IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 25, 2020, accepted December 19, 2020, date of publication December 29, 2020, date of current version February 3, 2021.

*Digital Object Identifier 10.1109/ACCESS.2020.3048018*

# Tax Default Prediction Using Feature Transformation-Based Machine Learning

**MOHAMMAD ZOYNUL ABEDIN[1], GUOTAI CHI[2], MOHAMMED MOHI UDDIN[3], MD. SHAHRIARE SATU[4], MD. IMRAN KHAN[5], AND PETR HAJEK[6]**

[1]Department of Finance and Banking, Hajee Mohammad Danesh Science and Technology University, Dinajpur 5200, Bangladesh
[2]School of Economics and Management, Dalian University of Technology, Dalian 116024, China
[3]Department of Accountancy, College of Business and Management, University of Illinois Springfield, Springfield, IL 62703, USA
[4]Department of Management Information Systems, Noakhali Science and Technology University, Noakhali 3814, Bangladesh
[5]Department of Computer Science and Engineering, Gono Bishwabidyalay, Dhaka 1344, Bangladesh
[6]Science and Research Centre, Faculty of Economics and Administration, University of Pardubice, 53210 Pardubice, Czech Republic

Corresponding author: Guotai Chi (chigt@dlut.edu.cn)

This work has been supported by the Key Projects of National Natural Science Foundation of China (71731003), the General Projects of National Natural Science Foundation of China (72071026, 71873103, 71971051, and 71971034), the Youth Projects of National Natural Science Foundation of China (71901055, and 71903019), the Major Projects of National Social Science Foundation of China (18ZDA095), the scientific research project of the Czech Sciences Foundation Grant No. 19-15498S. This paper has also been supported by the Bank of Dalian and Postal Savings Bank of China. We thank the organizations mentioned above.

**ABSTRACT** This study proposes to address the economic significance of unpaid taxes by using an automatic system for predicting a tax default. Too little attention has been paid to tax default prediction in the past. Moreover, existing approaches tend to apply conventional statistical methods rather than advanced data analytic approaches, including state-of-the-art machine learning methods. Therefore, existing studies cannot effectively detect tax default information in real-world financial data because they fail to take into account the appropriate data transformations and nonlinear relationships between early-warning financial indicators and tax default behavior. To overcome these problems, this study applies diverse feature transformation techniques and state-of-the-art machine learning approaches. The proposed prediction system is validated by using a dataset showing tax defaults and non-defaults at Finnish limited liability firms. Our findings provide evidence for a major role of feature transformation, such as logarithmic and square-root transformation, in improving the performance of tax default prediction. We also show that extreme gradient boosting and the systematically developed forest of multiple decision trees outperform other machine learning methods in terms of accuracy and other classification performance measures. We show that the equity ratio, liquidity ratio, and debt-to-sales ratio are the most important indicators of tax defaults for 1-year-ahead predictions. Therefore, this study highlights the essential role of well-designed tax default prediction systems, which require a combination of feature transformation and machine learning methods. The effective implementation of an automatic tax default prediction system has important implications for tax administration and can assist administrators in achieving feasible government expenditure allocations and revenue expansions.

**INDEX TERMS** Default prediction, corporate tax, machine learning, feature transformation.

## I. INTRODUCTION

World Bank statistics claim that approximately 40% of firms around the globe pay their taxes but 60% fail to pay their taxes, and these amounts might not be recovered during upcoming tax years. The statistics also report that rates of tax defaults are increasing worldwide [1]. Considering the economic importance of unpaid taxes (tax debts not paid by the due date), little research has been conducted in predicting the tax status of firms. Most studies have tended to focus on predicting other default risks, such as credit default [2] and corporate bankruptcy [3]. Measuring the tax default status of firms is considered to be an even more challenging prediction task because it is characterized by the presence of sample selection bias due to the small number of labelled data (known as tax default cases) [4]. The indicators of credit default and corporate bankruptcy have been extensively studied in earlier research [5], [6], but the indicators of tax default

The associate editor coordinating the review of this manuscript and approving it for publication was Najah Abuali.

are still poorly understood despite the fact that corporate tax documents contain much multifaceted information for assessing tax default risk prediction [7]. Predicting a tax default differs from predicting a tax evasion (tax fraud) [4], [8] or a tax avoidance [9]. In tax evasion, a taxpayer supplies intentionally incorrect or partial information to tax agencies to lessen the tax burden, and in tax avoidance, the taxpayer arranges the affairs of a company in such a way that the tax burden is reduced relative to the pretax income.

A preliminary study on tax default prediction was conducted in Margescu *et al.* [10]. However, its classification accuracy was only 61.6%, and only 16.4% of firms were correctly classified as default firms. This can be attributed to model underspecification (only four financial indicators were used) and the presence of nonlinear relationships between the financial indicators and the firms' tax status, which could not be detected using the logistic regression (LR) model. A substantially improved classification accuracy was achieved when a large number of financial indicators were employed [7]. It must be noted that the aim of this earlier study was to investigate the importance of early-warning financial indicators rather than to develop an accurate tax default prediction system. Therefore, a linear discriminant analysis (LDA) model was used to classify defaulting and nondefaulting firms; this indicates that solvency and liquidity are important features for detecting tax defaults. The main drawbacks of LDA are its underlying assumptions of multivariate normality, homoscedasticity, and absence of multicollinearity between explanatory variables. Although LR represents a more flexible statistical prediction model, it is sensitive to outliers and missing data and it does not allow the consideration of nonlinear and complex relationships between predictors and defaults [11].

To overcome the above limitations of previous studies, this study introduces an automated tax default prediction system by integrating state-of-the-art data analytic approaches with financial predictors extracted from corporate financial statements. We here show that the proposed system outperforms existing statistical approaches to tax default prediction. In addition, by measuring the prediction power of the financial indicators, this study also examines their importance and establishes a comprehensive early-warning system regarding the status of corporate tax payment.

The relatively small samples of tax default data mentioned above require careful feature selection and data preprocessing. The related literature on business risk prediction suggests that feature reduction and feature transformation may improve the performance of prediction systems [12]. There are two main motivations for reducing the number of features. First, irrelevant and redundant features are disregarded, and second, model explainability is increased. In the literature on credit default and corporate bankruptcy prediction, feature transformation techniques were applied to enhance the informational content of financial indicators by reducing their group-level heterogeneity [13] and distortions of financial ratio distributions [14].

Based on these considerations, the current study examines several approaches to feature transformation, which is a novel research domain in taxation and accounting fields. It refers to the conversion of original attributes to a form that optimizes the outcomes of a specific data analytic algorithm. The transformed datasets are extensively tested over 13 data analytic (machine learning) approaches trained to detect a tax default (1) in the default year and (2) 1 year prior to the default. Specifically, we here investigate the effects of feature transformation on financial ratio distributions and classification performances of the data analytic methods.

The current study provides several contributions and managerial insights into the existing literature on financial default prediction. The contributions of our study are threefold:

- A novel tax default prediction model is proposed that, as far as we know, uses, for the first time, advanced machine learning methods.
- This is also the first study investigating the fundamental role of feature transformation in tax default prediction.
- A real-world dataset of Finnish tax defaulted and nondefaulted firms is used to demonstrate the effectiveness of the proposed prediction system, indicating significant improvements of classification performance over existing tax default prediction models.

The organization of the rest of this study is as follows. Section 2 describes empirical literature related to tax default prediction. Section 3 outlines the proposed automatic tax default prediction system that includes feature transformation as well as machine learning methods. Section 4 presents the used dataset. Section 5 introduces the experimental design and presents the experimental results. Section 6 concludes the study by highlighting the study's limitations and offering future roadmaps for research.

## II. LITERATURE REVIEW

As has already been mentioned, default risk prediction is a well-studied problem in business finance literature, but existing studies are related to other areas, such as credit risk prediction [3] and corporate bankruptcy prediction [15]. In a recent study, Beutel *et al.* [16] employed LR and other data analytic approaches to predict banking crises. The comparative analysis found that it is difficult for machine learning methods to outperform the traditional LR model in terms of out-of-sample evaluations. This finding was robust for performance measures, sample sizes, and data transformations. Indeed, the LR models are popular in the financial distress prediction literature. For instance, Andrikopoulos and Khorasgani [17] applied LR to predict defaults by small enterprises by integrating accounting and market information. Unfortunately, the LR model failed to effectively handle the highly imbalanced dataset of defaulted and nondefaulted firms.

Huang and Yen [18] evaluated the forecasting performance of several data analytic approaches for corporate financial distress prediction, comparing algorithms with supervised

and unsupervised learning. Among the compared models, the extreme gradient boosting (XGBoost) method performed best for the sample of publicly listed Taiwanese companies. However, the sample size was limited to 64 companies and 16 financial indicators. Then, Xiao *et al.* [19] proposed several ensemble-based data analytic approaches for bank customer credit scoring. Applying public and real-company credit-scoring datasets, a semisupervised learning model had a better overall credit-scoring performance than supervised learning approaches. In addition, Song *et al.* [20] applied data analytic approaches based on ensemble learning for the peer-to-peer lending industry. These studies confirmed that ensemble learning approaches performed better in default prediction with imbalanced data compared with individual machine learning classifiers.

In the area of customer behavior analytics, Amin *et al.* [21] applied data analytic approaches for predicting customer churn in the field of telecommunications. In their study, coherence was established between data certainty and machine learning accuracy, showing that high classifier certainty indicated a great distance between churn and nonchurn customer behaviour. Later, Al-Mashraie *et al.* [22] compared multiple data analytic approaches for the prediction of telecommunication customer switching behaviour in the United States to demonstrate that the support vector machine (SVM) model outperformed the LR, random forest (RF), and decision tree (DT) models. Maldonado *et al.* [23] employed a profit-driven data analytic approach to improve the economic effectiveness of predictions of customer churn rates. In addition to the above default prediction domains, the existing literature also reported on financial statement fraud detection [24], [25].

Comparatively few empirical studies have been published on tax default prediction. This can be ascribed to inadequate access to corporate tax databases at the national level. Tax default forecasting differs from studies of ordinary tax noncompliance because in the latter, tax evaluations are infrequently planned events. Taking an example from the Finnish tax evaluation environment, Marghescu *et al.* [10] attempted to find the extent to which financial statement ratios could be applied for detecting the payment status of tax clients. They trained a binomial LR classifier with four attributes, achieving a low prediction accuracy of 61.6%. More recently, Höglund [7] applied a genetic algorithm (GA)–based analytic support system for detecting the payment status of tax clients. In his study, the LDA was compiled and the GA selected the features. Thus, a set of relevant features was identified for tax default prediction. This model also utilized the log-transformation approach to satisfy LDA assumptions and improve its prediction accuracy.

Before the above study, Yu *et al.* [26] had proposed a data-mining structural design for the problem of fraudulent tax declarations by Chinese commercial enterprises. It consisted of conversations with area specialists, selection by the data analytic approach, and the building of a system for detecting fraudulent tax declarations using the DT and incorporating expert domain knowledge. Gebauer *et al.* [27] experimented with the possibilities of value-added tax (VAT) evasion, indicating that tax defaults can be derived from the fraudulent preservation of revenues. Gupta and Nagadevara [28] executed eight data analytic approaches derived from diverse mixtures of DT, LDA, and LR. Their findings suggested that data analytic approaches performed better in predicting VAT evasion relative to automatic audit selection. Thus, a more effective audit selection and taxpayer compliance could be achieved. Wu *et al.* [29] used an association rule–based data analytic approach to augment the detection of tax evasion. This screening procedure was applied to filter noncomplaint VAT statements for consecutive audit processes. The use of data analytic methods significantly increased the accuracy of the detection of tax evasion. Rahimikia *et al.* [30] assessed the effectiveness of combining data analytic approaches (SVM, LR, and multilayer perceptron (MLP)) with evolutionary feature selection. For the situation of Iranian corporate tax evasion in the food and textile sectors, it was reported that the MLP-based hybrid algorithm outperformed other approaches, ensuring 90.1% and 82.5% accuracy, respectively. A sector-dependent unsupervised anomaly detection method was developed for VAT fraud detection within a real-world scenario of very few labelled samples [4]. This approach is fast and scalable but could only identify 20 out of 30 investigated firms. The MALDIVE approach was proposed for tax risk assessment by combining social network visualization from taxpayer data with LR, MLP, SVM, and RF data analytic approaches [9]. The RF method performed best, with 74.3% accuracy in detecting the positive and negative outcomes of a fiscal audit.

