let the second gene of the SVM, the fourth gene of the RF and third gene of the DT are mutated in the hyperparameter part, and let the first gene of the model part are mutated. Assume the specified lists of those genes are $[0.01, 0.001, 0.0001]$, $[True, False]$, $[2, 7]$, and $[0, 1]$, the mutation operator is illustrated in Fig. 3.

From Fig. 3, the values, including 0.01, 'True', 7, and 1, are selected to replace original genes. Then, the chromosome C_i' is generated. After mutation operation, the offspring C_i' will replace the C_i in the population.

IV. GENERALIZED WINE QUALITY PREDICTION FRAMEWORK AND PROPOSED APPROACH

In this section, the generalized wine quality prediction framework is presented in Section VI.A. Then, based on the framework, the proposed algorithm is described for obtaining the appropriate hybrid model using the GA for wine quality prediction in Section VI.B.

A. Generalized Wine Quality Prediction Framework

As mentioned in the previous section, the existing approaches focus on how to obtain a classifier that can have the best prediction ability on a specific dataset. As to the hyperparameters of the classifier, they can be discovered by different strategies, e.g., the grid search [30], or random search [31]. However, the wine datasets may be collected from different areas and countries, and hyperparameter discovery process may time consuming based on the given searching space. In this paper, we thus propose the generalized wine quality prediction framework for providing a mechanism that can automatically find not only the appropriate hybrid model but also the hyperparameters by the evolution-based algorithms. The framework is shown in Fig. 4.

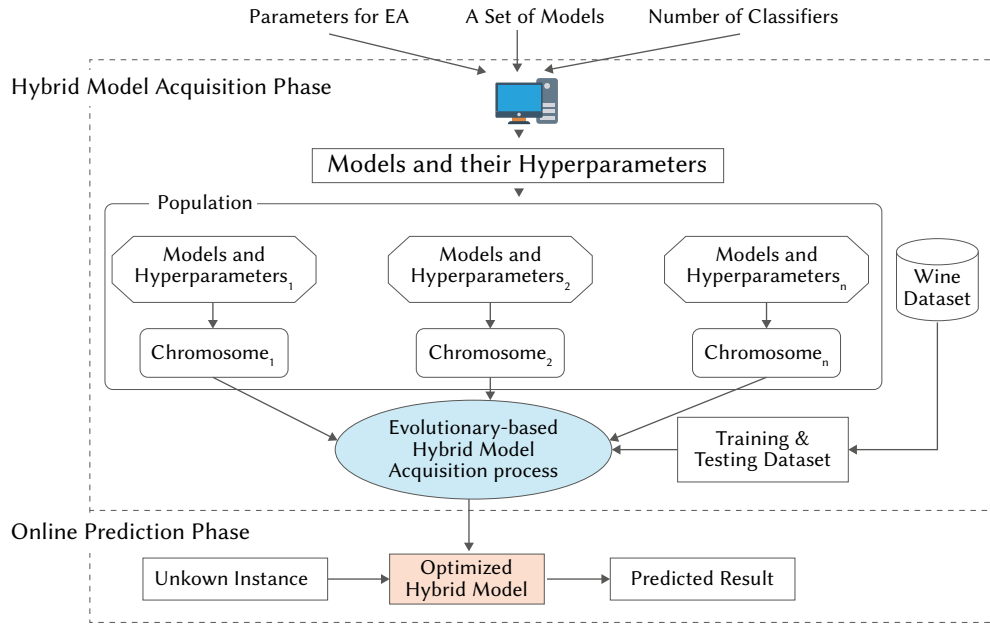


Fig. 4. The generalized wine quality prediction framework.

From Fig. 4, the proposed framework contains two phases, including the hybrid model acquisition and online prediction phases. In the first phase, according to the types of classifiers and their hyperparameters, the population is initialized. The initialized population is then sent into the evolutionary-based hybrid model acquisition module. Operations will generate possible offspring. After generations, the hybrid model with the best fitness value is outputted. Note that any evolutionary-based algorithms can be employed for searching the hybrid model as well as their hyperparameters. In the online prediction phase, the unknown instance can be identified by the optimized hybrid model.

B. GA-based Generalized Wine Quality Prediction Algorithm

Based on the proposed framework, the GA-based generalized wine quality prediction algorithm is stated in this section. The pseudocode of the proposed algorithm is shown in Table III.

