Separability-based Quadratic Feature Transformation to Improve Classification Performance

Usman Sudibyo, Supriadi Rustad, Pulung Nurtantio Andono, Ahmad Zainul Fanani
Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia

***Abstract*—Feature transformation is an essential part of data preprocessing to improve the predictive performance of machine learning (ML) algorithms. Box-Cox transformation with the goal of separability is proven to align with the performance improvement of ML algorithms. However, the features mapped using Box-Cox transformation preserve the order of the data, so it is ineffective when used to improve the separability of multimodal distributed features. This research aims to build a feature transformation method using quadratic functions to improve class separability that can adaptively change the order of the data when necessary. Fisher score (Fs) measures the separability level by maximizing the Fisher's Criteria of the quadratic function. In addition to increasing the Fs value of each feature, this method can also make the feature more informative, as evidenced by the increasing value of information gain, information gain ratio, Gini decrease, ANOVA, Chi-Square, reliefF, and FCBF. The increase in Fs is particularly significant for bimodally distributed features. Experiments were conducted on 11 public datasets with two statistical-based machine learning algorithms representing linear and nonlinear ML algorithms to validate the success of this method, namely LDA and QDA. The experimental results show an improvement in accuracy in almost all datasets and ML algorithms, where the highest accuracy improvement is 0.268 for LDA and 0.188 for QDA.**

***Keywords*—Separability; feature transformation; quadratic function; fisher's criterion; fisher score**

I. INTRODUCTION

Machine Learning (ML) is an essential branch of artificial intelligence widely used for pattern recognition, image processing, text classification, intrusion detection systems, etc. [1]. However, the ML algorithm's success depends on the features' quality. The resulting model is also good if the features are good [2]. Therefore, improving features to suit ML algorithms' needs is an essential topic in feature engineering [3]. Feature transformation is one of the feature engineering techniques that can be used to improve features before being input into the ML algorithm [2], [4]. There have been many studies that discuss feature improvement through transformation techniques, where with the proper feature transformation, feature quality can be improved [5], [6].

Feature transformations developed to improve feature quality are generally grouped into three: first, feature transformations that only change the scale (e.g., Min-Max normalization, Z-Score normalization) [7], second nonlinear feature transformations that do not change the order of the data (e.g. log transform [8], square root transform [9], Box-cox transform, Yeo-Johnson transform [10]), and third, nonlinear feature transformations that can change the order of the data (e.g. kernel function in SVM) [11], [12]. Some machine learning algorithms are sensitive to scale differences, so normalization or standardization is needed to uniform the scale of features. Paper in [13] discusses the effect of various data normalization methods on support vector machine (SVM) algorithms and technical indicators to predict stock index price movements. The result is a slight increase in accuracy performance. Paper in [6] proposed mixed feature transformation methods such as CDF transformation and Symmetric log1p transformation, where feature transformation can substantially improve the performance of neural ranking models compared to directly using raw features. Paper in [14] compares the effect of Box-Cox transformation to improve two-dimensional images with advanced low-light image enhancement techniques. Paper in [15] addresses issues in nonlinear stochastic degradation modeling and prognostics from the Box-Cox transform (BCT) perspective, where BCT is used to transform nonlinear degradation data into near-linear data. Adaptive Box-Cox (ABC) transformation was introduced by [16], where adaptive parameter tuning is used to normalize data in various distributions that cannot be properly normalized using conventional data transformation algorithms, including log and square root transformations.

In general, feature transformation aims to change the data distribution to be close to Gaussian in order to improve ML performance, such as log transformation [8], [17], square root transformation [18], Box-Cox transformation [14], [19]–[23], and Yeo-Johnson transformation [10], [24]. However, the experimental results of Bicego & Baldo, 2016 [20] showed different results. Their findings show that ML classification accuracy improves when the data distribution is far from Gaussian and is more related to the class separability problem. This finding is corroborated by [21], [25]–[27], which state that, based on the fisher criterion, features with greater class separability are considered more informative and can improve ML classification performance. Based on these findings, ML classification performance can be improved when features have large separability. This transformation can improve class separability, even in cases where the original dataset is not linearly separable.

Bicego and Baldo's research above uses the Box-Cox transformation, where this transformation is monotonous because it cannot change the order of the data [20]. This condition causes the Box-Cox transformation not to produce maximum separability. In addition, the result of the Box-Cox transformation is determined by a parameter that is searched using the grid search method so that getting the best parameter of each feature associated with the maximum fisher value

requires high computational costs [21]. Bicego and Baldo's empirical analysis of the behavior of the Box-Cox transform for pattern classification opens up opportunities for the analysis of different nonlinear data pre-processing methods that can improve class separability.

This research proposes a quadratic feature transformation for preprocessing that is directly designed to maximize the class separability of each feature. The idea is to optimize the quadratic function parameters using Fisher's optimization criterion to obtain maximum class separability. In this way, each feature is transformed using a quadratic function to have a higher fisher score compared to the original feature's fisher score. The quadratic transformation process is performed at the preprocessing stage, making it flexible to be combined with various ML algorithms. Since this technique is directly geared towards maximizing the fisher score, while the mean and variance of each class can be easily calculated, it has the potential to be applied to multi-class data. Various feature quality test metrics, such as Information Gain [28]–[30], Gain ratio [31], Gini Decrease [32], Anova [33], [34], Chi-Square [35], ReliefF [36], [37], and Fast Correlation-Based Feature selection (FCBF) [38], [39], are used to test the feature quality of the proposed Box-Cox transformation and Quadratic transformation, before finally comparing their respective performance.

The classification algorithms used in this study are Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA), which represent linear and nonlinear ML algorithms. Although LDA has proven to be an excellent classification and dimensionality reduction algorithm, it produces poor vector projections on multimodal data [40]–[42]. With feature transformation, it is expected that the multimodal influence on LDA can be reduced. QDA was chosen because it is one of the most commonly used classifiers in practice and is quite simple. In addition, QDA has been shown to improve performance when paired with the nonlinear feature extraction technique quadratic Fisher transformation [43].

The contributions of this research include:

1) Development of separability-based feature transformation to optimize the classification task of machine learning algorithms.

2) The chosen quadratic transformation is generally able to improve feature quality based on fisher score, information gain, chi-square, relief, and FCBC values on the dataset studied.

This paper is organized into several sections, starting with an introduction in Section I, followed by a brief explanation of fisher score and quadratic function in Section II, research methodology containing the dataset and the proposed method in Section III, results and discussion in Section IV, and then closed with a conclusion in Section V.

II. Preliminary Works

To prepare for a better understanding of this research, some feature transformation techniques, fisher score, and quadratic function definition and application will be introduced.

A. Feature Transformation Technique

In some literature, the use of the term feature transformation is often equated with feature engineering, feature extraction, and feature construction [44]. However, this study consistently uses the term feature transformation as a univariate feature engineering technique.

Feature transformations, which T. Verdonck et.all 2021[2] call feature engineering, are grouped into two, namely univariate and multivariate feature engineering techniques. Univariate feature transformations on continuous variables can improve symmetry, normality, or model fit, such as logarithmic transformation, Box-Cox transformation, and Yeo-Johnson transformation. Multivariate feature transformations aim to reduce the dimensionality of the data by creating new features that are linear combinations of the original variables, such as PCA, LDA, SVD, UV (non-negative) decomposition, and tensor decomposition.