The above literature review indicates that advanced data analytic approaches have not been considered for tax default prediction. The previous related literature highlighted the paramount importance of data transformation methods in enhancing the informational relevance of limited default data, and this inspired us to explore their use in tax default prediction models.

Regarding the data transformation literature, Zhang *et al.* [31] experimented with cross-project software defect prediction employing different feature transformation methods, including log, Box-Cox, and rank transformations, as well as their combinations. RF plus Box-Cox feature transformation outperformed the other methods, including the nontransformed data analytic approaches. Like this study, Amin *et al.* [32] applied several feature transformation methodologies for predicting the churn in telecommunications. The log, $Z$-score, rank, and Box-Cox transformations were tested in combinations with the data analytic approaches naïve Bayes (NB), $k$-nearest neighbour ($k$-NN), gradient boosting (GB), single rule induction, and deep neural network. For two real-world churn datasets, the results indicated that data transformations performed well, except the $Z$-score transformation. In the same application domain, Coussement *et al.* [12] explored three well-known feature transformation approaches for categorical attributes, namely, dummy coding,

incidence replacement, and weight-of-evidence conversion. By training the LR data analytic approach, it was found that feature transformation enhanced the churn prediction performance by as much as 14.5% in terms of the area under the receiver operating characteristic curve (AUROC). For continuous attributes, Al Shalabi and Shaaban [33] applied the $Z$-score, scaled transformation, and decimal scaling models for a classification dataset by considering DT learning parameters. They concluded that scaled transformation is the best data transformation methodology for the DT classifier. Huang and Dun [34] and Wang and Huang [35] also applied scaled transformation (normalization) to preprocess credit-scoring data for SVM-based classification. Recently, Singh and Singh [36] showed that scaled transformation has a positive impact on classification performance over multiple real and synthetic datasets. Based on these findings, we hypothesized that feature transformation may provide substantial benefits to data analytic methods in predicting the corporate tax default status.

## III. AUTOMATED TAX DEFAULT PREDICTION SYSTEM

This section presents the first automated tax default prediction system in the accounting and tax domain. The system utilizes diverse accounting data transformed to achieve high-performance data analytic approaches. Figure 1 illustrates the components of the proposed system. First, the accounting data are collected for two categories of firms, tax default and nondefault firms. The system was developed to perform tax default predictions for 1 year ahead; therefore, the data for the input attributes are obtained for the year prior to the tax default. Second, data are randomly divided into training and testing sets at a ratio of 4:1. Third, the training data are investigated using statistical methods to test for the feature significance and linear relationships among the features. Feature transformation methods are applied in the fourth step. To optimize the hyperparameters of machine learning methods, 10-fold cross-validation is performed in the next step. The machine learning–based models are trained to classify tax default and nondefault firms. The classification performance is evaluated using performance measures over the testing data.

### A. FEATURE TRANSFORMATION APPROACHES

The current study applies five dissimilar feature transformation techniques that preserve the information of the original features from different business analytic perspectives. Specifically, the following methods were examined: (1) log transformation (log-tr.), (2) Z-normalization transformation ($Z$-nor-tr.), (3) scaled transformation (scaled-tr.), (4) sine transformation (sine-tr.), and (5) square-root transformation (sqrt-tr.). The rationale behind the selection of these methods is that they have widely been applied in related application domains, including customer churn prediction [21], software metrics sustainability and normality [31], defect prediction [37], and dimensionality reduction [38]. This section briefly illustrates their main features. For details, interested readers are requested to review Zhang *et al.* [31] and Amin *et al.* [21].

#### 1) LOG TRANSFORMATION

Log transformation converts feature values by simply applying the natural logarithm; this makes it a popular data transformation technique with applications ranging from customer churn prediction [21], [32] to software defect prediction [31]. This transformation not only enhances an algorithm's predictability, but it also transforms a skewed data distribution to a normal or Gaussian pattern, which conforms to the normal distribution approximately as follows:

$$y = \ln(1 + x), \tag{1}$$

where $y$ is the transformed feature, $x$ is the original feature, and $\ln(x)$ refers to the natural logarithmic function. Owing to the constraint of the $\ln(x)$ function, however, the above formula converts only the numerical values that are higher than zero. To deal with the zero values, a constant is inserted, such as the $\ln(1 + x)$ function.

#### 2) $Z$-NORMALIZATION TRANSFORMATION

Normalized $Z$-scores are calculated by subtracting the average value from the raw value for each instance and dividing that by the standard deviation (SD) for the data as follows:

$$y = \frac{x - \bar{x}}{SD(x)}. \tag{2}$$

As a result, a common scale is obtained for y where the average value equals zero and the SD is 1, thus reducing the effect of outlying values [39]. This enables us to focus on the structural characteristics of the features, rather than on their variance. However, the scales are not identical for all features, and, moreover, if the variance is small, noise in the data may be amplified. Another problem may be the sensitivity of the mean and SD to outliers.

#### 3) SCALED TRANSFORMATION

Scaled transformation is performed by normalizing the values to equal scales, which can be useful for machine learning algorithms because features with larger scales may distort the results of their objective functions. Here we followed related literature [40] and used the most common min-max method to rescale the data to the range [0,1] as follows:

$$y = \frac{x - min(x)}{max(x) - min(x)}. \tag{3}$$

This transformation also accelerates the convergence of the gradient descent algorithm. However, as for $Z$-normalization transformation, data noise may be amplified.

#### 4) SINE TRANSFORMATION

Sine transformation allows us to overcome some disadvantages of the above transformation methods. Unlike traditional log transformation, features with negative values can

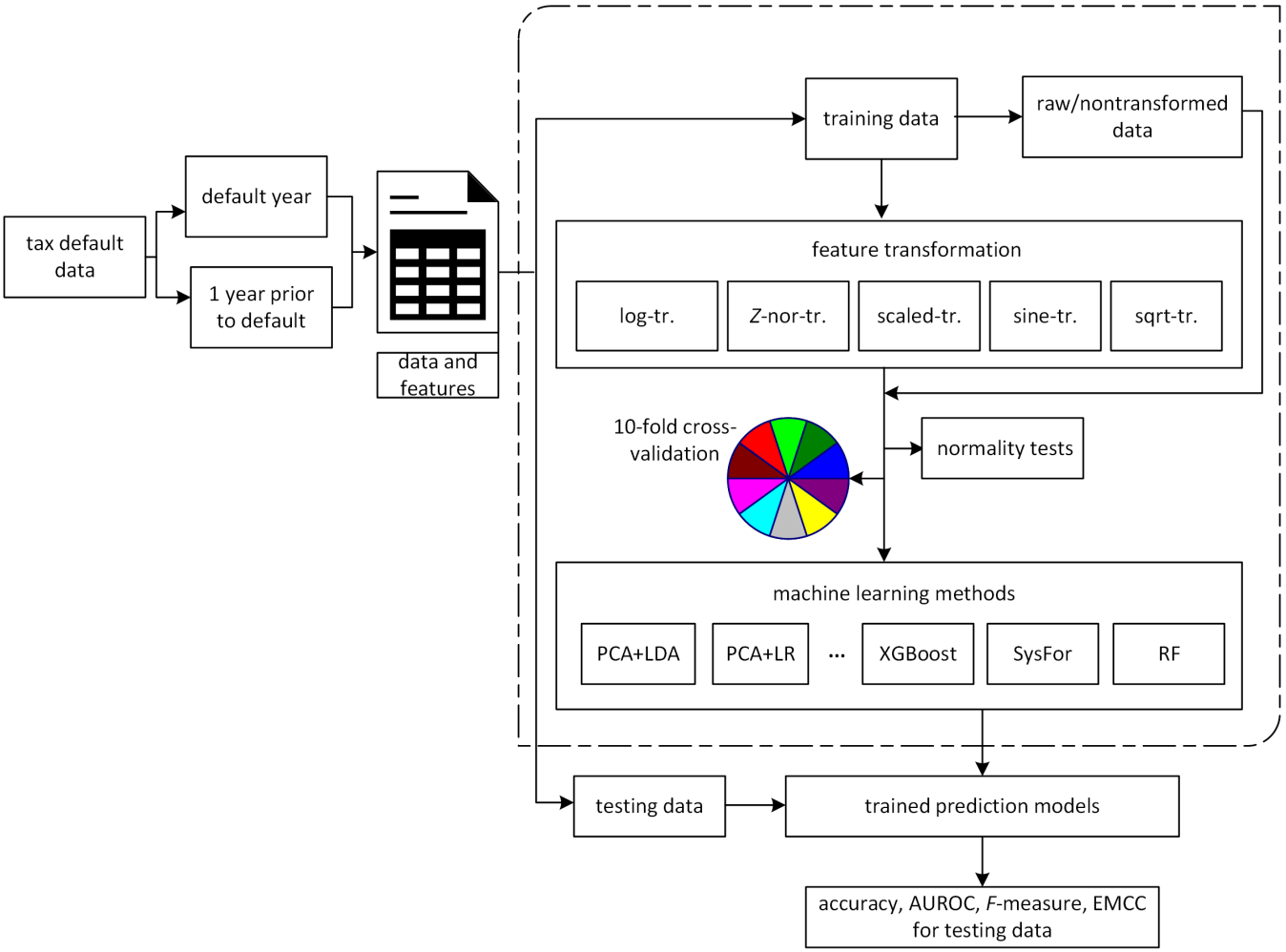



**FIGURE 1.** Automated tax default prediction system using feature transformation–based machine learning.

be processed [41]. The converted feature can be expressed as follows:

$$y = sin(x). \tag{4}$$

The application of the sine function results in the rescaling of the feature to the range [–1, 1] with the amplitude of 1, implying that the data intensity (amplitude) can be observed against the data frequency. In addition, as in scaled transformations, faster convergence rates were observed in related studies [42].

#### 5) SQUARE-ROOT TRANSFORMATION

Square-root (sqrt) transformation helps normalize a geometrically (Poisson or negative binomial) distributed feature. Poisson distributions frequently occur with features measured as counts. It carries a moderate effect on the distribution shape that is weaker than the logarithm and cube root. Eq. (5) illustrates the sqrt transformation:

$$y_{ij} = x_{ij}^{1/2}, \quad x \geq 0 \tag{5}$$

where $y_{ij}$ is the transformed value for the $i$-th feature and $j$-th sample, and $x_{ij}$ is the nontransformed feature. The transformed feature stabilizes the variance; this is particularly effective if the data variance is proportional to the mean value [43].

### B. MACHINE LEARNING METHODS

In existing studies, modelers appeared to focus on a high prediction rate in machine learning methods for default risk prediction. Depending on the domain, the modelers also utilized multiple datasets to appraise prediction models. Therefore, it is difficult to declare which classifiers provided the most accurate predictions. The stability of the proposed automated tax default prediction system can be ensured by investigating multiple classifiers in the decision support system. We trained 13 state-of-the-art machine learning methods from business intelligence domains, including both single classifiers (LDA, LR, $k$-NN, NB, MLP, SVM, extreme learning machine (ELM), and DT) and ensemble classifiers (RF, GB, XGBoost, SysFor, and the decision forest

by penalizing attributes (ForestPA)). Because of limited space considerations, this section presents only the most recent machine learning methods, namely, GB, XGBoost, SysFor, and ForestPA.

Concerning the remaining classifiers, LDA and LR are both discriminative classifiers that are linear in their parameters. The parameters of LR are estimated using maximum likelihood, while least squares estimation is used for LDA. The assumption of multivariate normal distribution is stronger for LDA. Moreover, identical covariance matrices are assumed for LDA, which makes LR more robust and less sensitive to outliers compared with LDA. The $k$-NN method is an example-based classifier, in which $k$ most-similar (typically in terms of Euclidean distance) instances are used to classify a new instance. In NB, posterior probabilities for classes are calculated given the values of independent variables, which are assumed to be conditionally independent given the class. MLP is a fully connected, feed-forward neural network. Gradient methods are used to learn the values of the MLP connection weights in order to minimize the training error. Unlike MLP, SVM aims to minimize the structural risk by maximizing margin between classes. To do so, a subset of the training instances (support vectors) is applied to determine the decision boundary. Similartly to MLP and SVM, universal approximation capability was proved for ELM. The parameters of ELM are typically learned in a single step with (constrained) random weights for connections between input and hidden layer. DT uses nodes to denote features chosen to maximize goodness of a split and branches to represent feature values dividing the data into subsets. To allow for a larger variance reduction, RF employs bagging (generating different samples from the training data) with random feature selection for an ensemble of DTs. Interested readers can find detailed descriptions of the above classifiers in the literature [44], [45].