In Table III, the proposed approach first divides the dataset into training and testing datasets (line 2). Then, the initial population is generated based on the given set of models M , the number of classifiers num_c , and the hyperparameters HP_M (lines 4~7). The fitness function defined in formula (1) is utilized to evaluate the quality of every chromosome (lines 9~12). Each chromosome represents a model M . Using the given training and testing datasets D_{train} and D_{test} , the fitness value $fValue$ of a chromosome is calculated (line 10). During this step, the encoded model is trained and tested, and a performance score for each model is returned as the fitness value in the end. Then the fitness value of a chromosome is updated the population (line 10). Note that other criteria can also be used as the fitness function, e.g., macro-F1 score. The two genetic operators are performed on the population to generate new offspring (lines 13~14). The newly generated population will replace the previous population (line 15). After reaching the predefined number of generations num_gene , the best chromosome is outputted as the final hybrid model (line 17). The best chromosome consists of hyperparameters for the hybrid model. Because many works reported that the classification techniques, including the SVM, RF and DT are commonly used for wine quality prediction. Therefore, in this paper, three models but not limited to are used as the set of models to construct the hybrid model. Although there are many existing wine quality prediction approaches, they focus only on certain dataset or classifiers. Thus, the proposed algorithm's advantage is that it is a general algorithm and can be employed to find the appropriate

hybrid model and its hyperparameters no matter what kinds of wine datasets are given.

TABLE III. PSEUDOCODE OF THE GA-BASED GENERALIZED WINE QUALITY PREDICTION ALGORITHM

The proposed algorithm: GA_hybrid()	
Input: dataset D , a set of models M , hyperparameter for models HP_M number of classifiers num_c Parameters: population size $pSize$, number of generations num_gene , crossover rate c_rate , mutation rate m_rate Output: The hybrid model and hyperparameters	
1	Procedure GA_hybrid(){
2	$D_{train}, D_{test} \leftarrow \text{trainingandTestingGeneration}(D)$
3	$population \leftarrow \phi$
4	For $i = 0$ to $pSize$
5	$C_i \leftarrow \text{generateModelandHyperpara}(M, num_c, HP_M)$
6	$population \leftarrow population \cup C_i$
7	End For i
8	For $j = 0$ to num_gene
9	For $i = 0$ to $pSize$
10	$fValue_i \leftarrow \text{calculateFitness}(C_i, M, D_{train}, D_{test})$
11	$Population \leftarrow \text{updateFitness}(population, fValue_i)$
12	End For i
13	$newPopulation \leftarrow \text{SSCO}(population, c_rate, pSize)$
14	$newPopulation \leftarrow \text{executeMutation}(newPopulation, m_rate)$
15	$population \leftarrow newPopulation$
16	End For j
17	$bestChromosome = \text{findBestSolution}(population)$
18	Output $bestChromosome$
19	}
20	

V. EXPERIMENTAL RESULTS

A. Dataset Descriptions and Baseline Models

The wine dataset from the UCI database consists of two sets of wine datasets that are red and white wine datasets [7]. The red wine and white datasets contain 1599 and 4898 instances, respectively. Both datasets contain 11 physiochemical variables, including fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, Sulphates, and alcohol. The attribute

“sensory” is a quality rating (class label) which is from 0 (very bad) to 10 (excellent).

The datasets were collected from May 2004 up to February 2007 using the only protected designation of origin samples that were tested at the official certification entity (CVRVV). The CVRVV is an inter-professional organization to improve the quality and marketing of Vinho Verde. The datasets were recorded by a computerized system (iLab), which automatically manages the wine sample testing process from producer requests to laboratory and sensory analysis [1]. The statistical details of the datasets for the physiochemical variables are shown in Table IV.

TABLE IV. THE STATISTICS SUMMARY FOR THE PHYSIOCHEMICAL FEATURES OF THE DATASETS

Attribute	Red Wine			White Wine		
	Min.	Max.	Mean	Min.	Max.	Mean
Fixed acidity	4.6	15.9	8.3	3.8	14.2	6.9
Volatile acidity	0.1	1.6	0.5	0.1	1.1	0.3
Citric acid	0	1	0.3	0	1.7	0.3
Residual sugar	0.9	15.5	2.5	0.6	65.8	6.4
Chlorides	0.01	0.61	0.08	0.01	0.35	0.05
Free sulfur dioxide	1	72	14	2	289	35
Total sulfur dioxide	6	289	46	9	440	138
Density	0.99	1.01	0.996	0.99	1.04	0.994
Sulphates	0.3	2	0.7	0.2	1.1	0.5
pH	2.7	4	3.3	2.7	3.8	3.1
Alcohol	8.4	14.9	10.4	8	14.2	10.4

To show the merits of the proposed approach, we compare the hybrid model against the SVM, RF and DT with the hyperparameters that were discovered using the grid search and random search. The n -fold cross-validation is utilized to construct models. The setting for each model is displayed in Table V.