The study of feature transformation has progressed quite well. The following studies are related to feature transformation. A feature transformation method based on Mutual Information (MI) is proposed by [45], where the Probability Density Function (PDF) of features in the class is assumed to be Gaussian. The gradient descent technique is used to maximize the mutual information between features and classes. Experimental results show that the proposed MI projection consistently outperforms other methods for various cases. Most of these Medical decision support systems (MDSS) focus on feature transformation-based methods and their integration with machine learning models for the prediction of risks associated with Heart failure (HF). However, the improvement in accuracy on test data is not followed by training data. This study proposes a more robust approach that integrates stacked autoencoder grids with neural network models to address the problem[46]. Most feature engineering in the input space relies on manually defined transformation functions. However, research [12] builds transformation functions automatically learned through autoencoders for latent representation extraction and multi-layer perceptron (MLP) regressors. The transformation function built in this way can also improve the performance of LSVM and JST, when embedded as a preprocessing step. A random projection-based feature transformation method using Metaheuristic Optimization Algorithm [47] is proposed to map data points from the original space to a new binary space, where the random projection process is formulated as an optimization problem. The transformation of features to binary space is needed when the system requires coarse quantization of measurements. A new guided FT method called minimax probabilistic feature transformation (MPFT) was proposed for multi-class datasets [48]. The idea of this method is based on trying to control the probability of correct classification of future test points as large as possible in the transformed feature space. Past tax default prediction by applying diverse feature transformation techniques and advanced machine learning approaches was proposed by [49]. A combination of feature transformations, such as logarithmic and square root transformations, is able to improve tax default prediction performance. A feature transformation method to improve classification performance referred to as weight-matrix

learning (WML), was proposed by[50]. The way this method works is that WML is identified as an off-center technique with a center of 0.5 similarity.

B. Fisher Score (Fs)

Fs belongs to the classical supervised feature selection filter method, which aims to score features based on the ratio of data scatter between classes and data scatter within classes. Fs is used to measure the class discriminant properties of each feature independently. Features with higher Fisher scores are more discriminant than features with lower scores [51]. Fisher score has been widely used for feature selection on gene microarray data [52]–[57]. This study uses the fisher score as a basis for improving separability because it is conceptually easy to understand, easy to implement on various functions, and Fs is an efficient approach to data dimensionality reduction [58]. Fisher score Fs(k) is used to measure the separability of classes at the kth feature of a dataset. Mathematically, Fs(k) is calculated using Eq. (1)[52].

$$Fs(k) = \frac{\sum_{i=1}^{c} n_i (\mu_i^k - \mu^k)^2}{\sum_{i=1}^{c} n_i (\sigma_i^k)^2} \tag{1}$$

$$\text{with} \quad \mu_i^k = \sum_{j=1}^{n_i} \frac{x_{ij}^k}{n_i} \quad ; \quad \mu^k = \sum_{j=1}^{N} \frac{x_j^k}{N}$$

$$\sigma_i^k = \sum_{j=1}^{n_i} \frac{(x_{ij}^k - \mu_i^k)^2}{n_i - 1} \tag{2}$$

where c, N, n_i, μ_i^k, μ^k, and σ_i^k respectively are the number of classes, the total number of samples, the number of samples of the i-th class, the mean value of the k-th feature of the i-th class, the mean value of the k-th feature for the whole class, and the variance of the k-th feature of the i-th class.

C. Quadratic Function

Throughout the literature review, no quadratic functions were found to be used for the purpose of increasing the separability of classes in features. However, there are many uses of quadratic functions for different purposes. Quadratic kernel-free non-linear support vector machine (QSVM) uses quadratic functions as decision boundaries that are able to separate data in a non-linear manner. The decision boundary is built from a multivariate quadratic function that can replace the kernel trick in SVM when faced with problems that cannot be separated linearly [59]. The QSVM method was successfully used for credit scoring models [60] and improved accuracy and efficiency. This method, called Quadratic Fisher Discriminant Analysis (QFDA), uses linear and quadratic basis functions to improve classification accuracy by considering data variance. This method aims to maximize the fisher criterion in the transformation space using a transformation matrix[61]. Before the transformation, each feature is squared, and the features are multiplied, resulting in a significant increase in the number of features and high computational cost. One disadvantage of the QFDA method is that it may only work well if the class mean values are equal or if the vital information for classification lies in the variance of the data rather than the mean value.

From a mathematical point of view, a quadratic function is a polynomial of degree 2. Its highest exponent on the independent variable (i.e., x) is 2. The general form of the parabolic quadratic function, as shown in Eq. (5), is defined as follows [62].

Definition (General form). For fixed constants b, c ∈ R and nonzero a ∈ R, the function f : R → R given by

$$f(x) = ax^2 + bx + c \tag{3}$$

is a (real) quadratic function written in general form.

Quadratic functions are capable of transforming data to be closer for data in the same group and further away for data from different groups. This ability is the basis for using quadratic functions for feature transformation.

III. Research Methodology

A. Datasets

This study used 11 datasets, which are presented in Table I. Ten datasets were downloaded from the Keel data repository (https://sci2s.ugr.es/keel/category.php?cat=clas), including Sonar, WDBC, Ringnorm, New Thyroid, Wisconsin, Parkinson's, Splice, Balance, Spectf, and Hayes Roth. One Hungarian dataset was downloaded from https://www.openml.org/. Table I consists of five columns containing dataset information (code, dataset, features, samples, and class) and one column containing different methods using the same data. The number of features is between 4 and 60, the sample size is between 160 and 7400, and the targets consist of two classes and three classes. The data type varies from numeric, categorical, binary, and there are no missing values in all datasets. The data information is summarized in Table I.

TABLE I. Profile Dataset and Accuracy Performance of State-of-the-Art Methods

Code	Dataset	Features	Samples	Class	Method
d-1	SONAR	60	208	2	2DCSCA[47]
d-2	WDBC	30	569	2	Hybrid GPFC[63]
d-3	Ringnorm	20	7400	2	EIDA[64]
d-4	New Thyroid	5	215	2	Hybrid GPFC[63]
d-5	Wisconsin	9	683	2	SSDA[65]
d-6	Parkinson	22	195	2	2DCSCA[47]
d-7	Hungarian	11	1190	2	RFSA -MCC[51]
d-8	Splice	60	3190	2	RMB-SSVM[66]
d-9	Balance	4	625	3	GPMO[67]
d-10	Spectf	44	267	2	RFSA -MCC[51]
d-11	Hayes-Roth	4	160	3	SSDA[65]

B. Proposed Method

The proposed method can find the best parameters of the quadratic function in Eq. (3) generated by maximizing the Fisher score [68], [69], which is the ratio of variance between classes and variance within classes. The result of maximizing the Fisher score is a closed-form solution of the quadratic function parameters described in Section III (B) (1), while the feature transformation procedure is described in Section III (B) (2).