#### 1) GRADIENT BOOSTING (GB) AND EXTREME GRADIENT BOOSTING (XGBoost)

The GB classifier follows a three-step configuration of the analytics mechanism, namely, (1) a loss function definition, (2) a weak model (tree) training, and (3) an additive technique that adjoins the weak trees to optimize the loss function. The loss function is defined and trained based on the characteristics of the underlying predictive task. Here we trained the softmax loss function defined as:

$$P_i = \frac{e^{f_{i,y_i}}}{\sum_{j=1}^{n} e^{f_{i,j}}}, \tag{6}$$

where $y_i$ is the target output, $i$ stands for the instance index, $f_{i,j}$ is the score of the $i$-th sample on the $j$-th class, $j$ is the class index, and $n$ is the number of classes. The GB algorithm trains many weak learners (shallow DTs) sequentially in order to minimize the loss function. Here we trained 2000 boosted trees in the greedy manner, which chooses the best split points based on the scores optimizing the losses. The boosted tree selects each input attribute as a feature. The leaf node herein symbolizes the softmax loss value. The old trees in the ensemble are not affected when the newly generated trees are inserted, and the loss function is optimized by the gradient descent method. This process continues until the classifier achieves the optimized performance score.

However, the greedy GB classifier may rapidly become overfitted for the given training set. The four following augmentations need to be considered to make the GB algorithm more effective, namely, the tree constraints, shrinkage, random sampling, and penalized learning. The XGBoost algorithm [46] is a scalable end-to-end tree-boosting technique that considers numerous adjustments to the GB algorithm. The most significant enhancement in XGBoost is that it augments a regularization factor to the loss function intended for creating ensembles that are straightforward and more generative. The regularization factor is added to the loss function to control the model complexity and, thus, avoid overfitting.

#### 2) SYSTEMATICALLY DEVELOPED FOREST OF MULTIPLE DECISION TREES (SysFor)

The SysFor algorithm was introduced by Islam and Giggins [47] to extract data patterns from low- to high-dimensional datasets. During the training process, the SysFor algorithm extracts the features that have a high predictive capability. SysFor produces a good set of trees even for low-dimensional data. The following steps illustrate the learning process of this algorithm:

**Step 1:** Select a good feature set and corresponding split points by applying user-defined goodness and separation thresholds.

**Step 2:** Choose each good feature one by one as the root feature (at level 0) of a tree when the feature set from Step 1 is larger than the number of trees defined by the user.

**Step 3:** If the number of trees is lower than that defined by the user, alternative good features (chosen at level 1) are used to construct additional trees.

**Step 4:** Return all trees generated in the two previous steps as a SysFor.

#### 3) DECISION FOREST BY PENALIZING ATTRIBUTES (ForestPA)

The ForestPA algorithm, proposed by Adnan and Islam [48], employs the entire feature set to produce the upcoming DT by imposing penalty weights on those features that emerged in the latest DTs. In addition, random weights are assigned to the used features depending on the levels in the DT. Thus, the diversity of the DTs is sustained. The learning steps of the ForestPA algorithm are as follows:

**Step 1:** Produce a bootstrap sample from the original training set. Thus, diversity is introduced into the base DTs [48], [49]. **Step 2:** Build a DT for the bootstrap sample by applying the feature weights. In doing so, it uses the classification and regression tree that applies merit values as the splitting criterion, which is calculated by multiplying the feature prediction ability with its weight.

**Step 3:** Update the feature weights and gradual weight augmentation values for the features in the newly generated trees. By contrast, feature weights remain identical if they do not emerge in the newly generated trees. The feature weights consider the tree levels for which they are tested over the newly generated trees.

The ForestPA algorithm ensures substantial diversity of the trained classifier by applying weight assignment as well as weight increment strategies towards the learned features.

### C. PERFORMANCE MEASURES

For tax default prediction, the class attribute is binary, with tax default as class 1 and tax nondefault as class 0. Concerning the predictive evaluation measures, this study applies standard performance measures such as accuracy, type I and type II errors, F-measure, and AUROC. This study refers to Moula *et al.* [50] for their detailed explanations.

In tax default applications, it is generally believed that the cost of failing to detect a default taxpayer $C_{12}$ is substantially larger than the cost of the wrong prediction of a nondefault taxpayer $C_{21}$. Based on this background, it is vital to make a viable trade-off between the cost of a ''false-positive'' error and the cost of a ''false-negative'' error with the following cost function [51]:

$$\text{EMCC} = C_{12}\lambda_2\,(q_2/Q_2) + C_{21}\lambda_1\,(q_1/Q_1) \tag{7}$$

where the expected misclassification cost (EMCC) ratio is 5:1, this is $C_{12} = 5$ and $C_{21} = 1$; $\lambda_1$ and $\lambda_2$ indicate the prior probabilities of nondefault and default taxpayers, respectively; the ratio $q_2/Q_2$ refers to the false-positive rate (the fraction of default taxpayers incorrectly predicted as nondefault taxpayers), and $q_1/Q_1$ denotes the false-negative rate (the fraction of nondefault taxpayers incorrectly predicted as default taxpayers).

Furthermore, the nonparametric Friedman test, along with the Iman-Davenport (FID) adjustment, was applied to evaluate the predictive performance of the used machine learning methods. In addition, the nonparametric Wilcoxon test was applied to compare the prediction performance between each pair of machine learning methods [52].

## IV. DATA

The proposed tax default prediction system was validated with a real-world tax payment dataset of Finnish limited liability firms originally compiled by Höglund [7]. Finland can serve as a relevant background for the study of tax default risk prediction because approximately 12% of all active Finnish enterprises had outstanding taxes at the end of the year 2015, with more than three billion Euros of overdue corporate taxes [7]. The total amount of settled taxes over the same period was 49 billion Euros. The Finnish tax bureau claims that only 20% of these unsettled state revenues may be recovered [7]. This huge amount of unpaid taxes leads to economic divergence.

The current study was designed for two experimental scenarios corresponding to the prediction of tax default (1) in the default year 2014 (i.e., using the financial indicators from the year 2014) and (2) 1 year prior to the default (using the data from 2013). Specifically, the tax default information is for defaults in the VAT and employer contribution tax in 2014. Of the total number of 768 firms, 384 firms defaulted and the remaining 384 fully paid their taxes. The nondefaulted firms were matched with the default sample for the year, industry, and size of total assets in the predefault year; see [7] for details. The experimental dataset is available at https://goo.gl/52cK41. For each firm, the database lists a total of 36 features, which are financial indicators collected for 2 years, 2013 and 2014. Two additional industry-related features, the industry payment default and industry bankruptcy risk, are reported for 2014. This database allowed us to construct two different datasets, one for 2013 (for the 1-year-ahead prediction) and the other for 2014 (for the tax default prediction during the same year). The datasets had no missing values. In addition to the original datasets, we generated five new datasets using the feature transformation techniques for each prediction horizon. The predictors of tax default risk are represented by financial indicators obtained from financial statements. Note that financial statement data are considered the most significant factors affecting the default risk in related business domains [5], [53]. The financial indicators applied in this study include (1) liquidity ratios (current ratio, quick ratio), (2) leverage ratios (debt-to-sales ratio, equity ratio), (3) firm size (total assets), (4) activity ratios (working capital–to-sales ratio, sales-to–total assets ratio, inventory turnover ratio, periods of trade receivables and payables), and (5) profitability ratios (operating income, operating margin, return on assets, return on investment). Table 1 presents descriptive statistics for the financial indicators (the values of features from 2013 were used for the 1-year-ahead prediction, and those from 2014 were used for the prediction in the default year), along with the independent sample $t$-test, to evaluate the statistical difference between the two sample categories (default vs. nondefault). Statistically significant differences were identified for all attributes except for the operating income. In addition, correlations among the attributes were investigated, as documented in Appendix A1; they indicate significant multicollinearity among some attributes, particularly in the year 2013. For traditional statistical models, this can cause a considerable problem with the interpretation of results. We also applied the Kaiser-Meyer-Olkin (KMO) test and Bartlett test of sphericity, obtaining the test statistics 0.507 and 38150.366 ($p < .01$), respectively. These results are presented in Appendix A2, and they suggest that factor analysis should be performed before using the parametric statistical prediction algorithms LR and LDA [54]. To overcome the problem of multicollinearity for LR and LDA, we employed principal component analysis (PCA). By considering the factors with eigenvalues higher than 1, 13 factors were extracted from the original 36 features that explain 79.51% of the cumulative variance (for details on cumulative variance and factor loading matrix, see Appendix A3). The factor loading

**TABLE 1. Descriptive statistics of the used financial indicators for the years 2013 (1 year prior to tax default) and 2014 (the year of tax default).**

| Feature | the year of tax default $t$ | | | 1 year prior to tax default $t$-1 | | |
|---|---|---|---|---|---|---|
| | Mean | SD | $t$-test | Mean | SD | $t$-test |
| Industry risk of payment default (A1 for $t$) | 0.0638 | 0.0285 | 62.015*** | NaN | NaN | NaN |
| Industry risk of bankruptcy (A2 for $t$) | 0.0071 | 0.0051 | 38.085*** | NaN | NaN | NaN |
| Sales / total assets (A3 for $t$, A4 for $t$-1) | 2.9015 | 3.1473 | 25.549*** | 2.9084 | 2.6492 | 30.425*** |
| Total assets (A5, A6) | 243.0116 | 503.6701 | 13.371*** | 247.6263 | 513.6732 | 13.360*** |
| Change in sales (A7, A8) | 0.2657 | 4.6136 | 1.596 | 0.2867 | 2.0855 | 3.810*** |
| Gross result / sales (A9, A10) | 0.6923 | 0.3531 | 54.339*** | 0.5937 | 4.7978 | 3.430*** |
| Operating margin (A11, A12) | 0.0541 | 0.3522 | 4.260*** | -0.0768 | 3.6230 | -0.587 |
| Operating income (A13, A14) | 0.0034 | 0.3591 | 0.259 | -0.1306 | 3.6222 | -0.999 |
| Quick ratio (A15, A16) | 2.2161 | 7.9457 | 7.729*** | 2.1236 | 6.1868 | 9.512*** |
| Current ratio (A17, A18) | 2.5599 | 7.8795 | 9.003*** | 2.5280 | 6.2891 | 11.140*** |
| Return on investment (A19, A20) | 0.0003 | 1.2003 | 0.007 | 0.0618 | 0.4673 | 3.666*** |
| Return on assets (A21, A22) | 0.0362 | 0.2510 | 3.994*** | 0.0498 | 0.2611 | 5.285*** |
| Equity ratio (A23, A24) | -0.0865 | 1.3724 | -1.747* | 0.0165 | 1.1592 | 0.395 |
| Net gearing (A25, A26) | 1.4414 | 17.4960 | 2.283** | 1.5150 | 49.1667 | 0.854 |
| Debt / sales (A27, A28) | 0.9811 | 3.6279 | 7.494*** | 1.1300 | 5.7095 | 5.485*** |
| Working capital / sales (A29, A30) | 0.0778 | 0.7109 | 3.033*** | 0.2299 | 3.6251 | 1.750* |
| Inventory turnover ratio (A31, A32) | 0.0952 | 0.2928 | 9.008*** | 0.2479 | 3.6676 | 1.873* |
| Collection period of trade receivables (A33, A34) | 31.3112 | 71.6465 | 12.111*** | 31.7357 | 61.5403 | 14.291*** |
| Payment period of trade payables (A35, A36) | 123.3880 | 337.6126 | 10.128*** | 175.4700 | 750.0660 | 6.483*** |

Note: the results of the Student's paired $t$-test show statistical differences between the default and nondefault target classes, ***significant at $p < .01$, **significant at $p < .05$, *significant at $p < .10$.

matrix can be used to interpret the 13 factors. For example, F1 represents activity ratios (A30, A32 and A7) and F2 stands for liquidity ratios (A15-A18).

