Applied Soft Computing Journal 97 (2020) 105524

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Investigating the impact of data normalization on classification performance

Dalwinder Singh, Birmohan Singh *

Department of Computer Science and Engineering, Sant Longowal Institute of Engineering and Technology, Longowal, Punjab, India

HIGHLIGHTS

- The impact of data normalization on classification performance is investigated empirically.
- Full feature set, feature selection and feature weighting are used for empirical analysis.
- A modified Ant Lion Optimization algorithm is presented for searching optimal solutions.
- A set of best and worst normalization methods are identified and recommended.

ARTICLE INFO

Article history:
Received 1 June 2017
Received in revised form 25 April 2019
Accepted 21 May 2019
Available online 23 May 2019

Keywords:
Ant lion optimization
Data normalization
Feature selection
Feature weighting
k-NN classifier

ABSTRACT

Data normalization is one of the pre-processing approaches where the data is either scaled or transformed to make an equal contribution of each feature. The success of machine learning algorithms depends upon the quality of the data to obtain a generalized predictive model of the classification problem. The importance of data normalization for improving data quality and subsequently the performance of machine learning algorithms has been presented in many studies. But, the work lacks for the feature selection and feature weighting approaches, a current research trend in machine learning for improving performance. Therefore, this study aims to investigate the impact of fourteen data normalization methods on classification performance considering full feature set, feature selection, and feature weighting. In this paper, we also present a modified Ant Lion optimization that search feature subsets and the best feature weights along with the parameter of Nearest Neighbor Classifier. Experiments are performed on 21 publicly available real and synthetic datasets, and results are analyzed based on the accuracy, the percentage of feature reduced and runtime. It has been observed from the results that no single method outperforms others. Therefore, we have suggested a set of the best and the worst methods combining the normalization procedure and empirical analysis of results. The better performers are *z*-Score and Pareto Scaling for the full feature set and feature selection, and *tanh* and its variant for feature weighting. The worst performers are Mean Centered, Variable Stability Scaling and Median and Median Absolute Deviation methods along with un-normalized data.

1. Introduction

The pre-processing of the data is an essential step to achieve good classification performance before evaluating the data on the machine learning algorithms. It includes discretization of data, removing outliers and noise from the data, integration of data from various sources, dealing with incomplete data and transformation of data to comparable dynamic ranges (viz. normalization) [1,2]. Among these, data normalization is an essential pre-processing step which involves the transformation of features in a common range so that greater numeric feature values cannot dominate the smaller numeric features values. The main aim is to minimize the bias of those features whose numerical contribution is higher in discriminating pattern classes. If the relative importance of the features are unknown, the features in the data have been made equally important while predicting the output class of an unknown instance [2]. It is very useful for the statistical learning methods since all of the features in the data contribute equally to the learning process.

The importance of the data normalization for constructing accurate predictive models has been examined for various machine learning algorithms such as *k*-Nearest Neighbors (*k*-NN) [3], Artificial Neural Networks (ANN) [4,5] and Support Vector Machines (SVM) [6]. Many authors have validated the impact of data normalization for improving classification performance in various fields, such as medical data classification [7,8], multimodal biometrics systems [9], vehicle classification [10], faulty

* Corresponding author.
E-mail addresses: dalwindercheema@outlook.com (D. Singh), birmohansingh@sliet.ac.in (B. Singh).

https://doi.org/10.1016/j.asoc.2019.105524
1568-4946/

motor detection [11], predicting stock market [12], leaf classification [13], credit approval data classification [14], genomics [15] and some other application areas [16,17].

Although data normalization ensures that features have an equal numerical contribution, it does not imply that these features are equally important for the classification decision. Some features in the data can have a varying degree of relevance while others are entirely irrelevant and redundant. The presence of unwanted features complicates the learning process and increases the feature space size. These features interfere with the useful features which confuse the learning algorithm and result in deterioration of the classification performance. It also increases the computational complexity of a machine learning algorithms [1] which depends on the number of features as well as instances in training data. Further, increasing the amount of these features leads to the curse of dimensionality, the problem which needs to be addressed appropriately for obtaining efficient predictive models of the learning algorithms. Two approaches have been studied extensively by the researchers to reduce the dimensions, i.e., Feature Selection and Feature Weighting. Feature selection aims to choose a relevant subset of the features while discarding irrelevant and redundant features. In contrast, feature weighting works on the concept of feature relevance where weight is assigned to each feature according to its relevance. Thus, the weight for unwanted features will be zero while it may vary from higher to lower for other features [18].

This work aims to investigate the impact of data normalization on classification performance. The main contributions of the paper are as follows:

- Empirical analysis of normalization methods in contrast to un-normalized data for determining the impact on the classification accuracy. In this work, fourteen normalization methods are considered, and k-Nearest Neighbor classifier is used for evaluating the data.
- Empirical analysis of normalization methods when feature selection and feature weighting approaches are utilized. The wrapper method is employed for the analysis where Ant Lion Optimization (ALO) algorithm is used for searching the relevant features subsets and optimal feature weights along with the best parametric value (i.e., k) of the NN learning algorithm. The performance of both approaches is measured in terms of classification accuracy, dimensionality reduction, and runtime.

These investigations will be helpful for identifying the best and the worst normalization methods in terms of their classification accuracy, feature reduction, and the runtime. Furthermore, these investigations are a roadmap for selecting the normalization method to obtain an effective predictive model when working with the data from various application areas.

The rest of the paper is organized as follows. Section 2 covers related work and Section 3 provides the details of the normalization methods. Section 4 explains the ALO based wrapper method that performs feature selection and feature weighting along with a parameter optimization of the k-NN classifier. Experimental results and discussions are covered in Section 5. Section 6 discusses the findings, and the last section provides the conclusion of the work.

2. Related work

The normalization techniques have been used by many researchers for the improvement of classification performance in different applications areas. We have categorized the work of researchers according to the data used in various fields such as medical, biometric, credit score, bioinformatics, stock market, object recognition, and others.

Biometric Data: In biometric systems, data normalization played a vital role in the integration of different features on a common scale. In 2005, Jain et al. [19] proposed a multimodal system that combined the features of different biometric sources for improving the performance of the biometric systems. They considered face, fingerprint and hand-geometry traits of the human and integrated the extracted information at the matching score level. Normalization methods were used for the fusion of heterogeneous matching score. A total of seven normalization methods were considered in this work. Min–max [0, 1] and z-score methods were reported better as compared to the other normalization methods. However, since min–max [0, 1] and z-score methods are sensitive to outliers, they recommended *tanh* normalization for information fusion in their system. In another work, Ross and Govindarajan [20] studied three different scenarios of information fusion at the feature level considering face and hand biometrics. In their work, the median normalization was preferred as compared to a min–max method to deal with the outliers. Snelick et al. [9] studied a combination of normalization and fusion methods at the matching score level to improve the performance of biometric authentication systems. Fingerprint and face biometric were considered to develop a system on large-scale population. They proposed an adaptive normalization; Quadric-Line-Quadric (QLQ) method and compared it with min–max, z-score and *tanh* normalization methods. It was observed that the min–max method was best for applications with open populations whereas the QLQ method was best for applications with closed populations. Ekenel and Stiefelhagen [21] studied the effects of feature selection and data normalization for face recognition. Block-based discrete cosine transform was used for feature extraction from face images which were normalized using the unit and coefficient normalization. The comparative analysis of the normalization methods showed that the unit method achieved the best performance as compared to un-normalized data as well as the normalized coefficients method. Kumar and Ravikanth [22] proposed a new biometric authentication system based on the finger back surface images. The features of finger knuckle bending and geometry of fingers were integrated to develop a personal authentication system. The geometric features were normalized before their integration with the knuckle surface based features because of different numeric ranges. Two normalization methods (min–max [0, 1] and z-score methods) were used in this work, and min–max method worked better than the z-score method.

Credit Score Data: Huang and Dun [23] used the min–max normalization that scaled the features in a range of [0, 1]. A wrapper based distributed Particle Swarm optimization-Support Vector Machines (PSO-SVM) method was proposed for finding feature subset along with the parameters of SVM. German credit score data together with simulated data were used for evaluating their method. However, in this work, they have not compared the results with the un-normalized data. Wang and Huang [14] performed evolutionary algorithm based feature selection for improving the classification of credit approval data. The data was pre-processed before evaluating it on the different classifiers. They used four pre-processing schemes that deal with the replacement of missing values, re-sampling for unbalanced data, data type transformation and min–max normalization [0, 1]. Two credit approval datasets, Australian and German, were used for experiments in their study, and the data were evaluated on nine classifiers. However, the results were not compared with the un-normalized data.

Publicly Available Datasets: Huang and Wang [24] applied a Genetic Algorithm (GA) for simultaneous feature selection and

parameter optimization of the SVM with Radial Basis Function (RBF) kernel. The data was normalized using min–max [0, 1] normalization method, but no comparison of performance in contrast to un-normalized data was reported. Shalabi and Shaaban [25] studied the impact of three normalization methods on the classification of HSV dataset with Induction Decision tree (ID3) classifier. The data was normalized with three methods namely, z-score, min–max, and decimal scaling. The outcomes showed that min–max normalization was the best performing method as compared to z-score and decimal method on ID3 as well as on the three other machine learning algorithms considered in this study. Lin et al. [26] used the min–max normalization $[-1, 1]$ for the classification of data on SVM with RBF kernel. PSO was used for feature selection as well as parameters determination of the SVM. However, their work did not report the amount of improvement made with the normalization method as compared to un-normalized data with and without feature selection.

Other research areas where data normalization was employed are discussed here. Berg et al. [27] analyzed the effect of data preprocessing considering eight normalization methods to improve the biological interpretation of metabolomics data. The data was analyzed using PCA to determine the effective normalization method among them, and min–max, as well as z-score method, performed better as compared to the other methods. Craig et al. [28] studied the effects of feature and instance normalization on simulated NMR spectroscopic metabonomic datasets. Mean-centered and z-score methods were considered for normalization, and the resultant data was analyzed with PCA. The authors reported that normalization helps for effective classification models, but these pre-processing methods cannot be generalized. Li and Liu [16] compared different normalization methods for the improved classification of intrusion data using SVM. A total of six normalization methods were considered in this study, but the experiments were performed using two normalization (min–max and max) methods only. An improvement in accuracy was reported with the min–max normalization method as compared to un-normalized data as well as max normalization. Kadir et al. [13] used the Probabilistic Neural Network (PNN) for the leaf classification where extracted features were normalized using min–max ([0, 1]) method. The improvement in accuracy was reported as compared to the previous works, but the outcomes were not compared with the un-normalized data. Esfahani et al. [11] used z-score normalized data for the detection of faulty motors in real-time condition monitoring of induction motors. The outcomes of the normalized data are not compared with un-normalized data. Wen et al. [10] improved the classification of vehicles from the images by using Haar-like features. The features were normalized with a normalization method that computed the magnitudes of the features and then, rescaled them using min–max normalization. The performance of their method was better than the un-normalized data, z-score and min–max methods. Su et al. [17] proposed anomaly android malapp detection system for the detection of malapp from the android operating system. A total of 4209 platform-based features were extracted, and data were normalized using min–max and Term Frequency–Inverse Document Frequency (TF–IDF) methods. PCA and k-NN classifiers were used for the classification of the apps. The combination of min–max [0, 1] method with k-NN classifier achieved better performance than other combinations, but the comparison of performance with un-normalized data was not reported. Pan et al. [12] investigated the effect of various data normalization methods for predicting stock index and its movements. The features set consisted of 27 technical indicators to predict the price movement of the stock index. The prediction models were constructed from SVM with RBF kernel using five different normalization methods. The data normalization helps to improve the classification performance as compared to un-normalized data, but no method emerged as superior. The decimal scaling method required the least processing time for learning the prediction model of their problem.

The previous studies show that normalizing data before evaluating it on classifier impact the classification performance. Data normalization helps in predicting a more accurate model of the machine learning algorithms, and therefore, these methods have been applied to data belonging to various fields. However, the observations from the literature show that the primary aim of utilizing normalization is to improve the performance. Therefore, the comprehensive analysis of normalization methods against the un-normalized data is not present. Furthermore, no study has been conducted that contrasts the classification performance after finding the relevant features either by feature selection or feature weighting. Hence, a comparison of different normalization methods is essential for in-depth understanding so that a better predictive model can be constructed. This study aims to focus on the analysis of the impact of the data normalization for the improvement of classification performance. We have considered fourteen data normalization methods in this study which are discussed in the next section.

3. Normalization methods

The normalization is an operation on raw data that either rescale or transform it such that each feature has a uniform contribution. It deals with two main issues of data which hinder the learning process of machine learning algorithms, i.e., the presence of dominant features and outliers. Many methods have been proposed to normalize the data within a specified range based on statistical measures from the raw (un-normalized) data. Consider a dataset with f features and N instances which is represented as: $D = \{x_{i,n}, y_n \mid i \in f \text{ and } n \in N\}$, where x represents the data to be learnt by learning algorithm and y is the corresponding class label. In this work, various normalization methods are considered to investigate their impact on classification performance. These methods are categorized based on how certain statistical characteristics of the raw data are used for normalizing the data. The methods are described as follows.

3.1. Mean and Standard Deviation Based Normalization Methods:

In these methods, the statistical mean and/or standard deviation of raw data are used to normalize the data. Different variants are presented by the researchers to rescale or transform the data with these measures. It includes:

3.1.1. Z-score Normalization (ZSN):

The mean and standard deviation measures are used to rescale the data such that resultant features have zero mean and a unit variance [29,30]. Each instance, $x_{i,n}$ of the data is transformed into $x'_{i,n}$ as follows:

$$x'_{i,n} = \frac{x_{i,n} - \mu_i}{\sigma_i} \tag{1}$$

where μ and σ denote the mean and standard deviation of ith feature respectively.

3.1.2. Mean Centered (MC):

It removes the offset from the data by subtracting the mean of a feature from each instance of that feature [28]. The method is defined as follows:

$$x'_{i,n} = x_{i,n} - \mu_i \tag{2}$$

3.1.3. Pareto Scaling (PS): This method is similar to a z-score method where the scaling factor is the square root of standard deviation [31,32]. Thus, the newer features have a variance equal to the standard deviation of un-normalized features. It is given as follows:

$$x'_{i,n} = \frac{x_{i,n} - \mu_i}{\sqrt{\sigma_i}} \tag{3}$$

The method improves the representation of lower concentrated features while minimizing the contribution of noise in the data. Further, it removes the limitation of the unit variance in contrast to the z-score method and keeps the structure of the data partially intact [27].

3.1.4. Variable Stability Scaling (VSS): The method extends the z-score normalization by introducing the Coefficient of Variation (CV) as a scaling factor [27]. The coefficient of variation is given as the ratio of the mean of data to its standard deviation which is defined as follows:

$$x'_{i,n} = \frac{(x_{i,n} - \mu_i)}{\sigma_i} \cdot \frac{\mu_i}{\sigma_i} \tag{4}$$

The coefficient of variation gives higher importance to those features which have a small standard deviation and lower importance to those that have a large standard deviation. The coefficient of variation gives higher importance to the features having small standard deviation and lower importance to the features having a large standard deviation.

3.1.5. Power Transformation (PT): This method transforms the data into homoscedasticity by reducing the effects of heteroscedasticity [33], and it is applied to the data that has standard deviation proportional to the root of its mean [27]. The raw data is transformed by calculating its square root and then rescaled using the mean centered method. It is given as follows:

$$x'_{i,n} = p_{i,n} - \mu_i^p, \quad \text{where} \quad p_{i,n} = \sqrt{\hat{x}_{i,n}} \tag{5}$$

The method cannot be used for transforming negative values into real values. Therefore, the values of such features are shifted before normalizing data as follows:

$$\hat{x}_{i,n} = x_{i,n} - min(x_i)$$

where min denotes the minimum value of ith feature. The data is relocated such that minimum value will coincide with zero and maximum value equals to the difference between two extreme values of the feature. The main disadvantage of power transformation is that it cannot make multiplicative effects on the data to be additive. The multiplicative effect occurs when data has a standard deviation proportional to its mean.

These methods help to reduce the effect of outliers from the data but do not overcome the problem of dominant features entirely except the z-score method. Nevertheless, all the methods mentioned above lack scaling or transforming of data into the same numerical range since mean and standard deviation measures may vary with time [12].

3.2. **Minimum–Maximum Value Based Normalization Methods:** The minimum and/or maximum values of the un-normalized data are used for rescaling. These methods include:

3.2.1. Min–Max Normalization (MMN): The method scales the un-normalized data to a predefined lower and upper bounds linearly [34]. The data is usually rescaled within the range of 0 to 1 or -1 to 1. The equation is given as follows:

$$x'_{i,n} = \frac{x_{i,n} - min(x_i)}{max(x_i) - min(x_i)}(nMax - nMin) + nMin \tag{6}$$

where min and max denotes the minimum and maximum value of i^{th} feature respectively. The lower and upper bounds to rescale the data are denoted with $nMin$ and $nMax$ respectively. In this work, both [0, 1] (MMN0) and [−1, 1] (MMN1) scales are considered to analyze the classification performance.