1) Separability-based quadratic feature transformation: Based on the Fisher Score function of Eq. (1) and the quadratic function of Eq. (3), Eq. (4) is generated as the basis for obtaining the best parameters.

$$Fs(k) = \frac{\sum_{i=1}^{c} n_i^k \left[a(\theta_i^k - \theta^k) + b(\mu_i^k - \mu^k) \right]^2}{\sum_{i=1}^{c} \left[n_i^k \sum_{j=1}^{n_i^k} \frac{(a((x_{ij}^k)^2 - \theta_i^k) + b(x_{ij}^k - \mu_i^k))^2}{n_i^k - 1} \right]} \quad (4)$$

The Mean square of the k-th feature, i-th class.

$$\theta_i^k = \sum_{j=1}^{n_i^k} \frac{(x_{ij}^k)^2}{n_i^k} \quad (5)$$

The Mean square of the k-th feature

$$\theta^k = \sum_{j=1}^{N} \frac{(x_j^k)^2}{N} \quad (6)$$

Arg_max(Fs(k)) Eq. (4) yields the optimal parameters a, b, and c formulated in Eq. (7) to Eq. (9).

$$a^k = 2R^k \quad (7)$$

$$b^k = -Q^k \pm \sqrt{(Q^k)^2 - 4R^k P^k} \quad (8)$$

$$c^k = 0 \quad (9)$$

Where is

$$A_i^k = (\theta_i^k - \theta^k); B_i^k = (\mu_i^k - \mu^k) \quad (10)$$

$$P^k = \sum_{i=1}^{c} A_i^k B_i^k \sum_{i=1}^{c} v((x_i^k)^2) - \sum_{i=1}^{c} A_i^2 \sum_{i=1}^{c} c(x_i^k, (x_i^k)^2) \quad (11)$$

$$Q^k = \sum_{i=1}^{c} (B_i^k)^2 \sum_{i=1}^{c} v((x_i^k)^2) - \sum_{i=1}^{c} (A_i^k)^2 \sum_{i=1}^{c} v(x_i^k) \quad (12)$$

$$R^k = \sum_{i=1}^{c} (B_i^k)^2 \sum_{i=1}^{c} c(x_i^k, (x_i^k)^2) - \sum_{i=1}^{c} (A_i^k B_i^k) \sum_{i=1}^{c} v(x_i^k) \quad (13)$$

Variance of the k-th feature of i-th class

$$v(x_i^k) = \frac{\sum_{h=1}^{n_i^k} (x_{h,i}^k - \mu_i^k)^2}{n_i^k - 1} \quad (14)$$

The Squared variance of kth feature, i-th class

$$v((x_i^k)^2) = \frac{\sum_{h=1}^{n_i^k} ((x_{h,i}^k)^2 - \theta_i^k)^2}{n_i^k - 1} \quad (15)$$

Covariance between x_i^k and $(x_i^k)^2$ of the kth feature of the i-th class

$$c(x_i^k, (x_i^k)^2) = \frac{\sum_{h=1}^{n_i^k} (x_{h,i}^k - \mu_i^k)((x_{h,i}^k)^2 - \theta_i^k)^2}{n_i^k - 1} \quad (16)$$

2) Transformation procedure: Each feature is separately parameterized using Eq. (7) to Eq. (9). The following procedure is used for feature transformation:

a) Select the kth feature for which parameters are to be calculated.

b) Split the feature into training and testing data.

c) Square each element of the feature (x^k) and add it as a new feature (x^k)^2.

d) Using the training data, calculate the mean of the kth feature of the i-th class, the average of the kth feature, the variance of the kth feature of the i-th class, the squared variance of the kth feature of the i-th class, the covariance between x_i^k and $(x_i^k)^2$ of the kth feature of the i-th class, respectively, using Eq. (2), (14), (15), and (16) to obtain the optimal parameters, b, and c in Eq. (7) to Eq. (9).

e) Transform each feature element x_i^k , both training and testing, using Eq. (3).

C. Experimental Design

All 11 datasets were preprocessed, which included converting categorical data types to numerical and standardizing the datasets in the range between 1 and 2 [19]. Fig. 1 presents the experimental design where the data is split into training and testing using the Cross-validation technique with k = 10. The training data was used to obtain the quadratic function parameters as described in detail in Section III (B) (2), and the Box-Cox transformation parameter λ in the range of -5 to +5 as in Bicego and Baldo's study, which chose the best λ based on the highest fisher score. These parameters are used to transform training and testing data features based on quadratic and Box-Cox functions. The quality of the transformed training data features was measured using the Fisher score, Information gain, Gain ratio, Gini Decrease, Anova, Chi-square, ReliefF, and FCBF to ensure they were good inputs for training the two statistical machine learning algorithms LDA [70] and QDA[71]. The resulting ML model was tested with the transformed testing data to evaluate model performance. Model performance is measured by the accuracy (Acc) metric using Eq. (17).

$$Acc = \left(\frac{Tp + Tn}{Tp + Tn + Fp + Fn} \right) * 100 \quad (17)$$

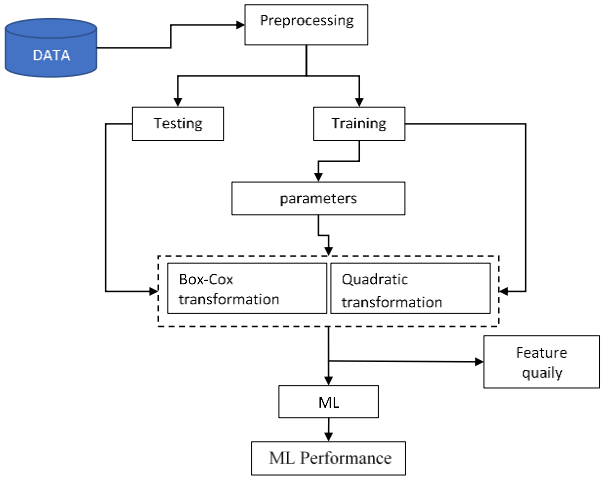

Fig. 1. Experimental design of quadratic and Box-Cox transformations.

where Tp, Tn, Fp, and Fn are components in the confusion matrix that respectively show the number of true positives, true negatives, false positives, and false negatives.

This study compares feature quality and ML performance before and after transformation to assess the efficacy of the proposed transformation, as well as comparisons with ML performance by other researchers. Seven metrics compare feature quality before and after transforming using quadratic and Box-Cox functions. The algorithm is trained using transformed training data and then tested using transformed testing data to determine ML performance. The results are compared with the performance before transformation. The accuracy of the proposed method is compared with the results of other researchers for the same dataset, including Huan Wan, et al 2018 [65], Jinsong Wang, et al 2020 [72], R Ksantini et al 2012 [73], Eslam Hamouda, et al 2021[47], Syed Muhammad Saqlain, et al 2019 [51], Jianbin Ma, et al 2019 [63], and Min Gan, et al 2021[74], and Peiyang Li, et al 2018 [75].

IV. Results and Discussion

A. Comparison of Feature Quality Before and After Transformation

The success of feature transformation is measured by comparing the quality of features before and after transformation. This research uses eight widely used feature quality measurement methods.

1) Fisher score: Since class separability is the basis for transforming features in this research, the higher the Fs, the better the feature quality. Fig. 2 presents the Fs of each feature from the d-3 (Ringnorm) dataset consisting of 20 features before and after transformation. As shown in Fig. 2, the Box-Cox and Quadratic transformations produce features with larger Fs than the original features, and the quadratic transformation produces better Fs improvement than the Box-Cox transformation on all features. Except for d-6 and d-9 datasets, the superiority of the quadratic transformation also occurs in other datasets, namely superiority in 41 out of 60 features, 25 out of 30 features, 3 out of 5 features, 9 out of 9 features, 8 out of 11 features, 38 out of 60 features, 37 out of 44 features, and 4 out of 4 features, respectively for d-1, d-2, d-4, d-5, d-7, d-8, d-10, and d-11. For the d-6 dataset of 22 features, the quadratic transform outperforms the Box-Cox in 11 features and vice versa in the other 11 features. For dataset d-9, the Box-Cox transform outperformed the quadratic transform for all features. In general, the quadratic transform produces more features with higher Fs for the whole dataset.