## V. EXPERIMENTAL RESULTS

### A. EXPERIMENTAL SETTINGS

We performed the feature transformation techniques in Python 3.6 using the scikit-learn 0.23 library. Similarly, we used the same library to implement the following machine learning methods: LDA, LR, $k$-NN, NB, SVM, ELM, DT, RF, GB, and XGBoost. Data normality and feature contributions were also run using the Python environment. To implement MLP, SysFor, and ForestPA, we used the Waikato Environment for Knowledge Analysis (WEKA) 3.8.0 packages. Nonparametric statistical tests were conducted using the Knowledge Extraction based on Evolutionary Learning (KEEL) GPLv3 modules. The results of the descriptive statistics and factor and correlation analysis were obtained using the Statistical Package for the Social Sciences (SPSS) 17.0 platform. All these experiments were carried out on a PC with a 3.10-GHz Intel Core i5-2400 CPU and 4-GB RAM in the Windows 10 operating system. For the machine learning methods, the grid search procedure performed over the 10-fold cross-validation was used to find the optimal values of training parameters. Table 2 shows the optimal parameter settings and respective references for which technical details are given to the readers.

**TABLE 2. Settings of parameters for machine learning methods.**

| Method | Parameters |
|---|---|
| LDA and LR | cut-off point = 0.5 [50] |
| $k$-NN | $k$ = 5, Euclidean distance [55] |
| NB | batch size = 100 [55] |
| MLP | no. of neurons in the hidden layer = (no. of features + no. of classes) / 2, learning rate = 0.1, no. of iterations = 500 |
| SVM | radial basis kernel function<br>default value of the complexity parameter $C$ (i.e., $C$ = 1) [55] |
| ELM | one hidden layer, no. of nodes in the hidden layer = 80<br>activation function = inv_multiquadric |
| DT | C4.5 algorithm, min. no. of instances per leaf = 2<br>confidence factor for pruning = 2 |
| RF | no. of trees = 80; no. of split features = 5<br>maximum tree depth $d$ = 25 [55] |
| GB | default booster, learning rate = 0.01, max. depth = 0.5 |
| XGBoost | learning rate = 0.1, gamma = 0<br>depth of trees = 6, no. of iterations = 200 |
| SysFor | confidence factor = 0.25, min. gain ratio = 0.3<br>min. number of instances in a leaf = 10, no. of trees = 60 |
| ForestPA | no. of trees = 10, min. no. of instances per leaf = 2<br>no. of pruning folds = 2 |

### B. NORMALITY TESTS

Normality tests are important to validate the contributions of the feature transformation methods. For data normality assessment, here we used two widely applied statistical characteristics, namely, skewness and kurtosis, to investigate the shape of the probability distributions. Skewness measures the degree of asymmetry in the probability distribution of the features. The symmetric distribution refers to zero skewness, and the positive (negative) skewness designates a long right (left) tail; the recommended range is –0.80 to +0.80. Kurtosis signifies the "peakness," that is, the extent to which the probabilities of features concentrate in the center, with the recommended range between –10 and +10.

The Q-Q (quantile-quantile) plots in Figure 2 illustrate the effect of the five feature transformations on the normality measures, whereas the boxplots in Figure 3 demonstrate the impact of feature transformations on the data distribution compared with the nontransformed (raw) features. Due to space constraints, only a limited number of features is

**TABLE 3.** Results of the Wilcoxon tests for data normality measures.

| normality measure | log-tr. vs. raw | $Z$-nor.-tr. vs. raw | scaled-tr. vs. raw | sine-tr. vs. raw | sqrt-tr. vs. raw |
|---|---|---|---|---|---|
| skewness | -1.980 (0.048)** | 0.000 (1.000) | 0.000 (1.000) | -1.720 (0.085)* | -1.744 (0.081)* |
| kurtosis | -4.870 (1.11E-6)*** | 0.000 (1.000) | 0.000 (1.000) | -5.232 (1.68E-7)*** | -5.216 (1.82E-7)*** |

Note: $p$ value of the Wilcoxon test are given in parenthesis, ***significant at $p < .01$, **significant at $p < .05$, *significant at $p < .10$.

presented in Figures 2 and 3. The remaining results (for features A2 to A36) are provided as online supplementary material 1. As can be seen from the Q-Q plots, we can assert that features satisfy the normal distribution after the application of the feature transformation techniques. For example, raw feature A2 has a skewness of 1.03 and a kurtosis of 3.14, whereas its square-root transformation carries a skewness of –0.41 and a kurtosis of –0.03. This indicates that A2 is normally distributed when using the square-root transformation. Figure 3 illustrates different effects of the feature transformation methods for different features. For example, the $Z$-normalization and scaled transformations for features A1 and A2 carry more outliers in the data. By contrast, the feature transformation methods reduce the effects of outliers for features A3 and A4.

Table 3 shows that significant statistical differences exist between the feature transformation methods in normality measures, as indicated by the results of the Wilcoxon nonparametric tests. Significantly improved measurements were achieved for both the skewness ($p < .10$) and kurtosis ($p < .05$). However, the statistical significance only applies to the log, sine, and square-root transformations; the remaining two techniques proved statistically insignificant. Therefore, our results support the conclusion that feature transformations significantly influence data normality in the tax default dataset.

### C. TAX DEFAULT PREDICTION

This section presents the performance of tax default prediction approaches and a comparative analysis of their performance using nonparametric tests to validate the proposed prediction architecture. First, we examine the effects of feature transformation methods on the prediction performance. Second, we compare the performance of machine learning methods in terms of various classification performance criteria. Finally, the importance of features is investigated separately for the default year and the 1 year prior to default.

#### 1) EFFECT OF FEATURE TRANSFORMATION METHODS

Note that here we present the main outcomes of the experimentation conducted across the five feature transformation methods. All the underlying results can be found in online supplementary material 2 (see supplementary Tables S1 to S6).

To rank the feature transformation methods, their performance was compared in terms of accuracy and $F$-measure. The Friedman ranking in Table 4 shows that the log-transformation method ranked first for both the performance criteria and the two prediction scenarios (i.e., for the default year and 1 year prior to default). The square-root transformation ranked second for all settings. The sine transformation ranked third for the default year, and the scaled transformation ranked third for the 1 year prior to default setting. By contrast, the $Z$-normalization transformation was the least effective approach among the tested feature transformation methods. The raw dataset was only competitive in case of the 1-year-ahead prediction. Overall, the results illustrate that feature transformation ensures better prediction performance for the machine learning methods. The differences among the feature transformation methods are statistically significant, as indicated by the result of the Friedman test ($p < .01$). The top-ranked feature transformation algorithm performed significantly better than the remaining methods in all the tested scenarios, as indicated by the Iman-Davenport test and Holm post-hoc analysis. Therefore, we reject the null hypothesis that the feature transformation methods perform similarly.

The nontransformed features performed significantly worse than their transformed counterparts. To highlight the contribution of the feature transformation methods over the raw untransformed data, we evaluated their competitive advantage ($CA$), obtained as follows:

$$CA_{\mathrm{ij}} = \left(\left(P_{\mathrm{ii}} - Q_{\mathrm{ij}}\right)/Q_{\mathrm{ij}}\right) * 100 \tag{8}$$

where $CA_{ij}$ is the competitive advantage for the $i$-th feature transformation method and $j$-th performance measure; $P_{ij}$ is the performance of the transformation method; and $Q_{ij}$ is the performance of the baseline method (no transformation). The Wilcoxon test was performed to evaluate the statistical significance of $CA$.

Table 5 shows that feature transformation provides substantial competitive gains across all machine learning methods. The most benefits were obtained for $k$-NN, NB, ELM, and SVM. Notably, a 38.94% and 29.55% increase in accuracy was achieved for ELM for the default year and 1 year prior to default, respectively. Regarding the $F$-measure, the highest improvement was obtained for SVM (46.30% and 47.41%, respectively). Without feature transformation, ELM, SVM, and NB performed worst, whereas LR, GB, and XGBoost were the best performers. After performing feature transformation, the performance of the poorly performing methods significantly improved but no significant difference was observed for the methods that had performed well. The low margins observed for RF, GB, XGBoost, SysFor, and

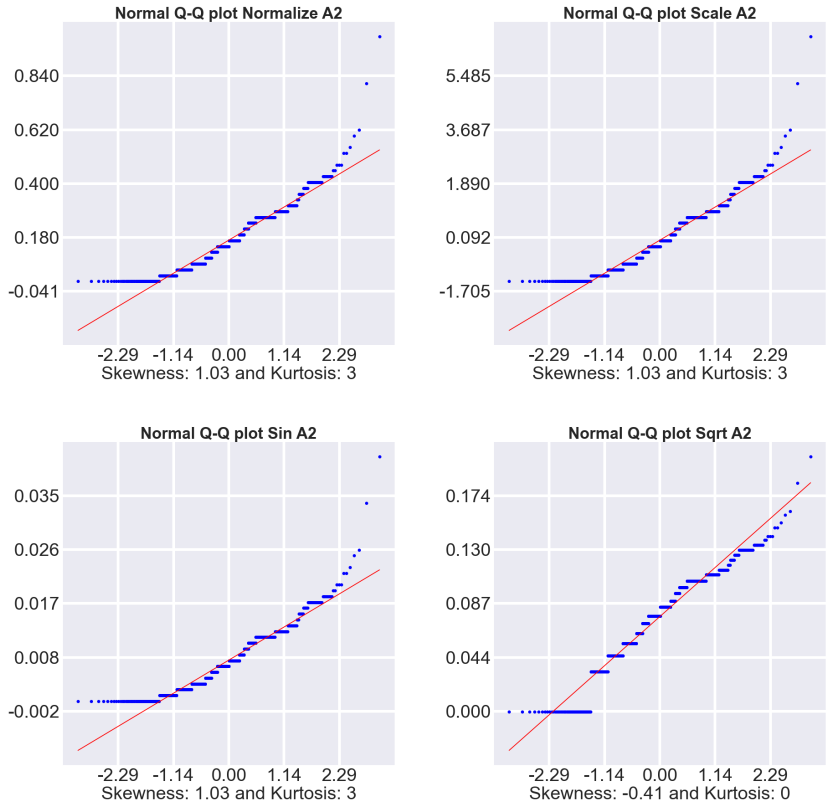



**FIGURE 2.** Normal Q-Q plots of feature A2 over the feature transformation methods.

**TABLE 4.** Results of the Friedman ranking tests and Holm post-hoc tests for the feature transformation methods.

| Feature transformation | in the default year | | | | 1 year prior to default | | | |
|---|---|---|---|---|---|---|---|---|
| | Accuracy | | $F$-measure | | Accuracy | | $F$-measure | |
| | Aver. Rank. (#) | Holm $p$-value | Aver. Rank. (#) | Holm $p$-value | Aver. Rank. (#) | Holm $p$-value | Aver. Rank. (#) | Holm $p$-value |
| log-tr. | 2.35 (#1) | ——— | 2.42 (#1) | ——— | 2.42 (#1) | ——— | 2.46 (#1) | ——— |
| sqrt-tr. | 2.65 (#2) | 0.050** | 2.69 (#2) | 0.050** | 2.88 (#2) | 0.050** | 3.00 (#2) | 0.050** |
| sine-tr. | 3.69 (#3) | 0.025** | 3.62 (#3) | 0.025** | 4.00 (#5) | 0.013** | 4.00 (#5) | 0.013** |
| scaled-tr. | 3.88 (#4) | 0.017** | 3.85 (#4) | 0.0167** | 3.31 (#3) | 0.025** | 3.27 (#3) | 0.025** |
| $Z$-nor-tr. | 3.96 (#5) | 0.013** | 3.96 (#5) | 0.013** | 4.62 (#6) | 0.010*** | 4.58 (#6) | 0.010*** |
| raw data | 4.46 (#6) | 0.010*** | 4.46 (#6) | 0.010*** | 3.77 (#4) | 0.017** | 3.69 (#4) | 0.017** |
| Iman-Davenport $p$-value | | 0.022** | | 0.036** | | 0.033** | | 0.055* |

Note: average Friedman ranking is reported together with the final ranking in parenthesis, ***significant at $p < .01$, **significant at $p < .05$, *significant at $p < .10$.

ForestPA signify that the ensemble machine learning methods provided a stable performance in the tested scenarios without the necessity for feature transformation.