TABLE V. HYPERPARAMETER SETTING FOR THE BASELINE MODELS

Model	Hyperparameters Setting
SVM	$C = 1500, \gamma = 0.0001,$ $kernel = rbf$
RF	$R_o = 1400, R_1 = 3, R_2 = 20,$ $R_3 = 4, R_4 = 3, R_5 = True$
DT	$D_o = gini, D_1 = 6, D_2 = 5,$ $D_3 = auto, D_4 = 3, D_5 = random$

B. Experimental Setting

In this section, we explain the experimental setting of the proposed algorithm. There is no specific rule to set proper hyperparameters for each model. It is a tedious but crucial task, as the performance of a classifier is highly dependent on the choice of hyperparameters. In order to find the appropriate initial parameter setting for each model, we executed a grid search and random search to find the possible parameters for each classifier. The ranges of hyperparameters for SVM i^{sum} , RF i^f , DT i^{dt} and weight option i^w are shown in Table VI.

Based on the parameters listed in Table VI, the number of chromosomes that can be created is 132,7104. This number is too large and unable to complete the evolution process in a reasonable time. Therefore, the population size of the proposed algorithm was set at 500. Hence, it randomly selected 500 chromosomes to form the initial population. The number of generations was set at 100. The crossover and mutation rates were set at 50% and 1%.

TABLE VI. THE RANGES OF HYPERPARAMETERS FOR THE USED MODELS

Model	Parameters	Ranges	Data types
i^{sum}	C	[500, 1000, 1500]	Integer
	γ	[0.01, 0.001, 0.0001]	Float
	$kernel$	['rbf', 'poly']	String
i^f	R_o	[650, 733, 800]	Integer
	R_1	[3, 'sqrt']	String
	R_2	[30, 45, None]	Integer
	R_3	[0.1, 2]	Float
	R_4	[1, 3]	Integer
	R_5	[False, True]	Boolean
i^{dt}	D_o	['entropy', 'gini']	String
	D_1	['best', 'random']	String
	D_2	[2, 7]	Integer
	D_3	[1, 2]	Integer
	D_4	['sqrt', 'auto']	String
	D_5	[9, 8]	Integer
i^w	w_s	[0, 1]	Integer
	w_F	[0, 1]	Integer
	w_T	[0, 1]	Integer

In the following, the performance measurements of a classifier are described. The accuracy of a classifier is one way to measure how often the algorithm correctly classifies an instance. The formula is shown as follows:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

where TP is the true positive, which means the number of positive instances that are classified to the positive class. TN is the true negative, which means the number of negative instances that are classified to the negative class correctly. FP is false positive, which means the number of negative instances that are classified to the positive class. FN is false negative, which means the number of positive instances that are classified to the negative class.

In the multi-class classification problem, micro and average accuracy, precision, and recall are always the same [32]. Therefore, we use the macro-averaging measurements that are macro-precision and macro-recall for additional measurement reference. Also, based on past work [11], it indicated the wine dataset is imbalanced, only using accuracy may not provide a clear picture. Thus, the macro-F1 score is also utilized for a more detailed comparison. The definition of precision to evaluate a multi-class classifier is shown as follows:

$$P_{Classifier} = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FP_c} \quad (5)$$

where TP_c and FP_c represent the true positive and the false positive for class c . When precision is one, it means the prediction ability of the classifier is perfect. The macro-precision will be lower than average precision. That is because although the model performs exceptionally well on some specific classes, it may perform poorly on some classes, hence downgrading the value of the macro-precision score. The macro-precision is given as follows:

$$Macro-Precision MP_{Classifier} = \frac{\sum_c P_c}{number\ of\ classes} \quad (6)$$

The macro-precision is performed by first computing the precision of every class, and then taking the average of all precisions.