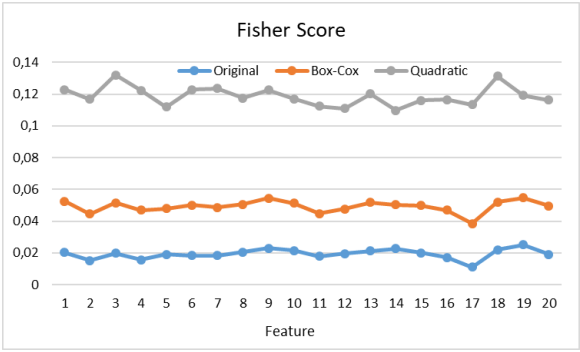

Fig. 2. Comparison of fisher score for d-3 (Ringnorm) dataset.

2) Information gain: Information Gain measurements for all datasets show that the Box-Cox transformation does not change the information gain value, meaning that the Box-Cox transformation does not increase the information gain value. On the other hand, except for datasets d-9 and d-11, the quadratic transformation generally produces features with greater information gain values, although this is not the case for every feature. Fig. 3 presents the information gain value of each feature for the d-3 (Ringnorm) dataset, which consists of 20 features. The increase in information gain in this dataset is observed for all features. An extreme case occurs in datasets d-9 and d-11, where the quadratic transformation does not increase the information gain value of each feature.

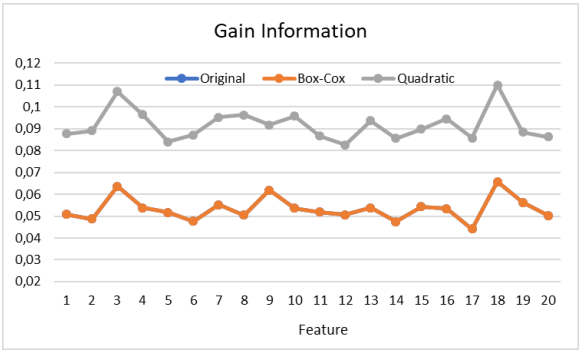

Fig. 3. Comparison of gain information for d-3 (Ringnorm) dataset.

3) Gain ratio: The Gain Ratio measurement shows similar results to the Information Gain, where the Box-Cox transformation does not change the Gain Ratio value. Except for d-9 and d-11 datasets that did not experience an increase in gain ratio in all features, the quadratic transformation generally increased the gain ratio value of a number of features in other datasets. The number of features that have

increased gain ratio values is different for each dataset, namely 29 out of 60, 13 out of 30, 20 out of 20, 2 out of 5, 4 out of 9, 8 out of 22, 3 out of 11, 15 out of 60, 18 out of 44 features, respectively for datasets d-1, d-2, d-3, d-4, d-5, d-6, d-7, d-8, d-10. Fig. 4 presents the Gain Ratio value of each feature for dataset d-3 (Ringnorm). It can be seen that the quadratic transformation increases the gain ratio of each feature.

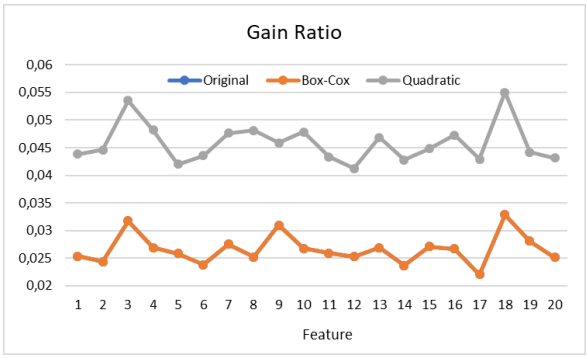

Fig. 4. Comparison of gain ratio for d-3 (Ringnorm) dataset.

4) Gini decrease: The Gini decrease measurement shows similar results to the Information Gain and Gain ratio, where the Box-Cox transformation does not change the Gini decrease value. Except for d-9 and d-11 datasets that did not experience an increase in gain ratio across all features, the quadratic transformation generally increased the Gini decrease value of a number of features in the other datasets. The number of features that have increased gain ratio values is different for each dataset, namely 27 out of 60, 13 out of 30, 20 out of 20, 2 out of 5, 5 out of 9, 9 out of 22, 5 out of 11, 15 out of 60, 17 out of 44 features, respectively for datasets d-1, d-2, d-3, d-4, d-5, d-6, d-7, d-8, d-10. Fig. 5 presents the Gini decrease value of each feature for dataset d-3 (Ringnorm). It can be seen that the quadratic transformation results in an increase in Gini decrease for all features.

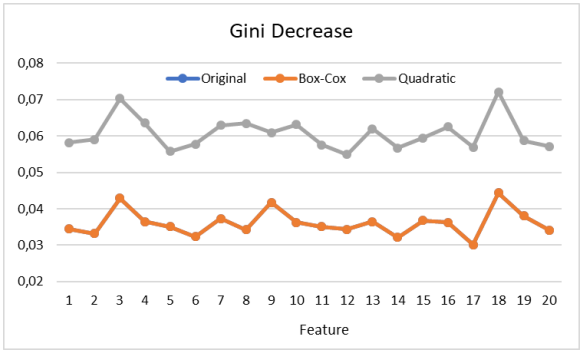

Fig. 5. Comparison of gini decreases for d-3 (Ringnorm) dataset.

5) Analysis of Variance (ANOVA): Except for the Hungarian and Balance datasets, ANOVA measurements show that the Box-Cox transformation improves most of the features in most datasets. The improvement in ANOVA values by Box-Cox transformation occurs in 58 out of 60, 25 out of 30, 20 out of 20, 3 out of 5, 7 out of 9, 19 out of 22, 60 out of 60, 31 out of 44, 4 out of 4 features for datasets d-1, d-2, d-3, d-4, d-5, d-6, d-8, d-10, d-11 respectively. The quadratic transformation gives slightly better ANOVA measurement results than the Box-Cox transformation. Except for dataset d-7, most datasets have an increase in ANOVA values. Even an increase in ANOVA values in each feature is observed in datasets d-3, d-8, d-9, and d-11. In the other six datasets, the increase in ANOVA values occurred in 59 out of 60, 27 out of 30, 3 out of 5, 8 out of 9, 19 out of 22, 30 out of 44 features, for datasets d-1, d-2, d-4, d-5, d-6, d-10, respectively. Fig. 6 presents the ANOVA values of each feature for dataset d-3 (Ringnorm) before and after undergoing Box-Cox and Quadratic transformations. It can be seen that the Quadratic transformation generally improves the ANOVA value better than the Box-Cox transformation.

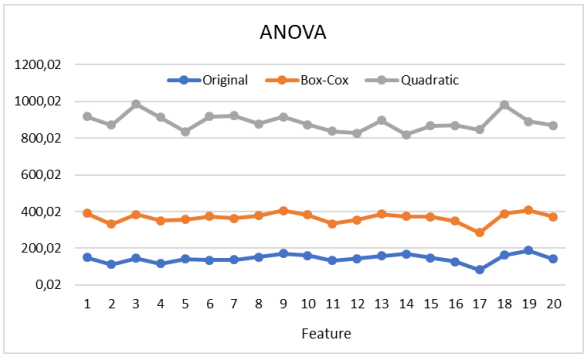

Fig. 6. Comparison of ANOVA for d-3 (Ringnorm) dataset.

6) Chi-Square: In Chi Square measurement in Box-Cox transformation, none of the features experienced changes in Chi Square value as produced by Information gain, Gain ratio, and Gini Decrease. For the quadratic transformation, except d-9 and d-11, the increase in Chi Square value occurred for 36 out of 60, 19 out of 30, 20 out of 20, 3 out of 5, 6 out of 9, 12 out of 22, 5 out of 11, 49 out of 60, 24 out of 44 features, respectively for datasets d-1, d-2, d-3, d-4, d-5, d-6, d-7, d-8, d-10. Fig. 7 presents the results of measuring the Chi-Square value of each feature for dataset d-3 (Ringnorm). It can be seen that the quadratic transformation results in an increase in the Chi-Square value of all features.