### 2) COMPARATIVE ANALYSIS OF MACHINE LEARNING METHODS

To compare the performance of the machine learning methods, all six scenarios (i.e., the untransformed data and five transformations) were considered. According to the Friedman ranking in Table 6, the XGBoost was the best algorithm, taking the top position in terms of accuracy and $F$-measure for the default year. SysFor performed best among the methods when the 1-year-prior-to-default scenario was considered. Besides, the performances of the top-ranked classifiers were statically significant based on the Iman-Davenport and Holm post-hoc tests. By contrast, $k$-NN, DT, and ELM ranked worst for both prediction scenarios. Overall, the ensemble learning methods outperformed the single classifiers, except for LR for the default year prediction. Indeed, LR seemed to be the best choice of the single-classifier approaches, a finding that is consistent with a recent work of Beutel *et al.* [16] for the default prediction domain.

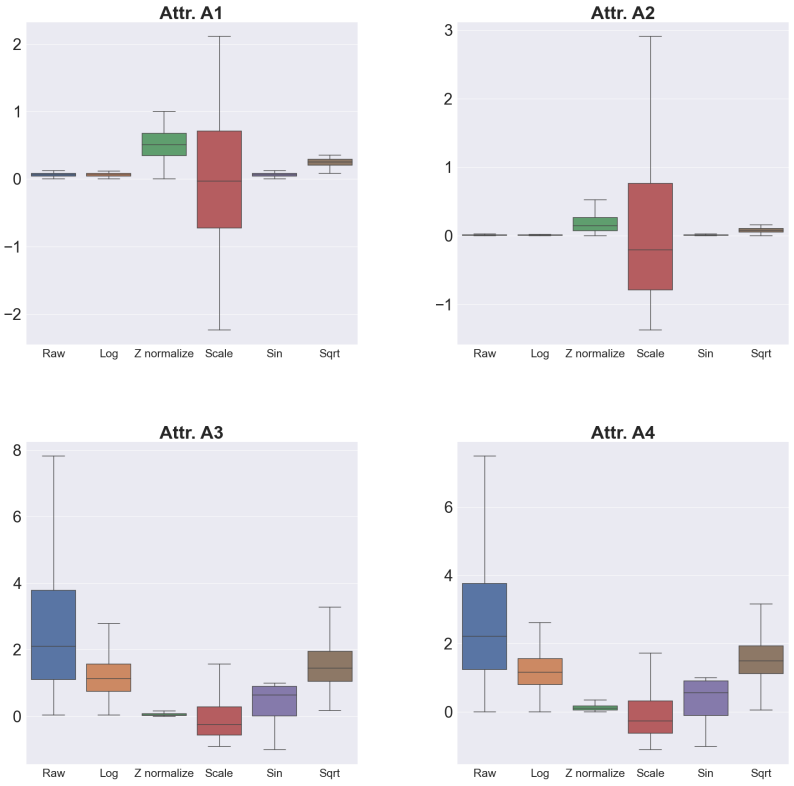



**FIGURE 3.** Boxplots for features A1 to A4 over the feature transformation methods.

The proposed prediction models also performed well in terms of time efficiency. The results of the training time criterion (measured as wall-clock time in seconds) showed that NB, LDA and $k$-NN performed best with average training times of 0.01 s, 0.03 s and 0.06 s, respectively. By contrast, GB and MLP performed worst with 2.3 and 3.0 s, respectively. Overall, we conclude that all the machine learning methods can be considered time efficient.

As emphasized above, different misclassification costs of false-positive and false-negative instances should be considered when evaluating the performance of a tax default prediction system. To assess the EMCC performance, the ratio between the false-positive and false-negative errors was set to 5:1. This ratio was normalized to obtain the following costs: $C_{12} = 0.833$ and $C_{21} = 0.167$. In Figure 4, we present the mean EMCC values achieved by the machine learning algorithms, whereas the detailed results for EMCC can be found in online supplementary material 2. XGBoost and GB achieved the lowest EMCC, thus providing the best trade-off between the false-positive and false-negative errors under different misclassification costs. Table 7 shows the pairwise comparison among the machine learning methods using the nonparametric Wilcoxon signed-rank (WSR) test. The bottom left corner and upper right corner of Table 7 show WSR test results for the default-year and 1-year-prior-to-default datasets, respectively. For the default year, only GB performed in a statistically similar manner

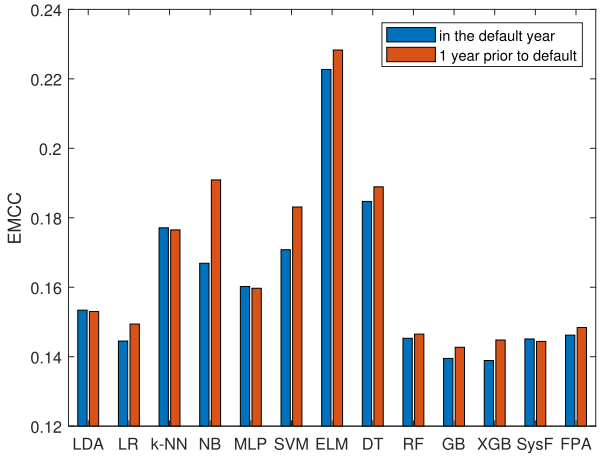



**FIGURE 4.** Mean EMCC values for the used machine learning methods.

as the best-performing XGBoost; all the tested ensemble learning methods performed similarly for the 1-year-ahead prediction scenario. In both scenarios, the following methods were significantly outperformed: SVM, NB, MLP, $k$-NN, DT, and ELM.

### 3) VERIFICATION OF FEATURE IMPORTANCE

To verify the feature importance for corporate tax default risk prediction, the trained RF model was utilized because it retains feature ranking. Indeed, feature validation through

**TABLE 5.** Competitive advantages of feature transformation over no transformation.

| Method | Accuracy | | $CA$ (%) | $F$-measure | | $CA$ (%) |
|---|---|---|---|---|---|---|
| | transformation | no transformation | | transformation | no transformation | |
| Panel A: in the default year | | | | | | |
| LDA | 0.7031 | 0.6771 | 3.8399 (0.407) | 0.7025 | 0.6768 | 3.7973 (0.508) |
| LR | 0.7070 | 0.7227 | -2.1724 (0.374) | 0.7066 | 0.7226 | -2.2142 (0.386) |
| $k$-NN | 0.6771 | 0.6172 | 9.7051 (0.033)** | 0.6764 | 0.6168 | 9.6628 (0.022)** |
| NB | 0.6966 | 0.6328 | 10.0822 (0.065)* | 0.6923 | 0.6198 | 11.6973 (0.028)** |
| MLP | 0.6641 | 0.6914 | -3.9485 (0.890) | 0.6640 | 0.6910 | -3.9074 (0.100)* |
| SVM | 0.7148 | 0.5417 | 31.9550 (0.005)*** | 0.7129 | 0.4873 | 46.2959 (0.005)*** |
| ELM | 0.6133 | 0.4414 | 38.9443 (0.028)** | 0.6130 | 0.4410 | 39.0023 (0.028)** |
| DT | 0.6315 | 0.6315 | —— | 0.6315 | 0.6315 | —— |
| RF | 0.7240 | 0.7083 | 2.2166 (0.906) | 0.7239 | 0.7083 | 2.2024 (0.959) |
| GB | 0.7318 | 0.7227 | 1.2592 (0.799) | 0.7317 | 0.7226 | 1.2593 (0.799) |
| XGBoost | 0.7188 | 0.7201 | -0.1805 (0.859) | 0.7185 | 0.7200 | -0.2083 (0.799) |
| SysFor | 0.7161 | 0.7096 | 0.9160 (0.890) | 0.7160 | 0.7100 | 0.8451 (1.00) |
| ForestPA | 0.7174 | 0.7031 | 2.0338 (0.890) | 0.7170 | 0.7030 | 1.9915 (0.69) |
| Panel B: 1 year prior to default | | | | | | |
| LDA | 0.7135 | 0.6862 | 3.9784 (0.093)* | 0.7131 | 0.6859 | 3.9656 (0.074)* |
| LR | 0.7174 | 0.6940 | 3.3718 (0.358) | 0.7170 | 0.6938 | 3.3439 (0.333) |
| $k$-NN | 0.6549 | 0.5924 | 10.5503 (0.059)* | 0.6549 | 0.5918 | 10.6624 (0.059)* |
| NB | 0.6823 | 0.5326 | 28.1074 (0.008)*** | 0.6793 | 0.4705 | 44.3783 (0.005)*** |
| MLP | 0.6862 | 0.6862 | —— | 0.6860 | 0.6860 | —— |
| SVM | 0.6927 | 0.5365 | 29.1146 (0.005)*** | 0.6912 | 0.4689 | 47.4088 (0.005)*** |
| ELM | 0.6055 | 0.4674 | 29.5464 (0.013)** | 0.6052 | 0.4662 | 29.8155 (0.013)** |
| DT | 0.6328 | 0.6159 | 2.7439 (0.610) | 0.6328 | 0.6159 | 2.7439 (0.610) |
| RF | 0.7148 | 0.7135 | 0.1822 (0.779) | 0.7147 | 0.7135 | 0.1682 (0.959) |
| GB | 0.7148 | 0.7174 | -0.3624 (1.000) | 0.7146 | 0.7173 | -0.3764 (0.878) |
| XGBoost | 0.7031 | 0.7135 | -1.4576 (0.386) | 0.7029 | 0.7132 | -1.4442 (0.386) |
| SysFor | 0.7148 | 0.7188 | -0.5565 (0.890) | 0.7150 | 0.7180 | -0.4178 (1.000) |
| ForestPA | 0.7057 | 0.7122 | -0.9127 (0.890) | 0.7060 | 0.7120 | -0.8427 (0.690) |

Note: $p$ value of the Wilcoxon test in parenthesis, ***significant at $p < .01$, **significant at $p < .05$, *significant at $p < .10$.

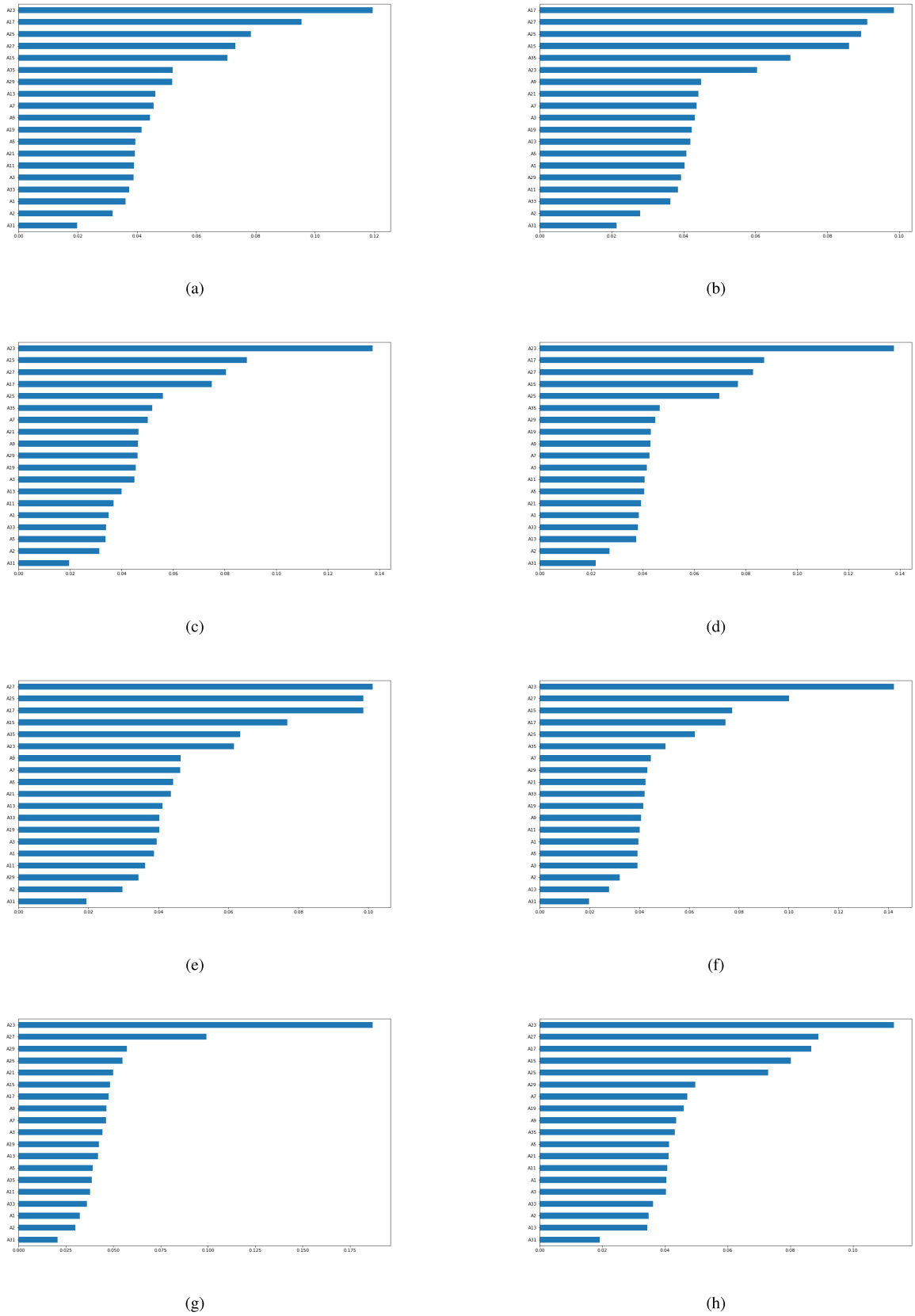
**TABLE 6.** Results of the Friedman ranking tests and Holm post-hoc tests for the used machine learning methods.