Another metric often used to evaluate performance other than

accuracy is the recall. There is a trade-off between precision and recall. It means higher the recall lower the precision and vice versa. The recall measures the percentage of total relevant results correctly classified by the algorithm. This value is an important indication of how many predictions are correctly predicted. The definition of recall to evaluate a multi-class classifier is shown as follows:

$$\text{recall } R_{\text{classifier}} = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FN_c} \quad (7)$$

where TP_c and FN_c represent the true positive and false negative for class c . When the recall is one, it means that all truly positive samples were predicted as the positive class. Similar to micro-precision, the value will be lower if one class performs poorly. The macro-recall is given as follows:

$$\text{Macro-Recall } MR_{\text{classifier}} = \frac{\sum_c R_c}{\text{number of classes}} \quad (8)$$

Accuracy is useful when the class distribution in the dataset is even, but F1 score is a better metric when the dataset has imbalanced classes. F1 score is simply a harmonic mean between precision and recall. The definition of F1 score to evaluate a multi-higher class classifier is shown as follows:

$$F1_{\text{classifier}} = \frac{2 \times (P_c \times R_c)}{(P_c + R_c)} \quad (9)$$

where P_c and R_c represent the precision and recall for class c . Maximizing the F1 score is like finding the best balancing value between precision and recall. Since we are processing multi-class dataset, we would prefer to use macro-F1 score for comparison. The macro-F1 score calculation is given as follows:

$$\text{Macro-F1 } F1_{\text{classifier}} = \frac{\sum_c F1_c}{\text{number of classes}} \quad (10)$$

There is no defined range of F1 score to determine the performance of the model. We can maximize the macro-F1 score to find the best-balanced value between precision and recall.

C. Experimental Results

Since most of the past works mainly focus on accuracy, we thus compare the accuracy of the proposed approach against others. Also, most works set the training and testing datasets ratio to 80% and 20%. Therefore, we also set the same ratio for the training process. For comparisons, we include three mentioned classification models, the SVM, RF, and DT, as the baseline models. We also compare the proposed approach to the works of Cortez et al. [7] and Appalasamy et al. [17] for performance evaluations. However, both works did not provide enough information to calculate precision and recall. Therefore, the comparison results of baseline and proposed approach in terms of accuracy, precision and recall on the testing datasets are shown in Table VII.

TABLE VII. COMPARISON OF PROPOSED APPROACH WITH DIFFERENT MODELS

Models	Red Wine			White Wine		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
SVM	0.57	0.39	0.37	0.51	0.41	0.35
RF	0.58	0.51	0.36	0.66	0.43	0.41
DT	0.53	0.37	0.34	0.58	0.32	0.32
Cortez et. al [7]	0.45	-	-	0.51	-	-
Appalasamy et. al [17]	0.62	-	-	0.65	-	-
Proposed Approach	0.72	0.34	0.36	0.68	0.57	0.37

From Table VII, we can observe that the accuracies of the proposed approach on red wine and white wine are 72% and 68% that is better than existing approaches. It is also interesting to see that the proposed model has lower precision and recall for red wine than most of the baseline models. For white wine, accuracy and precision are higher than all models, but the recall is slightly lower than the RF. These results indicated that using accuracy as fitness function for finding the hybrid models are good on white wine but a little worse on red wine dataset. To further examine the performances of the proposed approach, different training and testing datasets are used to obtain the hybrid models for red wine and white wine datasets. The results of them are shown in VIII and Table IX.

TABLE VIII. RESULT FOR DIFFERENT TESTING DATA RATIO (RED WINE)

Red Wine	Testing Dataset Size Percentage			
	10%	20%	30%	40%
Accuracy	0.73	0.72	0.69	0.67
Macro-Precision	0.42	0.34	0.34	0.34
Macro-Recall	0.47	0.36	0.36	0.33
Macro-F1 score	0.45	0.35	0.35	0.33

TABLE IX. RESULT FOR DIFFERENT TESTING DATA RATIO (WHITE WINE)

White Wine	Testing Dataset Size Percentage			
	10%	20%	30%	40%
Accuracy	0.70	0.68	0.67	0.66
Macro-Precision	0.64	0.57	0.49	0.52
Macro-Recall	0.62	0.37	0.37	0.36
Macro-F1 score	0.49	0.40	0.40	0.40

From Table VIII, when the testing ratios were set at 10% or 20%, the hybrid models provide the highest accuracies than others, and the accuracies were gradually decreased along with the increasing of ratios. The macro-precision and macro-recall are low and almost similar for red wine dataset. That means the amount of false-positive is very close or equal to the false negative. Besides, the macro-F1 score also dropped when the ratio larger and equal to 20%.