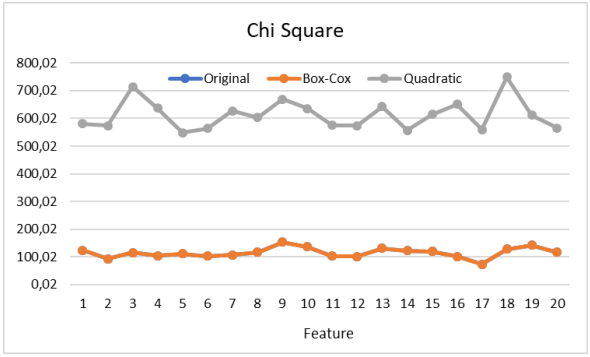

Fig. 7. Comparison of chi square for d-3 (Ringnorm) dataset.

7) ReliefF: Measurements using the ReliefF method provide varied results for both Box-Cox and Quadratic transformations, where in each dataset, features experience an increase in their ReliefF value. In Box-Cox transformation, the increase in reliefF value occurs in 35 out of 60, 28 out of 30, 2 out of 20, 3 out of 5, 8 out of 9, 19 out of 22, 4 out of 11, 29 out of 60, 2 out of 4, 25 out of 44, 3 out of 4 features, respectively for datasets d-1, d-2, d-4, d-5, d-6, d-7, d-8, d-9, d-10, d-11. The quadratic transformation generally gives slightly better ReliefF measurement results than the Box-Cox transformation. However, compared to the Box-Cox transformation, the number of features that have improved ReliefF values is less for datasets d-2, d-8, and d-10. However, the number of features that have improved ReliefF values is more in five datasets and the same in three other datasets. The increase in ReliefF value by Quadratic transformation occurs in 37 out of 60, 26 out of 30, 10 out of 20, 4 out of 5, 8 out of 9, 19 out of 22, 6 out of 11, 28 out of 60, 4 out of 4, 24 out of 44, 3 out of 4 features, respectively for datasets d-1, d-2, d-4, d-5, d-6, d-7, d-8, d-9, d-10, d-11. Fig. 8 presents the ReliefF value of each feature for dataset d-3 (Ringnorm) before and after undergoing Box-Cox and Quadratic transformations. On the d-3 dataset, the Quadratic transformation appears to increase the ReliefF value on ten features and decrease it on ten other features. In comparison, the Box-Cox transformation increases the ReliefF value on two features and decreases it on the other 18 features.

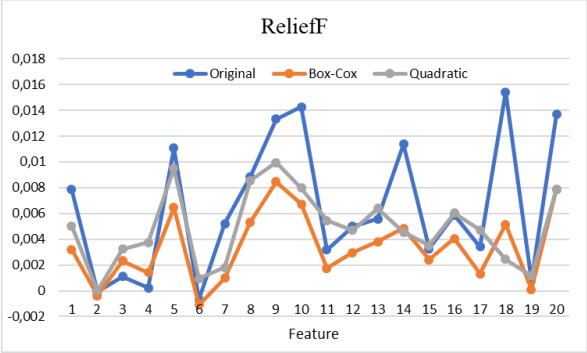

Fig. 8. Comparison of reliefF for d-3 (Ringnorm) dataset.

8) Fast Correlation Based Filter (FCBF): Measurement using FCBF gives very different results where Quadratic is better than Box-Cox regarding the number of features that have increased FCBF value. In quadratic transformation, out of 11 datasets, there are 10 datasets whose number of features has increased, where the increase in FCBF value occurs in 28 out of 60, 13 out of 30, 20 out of 20, 2 out of 5, 8 out of 9, 8 out of 22, 9 out of 11, 20 out of 60, 20 out of 44, 3 out of 4 features, respectively for datasets d-1, d-2, d-3, d-4, d-5, d-6, d-7, d-8, d-10, d-11. Except for datasets d-3, d-7, and d-11, the FCBF value from Box-Cox transformation has slightly increased in the other 8 datasets. The increase occurs in 2 out of 60, 3 out of 30, 3 out of 5, 1 out of 9, 2 out of 22, 2 out of 60, 4 out of 4, 1 out of 44 features, respectively, for datasets d-1, d-2, d-4, d-5, d-6, d-8, d-10. Fig. 9 presents the measurement results of each feature on dataset d-3 (Ringnorm), where all features have increased from 20 features.

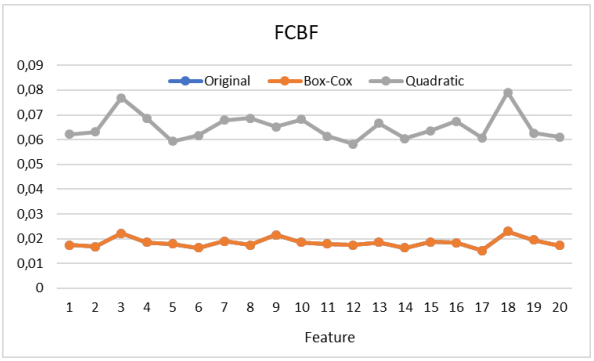

Fig. 9. Comparison of FCBF for d-3 (Ringnorm) dataset.

B. *Effects of Quadratic and Box-Cox Transformations on ML Performance*

This study uses two statistical-based machine learning algorithms, LDA and QDA. The performance results of each algorithm are presented in Tables II and III. Table II shows the comparison of LDA accuracy performance before and after Box-Cox and Quadratic transformations on 11 datasets. Experimental results on all datasets before and after quadratic transformation show an increase in accuracy performance. The highest increase in accuracy occurred on the Hayes Roth dataset, which was 0.268, and the lowest occurred on the Wisconsin dataset, which was 0.015. Experimental results on all datasets before and after the Box-Cox transformation also experienced an increase in accuracy performance. The highest increase in accuracy of 0.115 occurred on the Ringnorm dataset, and the lowest occurred on the WDBC dataset of 0.002. Compared to Box-Cox, Quadratic transformation generally produces a more significant increase in accuracy, ranging from 1.56% to 49.26%.

TABLE II. COMPARISON OF ACCURACY ALGORITHMS LDA BEFORE AND AFTER BOX-COX AND QUADRATIC TRANSFORMATION

dataset	Original	Box-Cox	Quadratic
Sonar	0.751	0.785	**0.834**
WDBC	0.956	0.958	**0.974**
Ringnorm	0.763	0.878	**0.941**
New Thyroid	0.934	0.939	**0.954**
Wisconsin	0.959	0.972	**0.974**
Parkinson	0.883	0.898	**0.908**
Hungarian	0.830	0.840	**0.848**
Splice	0.804	0.856	**0.884**
Balance	0.864	0.874	**0.912**
Spectf	0.746	0.765	**0.780**
Hayes Roth	0.544	0.606	**0.812**

Table III shows the comparison of QDA accuracy performance before and after Box-Cox and Quadratic transformations. Except for Ringnorm and Spectf datasets, where the accuracy is more or less the same, quadratic transformation improves QDA accuracy for all datasets. In comparison, Box-Cox transformation increases the accuracy on

seven datasets and decreases the accuracy on four datasets, namely Ringnorm, Balance, Spectf, and Hayes Roth. In general, both Box-Cox and Quadratic transformations improve accuracy on most datasets, whereas quadratic transformation improves accuracy on more datasets. Concurrent improvement in accuracy by Box-Cox and Quadratic transformations was observed in seven datasets, namely Sonar, WDBC, New Thyroid, Wisconsin, Parkinson, Hungarian, and Splice. On the Balance, Spectf, and Hayes-roth datasets, the Box-Cox transformation is shown to decrease accuracy, while the Quadratic transformation increases accuracy.