| Method | in the default year | | | | 1 year prior to default | | | |
|---|---|---|---|---|---|---|---|---|
| | Accuracy Aver. Rank. (#) | Holm $p$-value | $F$-measure Aver. Rank. (#) | Holm $p$-value | Accuracy Aver. Rank. (#) | Holm $p$-value | $F$-measure Aver. Rank. (#) | Holm $p$-value |
| XGBoost | 2.20 (#1) | ———— | 2.20 (#1) | ———— | 3.70 (#3) | 0.025** | 3.80 (#3) | 0.025** |
| GB | 2.60 (#2) | 0.050** | 2.60 (#2) | 0.050** | 2.50 (#2) | 0.050** | 2.60 (#2) | 0.050** |
| LR | 4.30 (#3) | 0.025** | 4.30 (#3) | 0.025** | 5.20 (#5) | 0.010*** | 5.20 (#5) | 0.010** |
| RF | 4.80 (#4) | 0.017** | 4.80 (#4) | 0.013** | 4.30 (#4) | 0.017** | 4.10 (#4) | 0.017** |
| ForestPA | 4.90 (#5) | 0.013** | 4.80 (#4) | 0.017** | 5.20 (#5) | 0.013** | 5.20 (#5) | 0.0130** |
| SysFor | 5.00 (#6) | 0.010*** | 5.00 (#6) | 0.010*** | 2.40 (#1) | ———— | 2.40 (#1) | ———— |
| LDA | 7.00 (#7) | 0.008*** | 7.00 (#7) | 0.008*** | 5.90 (#7) | 0.008*** | 6.20 (#7) | 0.008*** |
| SVM | 7.40 (#8) | 0.007*** | 7.50 (#8) | 0.007*** | 9.00 (#9) | 0.006*** | 9.00 (#9) | 0.006*** |
| NB | 8.80 (#9) | 0.006*** | 9.20 (#10) | 0.006*** | 10.10 (#10) | 0.006*** | 10.50 (#11) | 0.005*** |
| MLP | 9.00 (#10) | 0.006*** | 9.00 (#9) | 0.006*** | 8.50 (#8) | 0.007*** | 8.20 (#8) | 0.007*** |
| $k$-NN | 10.80 (#11) | 0.005*** | 10.60 (#11) | 0.005*** | 10.40 (#11) | 0.005*** | 10.20 (#10) | 0.006*** |
| DT | 11.20 (#12) | 0.005*** | 11.00 (#12) | 0.005*** | 11.20 (#12) | 0.005*** | 11.20 (#12) | 0.005*** |
| ELM | 13.00 (#13) | 0.004*** | 13.00 (#13) | 0.004*** | 12.60 (#13) | 0.004*** | 12.40 (#13) | 0.004*** |
| Iman-Davenport $p$-value | | 0.41E-10*** | | 0.68E-10*** | | 0.2E-11*** | | 0.1E-10*** |

Note: average Friedman ranking is reported together with the final ranking in parenthesis, ***significant at $p < .01$, **significant at $p < .05$, *significant at $p < .10$.

the RF classifier has been approved as a feasible approach in many feature ranking studies [56]–[58]. The feature importance in RF is verified by the disparity of a tree's out-of-bag error before and after random permutation of an explanatory feature. More precisely, the feature importance is calculated by how much the model error is increased when permutation of the feature has occurred, and the differences are averaged over all random trees generated. Random permutations also allow us to investigate the feature importance among correlated features.

Figures 5 and 6 demonstrate the average feature importance over six transformed and nontransformed datasets in



(a) (b) (c) (d) (e) (f) (g) (h)

**FIGURE 5.** Average feature importance for the default year prediction across the used machine learning methods.

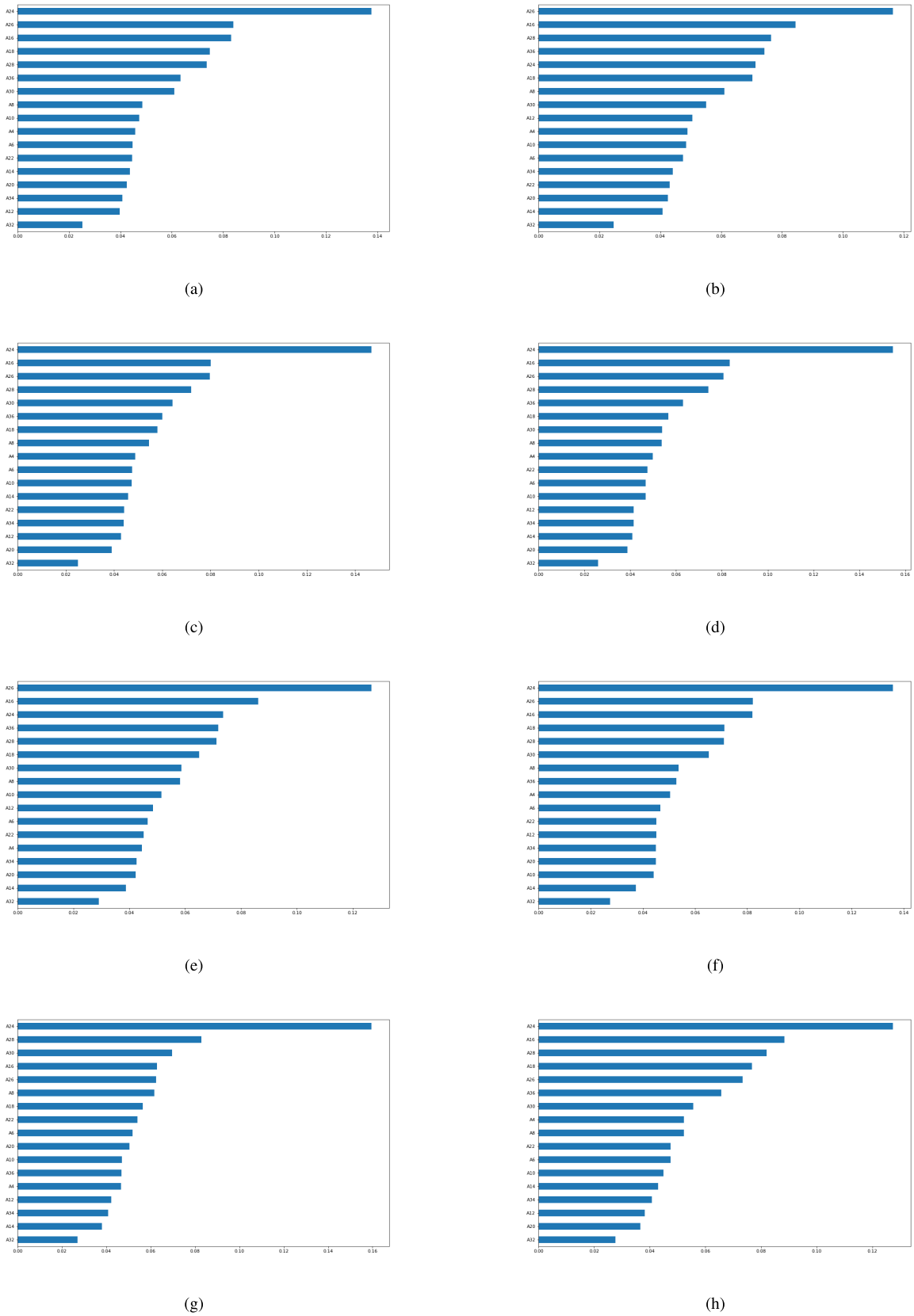


**FIGURE 6.** Average feature importance for the 1-year-ahead prediction across the used machine learning methods.

**TABLE 7.** Pairwise Wilcoxon signed rank test for EMCC (left bottom corner, default year; right upper corner, 1 year prior to default).

| | XGB | GB | LR | RF | ForestPA | SysFor | LDA | SVM | NB | MLP | $k$-NN | DT | ELM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XGB | | 0.173 | 0.172 | 0.416 | 0.207 | 0.249 | 0.115 | 0.028** | 0.027** | 0.027** | 0.028** | 0.028** | 0.028** |
| GB | 0.833 | | 0.046** | 0.104 | 0.249 | 0.600 | 0.075* | 0.028** | 0.028** | 0.028** | 0.028** | 0.027** | 0.028** |
| LR | 0.075* | 0.173 | | 0.116 | 0.345 | 0.116 | 1.000 | 0.028** | 0.028** | 0.116 | 0.028** | 0.027** | 0.028** |
| RF | 0.075* | 0.046** | 0.917 | | 0.249 | 0.028** | 0.138 | 0.027** | 0.043** | 0.028** | 0.028** | 0.028** | 0.028** |
| ForestPA | 0.027** | 0.046** | 0.917 | 0.600 | | 0.046** | 0.345 | 0.046** | 0.046** | 0.027** | 0.028** | 0.028** | 0.028** |
| SysFor | 0.028** | 0.028** | 0.917 | 0.917 | 0.463 | | 0.028** | 0.028** | 0.028** | 0.027** | 0.028** | 0.028** | 0.028** |
| LDA | 0.046** | 0.075* | 0.028** | 0.249 | 0.172 | 0.173 | | 0.075* | 0.027** | 0.027** | 0.028** | 0.028** | 0.028** |
| SVM | 0.028** | 0.046** | 0.080* | 0.116 | 0.028** | 0.116 | 0.463 | | 0.116 | 0.600 | 0.345 | 0.345 | 0.046** |
| NB | 0.027** | 0.028** | 0.028** | 0.046** | 0.046** | 0.028** | 0.027** | 0.753 | | 0.074* | 0.600 | 0.753 | 0.046** |
| MLP | 0.027** | 0.028** | 0.046** | 0.028** | 0.027** | 0.028** | 0.115 | 0.753 | 0.752 | | 0.046** | 0.028** | 0.028** |
| $k$-NN | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | 0.075* | 0.028** | 0.116 | | 0.046** | 0.028** |
| DT | 0.028** | 0.027** | 0.028** | 0.028** | 0.028** | 0.028** | 0.027** | 0.046** | 0.046** | 0.046** | 0.116 | | 0.028** |
| ELM | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | 0.028** | |

Note: **significant at $p < .05$, *significant at $p < .10$.