3.2.2. Max Normalization (MN): This method is a variant of min–max normalization where each feature is rescaled within the range of −1 to +1 [16]. It rescales the data by dividing each feature by its maximum value and is defined as follows:

$$x'_{i,n} = \frac{x_{i,n}}{max(|x_i|)} \tag{7}$$

These normalization methods are useful for preserving the relationships among the original input data, unlike normalization methods which are based on a mean and standard deviation of the data as these values may vary with time. The main drawback of this data normalization is its sensitivity towards outliers as well as extreme values present in un-normalized data [1]. Another drawback is the "out of the bound" problem that occurs when some values of the test data lie outside the range of the un-normalized data.

3.3. **Decimal Scaling Normalization (DSN):** This method normalizes each feature by measuring the maximum value of that feature which is equivalent to moving the decimal points of the instances values [34]. The method is useful for the data where variations in features are logarithmic [19]. Each instance, $x_{i,n}$ of the given data is rescaled into $x'_{i,n}$ as follows:

$$x'_{i,n} = \frac{x_{i,n}}{10^j} \tag{8}$$

where $j = log_{10}(max(x_i))$. This method is identical to max normalization and therefore, it faces the same limitations as does the MN method, i.e., the maximum values should be known beforehand and its sensitivity towards extreme values in the data. However, unlike the max method, it cannot rescale the negative values and therefore, one needs to relocate the data to positive scale similar to the power transformation method. Another disadvantage of decimal scaling normalization is that, if the features of the data itself are not on a logarithmic scale, there is no significance of this method. For example: if all features in the data have a common range in between [0, 10], then $j = 1$ and all of the dataset will be divided by 10. Hence, the data is changed to another range.

3.4. **Median and Median Absolute Deviation Normalization (MMADN):** In this method, the median, and the Median Absolute Deviation (MAD) values are measured from each feature to rescale the data [19]. The normalization is defined as follows:

$$x'_{i,n} = \frac{x_{i,n} - med_i}{MAD_i} \tag{9}$$

where med is the median value of each feature and $MAD = med(|x_{i,n} - med_i|)$. This method of normalization is alike z-score but uses the $median$ as a statistical property for

rescaling the data. This normalization method is more robust than *mean* based methods due to the insensitivity of *median* values towards the outliers as well as the extreme values of the data. However, similar to mean and standard deviation based normalization methods, this method also fails to rescale the data to the common numerical range when data varies with respect to the time.

3.5. ***Tanh* Based Normalization (TN):** This method was proposed by Hampel et al. [35] in which the transformation of data is based on the Hampel estimators. Each instance, $x_{i,n}$ is transformed as follows:

$$x'_{i,n} = \frac{1}{2}\{tanh(0.01(\frac{x_{i,n} - \mu_i^H}{\sigma_i^H})) + 1\} \tag{10}$$

where μ^H denotes the mean and σ^H denotes the standard deviation of the Hampel estimators. These estimators are based on influence function (ψ) which is defined as follows:

$$\psi(z) = \begin{cases} z & 0 \le |z| \le u_1, \\ u_1 \times sign(z) & u_1 < |z| \le u_2, \\ u_1 \times sign(z) \times (\frac{u_3 - |z|}{pu_3 - u_2}) & u_2 < |z| \le u_3, \\ 0 & |z| > u_3, \end{cases}$$

where $z_{i,n} = x_{i,n} - med(x_i)$, $sign(z) = 1$ for $z \ge 0$ and $sign(z) = -1$ for $z < 0$. The function helps to reduce the influence of those values that lie at the tails of the distribution (given by parameter u_1, u_2 and u_3) during the estimation of the location and scale parameters. Therefore, it is not sensitive to the outliers. However, these parameters must be chosen carefully which depends upon the amount of robustness required as well as noise in the training data. The values of these parameters are fixed to $u_1 = quantile_{0.7}(|z|)$, $u_2 = quantile_{0.85}(|z|)$ and $u_3 = quantile_0.95(|z|)$ as used by Jain et al. [19]. A **variant of *Tanh* normalization (VTN)** was used by Snelick et al. [9], where Hampel estimators are replaced by mean and standard deviation of each feature. The normalization is given as follows:

$$x'_{i,n} = \frac{1}{2}\{tanh(0.01(\frac{x_{i,n} - \mu_i}{\sigma_i})) + 1\} \tag{11}$$

3.6. **Sigmoidal Normalization:** It is a non-linear transformation for reducing the effects of the outlier in the data [36]. Two sigmoidal functions are widely used in literature for normalizing the data which are given as follows:

3.6.1. Logistic Sigmoid (LS) function based normalization: The transformation of data is based on the logistic sigmoid function which maps data within the range of 0 to 1 [37]. It squashes larger values present at the tails of the data while rest of the data is mapped linearly. It is computed as follows:

$$x'_{i,n} = \frac{1}{1 + e^{-q_{i,n}}}, \quad \text{where} \quad q_{i,n} = \frac{x_{i,n} - \mu_i}{\nu \cdot \sigma_i} \tag{12}$$

where ν is the parameter whose value is set to 1 during evaluation. This non-linear normalization method is used when data is not evenly distributed around its mean. The outliers which lie away from the mean of the data are squashed exponentially [37].

3.6.2. Hyperbolic Tangent (HT) function based normalization: The features are normalized within the range of -1 to 1 using hyperbolic tangent function [7]. It is given as follows:

$$x'_{i,n} = \frac{1 - e^{-q_{i,n}}}{1 + e^{-q_{i,n}}}, \quad \text{where} \quad q_{i,n} = \frac{x_{i,n} - \mu_i}{\sigma_i} \tag{13}$$

This normalization is suitable for scaling the outliers without affecting the rest of the data. It scales the normal data linearly, thus prevents the data being pushed into narrow range due to the presence of outliers.

This normalization preserves the significance of the normal data while dealing with the outliers. The values that lie within a standard deviation of the mean of the feature are mapped to almost linear region whereas outliers, as well as extreme values, are mapped along the tails of the range by the given function.

4. Feature selection and feature weighting

Feature selection and feature weighting are popular research topics for studying improvements in classification performance [38–40]. Feature Selection (FS) is based on the idea of selecting an effective subset of features from the set of features [41,42]. Redundant and irrelevant features are removed from the full set of features since these features either degrade or do not contribute to the learning process. It can also be viewed as a feature weighting where the assignment of weights is constricted to the binary values [18]. It is a combinatorial search problem where an optimal subset of features is required to achieve good classification results. Many methods have been proposed for finding an optimal subset which is classified into three categories; filter, wrapper, and hybrid. Filter methods are based on information-theoretic measures such as measuring interclass distance [43] or clustering [44] for finding feature subset, wrapper methods use classifier performance as a criterion of subset selection, and hybrid methods combine the strength of both filter and wrapper methods [45]. Various methodologies have been proposed for selecting subset of features which includes Mutual Information [46], Value-Difference Metric [47], Iterative RELIEF [43], Cross-category feature importance [48], feature selection using Particle swarm optimization [49], Differential evolution [50], Teacher learner-based optimization [51] and Ant colony optimization [52]. A comprehensive survey of feature selection using evolutionary computation methods can be seen in [53].

Feature Weighting (FW) works on the idea that the contribution of each feature is different for discriminating the pattern classes. It helps to improve the classification performance of those machine learning algorithms that utilize distance as a measure of classification decision. It is achieved by assigning weights to each feature according to its discrimination capability and helps to reduce the sensitivity of redundant and irrelevant features present in the data. The weights assignment is a continuous search space problem where relevant features have larger weights and vice-versa [54]. It results in a newer feature space where dimensions of the higher weighted features will expand while dimensions of lower weighted features will shrink. Wrapper methods are used to search for an optimal set of weights. Several optimization algorithms have been applied for this purpose which includes Genetic algorithm [55,56], Tabu search [57], Biogeography-based optimization [58], Differential evolution [39], Simulated annealing [59] and Gradient descent [60].

Although feature selection is a subset of feature weighting, these approaches differ from each other from the perspective of feature relevance. Feature selection performs better when data has features that are either entirely relevant or irrelevant. Feature weighting works better when features vary in their relevance. In this paper, we considered both approaches to analyze the impact on classification performance.

4.1. Nearest neighbor learning

The Nearest Neighbor (NN) [61] is a supervised pattern classification approach that belongs to the class of *lazy learning* algorithms. *k*-NN does not learn from the supplied training data and postpones the estimation of the classification model until testing instances are available [18]. This learning algorithm performs classification by obtaining *k* least distant instances to the unknown instance from the training data and predicting their majority class as an output class. If the results are tied, one of the most probable classes will be selected randomly to determine the output class [62]. The closest neighbors to the unknown instance are determined by the distance function such as Euclidean distance. The success of the classifier depends upon those instances which lie nearby to the actual class of the unknown instance. In this study, we have used *k*-NN classifier, because it is sensitive to the changes in the feature space and therefore, it can efficiently evaluate normalization methods that have different feature spaces. Also, it has performed well in many applications [63,64], and it requires only one parameter to optimize.

4.2. Ant lion optimization

The purpose of the optimization algorithms is to deal with complex problems that cannot be solved in polynomial time, probably by searching the best solution from a given set of solutions. Many algorithms have been developed in pursuit of the global solution but may entrap in local optimum solutions due to the complexity of the problem (i.e., the curse of dimensionality) [65] as well as exploration and exploitation capabilities [66] of these algorithms.

Ant Lion Optimization (ALO) is a nature-inspired algorithm which is proposed by Mirjalili [67] for solving optimization problems. This algorithm mimics the hunting behavior of antlions that catches their prey by building cone-shaped pits. The hunting process is modeled mathematically to develop the algorithm in which ants are required to move over the search space so that antlions will hunt them and become fitter. The size of antlion traps represent the fitness of solution; larger pits size mean a higher probability of catching ants. The algorithm has good exploration and exploitation capabilities of searching for the best solutions and requires fewer parameters to tune. This algorithm has outperformed the other optimization algorithms for many engineering problems [68–70].

Antlion optimization can be applied to feature subset selection and feature weighting problems for improving classification performance and reducing learning time of machine learning algorithms. In this work, a wrapper-based method is utilized where feature selection which is a combinatorial search problem requires a discrete version of ALO to find a subset of features, and feature weighting which is a continuous search problem requires a continuous version of ALO to find the best weights for the features.

Consider P number of antlions (AL) and ants (A), where the position of antlions and ants represents a potential solution in d-dimensional search space. The initial positions of the antlions are selected randomly within the search space of the problem. Then, the ants are required to move randomly for exploring the search space which resembles to their nature of searching for food. The random walk of an ant is performed around an antlion (AL_j) which is modeled as follows:

$$X^t = \Big[\, 0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \ldots, cumsum(2r(t_T) - 1) \,\Big] \tag{14}$$

where *cumsum* denotes the cumulative sum, t is current iteration, T is the maximum iterations, and $r(t)$ is a stochastic function which is defined as follows:

$$r(t) = \begin{cases} 1 & if\ rand > 0.5 \\ 0 & otherwise \end{cases} \tag{15}$$

where *rand* is a random number which is generated uniformly in the interval of [0, 1]. The random walks generated with Eq. (14) does not relate to the boundary conditions of the optimization problems. Since ants will update their positions with the random walk, therefore these random walks are kept inside the search space at each iteration. These walks are confined in-between the search range of problem by using min–max normalization method, and it is given as follows:

$$\hat{X}_i^t = \frac{(X_i^t - a_i)(d_i^t - c_i^t)}{b_i - a_i} + c_i^t \tag{16}$$

where a_i represents minimum and b_i represents maximum value of random walk's i^{th} dimension, d_i^t is the lower and c_i^t is the upper bound of i^{th} dimension at t^{th} iteration. Since, random walks of ants are affected by the traps of antlions, the lower and upper bounds are updated to simulate the entrapment of ants in the pits of antlions at each iteration. For an ant (A_m) at t^{th} iteration, it is defined as follows:

$$c_i^t = AL_j^t + c_i^t, \quad \text{and} \quad d_i^t = AL_j^t + d_i^t \tag{17}$$

where AL_j^t represents the position of j^{th} antlion at t^{th} iteration around which ants are trapping. The lower (c) and upper (d) bound vectors represent the walking of ants in the hypersphere around the antlion. The radius of the hypersphere is decreased adaptively to limit the search space for the ants. It is analogous to the ant which is trapped in a pit of antlion and is trying to escape from it, but antlion shoot sands to catch the ant. It is defined as follows:

$$c_i^t = \frac{c_i}{I}, \quad \text{and} \quad d_i^t = \frac{d_i}{I} \tag{18}$$

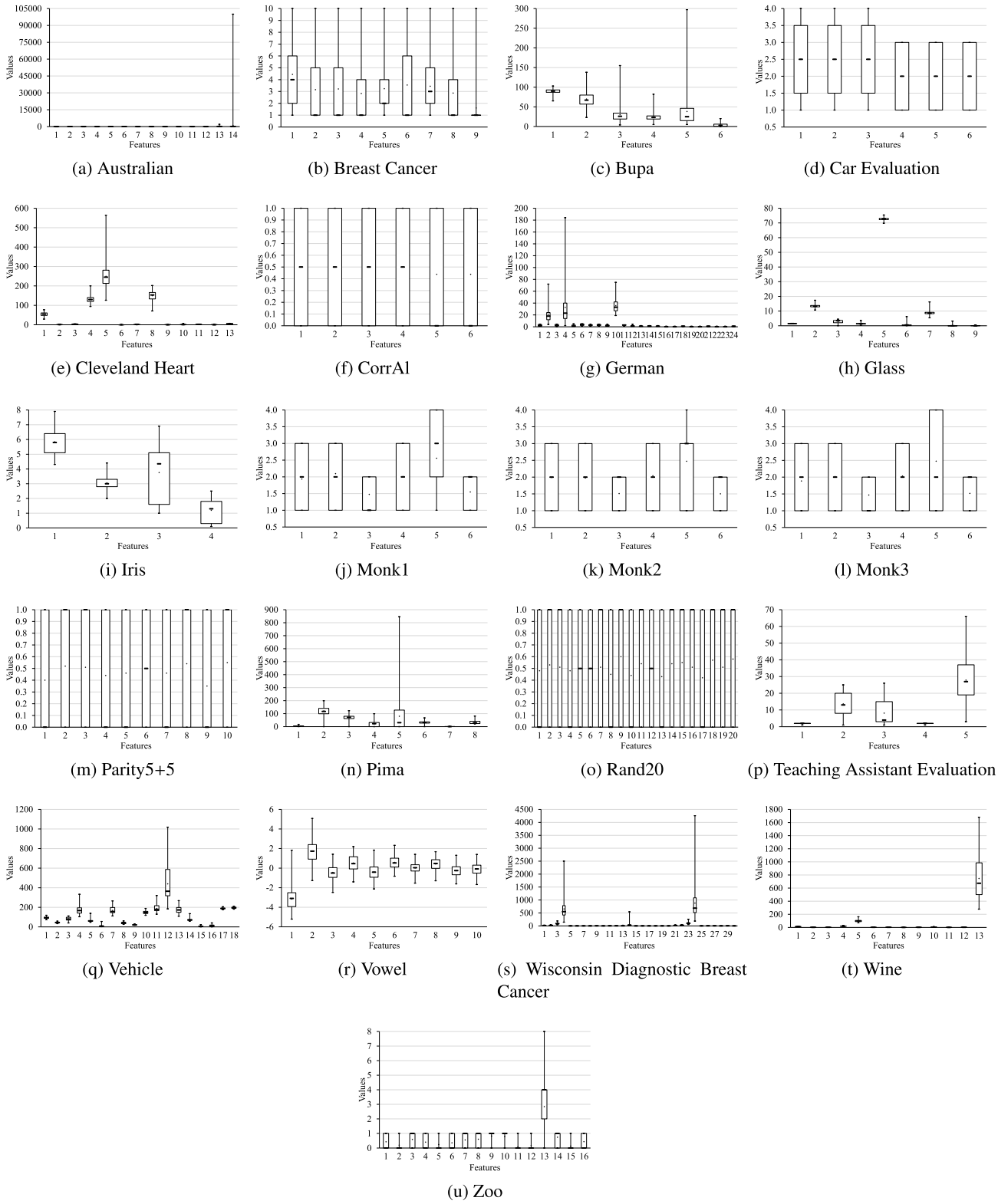
where c_i represents lower and d_i represents the upper bound of i^{th} dimension of the problem at t^{th} iteration and I is the ratio which is given as follows:

$$I = 10^w \frac{t}{T} \tag{19}$$

where w is a parameter which controls the level of exploitation. Elitism is used to sustain the best solutions in the algorithm at any iteration. The newer solutions are generated using two antlions; elite antlion (AL_E^t) obtained at t iteration and an antlion selected with roulette wheel approach (AL_S^t). The elite antlion affects all ants at each iteration whereas roulette wheel selected antlion affects only those ants which are nearby to its positions. In this paper, we use the blend crossover operator (BLX) [71] as a substitute of averaging operation for generating new solutions. This operator has been widely used in many evolutionary algorithms due to its success for convergence towards optimal solutions [72,73]. It enhances the exploration and exploitation capabilities of the ALO algorithm as compare to an averaging operation which gives deterministic positions. It generates new positions randomly from the expanded search range of elite and selected antlion. Suppose, $R_{m,j}^t$ represents the normalized random walk of m^{th} ant around j^{th} antlion at t^{th} iteration. Then, the *BLX* operation is used as follows:

$$\begin{aligned} y_1 &= min(R_{m,E}^t, R_{m,S}^t) - \beta \cdot m \\ y_2 &= max(R_{m,E}^t, R_{m,S}^t) + \beta \cdot m \end{aligned} \tag{20}$$

where $m{=}|R_{m,E}^t - R_{m,S}^t|$ and β is a positive parameter that controls the exploration and exploitation of the search space. Then, a



(a) Australian (b) Breast Cancer (c) Bupa (d) Car Evaluation

(e) Cleveland Heart (f) CorrAl (g) German (h) Glass

(i) Iris (j) Monk1 (k) Monk2 (l) Monk3

(m) Parity5+5 (n) Pima (o) Rand20 (p) Teaching Assistant Evaluation

(q) Vehicle (r) Vowel (s) Wisconsin Diagnostic Breast Cancer (t) Wine

(u) Zoo

Fig. 1. Boxplot indicating feature properties of datasets.

uniform random number is chosen to generate a solution as: $s = rnd(y_1, y_2)$, where rnd is random number generator. In case of feature weighting, the value of s is sufficient to define the new position of the ant.

$$A_i^t = s \tag{21}$$

But in case of feature subset selection, the conversion of continuous to binary values is necessary. For this purpose, we used S-shaped sigmoid function for obtaining discrete values which is the common choice in many wrapper-based feature selection methods [74,75]. It is given as follows:

$$v = \frac{1}{1 + e^{-s}} \tag{22}$$

Table 1
Datasets used in this work for performance analysis along with the abbreviations.