TABLE III. COMPARISON OF ACCURACY ALGORITHMS QDA BEFORE AND AFTER BOX-COX AND QUADRATIC TRANSFORMATION

dataset	Original	Box-Cox	Quadratic
Sonar	0.779	0.809	**0.818**
WDBC	0.956	0.961	**0.965**
Ringnorm	**0.979**	0.964	0.971
New Thyroid	0.967	0.985	**0.986**
Wisconsin	0.958	**0.972**	0.968
Parkinson	0.882	0.918	**0.923**
Hungarian	0.825	0.838	**0.844**
Splice	0.846	0.859	**0.889**
Balance	0.918	0.837	**0.963**
Spectf	**0.794**	0.791	**0.794**
Hayes Roth	0.649	0.609	**0.837**

C. Performance of LDA and QDA Based on Quadratic Transformation Compared to Other Methods

Table IV presents the performance comparison between LDA and QDA with methods from other researchers. It shows that the LDA algorithm in the 2nd column, six datasets have higher accuracy than other methods (OM) as listed in the 4th column of Table IV; they are Sonar, WDBC, Wisconsin, Parkinson, Hungarian, and Splice datasets. Compared to QDA, as shown in the 3rd column of Table 4, it indicates that QDA excels on seven datasets: Sonar, Ringnorm, New Thyroid, Parkinson, Splice, Balance, and Hayes Roth.

TABLE IV. COMPARISON OF THE OTHERS METHODS (OM) ACCURACY WITH QUADRATIC TRANSFORMATION

Dataset	LDA	QDA	OM
Sonar	**0.834**	0.818	0.768
WDBC	**0.974**	0.965	0.967
Ringnorm	0.941	**0.971**	0.953
New Thyroid	0.954	**0.986**	0.972
Wisconsin	**0.974**	0.968	0.970
Parkinson	0.908	**0.923**	0.862
Hungarian	**0.848**	0.844	0.845
Splice	0.884	**0.889**	0.868
Balance	0.912	**0.963**	0.939
Spectf	0.780	0.794	**0.827**
Hayes Roth	0.812	**0.837**	0.818

LDA and QDA outperformed other methods only on Sonar, Parkinson, and Splice datasets.

D. Discussion

The results of separability measurement using the Fisher score on each feature of the dataset before and after quadratic and Box-Cox transformation show that all features have increased separability. This result proves the success of the proposed method that aims to improve class separability. Compared to Box-Cox, the improvement in class separability of quadratic transformation is generally better. It happens because the quadratic transformation can change the order of the data that the Box-Cox does not have [20].

The results of measuring seven metrics show that the Box-Cox transformation cannot improve the quality of features in 4 matrices, namely Information gain, Gain ratio, Gini decrease, and Chi-Square. This finding indicates a relationship between transformations that do not change the order of the data and the ability of these transformations to improve the quality of the features of the four metrics above. This finding is reinforced by the results of the quadratic transformation on features that do not change the order of the data, where the feature quality also does not change. Thus, Box-Cox transformation is unsuitable for Information gain-based ML such as decision Tree and Random Fores. In contrast, quadratic transformation still has the opportunity to improve ML performance.

The quadratic transformation has been shown to improve separability, which in turn improves the performance of the LDA and QDA algorithms. Although the performance improvement varies for each dataset and algorithm used, it shows that the dataset's characteristics play a role. Likewise, the algorithm used where LDA provides a significant increase in accuracy compared to the QDA algorithm. It shows that the linear ML algorithm has an advantage over the QDA algorithm.

The quadratic transformation can work like the Box-Cox transformation in terms of transforming features to be more linearly separable between data groups, as shown by Bicego and Baldo. In some instances, the quadratic transformation can change the order of the data, which helps handle bimodal features, which Box-Cox lacks. The closed-form formulation to obtain the optimal quadratic parameters can be determined deterministically, whereas in Box-cox the optimal parameters are determined using Maximum Likelihood (MLE) or Grid search [14], [21]. It is clear that the computation time of the quadratic transform is more efficient.

Our research supports the findings of Bicego and Baldo that accuracy performance is more related to class separability than Gaussianity. We emphasize that from experimental results, using the quadratic transform where the separability effect is higher than Box-Cox also results in better accuracy. This result adds to the finding that Baldo's statement applies not only to the Box-Cox transform but also to the quadratic transform.

Based on the experimental results, we recommend the following for future research, namely:

- Quadratic transformation can detect the presence of bimodal distributed features by measuring the degree of change in the fisher score of features before and after

transformation. However, it needs to be researched more deeply on how much the right level of change is.

- Quadratic transformation can improve the Fisher score's ability to assess informative features.
- Overlapping features have a low Fisher score value. If this feature is transformed using a quadratic function, the new feature formed will also has a low Fisher score.
- Informative features with a high fisher score, when transformed using a quadratic function, the Fs value will not experience significant changes.

V. CONCLUSION

Feature engineering through quadratic transformation can be used to improve class separability. It can also be used to transform bimodal distributed data into unimodal. Unimodal features have a better fisher score than bimodal features, so if the dataset contains unimodal features, it can improve the performance of the ML algorithm. The proposed feature transformation method can improve the feature's fisher score and seven feature quality test metrics, i.e, Information Gain, Gen ratio, Gini Decrease, ANOVA, Chi-Square, ReliefF, and FCBF. Experimental results on 11 datasets using ML algorithms, namely LDA and QDA, show that feature transformation using quadratic functions can significantly improve the accuracy performance of ML algorithms.

REFERENCES

[1] Y. Wang, Z. Yang, and X. Yang, "Kernel-free quadratic surface minimax probability machine for a binary classification problem," Symmetry (Basel)., vol. 13, no. 8, 2021, doi: 10.3390/sym13081378.

[2] T. Verdonck, B. Baesens, M. Óskarsdóttir, and S. vanden Broucke, "Special issue on feature engineering editorial," Mach. Learn., no. 0123456789, 2021, doi: 10.1007/s10994-021-06042-2.

[3] M. Robnik Sikonja MarkoRobnik and friuni-ljsi IGOR Kononenko IgorKononenko, "Theoretical and Empirical Analysis of ReliefF and RReliefF," Mach. Learn., vol. 53, pp. 23–69, 2003, [Online]. Available: http://lkm.fri.uni-lj.si/xaigor/slo/clanki/MLJ2003-FinalPaper.pdf

[4] A. Zheng and A. Casari, Feature Engineering for Machine Learning. 2018. doi: 10.1201/9781315181080.

[5] K. J. Kim and W. B. Lee, "Stock market prediction using artificial neural networks with optimal feature transformation," Neural Comput. Appl., vol. 13, no. 3, pp. 255–260, 2004, doi: 10.1007/s00521-004-0428-x.

[6] H. Zhuang, X. Wang, M. Bendersky, and M. Najork, "Feature Transformation for Neural Ranking Models," SIGIR 2020 - Proc. 43rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., pp. 1649–1652, 2020, doi: 10.1145/3397271.3401333.

[7] W. Li and Z. Liu, "A method of SVM with normalization in intrusion detection," Procedia Environ. Sci., vol. 11, no. PART A, pp. 256–262, 2011, doi: 10.1016/j.proenv.2011.12.040.

[8] C. Feng et al., "Log-transformation and its implications for data analysis," Shanghai Arch. Psychiatry, vol. 26, no. 2, pp. 105–109, 2014, doi: 10.3969/j.issn.1002-0829.2014.02.