From Table IX, the hybrid model on white wine dataset shows a different result, where the macro-precision is always higher than macro-recall. When the ratio was set at 10%, the accuracy and macro-F1 score are at the highest, and the macro-precision and macro-recall are at the closest. The macro-F1 score is comparative constant with the change of ratio from 20% to 40%. When the macro-F1 score is low, the macro-precision and macro-recall score for red wine indicates the data is highly skewed on certain classes. The white wine dataset is also skewed, but the distribution is more even when comparing to red wine. It is interesting to note that when the testing and training ratio is 10% to 20%, the proposed approach can reach the best performance.

Since the work has proven the datasets for both red and white wine are imbalance [14], it is more reasonable to focus on macro-F1 score instead of accuracy. Macro-F1 score is instrumental in most scenarios when working with imbalanced datasets. Under this condition, we change the fitness function to focus on finding the hyper model that can provide the highest F1 score. The results using the proposed approach with the F1 score as a fitness function for red and white wine datasets are shown in Table X and Table XI.

The result shows that if we focused on improving the macro-F1 score, the accuracies would drop under all conditions. It is also interesting to see both red wine and white wine datasets behave similarly. That is when the increase in the ratios, the macro-F1 score and accuracy are decreasing. The results also show that when the ratio is 10%, the obtained hybrid model has the best performances of 0.59 and 0.58 on red and white wine datasets. Overall speaking, the

proposed approach using macro-F1 is better than that using accuracy as the fitness function.

TABLE X. RESULTS FOR USING F1 SCORE AS FITNESS SCORE (RED WINE)

Red wine	Testing dataset size percentage			
	10%	20%	30%	40%
Macro-F1	0.59	0.46	0.42	0.36
Accuracy	0.65	0.61	0.62	0.57
Macro-Precision	0.57	0.44	0.41	0.36
Macro-Recall	0.63	0.48	0.45	0.35

TABLE XI. RESULTS FOR F1 SCORE AS FITNESS SCORE (WHITE WINE)

White wine	Testing dataset size percentage			
	10%	20%	30%	40%
Macro-F1	0.58	0.41	0.40	0.38
Accuracy	0.65	0.68	0.67	0.65
Macro-Precision	0.66	0.51	0.50	0.52
Macro-Recall	0.55	0.37	0.37	0.35

D. Results of Wilcoxon Signed-Rank and Friedman Tests

We used the Wilcoxon signed-rank test to verify whether the proposed approach is statistically significance at a confidence level at 95%. Since we were unable to retain further experimental data from Cortez et al. [7] and APalasamy et al. [17], we compared the proposed model (P) with the baseline models (SVM, DT and RF). With the accuracies of each model A_{SVM} , A_{DT} , A_{RF} and A_p , the Wilcoxon signed-rank test results on red and white wines against the proposed model are summarized in Table XII and Table XIII.

TABLE XII. THE RESULTS OF THE WILCOXON SIGNED-RANK TEST (RED WINE)

	SVM- P	DT- P	RF- P
H_0 : Null hypothesis	$H_0: A_{SVM} = A_p$	$H_0: A_{DT} = A_p$	$H_0: A_{RF} = A_p$
H_a : Alternative hypothesis	$H_a: A_{SVM} < A_p$	$H_a: A_{DT} < A_p$	$H_a: A_{RF} < A_p$
z -value (two tail)	-2.0896	-1.9604	-1.6803
p -value (two tail)	0.0362	N/A	N/A
T_{wilcox}	$T_{wilcox}(10) = 6.5$	$T_{wilcox}(8) = 4$	$T_{wilcox}(8) = 6$

According to the Wilcoxon test for red wine, the p -value for the SVM- P pair is smaller than the threshold value of 0.05. Therefore, the null hypothesis is rejected. In addition, for SVM- P , the $T_{wilcox}(10)$ is 6.5 which is smaller than the critical value for Wilcoxon at $N = 10$ ($p < .05$) is 8. Since both p -value and T_{wilcox} all below the threshold, the null hypothesis can be rejected. That indicates the proposed approach is significantly better than the SVM. However, DT- P and RF- P pairs show different results, because the the Wilcoxon test data size N is 8, which is not large enough for the distribution of the Wilcoxon statistic to form a normal distribution. Therefore, it is not possible to calculate accurate p -value.