Dataset	Abbreviation	Classes	Features	Instances	Real/Synthetic	Dominance level (Features)	Outliers (Features)
Australian	AUS	2	14	690	Real	High (2)	Yes (5)
Breast Cancer	BC	2	9	683	Real	No	Yes (1)
Bupa	–	2	6	345	Real	low (3)	Yes (5)
Car Evaluation	CE	4	6	1728	Real	No	No
Cleveland Heart	CH	2	13	297	Real	High (3)	Yes (2)
CorrAl	–	2	6	64	Synthetic	No	No
German	–	2	24	1000	Real	Low (3)	Yes (5)
Glass	–	6	9	214	Real	No	Yes (7)
Iris	–	3	4	150	Real	No	No
Monk1	–	2	6	124	Synthetic	No	No
Monk2	–	2	6	169	Synthetic	No	No
Monk3	–	2	6	122	Synthetic	No	No
Parity 5+5	PART	2	10	100	Synthetic	No	No
Pima	–	2	8	768	Real	Moderate (4)	Yes (4)
Rand20	–	2	20	100	Synthetic	No	No
Teaching Assistant Evaluation	TAE	3	5	151	Real	Moderate (1)	No
Vehicle	–	4	18	846	Real	Moderate (7)	Yes (4)
Vowel	–	11	10	528	Real	No	Yes (1)
Wisconsin Diagnostic Breast Cancer	WDBC	2	30	569	Real	High (2)	Yes (22)
Wine	–	3	13	178	Real	High (1)	Yes (1)
Zoo	–	7	16	101	Real	No	No

The new position of the ant is used for evaluating the sigmoid function. The value of Eq. (22) is compared with a random number selected from a uniform distribution in the range of [0, 1] which produces binary value as follows:

$$A_m^t = \begin{cases} 1 & v > rand \\ 0 & otherwise \end{cases} \tag{23}$$

Then, the fitness of newer solutions is measured and the global best solution is updated if a solution with better fitness value is obtained. It is equivalent to the consumption of prey by antlion and laying of its new trap for next prey which is defined as follows:

$$AL_j^{t+1} = A_m^t \quad \text{if} \quad F(A_m^t) > F(AL_j^t) \tag{24}$$

where F is the fitness function that measures the classification accuracy using k-NN classifier. It is given as follows:

$$F = \frac{Correctly\ classified\ instances}{Total\ instances} \tag{25}$$

The pseudo code for finding feature subset and optimal feature weights using modified ALO algorithm is given in Algorithm 1.

5. Experiments and results

The proposed work aims to investigate the impact of data normalization methods on the classification performance. These normalization methods are evaluated considering the full set of features (FULLSET) as well as with feature selection and feature weighting approaches for thorough investigations. The work has been implemented in MATLAB 2016 development environment on a system having Intel Xeon CPU and 8 GB RAM. The publicly available datasets are obtained from the UCI machine learning repository [76]. In this work, a total of 21 datasets are opted to evaluate the impact of normalization methods on classification performance. The description of the datasets is provided in Table 1 which consists of both synthetic and real-world classification problems. The outliers from the features are detected with the help of Grubbs outlier detection test [77] at significance level α = 0.01. The dominant features are identified from each dataset by comparing the different ranges of features. These datasets are evaluated using 10-fold cross-validation on the k-NN classifier. Furthermore, we have performed 30 independent executions to remove the bias from the results which may occur due to the random success of a classifier.

5.1. Properties of data

Fig. 1 illustrates the feature properties of the datasets used in this study where the bottom of the lower whisker represents the minimum value and top of the upper whisker represents the maximum value. The center line represents the median of a feature, and a diamond-shaped marker represents the mean value of the feature. Top and bottom of the box represent 75 and 25 percentiles of a feature respectively. The problems of outliers and dominant features are observed in various datasets. Further, most of the real datasets have outliers as well as dominant features whereas synthetic datasets do not exhibit such problems.

5.2. Parameter settings

The parametric values of the optimization algorithm are kept fixed for all datasets to obtain the results. Following setting of the ALO algorithm is used for experiments: Maximum iterations are set to 200, and a total population of ants and antlions are set to 50 which results in 10,000 fitness function evaluations. Parameter w is set as: w=2 for $t > 0.1T$, w=3 for $t > 0.5T$, w=4 for $t > 0.75T$, w=5 for $t > 0.9T$ and w=6 for $t > 0.95T$ [67]. The value of the β is set to 0.5 as suggested by Eshelman and Schaffer [71]. The value of the k parameter of the nearest neighbor classifier is set within a range of 1 to 10 [78], which is optimized simultaneously along with the selection or weighting of features.

5.3. Impact of normalization on full feature set

Fig. 2 shows the plots of classification accuracy for un-normalized and normalized data with the NN learning algorithm. The outcomes shows that overall no method emerges as superior on most of the datasets. The outcomes also show

Algorithm 1. Feature selection and feature weighting using modified ALO

Input: D: Training data
P: Population of Antlions and Ants
T: Total iterations
Output: Optimal solution (AL_E) for feature subset selection or feature weighting

```
Begin
Randomly initialize a population of antlions
Calculate fitness of antlions using k-NN classifier (eq. 25)
Determine the elite antlion (AL_E^t)
while (fixed number of iterations, T)
   for i=1 to Number of ants (P)
      Select an antlion using Roulette wheel selection (AL_S^t)
      Update c and d vectors (eq. 17)
      Create a random walk around elite antlion and selected antlion using eq. 16
      Update the position of ant for feature weighting (eq. 21) and for feature selection (eq. 23)
   end for
Calculate fitness of all ants (eq. 25)
If an ant becomes fitter than any antlion, then update the position of antlion as:
AL_j^{t+1} = A_m^t   if   F(AL_j^t) < F(A_m^t)
If an antlion becomes fitter than elite antlion, then update the position of the elite as
AL_E^t = AL_j^t   if   F(AL_E^t) < F(AL_j^t)
end while
end
```

that classification performance with un-normalized data is not always poor as compared to normalized data. The normalization methods also have several shortcomings causing complications in data which results in poor performance. The accuracy declines with UN (AUS, CE, CH, Vehicle, WDBC, and Wine), MC (AUS, CE, CH, Vehicle, WDBC and Wine), PS (CH, WDBC and Wine), VSS (Bupa, CH, Glass, Iris, Monk3, PART and Zoo), PT (Vehicle, WDBC and Wine), MMN0 (Monk3), MMN1 (Monk3), MN (Vehicle), DSN (Vehicle) and MMADN (CE, Monk1, Monk3 and Zoo). Thus, sometimes normalization may also complicates the data.

The outcomes also show that the accuracy on the datasets improves with the increase in nearest neighbors (i.e., k) for both un-normalized and normalized data but with few exceptions. Six datasets (Glass, Monk2, Rand20, TAE, Vowel, and Zoo) shows a decline in accuracy with the increase in k-value. The improvement in accuracy with the increase of k value is due to either nonlinear separation of classes or the presence of noise in data. On the other hand, the decline in accuracy with the increase of k value is due to the marginal difference in decision boundaries of classes. The decision boundaries between classes depend upon the value of k where higher k value smoothens the boundaries while lower values produce uneven boundaries [61].

PT is the best performing method that obtains mean accuracy of 80.97% (considering maximum accuracy from all values of k) over all datasets whereas MMADN is the worst performing method that obtains mean accuracy of 77.33%. PT achieves maximum accuracy on five datasets, whereas VSS achieves minimum accuracy on seven datasets. The significant improvement in accuracy, considering first and second best rank method, is observed for Bupa (PT=71.40% and UN=68.53%), Monk2 (PT=87.62%, and TN=75.18%) and Rand20 (MMADN=67.91% and 63.94%) datasets. The significant difference between maximum and minimum accuracy is observed for Australian (15.19%), Bupa (10.85%), CH (16.12%), Glass (13.96%), Iris (9.51%), Monk1 (26.16%), Monk2 (24.87%), Monk3 (29.32%), Rand20 (7.07%), Vehicle (7.66%) and Wine (21.16%) datasets.

From the outcomes of full feature set, it is observed that data pre-processing using normalization is crucial for obtaining an effective predictive model of the classifier. Moreover, most of the methods helps to improve the accuracy, and un-normalized data does not produce least accuracy for all datasets Among the normalization methods, no single method outperforms others. It is because different features have different properties which a single normalization cannot handle. Some features may be normalized better but others are still causing problems in classification. Hence, these methods cannot be generalized, rather the choice of a method depends upon the properties of the data. Further, the impact of data normalization on classification performance is analyzed when feature selection and feature weighting approaches are utilized independently.

5.4. Impact of normalization with feature selection and feature weighting

In this section, the outcomes of feature selection and feature weighting approaches have been discussed in terms of classification accuracy, the percentage of the feature selected and runtime.

5.4.1. Classification accuracy

The minimum, maximum, mean and standard deviation of accuracies are measured to analyze the impact of feature selection and feature weighting on classification. Table 2 summarizes the outcomes of binary ALO that searches for a relevant subset of features from the un-normalized as well as normalized data. The important findings from the outcomes are as follows.

PT method is the best performer which obtains a mean accuracy of 84.39% on all datasets whereas UN data is the worst performer that obtains the mean accuracy of 82.77%. The major variations in the mean accuracies are observed on eight datasets; CE (TN=95.46% and UN=91.23%, difference of 4.22%), CH (TN=84.31% and MC= 81.41%, difference of 2.90%), German (PS=75.01% and MMADN=71.92%, difference of 3.09%), Glass (both LS and HT=80.55% and VSS=65.89%, difference of 14.66%), Monk2 (PT=87.62% while other methods have $\approx$ 65%–75%, difference of more than 12%), WDBC (MMN1=97.66% and UN=94.59%, difference of 3.07%), Wine (ZSN=99.47% and MC=95.86%, difference of 3.60%) and Zoo (Both MC and PS=100% and MMADN=96.41%, difference of 3.59%).

PS and MC methods achieve 100% mean accuracy on Zoo data. These results are interesting since the MC method has performed below average on other datasets. Furthermore, the PS method

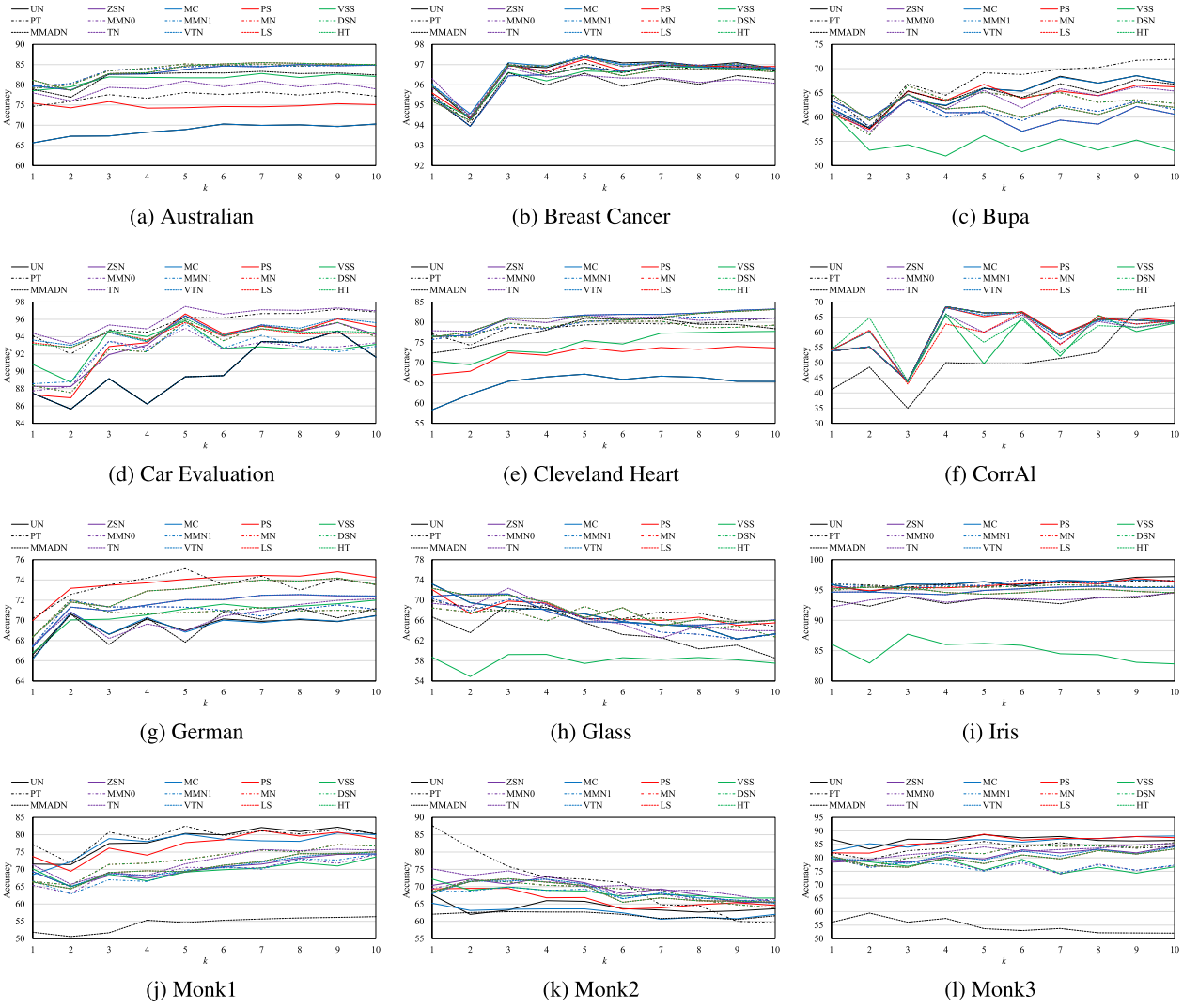


Fig. 2. A comparison of accuracy with un-normalized and normalized data for 21 datasets considering k parameter from 1 to 10. UN=Un-Normalized data, ZSN=Z-Score Normalization, MC=Mean Centered, PS=Pareto Scaling, VSS=Variable Stability Scaling, PT=Power Transformation, MMN0=Min–Max Normalization [0, 1], MMN1=Min–Max Normalization [−1, 1], MN=Max Normalization, DSN=Decimal Scaling Normalization, MMADN=Median and Median Absolute Deviation Normalization, TN=*Tanh* Normalization, VTN= Variant of *Tanh* Normalization, LS=Logistic Sigmoid and HT=Hyperbolic Tangent.