[9] C. A. Igbo and E. L. Otuonye, "The Result of a Square Root Transformation on the Error Part of the Additive Time Series Model," Am. J. Math. Stat., vol. 11, no. 4, pp. 73–93, 2021, doi: 10.5923/j.ajms.20211104.01.

[10] J. Raymaekers and P. J. Rousseeuw, "Transforming variables to central normality," Mach. Learn., no. May 2020, 2021, doi: 10.1007/s10994-021-05960-5.

[11] J. R. Berrendero and J. Cárcamo, Linear components of quadratic classifiers, vol. 13, no. 2. Springer Berlin Heidelberg, 2019. doi: 10.1007/s11634-018-0321-6.

[12] M. M. Elmorshedy, R. Fathalla, and Y. El-Sonbaty, "Feature Transformation Framework for Enhancing Compactness and Separability of Data Points in Feature Space for Small Datasets," Appl. Sci., vol. 12, no. 3, 2022, doi: 10.3390/app12031713.

[13] J. Pan, Y. Zhuang, and S. Fong, "The impact of data normalization on stock market prediction: Using SVM and technical indicators," Commun. Comput. Inf. Sci., vol. 652, pp. 72–88, 2016, doi: 10.1007/978-981-10-2777-2_7.

[14] A. Cheddad, "On Box-Cox Transformation for Image Normality and Pattern Classification," IEEE Access, vol. 8, pp. 154975–154983, 2020, doi: 10.1109/ACCESS.2020.3018874.

[15] X.-S. Si, T. Li, J. Zhang, and Y. Lei, "Nonlinear degradation modeling and prognostics: A Box-Cox transformation perspective," Reliab. Eng. Syst. Saf., vol. 217, p. 108120, 2022, doi: https://doi.org/10.1016/j.ress.2021.108120.

[16] H. Yu, P. Sang, and T. Huan, "Adaptive Box–Cox Transformation: A Highly Flexible Feature-Specific Data Transformation to Improve Metabolomic Data Normality for Better Statistical Analysis," Anal. Chem., vol. 94, no. 23, pp. 8267–8276, Jun. 2022.

[17] T. Zhan, "Log-based transformation feature learning for change detection in heterogeneous images," IEEE Geosci. Remote Sens. Lett., vol. 15, no. 9, pp. 1352–1356, 2018.

[18] F. Zhang, I. Keivanloo, and Y. Zou, "Data Transformation in Cross-project Defect Prediction," Empir. Softw. Eng., vol. 22, no. 6, pp. 3186–3218, 2017.

[19] C. C. Mason et al., "Bimodal distribution of RNA expression levels in human skeletal muscle tissue," BMC Genomics, vol. 12, no. February, 2011.

[20] M. Bicego and S. Baldo, "Properties of the Box–Cox transformation for pattern classification," Neurocomputing, vol. 218, pp. 390–400, 2016.

[21] L. Blum, M. Elgendi, and C. Menon, "Impact of Box-Cox Transformation on Machine-Learning Algorithms," Front. Artif. Intell., vol. 5, no. April, pp. 1–16, 2022.

[22] R. Van Der Heiden and F. C. A. Groen, "The Box-Cox metric for Nearest Neighbour classification improvement," Pattern Recognit., vol. 30, no. 2, pp. 273–279, 1997.

[23] M. M. Rahman, M. M. Hossain, M. K. Uddin, and A. K. Majumder, "Supervised Parametric Classification on Simulated Data via Box-Cox Transformation," Int. J. Adv. Sci. Tech. Res. Issue, vol. 3, no. 1, pp. 541–550, 2013.

[24] I. Yeo and R. A. Johnson, "A new family of power transformations to improve normality or symmetry," Biometrika, vol. 87, no. 4, pp. 954–959, Dec. 2000, doi: 10.1093/biomet/87.4.954.

[25] D. Charte, I. Sevillano-García, M. J. Lucena-González, J. L. Martín-Rodríguez, F. Charte, and F. Herrera, "Slicer: Feature Learning for Class Separability with Least-Squares Support Vector Machine Loss and COVID-19 Chest X-Ray Case Study BT - Hybrid Artificial Intelligent Systems," in International Conference on Hybrid Artificial Intelligence Systems, 2021.

[26] Z. Liu, "Fast kernel feature ranking using class separability for big data mining," *J. Supercomput.*, vol. 72, no. 8, pp. 3057–3072, 2016.

[27] S. Li, H. Zhang, R. Ma, J. Zhou, J. Wen, and B. Zhang, "Linear discriminant analysis with generalized kernel constraint for robust image classification," *Pattern Recognit.*, vol. 136, p. 109196, 2023.

[28] M. A. Muharram and G. D. Smith, "Evolutionary feature construction using information gain and gini index," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3003, pp. 379–388, 2004.

[29] Y. Liu, X. Yi, R. Chen, Z. Zhai, and J. Gu, "Feature extraction based on information gain and sequential pattern for English question classification," *IET Softw.*, 2018.

[30] G. Zhang, J. Hou, J. Wang, C. Yan, and J. Luo, "Feature Selection for Microarray Data Classification Using Hybrid Information Gain and a Modified Binary Krill Herd Algorithm," *… Sci. Comput. Life …*, 2020.

[31] A. A. Nababan, O. S. Sitompul, and Tulus, "Attribute Weighting Based K-Nearest Neighbor Using Gain Ratio," *J. Phys. Conf. Ser.*, vol. 1007, no. 1, 2018.

[32] H. Han, X. Guo, and H. Yu, "Variable selection using Mean Decrease Accuracy and Mean Decrease Gini based on Random Forest," *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, vol. 0, pp. 219–224, 2016.

[33] H. Nasiri and S. A. Alavi, "A Novel Framework Based on Deep Learning and ANOVA Feature Selection Method for Diagnosis of COVID-19 Cases from Chest X-Ray Images," *Comput. Intell. Neurosci.*, vol. 2, 2022.

[34] K. J. Johnson and R. E. Synovec, "Pattern recognition of jet fuels: Comprehensive GC × GC with ANOVA-based feature selection and principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 60, no. 1–2, pp. 225–237, 2002.

[35] X. Jin, A. Xu, R. Bie, and P. Guo, "Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3916 LNBI, pp. 106–115, 2006.

[36] A. K. F. Dornaika, "A hybrid discriminant embedding with feature selection : application to image categorization," *Appl. Intell.*, 2020.

[37] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *J. Biomed. Inform.*, vol. 85, no. July, pp. 189–203, 2018.

[38] D. Gharavian, M. Sheikhan, and S. S. Ghasemi, "Combined classification method for prosodic stress recognition in Farsi language," Int. J. Speech Technol., vol. 21, no. 2, pp. 333–341, 2018.

[39] L. Gao, M. Ye, and C. Wu, "Cancer classification based on support vector machine optimized by particle swarm optimization and artificial bee colony," *Molecules*, vol. 22, no. 12, 2017.

[40] T. Luo, C. Hou, F. Nie, and D. Yi, "Dimension Reduction for Non-Gaussian Data by Adaptive Discriminative Analysis," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 933–946, 2019.

[41] Y. R. Guo, Y. Q. Bai, C. N. Li, Y. H. Shao, Y. F. Ye, and C. zi Jiang, "Reverse nearest neighbors Bhattacharyya bound linear discriminant analysis for multimodal classification," *Eng. Appl. Artif. Intell.*, vol. 97, no. October 2020, p. 104033, 2021.