TABLE XIII. THE RESULTS OF THE WILCOXON SIGNED-RANK TEST (WHITE WINE)

	SVM- P	DT- P	RF- P
H_0 : Null hypothesis	$H_0: A_{SVM} = A_p$	$H_0: A_{DT} = A_p$	$H_0: A_{RF} = A_p$
H_a : Alternative hypothesis	$H_a: A_{SVM} < A_p$	$H_a: A_{DT} < A_p$	$H_a: A_{RF} < A_p$
z -value (two tail)	-2.8031	-2.8031	-2.8031
p -value (two tail)	0.00512	0.00512	0.00512
T_{wilcox}	$T_{wilcox}(10) = 0$	$T_{wilcox}(10) = 0$	$T_{wilcox}(10) = 0$

For white wine, it is interesting to see all null hypothesis can be rejected since $T_{wilcox}(10) = 0$ for all three sets of experiments. In short,

given the accuracy for each method on the same dataset, the proposed model performed better than all other models for white wine dataset. However, for red wine, the proposed model performed better than SVM, but cannot make a conclusion for DT and RF for red wine dataset. We can only conclude that the data size for red wine used for each test is small and does not provide enough information to make an effective conclusion.

A Friedman test was then conducted on ten runs for red and white wines to examine performances (accuracies) of the four different models on 10 datasets. The Friedman test is a non-parametric equivalent of the repeated measures ANOVA [33]. Results showed that different red wine models produce statistically significant differences in terms of accuracy with $Q = 149.64$ and $p < 0.000001$. For white wine, it also showed that different models also perform statistically significant difference in terms of accuracy with $Q = 168.48$ and $p < 0.000001$.

E. Discussion

There are some works in the recent two years that also conduct experiments on the same wine dataset. However, those studies divided the instances into two labels [12] or three labels, such as [34], [16], [22] and [20] for building models. Therefore, it makes the comparison slightly unfair due to the different standard. Other studies like [19], [15], [13] and [21] either have lower results than Appalasamy's work [17]. In this paper, we thus compared the proposed method with the models proposed by Cortez et al. [7] and Appalasamy et al. [17] because they provided detailed description of each evaluation measurement matrices.

Evolutionary Algorithms (EA) refers to a set of biologically-inspired algorithms, for example, the Genetic Algorithms (GA), the Particle Swarm Optimization (PSO) [35], etc. GA is a stochastic search method that mimics the metaphor of natural biological evolution. For the PSO, it is inspired by the social behaviours of animals, and by updating the position and velocity of each individual to find solutions. Recently, PSO gained some attention in the field of the next-generation wireless network [36].

The differences between the GA and PSO are stated as follows. Based on [37], the PSO performs better in terms of the computational efficiency than the GA for solving the unconstrained non-linear problems with continuous design variables. However, the GA performs better when applied to the constrained non-linear problems with continuous or discrete design variables. For the problem to be solved in this paper, variables are constrained, non-linear and discrete. Therefore, the GA is adopted to deal with the hybrid model optimization algorithm. However, if the problem can be mapped to the unconstrained non-linear problems, the PSO will be a good methodology to be employed for searching the solution. In the future, we will continue to enhance the framework and to try and design different approaches to tune the performances.

VI. CONCLUSION AND FUTURE WORK

In this paper, unlike most past works focusing on which classification model provides the best performance in predicting wine quality. Instead, we have proposed a generalized wine quality framework which consists of the hybrid model acquisition and online prediction phases. Based on the framework, the GA-based generalized wine quality prediction algorithm has been proposed. The proposed approach first encodes a set of classifiers and hyperparameters into a chromosome. The fitness functions including the accuracy and macro-F1 score are employed to evaluate the goodness of every chromosome. The steady-state crossover operator and uniform operator are applied on the population to generate new offspring.

After the evolution process, the appropriate hybrid model and the hyperparameters are used for wine quality prediction. Experiments on the red and white wine datasets indicate that the proposed approach is better than other existing approaches in terms of accuracy. In addition, when using macro-F1 score as the fitness function, although the accuracy of the hybrid model is decreasing, the macro-F1 score, macro-precision and macro-recall are increasing. In the future, under the proposed framework, other types of evolutionary algorithms can be employed to get a more solid classifier. In addition, more classifiers or other ML approach like different neural networks [38] can also be considered to construct the hybrid model.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Science and Technology of the Republic of China under grants MOST 108-2221-E-032-037 and 109-2622-E-027-032.