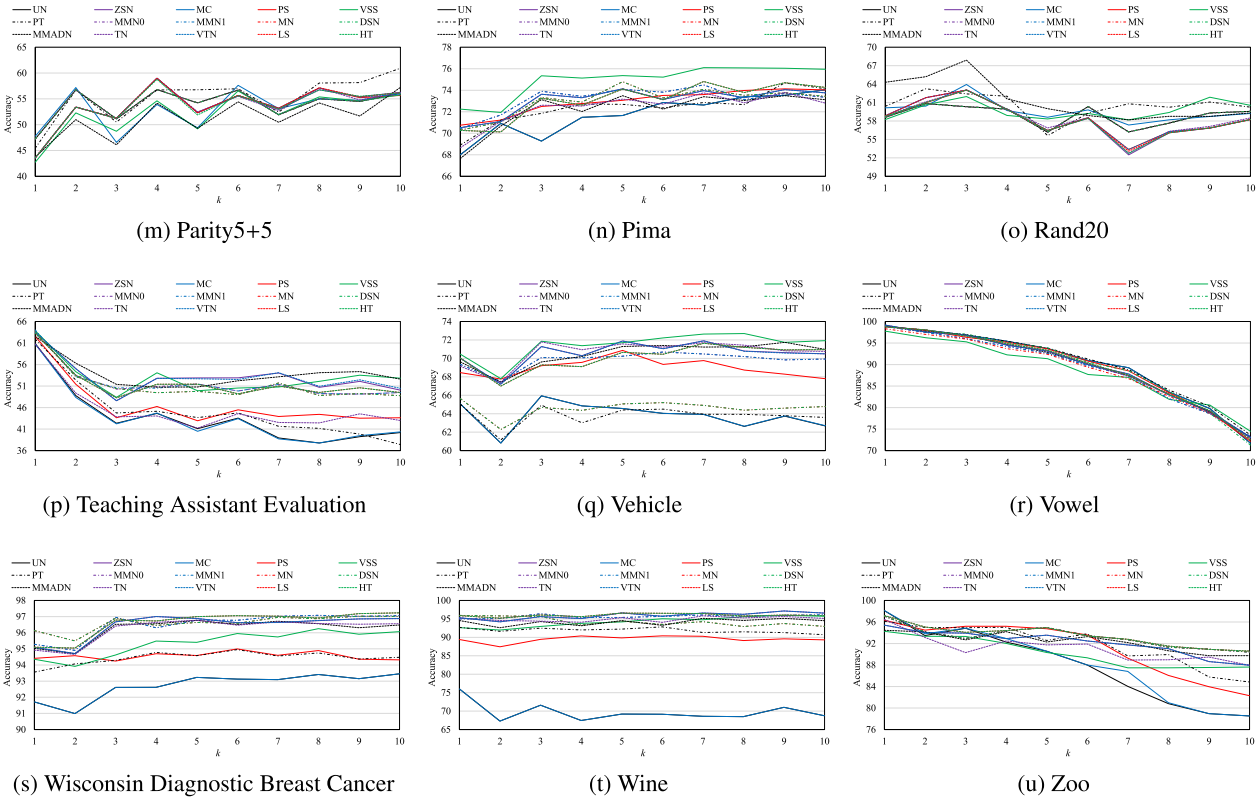
achieves maximum accuracy in five datasets whereas MMADN achieves minimum accuracy on six datasets surpassing UN data that achieves minimum accuracy on five datasets. The classification accuracy of 100% is achieved for Monk1, Wine and Zoo datasets at least once by 2, 8 and 7 (+UN) different normalization methods respectively.

Table 3 summarizes the outcomes of continuous ALO that searches for the best feature weights for un-normalized and normalized data. The important findings from the outcomes are as follows.

PT method is the best performer that obtains a mean accuracy of 87.41% on all datasets whereas MMADN method is the worst performer that obtains the mean accuracy of 85.97%. The significant variations in the mean accuracy observed in five datasets only; CH (TN=87.77% and MMADN=84.17%, difference of 3.59%), German (VTN=79.59% and MMADN=74.29%, difference of 5.30%), Glass (LS and MTS≈87%, and PT=77.58%, a difference of 10.07%), and Monk2 (PT=92.48% while the other methods have ≈ 68%–82%, a difference of minimum 10%). TN method performed well with feature weighting, obtains maximum accuracy on seven datasets. The classification accuracy of 100% is achieved on Monk1, Vowel, Wine and Zoo datasets at least once by different normalization methods. It is achieved by all methods (+UN) except VTN, LS and HT on Monk1 data, only HT method on Vowel data, and all methods along with UN on Wine data and all methods (+UN) except MMADN on Zoo data. However, the mean accuracy of 100% is not obtained on any dataset which shows that ALO algorithm stagnate at local optimal solution.

5.4.2. Search capabilities of ALO

The standard deviation of the outcomes is measured to determine the capability of ALO for searching optimal feature subsets and feature weights. Binary ALO shows better searchability with a mean standard deviation of around 0.78% for all datasets. The higher deviations are observed on CE, Monk2, Rand20, and TAE datasets. MN method exhibits the maximum standard deviation of 2.45% on CE dataset. On the other hand, the mean standard deviation with feature weighting is around 0.97% which also shows good convergence capabilities of ALO algorithm. The datasets with maximum deviations are Glass, Monk2, PART, Rand20, and



(m) Parity5+5 (n) Pima (o) Rand20

(p) Teaching Assistant Evaluation (q) Vehicle (r) Vowel

(s) Wisconsin Diagnostic Breast Cancer (t) Wine (u) Zoo

Fig. 2. (*continued*).

TAE. The maximum deviation of 3.55% is observed on Rand20 data with the LS method.

The variations in the outcomes indicate that ALO converges to local minima more often on the synthetic data as compared to real-world data. Furthermore, the mean accuracy of 100% is achieved on two datasets with binary ALO for feature selection while continuous ALO for feature weighting is unable to converge at optimal solutions for all independent runs. It is because searching for feature weights require larger search space as compared to a feature subset selection. Therefore, sometimes the optimization algorithms are unable to converge to the optimal solution as observed in the case of feature weighting.

5.4.3. Comparison of full feature set, feature selection, and feature weighting

Fig. 3 shows a comparison of classification accuracies for unnormalized and normalized data using the full set of features, feature selection, and feature weighting approaches. The improvements in the performance demonstrate the effectiveness of FS and FW approaches for obtaining better classification models as compared to FULLSET. Further, selecting a relevant feature subset or best feature weights along with the parameters of the classifier simultaneously also contribute to accurate learning. The outcomes reveal the merits of the FW approach in contrast to the FS approach when the goal is to maximize the classification performance. However, FW approach is susceptible to the local minimum problem as observed in Section 5.4.2.

5.4.4. Feature reduction

Removing irrelevant and redundant features from the data help to improve the classification performance and reduce the runtime of the machine learning algorithms. The impact of data normalization on the relevance of the features is measured by analyzing the amount of features reduced with feature selection and feature weighting approaches. Table 4 summarizes the mean percentage of feature reduced with feature selection approach over 30 independent runs. It is observed that a maximum features are reduced with MMADN method while the minimum features are reduced with MMN0 method from all datasets. The mean percentage of features reduced from MMADN method is 41.60%, and MMN0 method is 35.32%. Maximum features are reduced from Australian dataset whereas minimum features are reduced from the CE dataset. The significant variations in feature reduction is observed on Australian (MC=79.76% and MN=46.67%), CH (MC=53.85% and ZSN= 31.54%), Glass (VSS=48.89% and PS=23.70%), Iris (VSS=54.17% and ZSN=17.50%), Monk2 (MC=31.67% and PT= 0%) and Zoo (MMADN=45.83% and DSN=17.29%) datasets.

Table 5 summarizes the mean percentage of feature reduced with feature weighting approach. It is observed from the outcomes that maximum features from all datasets are reduced with MMADN method whereas minimum features are reduced with VTN method. The mean percentage of features reduced with MMADN data is 13.93%, and feature reduction with VTN method is 6.97%. Most of the features are reduced from the Monk1 dataset with all normalization methods. The maximum features (48.33%) are reduced from the Monk1 dataset when data is normalized with MMADN method. All features of CE dataset are relevant for weight assignment to achieve maximum accuracy. The notable variations in feature reduction are observed on Australian (MC=40.48% and ZSN=1.19%), CorrAL (LS=27.78 and MMADN=1.11%), Glass (MMADN=28.52% while other methods have reduced $\approx$ 1%–10% only) and Monk3 (MMADN=37.22% and TN=10.56%) datasets.

Fig. 4 shows a comparison of feature reduction with feature selection and feature weighting approaches. A large difference is observed in the outcomes as feature selection eliminates 3 to 5 times approximately more features as compared to feature weighting. It is because, in the feature selection, a feature is either selected or rejected whereas, in the feature weighting approach,

Table 2
Accuracies obtained with feature selection approach for un-normalized and normalized data.

Datasets																					
Methods		AUS	BC	Bupa	CE	CH	CorrAl	German	Glass	Iris	Monk1	Monk2	Monk3	PART	Pima	Rand20	TAE	Vehicle	Vowel	WDBC	Wine	Zoo
UN	Min.	83.33	96.78	65.80	90.28	79.63	68.21	71.90	72.91	96.67	95.83	65.99	90.9	63.94	73.43	72.95	59.42	71.86	98.85	94.02	95.03	99.00
	Max.	86.09	97.66	71.90	95.43	85.56	69.76	74.30	76.12	98.00	98.46	70.43	93.65	68.08	76.04	79.19	64.94	74.00	99.63	95.09	96.63	100
	Mean	85.23	97.27	68.51	91.23	81.56	68.96	73.12	74.47	97.18	98.31	67.88	92.81	66.14	74.61	76.03	63.02	72.84	99.30	94.59	95.88	99.29
	Std.	0.61	0.21	1.50	1.33	1.05	0.39	0.50	0.72	0.52	0.47	1.12	0.90	1.05	0.63	1.53	1.53	0.61	0.17	0.33	0.36	0.37
ZSN	Min.	85.65	96.93	65.82	90.51	82.22	68.04	73.10	78.50	96.00	96.67	67.48	90.83	64.35	74.11	72.99	60.18	73.42	98.51	97.19	98.89	98.18
	Max.	87.27	97.51	69.85	96.81	84.81	69.76	76.5	81.30	98.00	98.52	74.57	93.65	68.70	77.35	79.37	66.76	75.16	99.45	98.25	100	99.17
	Mean	86.56	97.18	67.75	92.65	83.37	68.93	74.43	79.57	96.87	98.02	70.86	92.43	66.34	75.68	76.37	64.25	74.33	99.26	97.58	99.47	99.01
	Std.	0.40	0.15	1.13	1.95	0.68	0.42	0.82	0.74	0.43	0.70	1.95	0.97	1.19	0.68	1.67	1.65	0.46	0.20	0.25	0.23	0.17
MC	Min.	84.20	96.93	66.41	90.28	79.26	68.21	71.90	72.91	96.00	95.83	65.13	91.03	64.92	73.30	73.61	60.09	71.85	98.85	93.85	95.09	100
	Max.	86.25	97.52	71.86	95.08	84.44	69.76	74.20	76.12	98.00	98.57	69.98	95.06	67.99	75.79	80.09	65.61	74.00	99.63	95.27	96.63	100
	Mean	85.31	97.28	68.89	91.59	81.41	68.94	73.05	74.47	97.00	98.15	66.91	92.79	66.24	74.74	76.56	63.62	72.88	99.30	94.63	95.86	100
	Std.	0.46	0.16	1.44	1.52	1.09	0.39	0.57	0.72	0.45	0.56	1.23	0.97	0.86	0.60	1.85	1.49	0.61	0.17	0.34	0.36	0.00
PS	Min.	84.49	96.93	67.53	90.28	79.63	68.21	73.40	72.48	96.00	96.67	66.89	91.73	64.93	74.48	74.29	59.29	72.47	98.68	95.96	96.54	100
	Max.	86.39	97.66	70.47	96.93	83.33	73.81	76.90	75.87	98.00	98.52	72.28	93.65	70.72	77.22	81.12	65.67	74.47	99.63	97.01	98.36	100
	Mean	85.36	97.38	68.97	92.97	81.6	69.51	75.01	74.10	97.09	98.01	69.49	93.09	66.76	75.81	76.88	62.94	73.45	99.28	96.51	97.66	100
	Std.	0.44	0.19	0.89	2.28	0.81	1.13	0.83	0.83	0.51	0.70	1.33	0.57	1.33	0.68	1.73	1.44	0.51	0.22	0.26	0.52	0.00
VSS	Min.	85.65	96.78	65.48	90.68	80.00	68.21	72.60	64.12	96.00	96.67	69.20	90.96	63.94	74.74	75.37	61.29	72.34	97.33	96.83	97.16	97.84
	Max.	87.85	97.37	70.39	96.12	84.44	70.48	75.10	68.72	97.33	98.52	75.08	93.65	68.08	77.73	80.21	66.17	74.70	98.88	97.72	97.80	99.00
	Mean	86.43	97.10	67.30	92.65	81.62	68.99	73.80	65.89	96.60	98.01	71.72	92.59	66.18	76.06	77.82	64.26	73.64	97.97	97.29	97.31	98.20
	Std.	0.45	0.16	1.08	1.78	0.81	0.48	0.70	1.22	0.47	0.70	1.55	0.97	1.18	0.72	1.23	1.38	0.57	0.37	0.23	0.21	0.31
PT	Min.	84.78	96.78	66.11	93.58	81.85	68.21	73.20	71.51	96.00	99.17	85.21	89.42	63.94	73.70	72.06	59.38	71.98	98.70	95.96	97.15	98.09
	Max.	87.10	97.66	73.62	97.45	86.67	69.76	76.10	76.24	98.00	100	89.99	93.65	68.30	77.08	80.41	65.43	74.72	99.81	96.84	98.33	100
	Mean	85.69	97.27	70.00	95.07	83.79	68.97	74.74	74.38	96.78	99.97	87.62	91.62	66.20	75.71	76.57	62.45	72.88	99.40	96.40	97.67	99.00
	Std.	0.58	0.21	1.75	1.30	0.98	0.40	0.71	1.30	0.35	0.15	1.16	1.07	1.20	0.80	1.96	1.40	0.58	0.31	0.25	0.29	0.31
MMN0	Min.	85.79	97.07	66.09	90.28	81.85	68.21	72.40	77.63	96.00	96.67	67.33	90.90	64.03	74.48	73.06	60.89	73.89	98.30	97.19	98.86	99.00
	Max.	88.11	97.66	69.93	95.20	86.30	69.76	75.90	80.38	97.33	98.52	74.60	93.65	69.30	76.70	80.90	66.89	75.75	99.27	98.25	100	100
	Mean	86.43	97.33	67.57	92.37	83.31	68.96	74.11	79.15	96.73	98.01	70.29	92.45	66.24	75.5	76.09	64.28	74.85	98.93	97.59	99.41	99.29
	Std.	0.51	0.15	0.99	1.35	0.90	0.39	0.69	0.81	0.40	0.70	1.9	0.88	1.17	0.62	1.72	1.56	0.47	0.22	0.24	0.25	0.40
MMN1	Min.	85.35	96.92	65.49	90.28	81.85	68.21	72.4	77.63	96.00	96.67	67.07	90.13	63.87	74.21	73.24	61.56	73.89	98.30	97.36	98.89	99.00
	Max.	87.68	97.51	70.68	96.01	84.81	70.48	76.20	80.38	97.33	98.52	71.63	93.65	68.08	76.17	77.37	67.56	75.88	99.27	98.24	100	100
	Mean	86.40	97.25	67.46	92.40	83.37	68.99	73.87	79.15	96.82	98.01	69.25	92.58	66.06	75.28	75.53	64.57	74.88	98.93	97.66	99.39	99.29
	Std.	0.56	0.19	1.12	2.03	0.78	0.48	0.75	0.81	0.34	0.70	1.23	1.11	1.13	0.54	1.18	1.56	0.46	0.22	0.21	0.22	0.40
MN	Min.	85.22	96.78	65.75	90.39	82.22	68.10	72.90	70.17	96.00	96.67	66.85	90.83	63.72	74.35	72.04	60.70	72.91	97.92	97.18	98.30	99.00
	Max.	87.38	97.66	69.88	96.70	85.19	69.76	75.80	73.84	97.33	98.52	72.66	93.53	68.08	76.69	79.28	66.8	75.20	98.88	98.06	99.47	100
	Mean	86.53	97.25	67.73	92.69	83.72	68.96	74.20	72.02	96.69	98.01	69.68	92.36	66.03	75.33	75.97	64.34	73.64	98.47	97.63	98.95	99.29
	Std.	0.55	0.19	1.10	2.45	0.87	0.41	0.69	0.97	0.37	0.70	1.68	1.01	1.16	0.53	1.48	1.56	0.53	0.23	0.22	0.30	0.40
DSN	Min.	85.64	96.92	64.97	90.39	82.22	68.21	73.10	70.17	96.00	96.67	66.47	90.90	63.83	74.09	72.22	60.70	72.79	98.67	97.36	98.30	99.00
	Max.	87.96	97.80	70.71	96.24	84.81	69.76	75.70	73.84	97.33	98.52	72.81	93.46	69.30	77.60	79.67	66.80	74.49	99.43	98.07	99.47	100
	Mean	86.56	97.31	67.64	92.48	83.27	68.96	74.39	71.92	96.73	98.01	69.27	92.40	66.27	75.20	76.09	64.33	73.64	99.16	97.64	98.93	99.29
	Std.	0.53	0.22	1.19	1.94	0.73	0.41	0.66	0.97	0.32	0.70	1.73	0.95	1.30	0.73	1.67	1.50	0.49	0.16	0.17	0.32	0.40
MMADN	Min.	87.39	96.48	65.55	90.28	80.00	65.89	71.00	78.55	96.00	98.33	63.24	90.06	62.79	74.22	73.90	61.33	72.94	98.86	97.02	98.33	95.96
	Max.	88.27	97.08	71.85	94.90	84.07	70.24	73.00	80.72	97.33	98.52	66.40	93.46	66.39	77.08	81.01	67.54	74.82	99.45	98.24	100	97.26
	Mean	87.92	96.81	69.10	91.35	82.19	68.69	71.92	79.58	96.69	98.39	64.94	91.87	64.25	75.61	76.64	65.13	73.93	99.20	97.55	99.04	96.41
	Std.	0.28	0.17	1.58	1.28	1.08	0.77	0.49	0.65	0.21	0.05	0.79	1.22	1.12	0.68	2.01	1.61	0.49	0.15	0.26	0.32	0.37
TN	Min.	85.37	96.92	67.51	93.75	83.33	68.21	73.50	70.58	96.00	98.33	72.18	90.06	64.25	73.69	73.26	59.99	72.59	98.86	97.18	98.86	97.98
	Max.	87.11	97.80	73.06	97.91	86.67	70.71	75.80	75.29	97.33	100	76.77	93.65	68.14	76.82	79.19	65.63	74.82	99.44	98.06	100	98.26
	Mean	85.94	97.28	69.31	95.46	84.31	69.13	74.73	73.58	96.56	99.59	74.94	92.13	66.31	75.44	76.73	63.42	73.53	99.23	97.48	99.16	98.12
	Std.	0.46	0.22	1.27	1.37	0.66	0.61	0.59	0.93	0.39	0.70	1.21	1.17	1.08	0.79	1.43	1.59	0.51	0.15	0.24	0.40	0.08
VTN	Min.	85.66	96.78	66.06	93.06	82.22	68.21	72.80	78.50	96.67	96.67	66.75	90.06	64.35	73.96	73.71	60.18	73.52	98.51	97.19	98.89	98.89
	Max.	88.11	97.66	70.71	96.47	85.93	70.71	75.80	81.30	97.33	98.52	75.34	93.65	68.12	77.08	81.96	66.29	75.39	99.45	98.24	100	99.17
	Mean	86.73	97.19	67.54	94.24	83.51	68.96	74.01	79.57	96.87	98.01	70.28	92.19	66.37	75.5	77.03	63.99	74.42	99.26	97.58	99.47	99.04
	Std.	0.59	0.19	1.29	0.96	0.86	0.50	0.66	0.74	0.31	0.70	2.26	1.09	1.12	0.60	1.87	1.57	0.45	0.20	0.24	0.23	0.08

