informatics

MDPI

Article

On Ensemble SSL Algorithms for Credit Scoring Problem

Ioannis E. Livieris [1,*], Niki Kiriakidou [2], Andreas Kanavos [1], Vassilis Tampakas [1] and Panagiotis Pintelas [3]

[1] Computer & Informatics Engineering Department, Technological Educational Institute of Western Greece, GR 263-34 Antirion, Greece; kanavos@ceid.upatras.gr (A.K.); vtampakas@teimes.gr (V.T.)
[2] Department of Statistics and Insurance Science, University of Piraeus, GR 185-34 Piraeus, Greece; kiriakidou@unipi.gr
[3] Department of Mathematics, University of Patras, Patras, GR 265-00, Greece; ppintelas@gmail.com
* Correspondence: livieris@teiwest.gr

Received: 17 September 2018; Accepted: 26 October 2018; Published: 28 October 2018

Abstract: Credit scoring is generally recognized as one of the most significant operational research techniques used in banking and finance, aiming to identify whether a credit consumer belongs to either a legitimate or a suspicious customer group. With the vigorous development of the Internet and the widespread adoption of electronic records, banks and financial institutions have accumulated large repositories of labeled and mostly unlabeled data. Semi-supervised learning constitutes an appropriate machine- learning methodology for extracting useful knowledge from both labeled and unlabeled data. In this work, we evaluate the performance of two ensemble semi-supervised learning algorithms for the credit scoring problem. Our numerical experiments indicate that the proposed algorithms outperform their component semi-supervised learning algorithms, illustrating that reliable and robust prediction models could be developed by the adaptation of ensemble techniques in the semi-supervised learning framework.

Keywords: semi-supervised learning; self-labeled methods; ensemble learning; credit scoring; classification

1. Introduction

In today's competitive economy, *credit scoring* constitutes one of the most significant and successful operational research techniques used in banking and finance. It was developed by Fair and Isaac in the early 1960s and corresponds to the procedure of estimating the risk related to credit products which is evaluated using applicants' credentials (such as annual income, job status, residential status, etc.) and historical data [1,2]. In simple terms, credit scoring produces a score which can be used to classify customers into two separate groups: the "credit-worthy" (likely to repay the credit loan), and the "non credit-worthy" (rejected due to its high probability of defaulting).

The *global financial crisis of 2008* (which resulted in the collapse of many venerable names in the industry) caused a ripple effect throughout the economy and demonstrated the potential large losses when a credit applicant defaults on a loan [3,4]. Therefore, the credit scoring systems are of great interest to banks and financial institutions, not only because they must measure credit risk, but also because any small improvement would produce great profits [5–7]. For this task, many researchers in the past have developed credit- scoring models by exploiting the knowledge acquired from individual and company records of past borrowing and repaying actions gathered by the banks and financial institutions [8–14]. In the field of credit scoring, imbalanced datasets frequently occur as the number of non-worthy applicants is usually much smaller than the number of worthy. In order to address this

difficulty, ensemble learning methods have been proposed as a new direction for obtaining a better composite global model with more accurate and reliable decisions than can be obtained from using a single model [15]. The basic idea of ensemble learning is the combination of a set of diverse prediction models for developing a prediction model with improved classification accuracy. Nevertheless, the vigorous development of the Internet, the emergence of vast collections and the widespread adoption of electronic records have led to the development of large repositories of labeled and mostly unlabeled data. Most conventional credit-scoring models are based on individual supervised classifiers or a simple combination of these classifiers which exploit only labeled data, ignoring the knowledge hidden in the unlabeled data.

Semi-Supervised Learning (SSL) algorithms constitute a hybrid model which comprises characteristics of both supervised and unsupervised learning algorithms. More specifically, these algorithms efficiently exploit the hidden knowledge in the unlabeled data with the explicit classification knowledge from the labeled data. To this end, they are generally considered as the appropriate machine learning methodology to build powerful classifiers by extracting information from both labeled and unlabeled data [16]. Self-labeled algorithms constitute the most popular and frequently used class of SSL algorithms, thus have been efficiently applied in several real-world problems [17–24]. These algorithms wrap around a supervised prediction base learner and exploit the unlabeled data via a self-learning philosophy. Recently, Triguero et al. [25] presented an in-depth taxonomy focusing on demonstrating their simplicity of implementation and their wrapper-based methodology [16,26].

In this work, we examine and evaluate the performance of two ensemble-based self-labeled algorithms for the credit risk scoring problem. The proposed algorithms combine/fuse the predictions of three of the most productive and frequently used self-labeled algorithms, using different methodologies. Our experimental results demonstrate the classification accuracy of the presented algorithms on three credit scoring datasets.

The remainder of this paper is organized as follows: Section 2 presents a survey of recent studies concerning the application of data mining in credit scoring problem. Section 3 presents a brief description of self-labeled methods and the proposed ensemble-based SSL algorithms. Section 4 presents a series of experiments to evaluate the accuracy of the proposed algorithms for the credit scoring problem. Finally, Section 5 sketches our concluding remarks and our future work.