REFERENCES

- [1] B. C. Smith, "Getting more out of wine: Wine experts, wine apps and sensory science," *Current Opinion in Food Science*, vol. 27, pp. 123-129, 2019.
- [2] J. M. Cardebat and F. Livat, "Wine expert's rating: A matter of taste?," *International Journal of Wine Business Research*, Vol. 28, pp. 43-58, 2016.
- [3] B. V. Canizo, L. B. Escudero, R. G. Pellerano, and R. G. Wuilloud, "10 - quality monitoring and authenticity assessment of wines: Analytical and chemometric methods," in *Quality control in the beverage industry*, A. M. Grumezescu and A. M. Holban, Eds.: Academic Press, pp. 335-384, 2019.
- [4] M. Yeo, T. Fletcher, and J. Shawe-Taylor, "Machine learning in fine wine price prediction," *Journal of Wine Economics*, Vol. 10, No. 2, pp. 151-172, 2015.
- [5] J. Ribeiro, J. Neves, J. Sanchez, M. Delgado, J. Machado and P. Novais, "Wine vinification prediction using data mining tools," *International conference on European computing conference*, pp. 78-85, 2009.
- [6] R. Andonie, A. M. Johansen, A. L. Mumma, H. C. Pinkart, and S. Vajda, "Cost efficient prediction of cabernet sauvignon wine quality," *IEEE Symposium Series on Computational Intelligence*, pp. 1-8, 2016.
- [7] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems*, Vol. 47, no. 4, pp. 547-553, 2009.
- [8] Y. Gupta, "Selection of important features and predicting wine quality using machine learning techniques," *Procedia Computer Science*, Vol. 125, pp. 305-312, 2018.
- [9] Z. Lingfeng, F. Feng, and H. Heng, "Wine quality identification based on data mining research," *International Conference on Computer Science and Education*, pp. 358-3612, 2017.
- [10] S. Bhattacharjee and M. R. Chaudhuri, "Understanding quality of wine products using support vector machine in data mining," *Prestige International Journal of Management & IT- Sanchayan*, Vol. 5, no. 1, pp. 67-80, 2016.
- [11] Y. Er and A. Atasoy, "The classification of white wine and red wine according to their physicochemical qualities," *International Journal of Intelligent Systems and Applications in Engineering*, pp. 23-23, 2016.
- [12] A. Trivedi and R. Sehrawat, "Wine quality detection through machine learning algorithms," *International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering*, pp. 1756-1760, 2018.
- [13] B. Shaw, A. K. Suman, and B. Chakraborty, "Wine quality analysis using machine learning," *Emerging technology in modelling and graphics: Springer*, pp. 239-247, 2020.
- [14] G. Hu, T. Xi, R. Mohammed, and H. Miao, "Classification of wine quality with imbalanced data," *IEEE International Conference on Industrial Technology*, pp. 1712-1217, 2016.
- [15] S. Aich, A. A. Al-Absi, K. L. Hui, J. T. Lee, and M. Sain, "A classification approach with different feature sets to predict the quality of different types of wine using machine learning techniques," *International Conference on Advanced Communication Technology*, pp. 139-143, 2018.
- [16] G. U. Mahima, Patidar Y., Agarwal A., Singh K.P., "Wine quality analysis using machine learning algorithms," *Micro-Electronics and Telecommunication Engineering: Proceedings of 3rd ICMETE 2019*, pp. 11-18, 2020.
- [17] P. Appalasamy, A. Mustapha, N. Rizal, F. Johari, and A. Mansor, "Classification-based data mining approach for quality control in wine production," *Journal of Applied Sciences*, Vol. 12, pp. 598-601, 2012.
- [18] S. Petropoulos, C. S. Karavas, A. T. Balafoutis, I. Paraskevopoulos, S. Kallithraka, and Y. Kotseridis, "Fuzzy logic tool for wine quality classification," *Computers and Electronics in Agriculture*, Vol. 142, pp. 552-562, 2017.
- [19] G. Agrawal and D.-K. Kang, "Wine quality classification with multilayer perceptron," (in En), *International Journal of Internet, Broadcasting and Communication*, Vol. 10, no. 2, pp. 25-30, 2018.
- [20] D. Sowmya, Sayyed Johar, M. Ganavi, and N. S. Nayak, "Analyzing wine types and quality using machine learning techniques," *International Journal of Engineering Applied Sciences and Technology*, Vol. 4, no. 5, pp. 519-529, 2019.
- [21] S. Kumar, K. Agrawal, and N. Mandan, "Red wine quality prediction using machine learning techniques," *International Conference on Computer Communication and Informatics*, pp. 1-6, 2020.