(*continued on next page*)

Table 2 (*continued*).

Datasets																					
Methods		AUS	BC	Bupa	CE	CH	CorrAl	German	Glass	Iris	Monk1	Monk2	Monk3	PART	Pima	Rand20	TAE	Vehicle	Vowel	WDBC	Wine	Zoo
LS	Min.	85.52	96.92	65.82	93.06	81.85	68.04	72.90	78.95	96.00	96.67	65.68	89.42	64.88	74.49	73.67	60.23	73.40	98.49	97.19	98.86	98.18
	Max.	87.69	97.81	69.89	95.66	85.19	69.76	75.80	82.29	97.33	98.52	74.07	93.59	68.76	76.95	80.10	66.76	75.77	99.43	98.07	100	99.23
	Mean	86.73	97.32	67.47	93.86	83.52	68.91	74.38	80.55	96.80	98.01	69.51	91.86	66.73	75.49	76.54	64.01	74.40	99.04	97.48	99.06	99.04
	Std.	0.59	0.22	0.95	0.67	0.82	0.38	0.66	0.86	0.41	0.70	2.07	1.03	1.24	0.53	1.50	1.61	0.60	0.20	0.20	0.30	0.18
HT	Min.	85.79	96.78	65.21	92.71	82.22	68.04	73.50	78.95	96.00	96.67	66.75	90.06	64.35	74.08	74.06	60.23	73.64	98.49	97.19	98.86	98.89
	Max.	87.83	97.52	71.06	96.93	84.81	72.38	75.50	82.29	97.33	98.52	72.55	93.53	68.70	76.70	80.21	66.76	75.40	99.43	97.89	100	99.23
	Mean	86.77	97.23	67.89	94.17	83.51	69.08	74.37	80.55	96.84	98.01	69.44	91.81	66.45	75.57	77.25	64.01	74.50	99.04	97.54	99.06	99.07
	Std.	0.52	0.16	1.35	1.32	0.63	0.79	0.56	0.86	0.46	0.70	1.65	1.02	1.20	0.65	1.50	1.61	0.45	0.20	0.20	0.29	0.07

UN=Un-Normalized data, *Z*SN=*Z*-Score Normalization, MC=Mean Centered, PS=Pareto Scaling, VSS=Variable Stability Scaling, PT=Power Transformation, MMN0=Min–Max Normalization [0, 1], MMN1=Min–Max Normalization [−1, 1], MN=Max Normalization, DSN=Decimal Scaling Normalization, MMADN=Median and Median Absolute Deviation Normalization, TN=*Tanh* Normalization, VTN=Variant of *Tanh* Normalization, LS=Logistic Sigmoid and HT=Hyperbolic Tangent.

Table 3
Accuracies obtained with feature weighting approach for un-normalized and normalized data.

Datasets																					
Methods		AUS	BC	Bupa	CE	CH	CorrAl	German	Glass	Iris	Monk1	Monk2	Monk3	PART	Pima	Rand20	TAE	Vehicle	Vowel	WDBC	Wine	Zoo
UN	Min.	85.51	97.66	71.33	97.63	84.81	68.57	76.20	75.47	97.33	98.45	73.53	93.53	67.15	77.6	76.17	66.8	74.24	99.25	95.79	97.68	99.00
	Max.	89.72	98.53	76.22	98.67	88.89	72.68	79.50	81.68	98.67	100	81.39	96.86	77.21	79.82	91.05	73.34	77.91	99.82	96.84	100	100
	Mean	87.53	98.03	73.03	98.22	87.06	69.84	78.13	78.58	98.11	99.82	77.81	95.62	71.51	78.69	82.81	70.68	76.16	99.61	96.34	98.53	99.85
	Std.	1.32	0.23	1.02	0.29	0.95	1.03	0.85	1.59	0.39	0.44	1.90	0.71	2.08	0.50	3.49	1.85	0.92	0.17	0.25	0.58	0.35
ZSN	Min.	87.97	97.51	71.95	97.74	85.93	68.81	76.30	84.58	96.67	97.74	72.86	93.40	67.87	77.22	78.52	69.14	75.07	99.23	97.73	98.89	99.00
	Max.	90.88	98.68	76.29	98.55	88.89	72.68	81.50	87.83	98.67	100	81.67	96.09	74.19	80.86	89.23	74.85	79.55	99.82	98.59	100	100
	Mean	89.48	98.01	74.38	98.10	87.17	69.96	79.05	86.33	97.87	99.79	78.33	95.22	71.02	79.33	83.11	71.53	77.30	99.64	98.11	99.56	99.82
	Std.	0.66	0.25	1.12	0.29	0.76	1.13	1.41	0.77	0.48	0.53	1.99	0.87	1.55	0.84	2.89	1.52	1.37	0.20	0.20	0.34	0.37
MC	Min.	85.49	97.66	71.26	97.63	83.33	68.33	76.90	76.51	97.33	98.46	72.64	94.29	67.08	77.48	79.59	66.39	74.23	99.23	95.78	97.11	99.00
	Max.	90.27	98.39	75.66	98.50	88.89	72.62	79.30	82.11	98.67	100	81.27	96.03	75.97	79.81	91.74	74.80	78.37	99.82	96.84	100	100
	Mean	87.44	97.99	72.97	98.17	86.84	69.61	78.18	78.91	98.04	99.87	77.94	95.47	71.36	78.81	84.58	70.71	76.22	99.61	96.33	98.6	99.82
	Std.	1.48	0.20	1.06	0.30	1.13	1.11	0.72	1.49	0.43	0.36	2.08	0.65	2.21	0.61	3.02	2.15	1.03	0.16	0.26	0.67	0.37
PS	Min.	86.10	97.66	72.47	97.22	84.44	68.81	77.40	75.87	97.33	98.45	74.73	93.53	66.06	78.39	77.10	67.43	75.06	99.25	96.67	98.36	98.33
	Max.	90.88	98.40	76.24	98.67	88.89	73.81	79.90	87.59	98.67	100	82.27	96.09	75.05	80.21	90.78	74.19	79.57	99.82	98.42	100	100
	Mean	89.39	97.99	74.21	98.12	87.09	70.10	78.77	80.04	98.04	99.82	78.61	95.46	70.92	79.34	83.80	71.42	76.75	99.61	97.48	99.39	99.86
	Std.	1.02	0.19	0.96	0.41	0.97	1.33	0.60	2.73	0.39	0.44	2.02	0.61	2.14	0.53	2.94	1.86	1.03	0.15	0.46	0.47	0.39
VSS	Min.	87.80	97.51	70.45	97.51	83.33	68.69	76.80	79.94	97.33	97.63	71.69	93.40	66.48	78.92	78.52	66.14	74.82	98.87	97.19	96.66	98.09
	Max.	90.14	98.25	75.16	98.44	88.52	72.68	80.00	85.52	98.67	100	81.57	96.09	75.90	80.60	90.05	73.57	77.55	99.82	98.60	100	100
	Mean	89.04	97.89	72.31	98.09	85.99	69.93	78.29	83.42	97.96	99.74	78.14	95.33	70.33	79.63	82.51	70.94	76.10	99.50	97.74	98.94	99.82
	Std.	0.53	0.22	1.14	0.26	1.34	1.11	0.87	1.28	0.52	0.66	2.09	0.77	1.72	0.40	2.90	1.82	0.62	0.27	0.35	0.78	0.50
PT	Min.	87.69	97.80	72.71	97.69	85.93	68.57	77.90	75.27	97.33	99.17	90.75	93.46	67.04	77.86	78.84	67.02	75.30	99.23	96.48	97.81	98.09
	Max.	90.72	98.39	76.47	98.84	89.26	72.68	80.40	80.36	98.67	100	94.70	96.86	75.90	80.20	89.25	73.42	79.90	99.82	98.42	100	100
	Mean	89.23	98.08	74.66	98.29	87.40	70.01	79.16	77.58	97.89	99.92	92.48	95.27	70.82	78.95	83.03	70.52	76.99	99.52	97.29	99.08	99.45
	Std.	0.76	0.17	1.06	0.41	0.81	1.16	0.74	1.39	0.31	0.25	1.01	0.94	1.88	0.50	2.97	1.55	1.08	0.20	0.51	0.59	0.57
MMN0	Min.	87.82	97.66	71.03	97.57	85.93	68.57	76.70	83.23	97.33	97.63	74.15	93.40	67.82	78.25	75.64	68.60	75.04	99.22	97.89	98.86	98.00
	Max.	90.15	98.39	76.25	98.90	88.52	72.68	79.80	87.58	98.67	100	82.26	96.73	74.01	80.86	88.34	74.14	78.86	99.82	98.59	100	100
	Mean	88.80	98.00	74.25	98.09	87.30	69.97	78.38	85.56	97.93	99.74	77.81	95.22	70.94	79.41	83.63	71.26	76.78	99.57	98.14	99.38	98.99
	Std.	0.57	0.17	1.21	0.33	0.83	1.22	0.80	1.21	0.37	0.66	1.73	0.94	1.67	0.68	3.04	1.57	0.95	0.20	0.17	0.30	0.85
MMN1	Min.	87.40	97.66	71.29	97.57	85.19	68.81	77.10	82.26	97.33	97.63	72.68	93.40	67.17	78.27	79.48	67.68	75.40	99.05	97.71	98.89	98.09
	Max.	90.30	98.39	77.66	98.56	88.89	72.68	80.50	88.29	98.67	100	80.98	96.79	74.96	80.47	89.85	74.19	79.09	99.82	98.59	100	100
	Mean	88.52	98.03	73.85	98.05	87.05	70.15	78.71	85.62	98.00	99.74	77.02	95.13	71.37	79.39	84.49	71.30	76.81	99.50	98.16	99.46	99.42
	Std.	0.54	0.18	1.23	0.30	0.96	1.18	0.88	1.41	0.35	0.66	1.94	0.91	1.80	0.51	2.76	1.62	0.94	0.21	0.20	0.34	0.62
MN	Min.	87.67	97.65	70.76	97.46	85.93	68.75	77.90	74.69	97.33	97.63	72.94	93.46	65.95	78.25	78.99	67.97	75.17	98.85	97.72	98.33	97.98
	Max.	90.44	98.39	76.22	98.84	88.89	73.81	81.00	81.60	98.67	100	81.88	96.73	74.23	80.85	93.16	74.85	78.74	99.82	98.60	100	100
	Mean	88.74	98.01	74.01	98.29	87.40	69.94	78.96	78.39	98.00	99.74	78.18	95.33	70.74	79.37	84.72	71.51	77.42	99.45	98.12	99.35	99.24
	Std.	0.62	0.19	1.45	0.40	0.83	1.31	0.83	1.69	0.50	0.66	1.94	0.78	1.94	0.70	3.26	1.74	0.82	0.26	0.20	0.47	0.68
DSN	Min.	87.81	97.51	71.62	97.51	85.93	68.57	77.00	74.04	97.33	97.63	73.33	93.59	67.08	77.47	79.79	67.43	74.83	99.22	97.89	98.30	97.98
	Max.	90.29	98.39	75.93	98.84	88.89	72.68	81.30	83.64	98.67	100	81.57	96.73	73.03	81.64	89.34	74.85	79.33	99.82	98.42	100	100
	Mean	88.75	97.99	74.11	98.15	87.36	69.89	78.94	78.67	98.02	99.71	78.23	95.60	70.41	79.23	84.23	71.37	77.23	99.57	98.14	99.30	98.95
	Std.	0.61	0.17	1.00	0.37	0.71	1.14	1.25	2.24	0.48	0.66	2.03	0.75	1.79	0.75	2.73	1.98	1.10	0.17	0.14	0.43	0.57
MMADN	Min.	86.38	97.22	71.88	97.80	83.33	67.80	72.90	80.31	97.33	97.56	66.81	93.40	66.94	77.87	80.03	64.31	75.19	98.87	97.89	98.89	95.87
	Max.	89.29	97.96	76.25	98.73	84.81	70.65	75.50	86.94	98.67	100	69.39	96.03	72.94	81.64	87.87	73.46	78.36	99.81	98.77	100	97.33
	Mean	88.40	97.51	74.47	98.2	84.17	68.81	74.29	84.03	98.09	99.26	68.15	94.90	70.19	79.34	84.28	70.35	76.81	99.58	98.19	99.61	96.73
	Std.	0.61	0.21	1.13	0.24	0.35	0.69	0.66	1.63	0.42	0.89	0.70	0.86	1.76	0.92	2.19	1.90	0.89	0.24	0.24	0.41	0.54
TN	Min.	87.67	97.52	71.31	98.09	86.67	68.69	77.40	76.41	97.33	98.46	78.90	94.17	66.17	77.99	78.94	66.76	75.04	99.03	97.70	98.82	98.18
	Max.	90.87	98.53	75.92	99.36	89.26	75.24	80.50	81.08	98.67	100	87.69	96.73	77.21	80.86	88.23	74.86	78.50	99.82	98.94	100	100
	Mean	89.43	98.09	74.18	98.73	87.77	70.62	79.13	78.96	97.89	99.87	82.29	95.62	71.40	79.35	84.08	71.69	76.93	99.61	98.21	99.57	99.79
	Std.	0.75	0.20	1.15	0.37	0.62	1.93	0.73	1.38	0.39	0.37	2.78	0.58	2.27	0.67	2.41	1.83	0.87	0.20	0.32	0.38	0.46
VTN	Min.	88.11	97.66	71.56	97.86	86.30	68.81	77.10	84.08	97.33	97.56	74.10	93.40	67.96	77.98	77.40	66.76	75.19	99.04	97.72	98.86	98.00
	Max.	90.87	98.25	77.08	98.67	88.89	75.24	81.30	87.98	98.67	99.17	81.88	96.86	74.30	80.59	88.94	73.63	79.08	99.82	98.78	100	100
	Mean	89.50	97.98	74.04	98.14	87.36	70.29	79.59	86.48	98.02	98.31	77.49	95.44	71.17	79.35	83.82	71.21	76.97	99.60	98.20	99.65	99.45
	Std.	0.62	0.16	1.26	0.25	0.66	1.65	1.12	1.09	0.48	0.32	1.76	0.85	1.69	0.61	3.25	1.68	1.05	0.24	0.24	0.37	0.68

(*continued on next page*)

Table 3 (*continued*).