**TABLE 8.** Feature importance over the feature transformation methods.

| feature | feature importance (%) (ranking #) for the default year prediction | | | | | | feature importance (%) (position #) for the 1-year-ahead prediction | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | log-tr. | sqrt-tr. | sine-tr. | scaled-tr. | $Z$-nor-tr. | raw data | log-tr. | sqrt-tr. | sine-tr. | scaled-tr. | $Z$-nor-tr. | raw data |
| A1 | 3.50 (#15) | 4.04 (#14) | 3.23 (#17) | 3.96 (#14) | 3.85 (#15) | 3.60 (#17) | NaN | NaN | NaN | NaN | NaN | NaN |
| A2 | 3.14 (#18) | 3.47 (#17) | 3.01 (#18) | 3.21 (#17) | 2.71 (#18) | 3.18 (#18) | NaN | NaN | NaN | NaN | NaN | NaN |
| A3, A4 | 4.50 (#12) | 4.03 (#15) | 4.43 (#10) | 3.92 (#16) | 4.16 (#11) | 3.89 (#15) | 4.87 (#9) | 5.23 (#8) | 4.66 (#13) | 5.04 (#9) | 4.99 (#9) | 4.57 (#10) |
| A5, A6 | 3.37 (#17) | 4.13 (#11) | 3.91 (#13) | 3.93 (#15) | 4.10 (#13) | 3.94 (#12) | 4.74 (#10) | 4.75 (#10) | 5.18 (#9) | 4.66 (#10) | 4.67 (#11) | 4.48 (#11) |
| A7, A8 | 5.02 (#7) | 4.72 (#7) | 4.62 (#9) | 4.46 (#7) | 4.26 (#10) | 4.56 (#9) | 5.45 (#8) | 5.22 (#9) | 6.17 (#6) | 5.38 (#7) | 5.38 (#8) | 4.85 (#8) |
| A9, A10 | 4.63 (#9) | 4.36 (#9) | 4.63 (#8) | 4.07 (#12) | 4.29 (#9) | 4.44 (#10) | 4.73 (#11) | 4.49 (#11) | 4.71 (#11) | 4.40 (#14) | 4.67 (11) | 4.72 (#9) |
| A11, A12 | 3.69 (#14) | 4.07 (#13) | 3.79 (#15) | 4.01 (#13) | 4.08 (#12) | 3.90 (#14) | 4.29 (#15) | 3.82 (#14) | 4.21 (#14) | 4.51 (#12) | 4.15 (#12) | 3.98 (#16) |
| A13, A14 | 3.99 (#13) | 3.43 (#18) | 4.19 (#12) | 2.79 (#18) | 3.75 (#17) | 4.61 (#8) | 4.59 (#12) | 4.31 (#12) | 3.80 (#16) | 3.72 (#15) | 4.10 (#14) | 4.37 (#13) |
| A15, A16 | **8.87 (#2)** | **8.01(#4)** | 4.82 (#6) | **7.72 (#3)** | **7.70 (#4)** | **7.04 (#5)** | **8.02 (#2)** | **8.84(#2)** | **6.29 (#4)** | **8.20 (#3)** | **8.34 (#2)** | **8.30 (#3)** |
| A17, A18 | **7.50 (#4)** | **8.67(#3)** | 4.75 (#7) | **7.47 (#4)** | **8.72 (#2)** | **9.55 (#2)** | 5.81 (#7) | **7.68(#4)** | 5.65 (#7) | **7.13 (#4)** | 5.67 (#6) | **7.48 (#4)** |
| A19, A20 | 4.55 (#11) | 4.60 (#8) | 4.26 (#11) | 4.15 (#11) | 4.31 (#8) | 4.15 (#11) | 3.90 (#16) | 3.66 (#15) | 5.04 (#10) | 4.50 (#13) | 3.89 (#15) | 4.24 (#14) |
| A21, A22 | 4.65 (#8) | 4.11 (#12) | **4.99 (#5)** | 4.25 (#9) | 3.94 (#14) | 3.92 (#13) | 4.41 (#13) | 4.75 (#10) | 5.41 (#8) | 4.52 (#11) | 4.76 (#10) | 4.45 (#12) |
| A23, A24 | **13.74 (#1)** | **11.31(#1)** | **18.70(#1)** | **14.23 (#1)** | **13.77 (#1)** | **11.94 (#1)** | **14.68 (#1)** | **12.74(#1)** | **15.95(#1)** | **13.59 (#1)** | **15.46 (#1)** | **13.76 (#1)** |
| A25, A26 | **5.61 (#5)** | **7.30(#5)** | **5.49 (#4)** | **6.23 (#5)** | **6.98 (#5)** | **7.84 (#3)** | **7.97 (#3)** | **7.34(#5)** | **6.23 (#5)** | **8.21 (#2)** | **8.07 (#3)** | **8.40 (#2)** |
| A27, A28 | **8.05 (#3)** | **8.90(#2)** | **9.92 (#2)** | **10.02 (#2)** | **8.29 (#3)** | **7.31 (#4)** | **7.19 (#4)** | **8.21(#3)** | **8.28 (#2)** | **7.10 (#5)** | **7.42 (#4)** | **7.35 (#5)** |
| A29, A30 | 4.62 (#10) | 4.97 (#6) | **5.72 (#3)** | 4.32 (#8) | 4.48 (#7) | 5.19 (#7) | **6.43 (#5)** | 5.57 (#7) | **6.96 (#3)** | 6.52 (#6) | 5.40 (#7) | 6.09 (#7) |
| A31, A32 | 1.96 (#19) | 1.91 (#19) | 2.06 (#19) | 1.99 (#19) | 2.17 (#19) | 1.97 (#19) | 2.50 (#17) | 2.76 (#16) | 2.70 (#17) | 2.74 (#16) | 2.59 (#16) | 2.52 (#17) |
| A33, A34 | 3.40 (#16) | 3.62 (#16) | 3.62 (#16) | 4.22 (#10) | 3.81 (#16) | 3.73 (#16) | 4.40 (#14) | 4.08 (#13) | 4.07 (#15) | 4.50 (#13) | 4.15 (#13) | 4.08 (#15) |
| A35, A36 | 5.20 (#6) | 4.31 (#10) | 3.86 (#14) | 5.04 (#6) | 4.66 (#6) | 5.20 (#6) | 6.01 (#6) | 6.66 (#6) | 4.69 (#12) | 5.28 (#8) | **6.31 (#5)** | 6.34 (#6) |

Note: A1: Industry risk of payment default; A2: industry risk of bankruptcy; A3 sales / total assets for $t$, A4 sales / total assets for $t$-1; A5, A6: total assets; A7, A8: change in sales; A9, A10: gross result / sales; A11, A12: operating margin; A13, A14: operating income; A15, A16: quick ratio; A17, A18; current ratio; A19, A20: return on investment; A21, A22: return on assets; A23, A24: equity ratio; A25, A26: net gearing; A27, A28: debt / sales; A29, A30: working capital / sales; A31, A32: inventory turnover ratio; A33, A34: collection period of trade receivables; A35, A36: payment period of trade payables.

the default year and 1 year prior to default, respectively. The equity ratio was the most important feature for separating nondefault taxpayers from their default peers, being a critical early-warning indicator of tax default. The liquidity ratios (quick ratio and current ratio) and debt-to-sales ratio were also important, signifying that liquidity is another key indicator. Less indebted firms with more current assets and liquid reserves are less likely to fail to pay taxes. By contrast, the inventory turnover ratio had the smallest effect on the RF prediction accuracy. Table 8 shows that the underlying features contributed more than the machine learning methods for each of the feature transformation methods. The top five contributing features are in boldfont type, confirming the crucial role of the equity ratio across all feature transformations. Note that the feature ranking is relatively stable across the feature transformation methods; this validates the importance of the leverage and liquidity ratios for tax default prediction.

## VI. CONCLUSION

The importance of business analytics in the data-driven era of accounting has been recognized by the key accounting professional organizations. Notably, the American Institute of Certified Public Accountants and the Institute of Internal Auditors acknowledged business analytics as one of the top innovations and research priorities in the accounting domain. Machine learning methods are increasingly employed to predict financial statement anomalies. These technologies not only save time and money for audit stakeholders, but also reduce the risk of conspiracy and human error in the discovery of fraudulent and default events. Here we have presented a novel tax default prediction system using feature transformation–based machine learning. Returning to the questions posed at the beginning of this study, it is now possible to state that feature transformation significantly influences the tax default data normality; this implies that it is critically important for traditional statistical prediction

methods. In addition, this study has found that in general, feature transformation methods substantially improve the prediction performance of machine learning methods regardless of whether they are single or ensemble classifiers. This indicates that the application of feature transformations has significant advantages over the application of non-transformed raw data. Log and square-root transformations emerged as reliable feature transformation methods across the used machine learning methods. These findings clearly support the relevance of feature transformation in tax default prediction.

Concerning the effectiveness of the used machine learning methods, the results suggest that XGBoost outperformed the remaining methods in the default year sample and that SysFor dominated for the 1-year-ahead tax default prediction. The LR model also performed well with consistent outcomes, which is in agreement with the findings of Beutel *et al.* [16] and Höglund [7]. The present study provides additional evidence with respect to the effectiveness of ensemble-based machine learning methods in financial default prediction. We have shown that XGBoost is particularly effective in predicting tax defaults in the real-world scenario of different misclassification costs, which implies a substantial cost reduction for decision makers. Another considerable managerial implication is related to the investigated feature importance. Our study suggests that the equity ratio should be considered as the most informative indicator distinguishing tax nondefault firms from their default counterparts. Besides, liquidity ratios and other leverage ratios represent other important financial indicators; this corroborates the findings for alternative default prediction models [5], [6], [53].

From the managerial point of view, the insights of this study may assist financial managers of firms, tax administrators, tax auditors, suppliers, creditors, regulators, and government employees by providing guidelines to minimize tax default probability. The automatic early-warning tax default detection system can be used to reduce the work burden of tax administrators and enhance administrative efficiency. Eventually, the system may also help government workers to collect more of the taxes owed. Our findings may also support auditors and borrowers in their appraisal of the possibility of tax default, which may aid them in choosing a reliable firm. Additionally, firms could benefit from the automatic assessment of their tax documents so as to manage their financial risks more effectively.

The current investigation was limited by the use of a single dataset. Indeed, it is difficult to collect a reliable dataset due to the unavailability of data on the official tax default status of firms. Hence, the importance of features may vary with different accounting and tax systems. A cross-country comparative study is therefore recommended for future studies. The ensemble approaches used in this study were homogeneous, and similar base learners were employed in the ensembles. Using alternative heterogeneous approaches to ensemble learning combining the outcomes of different machine learning methods is another potential line of future research. Finally, fuzzy rule-based systems could improve the interpretability of tax default prediction.

## ACKNOWLEDGMENT

The authors thank the organizations.

## REFERENCES

[1] *Paying Taxes 2018*, World Bank Group, PwC, Washington, DC, USA, 2018, vol. 112.
[2] X. Hu, H. Huang, Z. Pan, and J. Shi, "Information asymmetry and credit rating: A quasi-natural experiment from China," *J. Banking Finance*, vol. 106, pp. 132–152, Sep. 2019.
[3] Y. Jiang and S. Jones, "Corporate distress prediction in China: A machine learning approach," *Accounting Finance*, vol. 58, no. 4, pp. 1063–1109, Dec. 2018.
[4] J. Vanhoeyveld, D. Martens, and B. Peeters, "Value-added tax fraud detection with scalable anomaly detection techniques," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105895.
[5] P. Hajek and K. Michalak, "Feature selection in corporate credit rating prediction," *Knowl.-Based Syst.*, vol. 51, pp. 72–84, Oct. 2013.
[6] W.-C. Lin, Y.-H. Lu, and C.-F. Tsai, "Feature selection in single and ensemble learning-based bankruptcy prediction models," *Expert Syst.*, vol. 36, no. 1, Feb. 2019, Art. no. e12335.
[7] H. Höglund, "Tax payment default prediction using genetic algorithm-based variable selection," *Expert Syst. Appl.*, vol. 88, pp. 368–375, Dec. 2017.
[8] J. Ruan, Z. Yan, B. Dong, Q. Zheng, and B. Qian, "Identifying suspicious groups of affiliated-transaction-based tax evasion in big data," *Inf. Sci.*, vol. 477, pp. 508–532, Mar. 2019.
[9] W. Didimo, L. Grilli, G. Liotta, L. Menconi, F. Montecchiani, and D. Pagliuca, "Combining network visualization and data mining for tax risk assessment," *IEEE Access*, vol. 8, pp. 16073–16086, 2020.
[10] D. Marghescu, M. Kallio, and B. Back, "Using financial ratios to select companies for tax auditing: A preliminary study," in *Organizational, Business, and Technological Aspects of the Knowledge Society*. Berlin, Germany: Springer, 2010, pp. 393–398.
[11] M. Moscatelli, F. Parlapiano, S. Narizzano, and G. Viggiano, "Corporate default forecasting with machine learning," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113567.
[12] K. Coussement, S. Lessmann, and G. Verstraeten, "A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry," *Decis. Support Syst.*, vol. 95, pp. 27–36, Mar. 2017.
[13] M. Niemann, J. H. Schmidt, and M. Neukirchen, "Improving performance of corporate rating prediction models by reducing financial ratio heterogeneity," *J. Banking Finance*, vol. 32, no. 3, pp. 434–446, Mar. 2008.
[14] A. A. Ding, S. Tian, Y. Yu, and H. Guo, "A class of discrete transformation survival models with application to default probability prediction," *J. Amer. Stat. Assoc.*, vol. 107, no. 499, pp. 990–1003, Sep. 2012.
[15] J. Q. Dong and C.-H. Yang, "Business value of big data analytics: A systems-theoretic approach and empirical test," *Inf. Manage.*, vol. 57, no. 1, Jan. 2020, Art. no. 103124.
[16] J. Beutel, S. List, and G. von Schweinitz, "Does machine learning help us predict banking crises?" *J. Financial Stability*, vol. 45, Dec. 2019, Art. no. 100693.
[17] P. Andrikopoulos and A. Khorasgani, "Predicting unlisted SMEs' default: Incorporating market information on accounting-based models for improved accuracy," *Brit. Accounting Rev.*, vol. 50, no. 5, pp. 559–573, Sep. 2018.
[18] Y.-P. Huang and M.-F. Yen, "A new perspective of performance comparison among machine learning algorithms for financial distress prediction," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105663.
[19] J. Xiao, X. Zhou, Y. Zhong, L. Xie, X. Gu, and D. Liu, "Cost-sensitive semi-supervised selective ensemble model for customer credit scoring," *Knowl.-Based Syst.*, vol. 189, Feb. 2020, Art. no. 105118.
[20] Y. Song, Y. Wang, X. Ye, D. Wang, Y. Yin, and Y. Wang, "Multi-view ensemble learning based on distance-to-model and adaptive clustering for imbalanced credit risk assessment in P2P lending," *Inf. Sci.*, vol. 525, pp. 182–204, Jul. 2020.
[21] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar, "Customer churn prediction in telecommunication industry using data certainty," *J. Bus. Res.*, vol. 94, pp. 290–301, Jan. 2019.