- [22] A. E. Ozalp and I. Askerzade, "A data science study for determining food quality: An application to wine," *Communications*, Vol. 68, No. 1, pp. 762-770, 2018.
- [23] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions On Neural Networks*, Vol. 10, No. 5, pp. 988-999, 1999.
- [24] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R news*, Vol. 2, pp. 18-22, 2002.
- [25] J. H. Holland, "Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence". *MIT press*, 1992.
- [26] A. Rogers and A. Prügel-Bennett, "Modelling the dynamics of a steady-state genetic algorithm," *Foundations of genetic algorithms*, Vol. 5, pp. 57-68, 1999.
- [27] W. M. Spears and K. D. De Jong, "On the virtues of parameterized uniform crossover," *Naval Research Lab Washington DC*, 1995.
- [28] I. Korejo, S. Yang, K. Brohi, and Z. Khuuro, "Multi-population methods with adaptive mutation for multi-modal optimization problems," *International Journal on Soft Computing, Artificial Intelligence and Application*, Vol. 2, pp. 1-19, 2013.
- [29] N. Soni and T. Kumar, "Study of various mutation operators in genetic algorithms," *International Journal of Computer Science and Information Technologies*, Vol. 5, no. 3, pp. 4519-4521, 2014.
- [30] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," *National Taiwan University*, 2003.
- [31] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, Vol. 13, pp. 281-305, 2012.
- [32] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, Vol. 45, No. 4, pp. 427-437, 2009.
- [33] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, Vol. 7, pp. 1-30, 2006.
- [34] D. Bhagyalaxmi, R. Manjula, and V. Ramanababu, "A new framework approach to predict the wine dataset using cluster algorithm," *International Conference on Computational Intelligence and Informatics*, pp. 39-48, 2020.
- [35] H. Al-Sahaf, Y. Bi, Q. Chen and A. Lensen, "A survey on evolutionary machine learning," *Journal of the Royal Society of New Zealand*, Vol. 49, No. 2, pp. 205-228, 2019.
- [36] J. C. W. Lin, G. Srivastava, Y. Zhang, Y. Djenouri, and M. Aloqaily, "Privacy preserving multi-objective sanitization model in 6g iot environments," *IEEE Internet of Things Journal*, pp. 1-1, 2020.
- [37] R. Hassan, B. Cohanin, O. De Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," *AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference*, AIAA 2005- 1897, 2005.
- [38] J. C.-W. Lin, Y. Shao, Y. Djenouri, and U. Yun, "ASRNN: A recurrent neural network with an attention model for sequence labeling," *Knowledge-Based Systems*, Vol. 212, 106548, 2020.



Terry Hui-Ye Chiu

Terry Hui-Ye Chiu received her B.S. in Information Technology and M.S. in Computer Science from the University of Auckland, New Zealand, in 2001. She is pursuing a Ph.D degree in the Department of Information and Finance Management at National Taipei University of Technology, Taipei, Taiwan. Her research interests include machine learning, data mining, financial technology

application.



Chienwen Wu received

Chienwen Wu received his PhD degree in Computer Science from the University of Illinois at Urbana-Champaign in Illinois. Currently, Chienwen Wu is an Associate Professor of Information and Finance Management at National Taipei University of Technology. His research areas include warehouse management, combinatorial optimization, meta-heuristics and machine learning.



Chun-Hao Chen

Chun-Hao Chen is an associate professor at Department of Information and Finance Management at National Taipei University of Technology, Taipei, Taiwan. Dr. Chen received his Ph.D. degree with major in computer science and information engineering from National Cheng Kung University, Taiwan, in 2008. He has a wide variety of research interests covering data mining, time series, machine learning, evolutionary algorithms, and fuzzy theory. Research topics cover portfolio selection, trading strategy, business data analysis, time series pattern discovery, etc. He serves as the associate editor of the International Journal of Data Science and Pattern Recognition, and IEEE Access.