Datasets																					
Methods		AUS	BC	Bupa	CE	CH	CorrAl	German	Glass	Iris	Monk1	Monk2	Monk3	PART	Pima	Rand20	TAE	Vehicle	Vowel	WDBC	Wine	Zoo
LS	Min.	88.26	97.51	71.55	97.69	85.56	68.57	77.30	85.43	96.67	96.73	73.48	93.40	69.06	78.13	76.02	66.10	73.40	98.87	97.88	98.33	98.09
	Max.	91.18	98.25	76.52	98.67	88.89	73.81	80.60	89.40	98.67	99.23	84.29	96.73	75.23	81.38	91.32	73.63	79.19	99.82	98.60	100	100
	Mean	89.63	97.98	73.96	98.23	87.17	70.02	78.55	87.40	97.78	98.36	79.71	95.25	71.30	79.15	84.00	70.93	76.47	99.55	98.17	99.20	99.34
	Std.	0.71	0.19	1.20	0.29	0.81	1.36	0.87	0.90	0.44	0.47	2.88	0.98	1.43	0.73	3.55	1.72	1.17	0.22	0.22	0.38	0.60
HT	Min.	88.25	97.51	72.18	97.69	85.93	68.57	76.90	85.62	96.67	96.73	74.04	93.40	67.34	77.61	77.59	65.47	74.36	99.23	97.89	98.33	99.00
	Max.	90.87	98.25	75.68	98.73	88.89	72.68	81.10	90.19	98.67	99.23	85.79	96.73	77.21	80.08	90.96	74.73	78.38	100	98.59	100	100
	Mean	89.45	97.92	74.15	98.12	87.17	69.85	79.23	87.66	97.84	98.37	79.60	95.31	71.48	79.26	83.83	70.73	76.42	99.59	98.21	99.20	99.58
	Std.	0.58	0.18	1.08	0.33	0.71	1.15	1.06	1.04	0.38	0.45	3.01	0.74	2.06	0.52	3.18	1.99	0.95	0.22	0.17	0.50	0.46

UN=Un-Normalized data, *Z*SN=*Z*-Score Normalization, MC=Mean Centered, PS=Pareto Scaling, VSS=Variable Stability Scaling, PT=Power Transformation, MMN0=Min–Max Normalization [0, 1], MMN1=Min–Max Normalization [−1, 1], MN=Max Normalization, DSN=Decimal Scaling Normalization, MMADN=Median and Median Absolute Deviation Normalization, TN=*Tanh* Normalization, VTN=Variant of *Tanh* Normalization, LS=Logistic Sigmoid and HT=Hyperbolic Tangent.

features are weighted according to their relevance. Therefore, only those features that have zero weight values are eliminated. Furthermore, among normalization methods, overall maximum features are eliminated with MMADN method and minimum features with MN method.

5.5. Runtime

The runtime is measured by incorporating the data normalization with methods and data evaluation with 10-fold cross validation for 30 independent runs. Fig. 5 shows the runtime for FULLSET, FS and FW approaches in *milliseconds* (ms). The average time for FULLSET, FS and FW is 8.78 ms, 6.56 ms, and 8.55 ms respectively. Due to the reduction of features from the original data, the runtime from lower to higher is FS, FW, and FULLSET for all methods except MN and DSN. These two methods exhibit the same traits for normalizing the data. UN requires minimum time in FS and PT requires minimum time in FW as well as in FULLSET whereas TN method requires maximum time due to the additional processing required for the estimation of Hampel estimators from the original data. Moreover, FS has 3–6 times fewer features than FW but its execution time is only 0.25 times slower.

6. Findings and Discussions

6.1. Comparison of un-normalized data against normalization methods

Wilcoxon signed-rank test [79] is used to determine the significance of data normalization for improving classification performance. The outcomes of un-normalized data are compared with the normalization methods considering the full set of features, feature selection, and feature weighting approaches to determine the significant differences in the classification accuracy. It is a non-parametric test which compares the performances of two methods over M datasets. This test is safer and robust than the parametric tests for comparing the classification performances of two methods as suggested in [80]. It measures the differences in the performance of two methods (un-normalized data and a normalization method in this case) for all datasets and compares the ranks based on the positive and the negative differences. The significance level for this test is set to 0.05 as well as 0.01. Table 6 shows the outcomes with this test for the pairwise comparisons of un-normalized data against the normalization methods in terms of average accuracies.

At the significance level 0.05, it is observed that the improvements made with PS and PT methods are significantly better than the un-normalized data when all features are used for classification. All methods except MC, VSS, MN, DSN, and MMADN performed significantly better than the un-normalized data when feature selection is employed. Eight methods performed significantly better than the un-normalized data when feature weighting is employed.

At the significance level 0.01, when all features are used for classification, no method performed significantly better than the un-normalized data. In the case of feature selection, PS, LS and HT methods performed significantly better whereas, in case of feature weighting, PS and TN method performed better than the un-normalized data significantly.

6.2. Comparison of normalization methods

The normalization methods are compared based on the ranking considering mean accuracies obtained with a full set of features, feature selection, and feature weighting approaches. The ranks of normalization methods are calculated and Friedman test [81] is used to determine the statistical differences between the ranks of different normalization methods. This test compares the ranks of the J methods over M datasets. The rank for each method is measured for each dataset where the best-performing method obtains the first rank, the second best method obtains the next rank and so on. The tied outcomes are resolved with the average rank. This test compares the average ranks of J methods on all datasets to test the null hypothesis, i.e., the performance of these methods are not significantly different. The significance level for the acceptance and rejection of the null hypothesis is set to 0.05 in this study.

Fig. 6 shows the mean ranks for FULLSET, FS, and FW approaches. It is observed that top performing methods in all three approaches are different, and no method outperforms others methods. ZSN, PS, and TN obtain best ranks in FULLSET, FS, and FW approaches respectively. Furthermore, the PT method unable to achieve first rank in FULLSET, FS, and FW, despite achieving maximum mean accuracy in all three approaches. PT obtains the 4, 6 and 6 ranks in FULLSET, FS and FW approaches respectively, which is an interesting finding because it outshines other methods on some of the datasets while achieving moderate accuracies on the rest of datasets. In the case of full feature set and feature selection approach, the ranks of the best methods (ZSN and PS respectively) are very close to the ranks of next best-performing methods which shows these methods are also very competitive. In feature weighting, the TN method outshines other methods by larger margin rank-wise, yet does not attain overall highest accuracy.

The outcomes also indicate that UN data is not always the worst representation for the classification, as it has achieved least rank in FS approach only. Other methods such as MC, VSS, and MMADN make the data more complex and complicated for learning as compared to the UN. VSS method attains the least rank in both FULLSET and FW. Furthermore, the performances of min–max based methods and DSN are average on all three approaches.

The data transformation methods (PT, TN, VTN, LS, and HT) have performed better than the scaling methods in case of full feature set and feature selection. However, in the case of feature weighting, the LS method lacks as compared to other transformation methods. One of these methods attain overall best rank in FW approach (TN), but also are very competitive in contrast to FS approach. From the scaling methods, ZSN and PS have performed better than other methods and achieve the best rank in FULLSET and FS approach respectively.

The obtained p-value with the Friedman test is 0.037, 0.059 and 2E-04 for FULLSET, FS, and FW approaches respectively. It shows that the null hypothesis is accepted for FS approaches and therefore, the performance of these normalization methods on feature subset selection is similar. On the other hand, the null hypothesis is rejected for both FULLSET and FW approaches which means the outcomes of the normalization methods are significantly different.

6.3. Findings

In the field of machine learning, the features are extracted from multiple sources to represent the object. For example, in biometric systems, the features are extracted from various

Table 4
Mean percentage of features reduced with feature selection approach.

Datasets	Methods														
	UN	ZSN	MC	PS	VSS	PT	MMN0	MMN1	MN	DSN	MMADN	TN	VTN	LS	HT
AUS	75.24	48.10	79.76	70.24	52.62	61.43	52.38	48.33	46.67	49.29	51.19	63.57	47.86	49.05	50.24
BC	26.67	35.56	32.22	41.11	36.67	35.19	29.26	28.89	28.15	27.04	37.78	34.81	37.04	34.07	37.41
Bupa	30.00	40.56	29.44	26.11	35.56	22.78	33.89	32.78	37.78	31.67	23.89	30.00	37.78	26.67	26.11
CE	15.00	10.00	12.78	8.33	1.11	0.00	7.78	10.00	9.44	9.44	12.78	0.00	0.00	0.56	1.11
CH	51.79	31.54	53.85	43.85	51.79	38.21	48.21	42.05	42.31	38.97	39.74	36.15	39.23	38.46	38.72
CorrAl	46.67	47.78	52.78	35.56	49.44	53.89	46.11	46.11	46.11	51.67	45.00	48.33	48.89	49.44	53.33
German	45.97	43.61	47.36	41.67	47.78	38.61	46.25	45.83	46.67	50.14	51.25	45.00	48.61	44.72	46.53
Glass	28.89	40.00	28.52	23.70	48.89	47.78	34.44	34.44	40.74	42.59	45.19	42.22	40.37	48.52	48.52
Iris	20.83	39.17	17.50	28.33	54.17	28.33	25.00	38.33	34.17	39.17	43.33	45.00	40.00	39.17	36.67
Monk1	50.00	47.78	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
Monk2	10.00	5.56	31.67	13.89	0.56	0.00	9.44	10.00	10.56	12.78	43.33	0.56	13.33	8.89	12.22
Monk3	48.89	52.78	46.11	54.44	55.56	55.00	52.22	58.33	53.89	52.22	50.56	53.33	53.89	53.89	50.56
PART	57.33	49.33	55.67	51.33	58.67	56.33	56.67	58.33	58.33	57.33	58.67	56.67	52.33	51.00	51.33
Pima	36.25	33.33	38.75	38.33	33.75	40.42	37.50	37.08	33.75	35.83	40.83	39.58	41.25	37.92	40.83
Rand20	45.00	48.17	43.67	47.17	52.33	43.67	45.50	45.50	41.67	47.83	45.83	50.67	46.33	48.33	50.17
TAE	40.00	28.00	37.33	22.00	28.00	30.00	26.00	21.33	26.00	26.67	40.00	22.00	24.67	28.00	28.00
Vehicle	46.11	46.11	46.48	44.07	36.30	45.93	35.00	36.85	45.19	48.52	46.67	44.63	46.67	50.00	49.07
Vowel	20.33	10.67	20.33	23.67	16.00	29.00	12.00	13.33	15.00	17.67	21.00	18.33	10.67	17.67	17.33
WDBC	50.89	43.56	48.67	34.22	35.78	32.00	36.67	40.44	41.11	41.11	44.11	42.78	40.44	44.44	40.89
Wine	48.46	33.33	47.95	53.33	39.23	44.87	36.92	37.95	35.64	34.87	36.67	35.38	33.08	38.72	38.72
Zoo	29.38	36.25	28.96	35.42	40.42	32.92	20.42	19.17	17.71	17.29	45.83	41.67	36.46	33.75	35.42

UN=Un-Normalized data, ZSN=*Z*-Score Normalization, MC=Mean Centered, PS=Pareto Scaling, VSS=Variable Stability Scaling, PT=Power Transformation, MMN0=Min–Max Normalization [0, 1], MMN1=Min–Max Normalization [−1, 1], MN=Max Normalization, DSN=Decimal Scaling Normalization, MMADN=Median and Median Absolute Deviation Normalization, TN=*Tanh* Normalization, VTN=Variant of *Tanh* Normalization, LS=Logistic Sigmoid and HT=Hyperbolic Tangent.

Table 5
Mean percentage of features reduced with feature weighting approach.

Datasets	Methods														
	UN	ZSN	MC	PS	VSS	PT	MMN0	MMN1	MN	DSN	MMADN	TN	VTN	LS	HT
AUS	30.95	1.19	40.48	5.00	8.33	4.05	4.05	5.48	3.57	3.10	40.00	4.29	3.10	2.38	4.05
BC	0.74	4.07	1.48	1.85	2.22	2.22	1.48	4.07	1.85	4.07	5.56	2.96	3.33	4.07	2.96
Bupa	1.11	1.67	1.11	1.11	7.22	1.11	0.00	1.11	3.33	2.78	0.00	0.00	0.00	1.11	0.00
CE	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CH	18.21	0.51	13.85	5.64	5.13	2.31	2.31	4.10	4.10	4.62	18.21	1.28	0.77	2.56	1.79
CorrAl	21.67	17.22	23.89	25.56	24.44	26.11	23.89	23.89	18.89	22.78	1.11	10.00	16.11	27.78	23.89
German	0.42	0.14	0.14	0.14	1.25	0.14	0.14	0.28	0.00	0.14	9.44	0.00	0.28	0.00	0.00
Glass	3.70	3.70	2.22	4.07	2.22	6.30	2.22	1.85	10.00	4.81	28.52	7.41	2.96	2.22	2.22
Iris	4.17	13.33	0.00	5.00	20.00	2.50	10.00	5.00	1.67	2.50	6.67	14.17	10.83	13.33	13.33
Monk1	42.22	45.56	43.33	41.11	47.22	41.11	42.78	42.78	41.67	41.11	48.33	47.78	40.00	37.78	33.33
Monk2	1.11	0.00	0.56	0.00	1.67	0.00	0.00	0.56	0.00	2.22	20.56	0.00	0.00	0.00	0.00
Monk3	12.22	17.22	12.22	12.78	17.78	13.89	18.33	21.67	18.89	14.44	37.22	10.56	13.89	17.78	12.22
PART	12.33	11.00	13.67	12.00	17.00	15.33	18.33	11.33	15.00	19.00	19.00	10.67	11.00	11.67	12.67
Pima	4.17	1.25	1.67	2.08	0.00	1.67	1.25	0.83	0.83	0.83	0.42	0.83	1.67	0.00	0.42
Rand20	21.83	13.00	14.50	13.83	16.83	13.00	19.50	16.17	10.83	10.00	12.50	16.50	10.67	11.33	12.50
TAE	4.67	0.00	8.00	0.67	0.00	1.33	0.00	0.00	0.00	0.00	6.67	0.00	0.00	0.00	1.33
Vehicle	2.04	0.00	2.78	0.74	0.19	1.48	1.30	1.30	0.93	1.11	0.37	0.00	0.00	0.56	1.30
Vowel	3.00	3.00	1.67	2.33	0.67	2.00	0.67	2.67	0.67	3.00	1.67	2.67	1.33	2.67	3.33
WDBC	4.22	1.78	4.44	6.22	3.67	6.56	3.89	3.22	1.56	1.89	5.22	2.89	2.22	3.78	3.56
Wine	10.26	6.67	7.69	5.38	7.18	5.13	12.31	12.31	7.69	10.26	9.74	7.18	6.92	11.28	8.21
Zoo	20.42	30.00	22.08	24.17	36.88	26.46	21.04	20.83	18.13	12.08	21.25	28.96	21.25	17.08	21.46

UN=Un-Normalized data, ZSN=*Z*-Score Normalization, MC=Mean Centered, PS=Pareto Scaling, VSS=Variable Stability Scaling, PT=Power Transformation, MMN0=Min–Max Normalization [0, 1], MMN1=Min–Max Normalization [−1, 1], MN=Max Normalization, DSN=Decimal Scaling Normalization, MMADN=Median and Median Absolute Deviation Normalization, TN=*Tanh* Normalization, VTN=Variant of *Tanh* Normalization, LS=Logistic Sigmoid and HT=Hyperbolic Tangent.

sources such as the face, fingerprint, hand-geometry, finger-knuckle bending, and iris. Similarly, in computer vision and multimedia applications, classification of objects such as natural scene requires shape, color or texture features. In the field of medical applications, features can comprise of patient's medical records, age, gender, features from the sound recordings, ElectroEncephaloGram EEG or ElectroCardioGram (ECG).

In statistical terms, the feature properties of the data will be different from each other depending upon the source of the data. Moreover, the properties of the features may also vary within a source due to the availability of the number of feature extraction methods that captures the same information but in different ways. For examples; the texture information regarding the object can be captured with the Local Binary Pattern (LBP) features, Gray-Level Co-Occurrence Matrix (GLCM) features or Gabor features. Therefore, it can be concluded that the features can have different statistical properties.