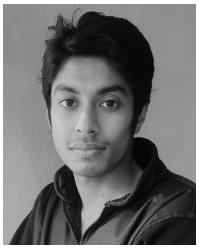
[22] M. Al-Mashraie, S. H. Chung, and H. W. Jeon, "Customer switching behavior analysis in the telecommunication industry via push-pull-mooring framework: A machine learning approach," *Comput. Ind. Eng.*, vol. 144, Jun. 2020, Art. no. 106476.
[23] S. Maldonado, J. López, and C. Vairetti, "Profit-based churn prediction based on minimax probability machines," *Eur. J. Oper. Res.*, vol. 284, no. 1, pp. 273–284, Jul. 2020.
[24] P. Hajek and R. Henriques, "Mining corporate annual reports for intelligent detection of financial statement fraud—A comparative study of machine learning methods," *Knowl.-Based Syst.*, vol. 128, pp. 139–152, Jul. 2017.
[25] Y.-J. Chen, W.-C. Liou, Y.-M. Chen, and J.-H. Wu, "Fraud detection for financial statements of business groups," *Int. J. Accounting Inf. Syst.*, vol. 32, pp. 1–23, Mar. 2019.
[26] F. Yu, Z. Qin, and X.-L. Jia, "Data mining application issues in fraudulent tax declaration detection," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 4, 2003, pp. 2202–2206.
[27] A. Gebauer, C. W. Nam, and R. Parsche, "Can reform models of value added taxation stop the VAT evasion and revenue shortfalls in the EU?" *J. Econ. Policy Reform*, vol. 10, no. 1, pp. 1–13, Mar. 2007.
[28] M. Gupta and V. Nagadevara, "Audit selection strategy for improving tax compliance: Application of data mining techniques," in *Proc. Found. Risk-Based Audits 11th Int. Conf. E-Governance*, Hyderabad, India, Dec. 2007, pp. 28–30.
[29] R.-S. Wu, C. S. Ou, H.-Y. Lin, S.-I. Chang, and D. C. Yen, "Using data mining technique to enhance tax evasion detection performance," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 8769–8777, Aug. 2012.
[30] E. Rahimikia, S. Mohammadi, T. Rahmani, and M. Ghazanfari, "Detecting corporate tax evasion using a hybrid intelligent system: A case study of iran," *Int. J. Accounting Inf. Syst.*, vol. 25, pp. 1–17, May 2017.
[31] F. Zhang, I. Keivanloo, and Y. Zou, "Data transformation in cross-project defect prediction," *Empirical Softw. Eng.*, vol. 22, no. 6, pp. 3186–3218, Dec. 2017.
[32] A. Amin, B. Shah, A. M. Khattak, F. J. Lopes Moreira, G. Ali, A. Rocha, and S. Anwar, "Cross-company customer churn prediction in telecommunication: A comparison of data transformation methods," *Int. J. Inf. Manage.*, vol. 46, pp. 304–319, Jun. 2019.
[33] L. Al Shalabi and Z. Shaaban, "Normalization as a preprocessing engine for data mining and the approach of preference matrix," in *Proc. Int. Conf. Dependability Comput. Syst.*, May 2006, pp. 207–214.
[34] C.-L. Huang and J.-F. Dun, "A distributed PSO–SVM hybrid system with feature selection and parameter optimization," *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1381–1391, 2008.
[35] C.-L. Huang and C.-J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Syst. Appl.*, vol. 31, no. 2, pp. 231–240, 2006.
[36] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 105524.
[37] T. Fukushima, Y. Kamei, S. McIntosh, K. Yamashita, and N. Ubayashi, "An empirical study of just-in-time defect prediction using cross-project models," in *Proc. 11th Work. Conf. Mining Softw. Repositories*, 2014, pp. 172–181.
[38] M. Danubianu, S. G. Pentiuc, and D. M. Danubianu, "Data dimensionality reduction for data mining: A combined filter-wrapper framework," *Int. J. Comput. Commun. Control*, vol. 7, no. 5, pp. 824–831, 2012.
[39] P. Pławiak, M. Abdar, J. Pławiak, V. Makarenkov, and U. R. Acharya, "DGHNL: A new deep genetic hierarchical network of learners for prediction of credit scoring," *Inf. Sci.*, vol. 516, pp. 401–418, Apr. 2020.
[40] A. Kim and S.-B. Cho, "An ensemble semi-supervised learning method for predicting defaults in social lending," *Eng. Appl. Artif. Intell.*, vol. 81, pp. 193–199, May 2019.
[41] M. Pompella and A. Dicanio, "Ratings based inference and credit risk: Detecting likely-to-fail banks with the PC-mahalanobis method," *Econ. Model.*, vol. 67, pp. 34–44, Dec. 2017.
[42] Z. Hua, B. Zhou, and Y. Zhou, "Sine-transform-based chaotic system with FPGA implementation," *IEEE Trans. Ind. Electron.*, vol. 65, no. 3, pp. 2557–2566, Mar. 2018.
[43] D. E. Goin, K. E. Rudolph, and J. Ahern, "Predictors of firearm violence in urban communities: A machine-learning approach," *Health Place*, vol. 51, pp. 61–67, May 2018.
[44] H. Son, C. Hyun, D. Phan, and H. J. Hwang, "Data analytic approach for bankruptcy prediction," *Expert Syst. Appl.*, vol. 138, Dec. 2019, Art. no. 112816.
[45] A. Petropoulos, V. Siakoulis, E. Stavroulakis, and N. E. Vlachogiannakis, "Predicting bank insolvencies using machine learning techniques," *Int. J. Forecasting*, vol. 36, no. 3, pp. 1092–1113, Jul. 2020.
[46] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.
[47] Z. Islam and H. Giggins, "Knowledge discovery through SysFor: A systematically developed forest of multiple decision trees," in *Proc. 9th Austral. Data Mining Conf.*, vol. 121, 2011, pp. 195–204.
[48] M. N. Adnan and M. Z. Islam, "Forest PA : Constructing a decision forest by penalizing attributes used in previous trees," *Expert Syst. Appl.*, vol. 89, pp. 389–403, Dec. 2017.
[49] G. Martínez-Muñoz and A. Suárez, "Out-of-bag estimation of the optimal sample size in bagging," *Pattern Recognit.*, vol. 43, no. 1, pp. 143–152, Jan. 2010.
[50] F. E. Moula, C. Guotai, and M. Z. Abedin, "Credit default prediction modeling: An application of support vector machine," *Risk Manage.*, vol. 19, no. 2, pp. 158–187, May 2017.
[51] K. W. De Bock, K. Coussement, and S. Lessmann, "Cost-sensitive business failure prediction when misclassification costs are uncertain: A heterogeneous ensemble selection approach," *Eur. J. Oper. Res.*, vol. 285, no. 2, pp. 612–630, Sep. 2020.
[52] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, May 2010.
[53] C. Serrano-Cinca, B. Gutiérrez-Nieto, and M. Bernate-Valbuena, "The use of accounting anomalies indicators to predict business failure," *Eur. Manage. J.*, vol. 37, no. 3, pp. 353–375, Jun. 2019.
[54] Z. Jing and Y. Fang, "Predicting US bank failures: A comparison of logit and data mining models," *J. Forecast.*, vol. 37, no. 2, pp. 235–256, 2018.
[55] M. Z. Abedin, C. Guotai, F.-E. Moula, A. S. Azad, and M. S. U. Khan, "Topological applications of multilayer perceptrons and support vector machines in financial decision support systems," *Int. J. Financ. Econ.*, vol. 24, no. 1, pp. 474–507, 2019.
[56] Y. Zhu, L. Zhou, C. Xie, G.-J. Wang, and T. V. Nguyen, "Forecasting SMEs' credit risk in supply chain finance with an enhanced hybrid ensemble machine learning approach," *Int. J. Prod. Econ.*, vol. 211, pp. 22–33, May 2019.
[57] S. Jones, "Corporate bankruptcy prediction: A high dimensional analysis," *Rev. Accounting Stud.*, vol. 22, no. 3, pp. 1366–1422, 2017.
[58] M. Mercadier and J.-P. Lardy, "Credit spread approximation and improvement using random forest regression," *Eur. J. Oper. Res.*, vol. 277, no. 1, pp. 351–365, 2019.

**MOHAMMAD ZOYNUL ABEDIN** received the B.B.A. and M.B.A. degrees in finance from the University of Chittagong, Chittagong, Bangladesh, and the D.Phil. degree in investment theory from the Dalian University of Technology, Dalian, China. From October 2018 to October 2020, he held a postdoctoral position with the Dalian Maritime University, China. He is currently an Associate Professor with the Department of Finance and Banking, Hajee Mohammad Danesh Science and Technology University, Dinajpur, Bangladesh. His research interests include business analytics and computational finance.

**GUOTAI CHI** is currently a Professor of finance and a Ph.D. Adviser with the Dalian University of Technology, Dalian, China. His research interests include financial risk management and asset—liability management.

**MOHAMMED MOHI UDDIN** received the Ph.D. degree in accounting from Aston University. He is currently an Assistant Professor of accountancy with the College of Business and Management. His research interests include accounting, accountability, and performance management in nonprofit/non-governmental organizations (NGOs).

**MD. SHAHRIARE SATU** received the B.Sc. and M.Sc. degrees in information technology from Janagirnagar University, in 2015 and 2017, respectively. He is currently working as a Lecturer with the Department of Management Information Systems, Noakhali Science and Technology University, Noakhali, Bangladesh. From 2016 to 2018, he worked as a Lecturer with the Department of Computer Science and Engineering, Gono Bishwabidyalay, Bangladesh. His research interests include machine learning, health informatics, business analytics, deep learning, and big data analytics.



**MD. IMRAN KHAN** received the B.Sc. degree in computer science and engineering from Gono Bishwabidyalay, in 2019. His research interests include machine learning, business analytics, deep learning, and health informatics.

**PETR HAJEK** was born in Duchcov, Czech Republic, in 1980. He received the B.S. and M.S. degrees in economic policy and the Ph.D. degree in system engineering and informatics from the University of Pardubice, Pardubice, Czech Republic, in 2003 and 2006, respectively.

From 2006 to 2012, he was a Senior Lecturer with the Institute of System Engineering and Informatics. Since 2012, he has been an Associate Professor with the Science and Research Centre, Faculty of Economics and Administration, University of Pardubice. He is the author of three books and more than 90 articles. His research interests include soft computing, machine learning, and economic modeling. He is an Associate Editor of four journals. He was a recipient of the Rector Award for Scientific Excellence in 2018 and 2019, respectively, and six Best Paper Awards at international scientific conferences.