[42] H. Wan, H. Wang, B. Scotney, J. Liu, and W. W. Y. Ng, "Within-class multimodal classification," *Multimed. Tools Appl.*, vol. 79, no. 39–40, pp. 29327–29352, 2020.

[43] M. A. Duarte-Mermoud, N. H. Beltrán, and M. A. Bustos, "Chilean wine varietal classification using quadratic Fisher transformation," *Pattern Anal. Appl.*, vol. 13, no. 2, pp. 181–188, 2010.

[44] H. Liu and H. Motoda, "Feature transformation and subset selection," *IEEE Expert*, vol. 13, no. 2, pp. 26–28, 1998.

[45] S. M. Bassir, A. Akbari, and B. Nassersharif, "An improved feature transformation method using mutual information," *Int. J. Speech Technol.*, vol. 17, no. 2, pp. 107–115, 2014.

[46] A. Noor, L. Ali, H. T. Rauf, U. Tariq, and S. Aslam, "An integrated decision support system for heart failure prediction based on feature transformation using grid of stacked autoencoders," *Measurement*, vol. 205, p. 112166, 2022.

[47] E. Hamouda, A. S. Abohamama, and M. Tarek, "Random Projection-Based Feature Transformation Using Metaheuristic Optimization Algorithm," *Arab. J. Sci. Eng.*, vol. 46, no. 9, pp. 8345–8353, 2021.

[48] Z. Deng, S. Wang, and F. L. Chung, "A minimax probabilistic approach to feature transformation for multi-class data," *Appl. Soft Comput. J.*, vol. 13, no. 1, pp. 116–127, 2013.

[49] M. Z. Abedin, G. Chi, M. M. Uddin, M. S. Satu, M. I. Khan, and P. Hajek, "Tax Default Prediction Using Feature Transformation-Based Machine Learning," *IEEE Access*, vol. 9, pp. 19864–19881, 2021.

[50] D. Yan, X. Zhou, X. Wang, and R. Wang, "An off-center technique: Learning a feature transformation to improve the performance of clustering and classification," *Inf. Sci. (Ny).*, vol. 503, pp. 635–651, 2019.

[51] S. M. Saqlain *et al.*, "Fisher score and Matthews correlation coefficient-based feature subset selection for heart disease diagnosis using support vector machines," *Knowl. Inf. Syst.*, vol. 58, no. 1, pp. 139–167, 2019.

[52] L. Sun, X.-Y. Zhang, Y.-H. Qian, J.-C. Xu, S.-G. Zhang, and Y. Tian, "Joint neighborhood entropy-based gene selection method with fisher score for tumor classification," *Appl. Intell.*, vol. 49, no. 4, pp. 1245–1259, 2019.

[53] C. Li and J. Xu, "Feature selection with the Fisher score followed by the Maximal Clique Centrality algorithm can accurately identify the hub genes of hepatocellular carcinoma," *Sci. Rep.*, vol. 9, no. 1, 2019.

[54] M. Dashtban and M. Balafar, "Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts," *Genomics*, vol. 109, no. 2, pp. 91–107, 2017.

[55] M. N. K.P. and T. P., "Feature selection using efficient fusion of Fisher Score and greedy searching for Alzheimer's classification," *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2021.

[56] J. Yang, Y. L. Liu, C. S. Feng, and G. Q. Zhu, "Applying the fisher score to identify Alzheimer's disease-related genes," *Genet. Mol. Res.*, vol. 15, no. 2, 2016.

[57] G. S. Saragih and Z. Rustam, "Support Vector Machine with Fisher Score Feature Selection to Predict Disease-Resistant Gene in Rice," *J. Phys. Conf. Ser.*, vol. 1108, no. 1, 2018.

[58] L. Sun, X. Zhang, Y. Qian, J. Xu, and S. Zhang, "Feature selection using neighborhood entropy-based uncertainty measures for gene expression data classification," *Inf. Sci. (Ny).*, vol. 502, pp. 18–41, 2019.

[59] I. Dagher, "Quadratic kernel-free non-linear support vector machine," *J. Glob. Optim.*, vol. 41, no. 1, pp. 15–30, 2008.

[60] Y. Tian, Z. Yong, and J. Luo, "A new approach for reject inference in credit scoring using kernel-free fuzzy quadratic surface support vector machines," *Appl. Soft Comput. J.*, vol. 73, pp. 96–105, 2018.

[61] M. A. Bustos, M. A. Duarte-Mermoud, and N. H. BeltrÁn, "Nonlinear feature extraction using fisher criterion," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 22, no. 6, pp. 1089–1119, 2008.

[62] S. S. Wirts, "Quadratic Formula: Revisiting a Proof Through the Lens of Transformations," 2020, [Online]. Available: http://arxiv.org/abs/2010.14251.

[63] J. Ma and G. Teng, "A hybrid multiple feature construction approach for classification using Genetic Programming," *Appl. Soft Comput. J.*, vol. 80, pp. 687–699, 2019.

[64] L. Li, L. Du, W. Zhang, H. He, and P. Wang, "Enhancing information discriminant analysis: Feature extraction with linear statistical model and information-theoretic criteria," *Pattern Recognit.*, vol. 60, pp. 554–570, 2016.

[65] H. Wan, H. Wang, G. Guo, and X. Wei, "Separability-Oriented Subclass Discriminant Analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 409–422, 2018.

[66] X. Wang, S. Wang, Z. Huang, and Y. Du, "Condensing the solution of support vector machines via radius-margin bound," *Appl. Soft Comput.*, vol. 101, p. 107071, 2021.

[67] J. Ma and X. Gao, "Designing genetic programming classifiers with feature selection and feature construction," *Appl. Soft Comput. J.*, vol. 97, 2020.

[68] L. Ali, I. Wajahat, N. Amiri Golilarz, F. Keshtkar, and S. A. C. Bukhari, "LDA–GA–SVM: improved hepatocellular carcinoma prediction through dimensionality reduction and genetically optimized support vector machine," *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2783–2792, 2020.

[69] R. Fu, M. Han, Y. Tian, and P. Shi, "Improvement motor imagery EEG classification based on sparse common spatial pattern and regularized discriminant analysis," *J. Neurosci. Methods*, vol. 343, no. June, p. 108833, 2020.

[70] Y. Aliyari Ghassabeh, F. Rudzicz, and H. A. Moghaddam, "Fast incremental LDA feature extraction," Pattern Recognit., vol. 48, no. 6, pp. 1999–2012, 2015, doi: 10.1016/j.patcog.2014.12.012.

[71] B. Ghojogh and M. Crowley, "Linear and Quadratic Discriminant Analysis: Tutorial," no. 4, pp. 1–16, 2019, [Online]. Available: http://arxiv.org/abs/1906.02590.

[72] J. Wang, J. Liao, and W. Huang, "A density-based maximum margin machine classifier," *Cluster Comput.*, vol. 23, no. 4, pp. 3069–3078, 2020, doi: 10.1007/s10586-020-03070-w.

[73] R. Ksantini and B. Boufama, "Combining partially global and local characteristics for improved classification," *Int. J. Mach. Learn. Cybern.*, vol. 3, no. 2, pp. 119–131, 2012, doi: 10.1007/s13042-011-0045-9.

[74] M. Gan and L. Zhang, "Iteratively local fisher score for feature selection," *Appl. Intell.*, vol. 51, no. 8, pp. 6167–6181, 2021, doi: 10.1007/s10489-020-02141-0.

[75] P. Li *et al.*, "Improved Graph Embedding for Robust Recognition with outliers," *Sci. Rep.*, vol. 8, no. 1, pp. 1–11, 2018, doi: 10.1038/s41598-018-22207-x.