The normalization of data requires analysis of the features properties to decide on a suitable method. However, the selection of a normalization method is not as straightforward as analyzed from the theoretical aspects and the empirical outcomes of various normalization methods considered in this study. The different normalization methods extract different features properties for normalizing the raw data. In this study, the normalization methods have used, mean, standard deviation, minimum, maximum and median statistical measures of the features. The analysis of the data normalization methods and the experimental outcomes reveal the following findings:

- **Equivalency of MN and DSN:** The outcomes of the MN and DSN are similar in all datasets except Vowel. Both of

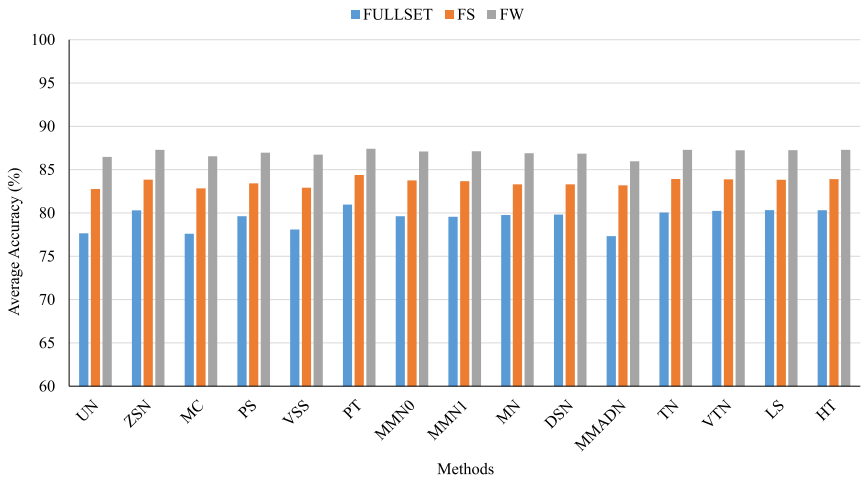


Fig. 3. Comparison of average accuracies for un-normalized and normalized data considering FULLSET, FS and FW. UN=Un-Normalized data, ZSN=*Z*-Score Normalization, MC=Mean Centered, PS=Pareto Scaling, VSS=Variable Stability Scaling, PT=Power Transformation, MMN0=Min–Max Normalization [0, 1], MMN1=Min–Max Normalization [−1, 1], MN=Max Normalization, DSN=Decimal Scaling Normalization, MMADN=Median and Median Absolute Deviation Normalization, TN=*Tanh* Normalization, VTN= Variant of *Tanh* Normalization, LS=Logistic Sigmoid and HT=Hyperbolic Tangent.

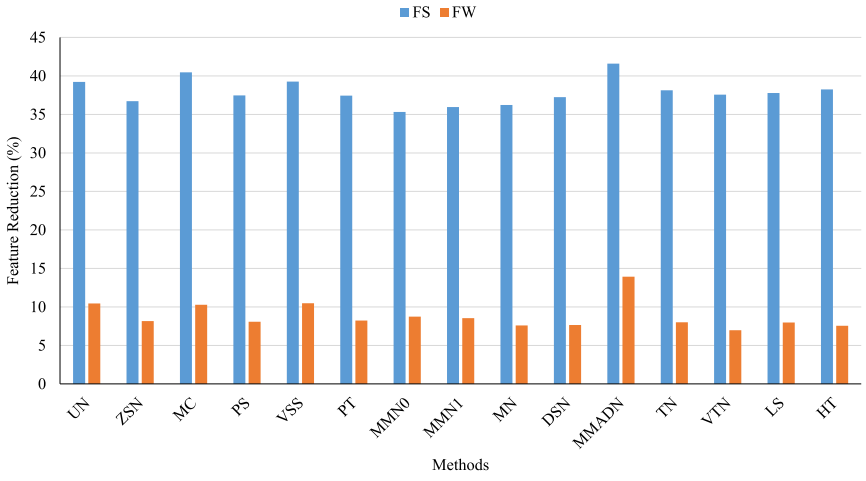


Fig. 4. Average feature reduced for un-normalized and normalized data with feature selection and feature weighting. UN=Un-Normalized data, ZSN=*Z*-Score Normalization, MC=Mean Centered, PS=Pareto Scaling, VSS=Variable Stability Scaling, PT=Power Transformation, MMN0=Min–Max Normalization [0, 1], MMN1=Min–Max Normalization [−1, 1], MN=Max Normalization, DSN=Decimal Scaling Normalization, MMADN=Median and Median Absolute Deviation Normalization, TN=*Tanh* Normalization, VTN= Variant of *Tanh* Normalization, LS=Logistic Sigmoid and HT=Hyperbolic Tangent.

Table 6
Outcomes of Wilcoxon signed-ranks test for pairwise comparison of un-normalized data against the normalization methods in terms of accuracies.

Methods	FULLSET		FS		FW	
	z-value	*p*-value	*z*-value	*p*-value	*z*-value	*p*-value
ZSN	−1.344	0.179	−2.520	0.012**	−2.520	0.012**
MC	+0.459	0.646	−0.966	0.334	0.295	0.768
PS	−2.172	0.030**	−3.146	0.002*	−2.624	0.009*
VSS	−0.261	0.794	−1.269	0.205	−0.365	0.715
PT	−2.165	0.030**	−2.450	0.014**	−1.616	0.106
MMN0	−0.893	0.372	−2.555	0.011**	−1.999	0.046**
MMN1	−1.023	0.306	−2.172	0.030**	−1.894	0.058
MN	−0.806	0.420	−1.493	0.135	−2.138	0.033**
DSN	−0.893	0.372	−1.867	0.062	−1.964	0.050**
MMADN	−0.149	0.881	−0.921	0.357	+0.469	0.639
TN	−1.095	0.274	−2.555	0.011**	−3.354	0.001*
VTN	−1.234	0.217	−2.520	0.012**	−2.033	0.042**
LS	−1.442	0.149	−2.589	0.010*	−1.929	0.054
HT	−1.373	0.170	−2.659	0.008*	−1.825	0.068

*Significance Level at $\alpha = 0.01$.
**Significance Level at $\alpha = 0.05$.

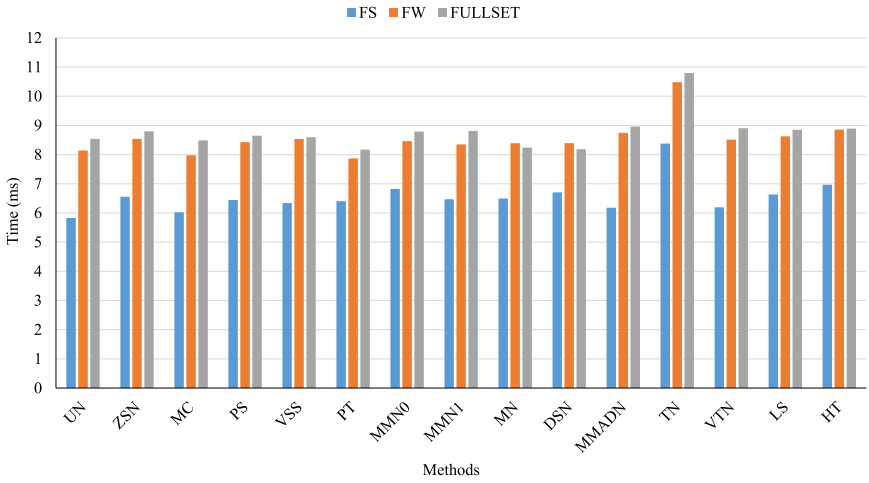
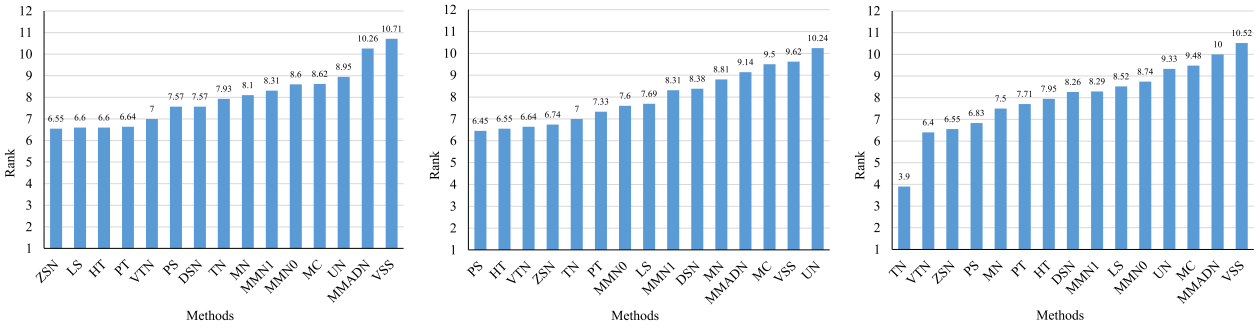


Fig. 5. Runtime for un-normalized and normalized data with full feature set, feature selection and feature weighting. UN=Un-Normalized data, *Z*SN=*Z*-Score Normalization, MC=Mean Centered, PS=Pareto Scaling, VSS=Variable Stability Scaling, PT=Power Transformation, MMN0=Min–Max Normalization [0, 1], MMN1=Min–Max Normalization [−1, 1], MN=Max Normalization, DSN=Decimal Scaling Normalization, MMADN=Median and Median Absolute Deviation Normalization, TN=*Tanh* Normalization, VTN= Variant of *Tanh* Normalization, LS=Logistic Sigmoid and HT=Hyperbolic Tangent.



(a) Full Feature Set (b) Feature Selection (c) Feature Weighting

Fig. 6. Average ranks for un-normalized data and normalization methods. UN=Un-Normalized data, *Z*SN=*Z*-Score Normalization, MC=Mean Centered, PS=Pareto Scaling, VSS=Variable Stability Scaling, PT=Power Transformation, MMN0=Min–Max Normalization [0, 1], MMN1=Min–Max Normalization [−1, 1], MN=Max Normalization, DSN=Decimal Scaling Normalization, MMADN=Median and Median Absolute Deviation Normalization, TN=*Tanh* Normalization, VTN= Variant of *Tanh* Normalization, LS=Logistic Sigmoid and HT=Hyperbolic Tangent.

these methods share the common traits for normalizing the data but in different ways. The MN method divides each feature by its maximum value. DSN method, on the other hand, is based on moving the decimal places so that features are rescaled in the range of [0, 1]. The only difference between these two methods is that the latter method does not rescale the negative values. The Vowel dataset has negative feature values, therefore data is relocated to positive scale before normalization and differences in the accuracies of both methods are observed.

- **Equivalency of Min–Max normalization methods:** In this paper, two variants of MMN methods are considered which rescales the data in the range of [0, 1] (MMN0) and [−1, 1] (MMN1). The outcomes of these methods are similar on most of the datasets with fewer exceptions. The outcomes are similar to the datasets that have binary and real-valued features. In the case of binary-valued features, MMN0 scale the feature with values 0 and 1, whereas MMN1 rescale the feature with values -1 and +1. The latter method only changes the proportions of the values which will be same for all features. Hence, there will be no change in Euclidean distance which KNN used for classification. In the case of real-valued features, only scale of features is changed which also does not affect the Euclidean distance. The only difference in the outcomes of these two methods is observed for the datasets that have features of integer-type only.
- **Equivalency of Sigmoidal normalization methods:** In this paper, two variants of sigmoidal normalization methods have been considered; LS and HT. The outcomes of these methods are also similar on all datasets except only three. In the current setting where the parameter ν of LS method is set to 1, the only difference is the range of transforming the data which is given as follows:

$$HT = LS \times (1 - e^{-q_{i,n}})$$

Therefore, it can be argued that this scaling difference does not affect the Euclidean distance, and the resulting outcomes are same for both methods.

- **Equivalency of MC and MMADN methods:** The MC and MMADN methods may produce similar normalized data when a feature has similar mean and median values, and the MAD is one. Consider the example of CE dataset which has no dominant feature or outlier, and its feature properties have been shown in Fig. 1(d). The outcomes of the UN, MC, and MMADN are similar on this dataset. MC method removes the offset from the features only, but this method has no dominant feature or outlier which is the cause for the

similar performance of both methods. The values of mean and median are similar, and the value of MAD is also one for all features which result in normalization of data with MMADN Eq. (9) method similar to the MC Eq. (2) method, thereby resulting in the same accuracy.

- **Equivalency of ZSN and PS methods:** The only difference between the ZSN and PS method is that the former method uses the standard deviation while the latter method uses the square root of standard deviation for normalization. Therefore, if the standard deviation of feature is one or all feature have the same standard deviation, then the outcomes of classification will be the same for both methods. It has been confirmed from the Partity5+5 and Rand20 synthetic datasets where the standard deviation of all features is approximately 0.50. Therefore, in case of these datasets, ZSN and PS method have different scales which is given as $PS = ZSN \times sqrt(0.5)$.
- **Limitations of the MC method:** With the MC method, only offset is removed from the data. Therefore, the problem of feature dominance is not removed at all, especially if features values have higher numeric range. Consider the Australian dataset, whose last feature is highly dominant and have a range of [1, 100001] with a mean value of 1018.39. Subtracting mean from this range does not affect the feature dominance, thereby leading to classification accuracy similar to the un-normalized data.
- **Problems with Coefficient of Variation**: The presence of Coefficient of Variation (CV) in the VSS make the features highly dominant if original features have a low standard deviation. Therefore, this method forms the dominant features when such conditions occur instead of tackling it. The same has been observed from the datasets considered in this study where the value of CV is greater than 5 from many features in 10 datasets.
- **Limitations of Power Transformation:** This method is ideal for reducing the effects of heteroscedasticity from the un-normalized data. However, it cannot be assured that data always contains features of such characteristics. The same has been observed from the outcomes of the experiments where this method achieves the best accuracy on five datasets, but the outcomes on rest of datasets are moderate.
- **Limitations of MMADN method:** The data normalization with MMADN method requires a MAD statistical measure that has a breakdown point [82] (i.e., $MAD = 0$), when the values of the features are identical (above 50%). However, the probability is higher for the binary or integer type features as compared to the real-valued features. In the context of this work, zero MAD values have been found from several features in the thirteen datasets. Therefore, it causes the degradation of the classification performance as observed from the outcomes of the experiments.
- **Limitations of TN and its variant:** These method pushes the features in a very narrow range which helps to alleviate the problems of feature dominance and outlier. But, the narrow range affects the relevance of features which causes performance issues. These methods does not perform better until feature weighting, which better suits for the data where feature relevance varies [18], is used. Therefore, TN and VTN is more preferred with feature weighting instead of full set of features and feature selection.
- **Influence on feature relevance:** The different outcomes of the normalization methods in terms of classification accuracy and feature reduction signify that these methods have changed the relevance of the features. If the relevance is not altered, the ranks of the methods on features selection and feature weighting would be same to full feature set. Data normalization affects the features properties of the datasets which causes the change in the feature relevance.

From the above findings, it is observed that the normalization methods have different pros and cons for dealing with the problem in the features such as dominance and the presence of outliers. Though data containing all of the ideal features for a particular normalization is not possible, these methods are estimated to have certain limitations as observed from the theoretical and experimental outcomes. Therefore, important observations regarding data normalization are reported as follows:

- It is not necessary that the data normalization always change the statistical properties of the features. Sometimes, the different methods may normalize the data to the same range. Consider the example of CE dataset, where MC and MMADN methods normalize the data to the same range. MN and DSN methods also normalize the data in the same range when features lie on a positive scale and therefore share the common traits.
- Some of the normalization methods share similar traits for normalizing data such as MN and DSN, MMN0 and MMN1, and LS and HT, but have different statistical properties. Therefore, only one method should be utilized for normalizing data if both options are available.
- The data normalization alters the relevance of the features as observed from the outcomes of feature selection and feature weighting approach. Different methods have selected different feature subsets or feature weights for improving the classification performance.

6.4. Discussions

The process of data normalization will remain subjective rather than the objective, as the machine learning applications usually employ features that have different statistical properties. However, a general trend is observed from the outcomes which show that measures (mean and standard deviation) are more capable than the others for effective data normalization.

The MMADN method is not successful due to certain limitations discussed above; hence, median-based measures should not be used for normalizing data. Further, min–max based normalization methods and decimal scaling method are unable to achieve the best ranks; it is observed that the minimum and maximum statistical measures are not useful for normalizing data. The best-performing methods in a full set of features, feature selection, and feature weighting approaches have utilized the mean, standard deviation or the combination of both measures for data normalization. Therefore, these measures are more effective for data normalization as compared to other measures.

The impact of fourteen different normalization methods on the classification performance has been observed based on classification accuracy, the percentage of feature reduced and runtime. Since k-NN classifier is sensitive to changes in feature space and the representation of each feature in data varies with normalization methods, the disparities in performances are evident as observed from the results. The scaling methods ZSN and PS and are best since these achieve good classification performance and handle the outliers more effectively than other scaling methods. These methods suit better for the full feature set and feature selection approach. The transformation methods such as TN and VTN achieve good classification performance with feature weighting because these methods effectively handle the problem of outliers and feature dominance. The worst methods that should be avoided are MC, VSS, and MMADN.

The investigations show that there exist many normalization methods which reduce the effect of dominant features but respond poorly when data has both dominant features as well as outliers. The presence of the outliers pushes the rest of the feature

values in a narrow range. The methods will handle outliers better by minimizing the standard deviation or utilizing the transformation methods that directly deal with the outliers either by eliminating or pushing them towards the rest of the data. Further findings from the full feature set, feature selection, and feature weighting also indicate that normalization changes the relevance of features. Therefore, an ideal representation for the data by a single method is not possible, and no method is superior in full feature set, feature selection and feature weighting.

The normalization is a complex data pre-processing approach that requires careful analysis of properties of data to choose a pre-processing method before evaluating data on the classifier. This work is more subjective and requires expertise to decide on a method. However, combining the aspects of these methods for normalizing the data and empirical analysis of the results, a set of best and worst methods are pointed out.

7. Conclusion

This work investigates one of the data pre-processing approach, i.e., normalization for the improvement of classification performance. It will help in better understanding of normalization methods without and with feature reduction approaches. We have considered fourteen normalization methods from different research areas for the empirical analysis on publicly available datasets. This paper indicates three important points regarding data normalization. First, some methods complicate the normalized data more than the un-normalized data on some classification problems. The classification performances lower than the un-normalized data indicate exceptions in data normalization which otherwise believed to improve the predictive power of a classifier in all cases. Second, the mean and standard deviation measures are more suitable for data normalization as compared to min–max and median measures. The best methods in full feature set, feature selection or feature weighting approaches have employed these two measures while method based on other measures lacks in better ranks. Last, the data normalization changes the relevance of the features which causes different outcomes in terms of accuracy and feature reduction while working with feature selection and feature weighting. The scaling methods, *Z*-Score and Pareto Scaling, have performed well with the full feature set and feature selection. The transformation methods, *tanh*, and its variant have performed well with feature weighting approach. Another method, power transformation, is an exceptional method that suits best for solving a particular problem in un-normalized data. Mean Centered, Variable Stability Scaling, and Median and Median Absolute Deviation method along with un-normalized data are the least performing methods that should be avoided. In conclusion, data normalization has a subjective, rather than objective, nature due to the different features properties that data may have. However, this work has suggested some methods that should be opted depending on the approach. These methods are supported by the theoretical as well as empirical analysis to tackle issues in normalizing data.

The issues in data that make normalization a complex data pre-processing approach are the features that exhibit different statistical properties and the presence of outliers as well as dominant features. With the increase of these problems in data, the choice of a normalization method tends to a hard problem because a single method cannot tackle all the problems. It may also be possible that one subset of features can be best normalized by one method while the other subset of features by some other methods. The combination of the various methods on the single dataset can be tried in the future to get more insights into normalization. This study can also be extended to the other machine learning algorithms for performance analysis.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.105524.

References

[1] Geoff Dougherty, Pattern Recognition and Classification: an Introduction, Springer Science & Business Media, 2012.
[2] Salvador García, Julián Luengo, Francisco Herrera, Data Preprocessing in Data Mining, Springer, 2015.
[3] Selim Aksoy, Robert M. Haralick, Feature normalization and likelihood-based similarity measures for image retrieval, Pattern Recognit. Lett. 22 (5) (2001) 563–582.
[4] Warren S. Sarle, Neural Network FAQ, periodic posting to the Usenet newsgroup comp. ai.neural-nets , 1997, ftp://ftp.sas.com/pub/neural/FAQ.html.
[5] J. Sola, Joaquin Sevilla, Importance of input data normalization for the application of neural networks to complex industrial problems, IEEE Trans. Nucl. Sci. 44 (3) (1997) 1464–1468.
[6] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al., A practical guide to support vector classification, Taipei, 2003.
[7] T. Jayalakshmi, A. Santhakumaran, Statistical normalization and back propagation for classification, Int. J. Comput. Theory Eng. 3 (1) (2011) 1793–8201.
[8] U Rajendra Acharya, Sumeet Dua, Xian Du, Chua Kuang Chua, et al., Automated diagnosis of glaucoma using texture and higher order spectra features, IEEE Trans. Inform. Technol. Biomed. 15 (3) (2011) 449–455.
[9] Robert Snelick, Umut Uludag, Alan Mink, Mike Indovina, Anil Jain, Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 450–455.
[10] Xuezhi Wen, Ling Shao, Wei Fang, Yu Xue, Efficient feature selection and classification for vehicle detection, IEEE Trans. Circuits Syst. Video Technol. 25 (3) (2015) 508–517.
[11] Ehsan Tarkesh Esfahani, Shaocheng Wang, V. Sundararajan, Multisensor wireless system for eccentricity and bearing fault detection in induction motors, IEEE/ASME Trans. Mechatronics 19 (3) (2014) 818–826.
[12] Jiaqi Pan, Yan Zhuang, Simon Fong, The impact of data normalization on stock market prediction: using svm and technical indicators, in: International Conference on Soft Computing in Data Science, Springer, 2016, pp. 72–88.
[13] Abdul Kadir, Lukito Edi Nugroho, Adhi Susanto, Paulus Insap Santosa, Leaf classification using shape, color, and texture features, Int. J. Comput. Trends Technol. 2 (1) (2011) 225–230.
[14] Chia-Ming Wang, Yin-Fu Huang, Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data, Expert Syst. Appl. 36 (3) (2009) 5900–5908.
[15] Wei Wu, Eric P Xing, Connie Myers, I Saira Mian, Mina J Bissell, Evaluation of normalization methods for cDNA microarray data by *k*-NN classification, BMC Bioinform. 6 (1) (2005) 191.
[16] Weijun li, Zhenyu Liu, A method of SVM with normalization in intrusion detection, Procedia Environ. Sci. 11 (2011) 256–262.
[17] Dan Su, Wei Wang, Xing Wang, Jiqiang Liu, Anomadroid: Profiling android applications' behaviors for identifying unknown malapps, in: Proceedings of IEEE Trustcom/BigDataSE/ISPA, IEEE, 2016, pp. 691–698.
[18] Dietrich Wettschereck, David W. Aha, Takao Mohri, A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, Artif. Intell. Rev. 11 (1–5) (1997) 273–314.
[19] Anil Jain, Karthik Nandakumar, Arun Ross, Score normalization in multimodal biometric systems, Patter. Recog. 38 (12) (2005) 2270–2285.
[20] Arun A. Ross, Rohin Govindarajan, Feature level fusion of hand and face biometrics, in: Biometric Technology for Human Identification II, vol. 5779, 2005, pp. 196–205.
[21] Hazim Kemal Ekenel, Rainer Stiefelhagen, Analysis of local appearance-based face recognition: Effects of feature selection and feature normalization, in: Conference on Computer Vision and Pattern Recognition Workshop, IEEE, 2006, 34–34.
[22] Ajay Kumar, Ch Ravikanth, Personal authentication using finger knuckle surface, IEEE Trans. Inform. Forensics Secur. 4 (1) (2009) 98–110.
[23] Cheng-Lung Huang, Jian-Fan Dun, A distributed PSO-SVM hybrid system with feature selection and parameter optimization, Appl. Soft Comput. 8 (4) (2008) 1381–1391.
[24] Cheng-Lung Huang, Chieh-Jen Wang, A GA-based feature selection and parameters optimization for support vector machines, Expert Syst. Appl. 31 (2) (2006) 231–240.

[25] Luai Al Shalabi, Zyad Shaaban, Normalization as a preprocessing engine for data mining and the approach of preference matrix, in: International Conference on Dependability of Computer Systems, IEEE, 2006, pp. 207–214.
[26] Shih-Wei Lin, Kuo-Ching Ying, Shih-Chieh Chen, Zne-Jung Lee, Particle swarm optimization for parameter determination and feature selection of support vector machines, Expert Syst. Appl. 35 (4) (2008) 1817–1824.
[27] Robert A van den Berg, Huub CJ Hoefsloot, Johan A Westerhuis, Age K Smilde, Mariët J van der Werf, Centering, scaling, and transformations: improving the biological information content of metabolomics data, BMC Genomics 7 (1) (2006) 142.
[28] Andrew Craig, Olivier Cloarec, Elaine Holmes, Jeremy K Nicholson, John C Lindon, Scaling and normalization effects in NMR spectroscopic metabonomic data sets, Anal. Chem. 78 (7) (2006) 2262–2267.
[29] Keinosuke Fukunaga, Introduction to Statistical Pattern Recognition, Elsevier, 2013.
[30] Antonio Reverter, Wes Barris, Sean Mcwilliam, Keren A Byrne, Yong H Wang, Siok H Tan, Nick Hudson, Brian P Dalrymple, Validation of alternative methods of data normalization in gene co-expression studies, Bioinformatics 21 (7) (2004) 1112–1120.
[31] Isao Noda, Scaling techniques to enhance two-dimensional correlation spectra, J. Molecul. Struct. 883 (2008) 216–227.
[32] Lennart Eriksson, Joanna Jaworska, Andrew P Worth, Mark TD Cronin, Robert M McDowell, Paola Gramatica, Methods for reliability and uncertainty assessment and for applicability evaluations of classification-and regression-based QSARs, Environ. Health Perspect. 111 (10) (2003) 1361.
[33] Olav M. Kvalheim, Frode Brakstad, Yizeng Liang, Preprocessing of analytical profiles in the presence of homoscedastic or heteroscedastic noise, Anal. Chem. 66 (1) (1994) 43–51.
[34] Jiawei Han, Jian Pei, Micheline Kamber, Data mining: concepts and techniques, Elsevier, 2011.
[35] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, Werner A Stahel, Robust statistics: the approach based on influence functions, vol. 196, John Wiley & Sons, 2011.
[36] Kevin L. Priddy, Paul E. Keller, Artificial neural networks: an introduction, vol. 68, SPIE press, 2005.
[37] Sergios Theodoridis, Konstantinos Koutroumbas, Pattern Recognition, fourth ed., Academic Press, Inc., Orlando, FL, USA, 2008.
[38] Joaquín Derrac, Isaac Triguero, Salvador García, Francisco Herrera, Integrating instance selection, instance weighting, and feature weighting for nearest neighbor classifiers by coevolutionary algorithms, IEEE Trans. Syst. Man Cybern. B 42 (5) (2012) 1383–1397.
[39] Javier Pérez-Rodríguez, Alexis Germán Arroyo-Peña, Nicolás García-Pedrajas, Simultaneous instance and feature selection and weighting using evolutionary computation: Proposal and study, Appl. Soft Comput. 37 (2015) 416–443.
[40] Daniel Mateos-García, Jorge García-Gutiérrez, José C Riquelme-Santos, An evolutionary voting for *k*-nearest neighbours, Expert Syst. Appl. 43 (2016) 9–14.
[41] Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, Vladimir Vapnik, Feature selection for SVMs, in: Advances in Neural Information Processing Systems, 2001, pp. 668–674.
[42] Isabelle Guyon, André Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (Mar) (2003) 1157–1182.
[43] Yijun Sun, Iterative RELIEF for feature weighting: algorithms, theories, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007).
[44] Qinbao Song, Jingjie Ni, Guangtao Wang, A fast clustering-based feature subset selection algorithm for high-dimensional data, IEEE Trans. Knowl. Data Eng. 25 (1) (2013) 1–14.
[45] H Hannah Inbarani, Ahmad Taher Azar, G. Jothi, Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis, Comput. Method. Progr. Biomed. 113 (1) (2014) 175–185.
[46] Hanchuan Peng, Fuhui Long, Chris Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[47] Craig Stanfill, David Waltz, Toward memory-based reasoning, Commun. ACM 29 (12) (1986) 1213–1228.
[48] Robert H Creecy, Brij M Masand, Stephen J Smith, David L Waltz, Trading MIPS and memory for knowledge engineering, Commun. ACM 35 (8) (1992) 48–64.
[49] Yuanning Liu, Gang Wang, Huiling Chen, Hao Dong, Xiaodong Zhu, Sujing Wang, An improved particle swarm optimization for feature selection, J. Bionic Eng. 8 (2) (2011) 191–200.
[50] Xingshi He, Qingqing Zhang, Na Sun, Yan Dong, Feature selection with discrete binary differential evolution, in: IEEE International Conference on Artificial Intelligence and Computational Intelligence, vol. 4, IEEE, 2009, pp. 327–330.
[51] Suresh C. Satapathy, Anima Naik, K. Parvathi, Rough set and teaching learning based optimization technique for optimal features selection, Central Eur. J. Comput. Sci. 3 (1) (2013) 27–42.
[52] Md Monirul Kabir, Md Shahjahan, Kazuyuki Murase, A new hybrid ant colony optimization algorithm for feature selection, Expert Syst. Appl. 39 (3) (2012) 3747–3763.
[53] Bing Xue, Mengjie Zhang, Will N. Browne, Xin Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2016) 606–626.
[54] Bo Chen, Hongwei Liu, Jing Chai, Zheng Bao, Large margin feature weighting method via linear programming, IEEE Trans. Knowl. Data Eng. 21 (10) (2009) 1475–1488.
[55] James D. Kelly Jr, Lawrence Davis, A Hybrid Genetic Algorithm for Classification, vol. 91, IJCAI, 1991, pp. 645–650.
[56] L.F. Giraldo, E. Delgado, C.G. Castellanos, Feature weighting and selection using a hybrid approach based on rademacher complexity model selection, in: Computers in Cardiology, IEEE, 2007, pp. 257–260.
[57] Muhammad Atif Tahir, Ahmed Bouridane, Fatih Kurugollu, Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier, Pattern Recognit. Lett. 28 (4) (2007) 438–446.
[58] Ahmad A. Kardan, Atena Kavian, Amir Esmaeili, Simultaneous feature selection and feature weighting with k selection for KNN classification using BBO algorithm, in: IEEE 5th Conference on Information and Knowledge Technology, IEEE, 2013, pp. 349–354.
[59] Adélia C.A. Barros, George D.C. Cavalcanti, Combining global optimization algorithms with a simple adaptive distance for feature selection and weighting, in: IEEE International Joint Conference on Neural Networks, IEEE, 2008, pp. 3518–3523.
[60] Roberto Paredes, Enrique Vidal, Learning weighted metrics to minimize nearest-neighbor classification error, IEEE Trans. Pattern Anal. Mach. Intell. (7) (2006) 1100–1110.
[61] Thomas Cover, Peter Hart, Nearest neighbor pattern classification, IEEE Trans. Inform. Theory 13 (1) (1967) 21–27.
[62] Li-Yeh Chuang, Sheng-Wei Tsai, Cheng-Hong Yang, Improved binary particle swarm optimization using catfish effect for feature selection, Expert Syst. Appl. 38 (10) (2011) 12699–12707.
[63] Nicolás García-Pedrajas, Juan A Romero del Castillo, Gonzalo Cerruela-García, A proposal for local *k* values for *k*-nearest neighbor rule, IEEE Trans. Neural Netw. Learn. Syst. 28 (2) (2017) 470–475.
[64] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, Ruili Wang, Efficient knn classification with different numbers of nearest neighbors, IEEE Trans. Neural Netw. Learn. Syst. 29 (5) (2018) 1774–1785.
[65] Frans Van den Bergh, Andries Petrus Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. Evolu. Comput. 8 (3) (2004) 225–239.
[66] Ahmad Nickabadi, Mohammad Mehdi Ebadzadeh, Reza Safabakhsh, A novel particle swarm optimization algorithm with adaptive inertia weight, Appl. Soft Comput. 11 (4) (2011) 3658–3670.
[67] Seyedali Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.
[68] Waleed Yamany, Alaa Tharwat, Mohammad F Hassanin, Tarek Gaber, Aboul Ella Hassanien, Tai-Hoon Kim, A new multi-layer perceptrons trainer based on ant lion optimization algorithm, in: 2015 Fourth International Conference on Information Science and Industrial Applications (ISI), IEEE, 2015, pp. 40–45.
[69] E.S. Ali, S.M.Abd Elazim, A.Y. Abdelaziz, Ant lion optimization algorithm for optimal location and sizing of renewable distributed generations, Renew. Energy 101 (2017) 1311–1324.
[70] Konidala Ratna Subhashini, Jitendriya Kumar Satapathy, Development of an enhanced ant lion optimization algorithm and its application in antenna array synthesis, Appl. Soft Comput. 59 (2017) 153–173.
[71] Larry J. Eshelman, J.D.avid Schaffer, Real-coded genetic algorithms and interval-schemata, in: Foundations of Genetic Algorithms, vol. 2, Elsevier, 1993, pp. 187–202.
[72] Irina Ciornei, Elias Kyriakides, Hybrid ant colony-genetic algorithm (GAAPI) for global continuous optimization, IEEE Trans. Syst. Man Cybern. B 42 (1) (2012) 234–245.
[73] Min Li, Frederico Guimarães, David A. Lowther, Competitive co-evolutionary algorithm for constrained robust design, IET Sci., Measur. Technol. 9 (2) (2015) 218–223.
[74] Yong Zhang, Dunwei Gong, Ying Hu, Wanqiu Zhang, Feature selection algorithm based on bare bones particle swarm optimization, Neurocomputing 148 (2015) 150–157.
[75] Yong Zhang, Xian-fang Song, Dun-wei Gong, A return-cost-based binary firefly algorithm for feature selection, Inform. Sci. 418 (2017) 561–574.
[76] Arthur Asuncion, David Newman, UCI Machine learning repository, 2007.
[77] Frank E. Grubbs, Procedures for detecting outlying observations in samples, Technometrics 11 (1) (1969) 1–21.
[78] Da-You Liu, Hui-Ling Chen, Bo Yang, Xin-En Lv, Li-Na Li, Jie Liu, Design of an enhanced fuzzy *k*-nearest neighbor classifier based computer aided diagnostic system for thyroid disease, J. Med. Syst. 36 (5) (2012) 3243–3254.
[79] Frank Wilcoxon, Individual comparisons by ranking methods, Biomet. Bull. 1 (6) (1945) 80–83.

[80] Janez Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.
[81] Milton Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1) (1940) 86–92.
[82] Peter J. Rousseeuw, Christophe Croux, Alternatives to the median absolute deviation, J. Am. Statist. Associat. 88 (424) (1993) 1273–1283.