2. Related Work

During the last decades, the developments and advances of machine learning systems in credit decision making have gained popularity, addressing many issues in banking and finance. Louzada et al. [27] presented an extensive review, discussing the chronicles of recent credit scoring financial analysis and developments and analyze the outcomes produced by a machine learning approach. Additionally, they described in detail the most accurate prediction models used for gaining significant insights on credit scoring problem and conducted a variety of experiments, using three real-world datasets (Australian credit scoring, Japanese credit scoring and German credit scoring). A number of rewarding studies have been carried out in recent years; some useful outcomes of them are briefly presented below.

Kennedy et al. [28] evaluated the suitability of semi-supervised one-class classification algorithms against supervised two-class classification algorithms on low-default portfolio problem. Nine banking datasets were used and class imbalance is artificially created by removing 10%of the defaulting observations from the training set after each run. Additionally, they also investigated the suitability of oversampling, which constitutes a common approach to dealing with low-default portfolios. Their experimental results demonstrated that semi-supervised techniques should not be expected to outperform the supervised two-class classification techniques and they should be used only in the near or complete absence of defaulters. Moreover, although oversampling improved the performance of some two-class classifiers, it does not lead to an overall improvement of the best performing classifiers.

Alaraj and Abbod [29] introduced a model based on the combination of hybrid and ensemble methods for credit scoring. Firstly, they combined filtering and feature selection methods to develop an effective pre-processor for machine learning models. In addition, they proposed a new classifier combination rule based on the consensus approach of different classification algorithms, during the ensemble modeling phase. Their experimental analysis on seven real-world credit datasets illustrated that the proposed model exhibited better predictive performance than the individual classifiers.

Abellán and Castellano [30] performed a comparative study on several base classifiers used in different ensemble schemes for credit scoring tasks. Additionally, they evaluated the performance of Credal Decision Tree (CDT) which uses imprecise probabilities and uncertainty measures to build a decision tree. Via an experimental study, they concluded that all the investigated ensemble schemes present better performance when they use CDT model as a base learner on credit scoring problems.

In more recent works, Tripathi et al. [31] proposed a hybrid credit scoring model based on dimensionality reduction by Neighborhood Rough Set algorithm for feature selection and layered ensemble classification with weighted voting approach to enhance the classification performance. They have proposed a novel classifier ranking algorithm as an underlying model for representing ranks of the classifiers based on classifier accuracy. The experimental results revealed the efficacy and robustness of the proposed method in two benchmarked credit scoring datasets.

Zhang et al. [32] proposed a new predictive model which is based on a novel technique for selecting classifiers using a genetic algorithm, considering both the accuracy and diversity of the ensemble. They conducted a variety of experiments, using three real-world datasets (Australian credit scoring, Japanese credit scoring and German credit scoring) to explore the effectiveness of their proposed model. Based on their numerical experiments the authors concluded that their proposed ensemble method outperforms classical classifiers in terms of prediction accuracy.

J. Levatić et al. [33] proposed method for semi-supervised learning of classification trees. The trees can be trained with nominal and/or numeric descriptive attributes on binary and multi-class classification datasets. Additionally, they performed an extensive empirical evaluation of their framework using an ensemble of decision trees as base learners obtaining some interesting results. Along this line, they extended their work, presenting some ensemble-based algorithms for multi-target regression problems [33,34].

3. A Review of Semi-Supervised Self-Labeled Classification Methods

The basic aim of semi-supervised self-labeled methods is the enrichment of the initial labeled data through labeling of unlabeled data via a self-learning process based on supervised prediction models. In the literature, several self-labeled methods have been proposed each one based on a different philosophy on exploiting knowledge from unlabeled data. In the sequel, we briefly describe the popular and frequently used semi-supervised self-labeled methods.

3.1. Self-Labeled Methods

Self-training [35] is a semi-supervised learning algorithm characterized by its simplicity and its good classification performance. In self-training, a supervised base learner is trained using the labeled data and iteratively augmented its training set gradually with the most confident predictions on unlabeled examples and re-trained. Nevertheless, this methodology can lead to erroneous predictions if noisy examples are classified as the most confident examples and incorporated into the labeled training set.

To address this difficulty, Li and Zhou [36] proposed *SETRED* (Self-trained with editing), which is-based on the adaptation of data editing in the self-training framework. This method constructs a neighboring graph in D-dimensional feature space and at each iteration a hypothesis test filters the candidate unlabeled instances. Finally, the unlabeled instances which have successfully passed the test, are added in the training set.

The standard *Co-training* [37] is a multi-view learning method, based on the assumption that the feature space can be split into two different views which are conditionally independent. Under the assumption about the existence of sufficient and redundant views, Co-training trains separately a base learner in each specific view. Then, iteratively each base learner teaches the other with its most confidently predicted unlabeled examples, hence augmenting their training sets. However, in most real-case scenarios this assumption is a luxury hardly met [21].

Zhou and Goldman [38] have adopted the idea of incorporating majority voting in the semi-supervised framework and proposed *Democratic co-learning* (Demo-Co) algorithm. This algorithm although it belongs to the multi-view algorithm, it operates in a different manner. Instead of demanding for multiple views of the corresponding data, it uses multiple algorithms for producing the necessary information and endorses a voted majority process for the final decision. Based on the previous work, Li and Zhou [39] proposed *Co-Forest*, in which a number of Random trees are trained on bootstrap data from the dataset and the output is defined as the combined individual prediction of each tree, via a simple majority voting. The basic idea behind this algorithm is that during the training process, the algorithm assigns a few unlabeled instances to each Random tree. Notice that, the efficiency of Co-Forest is mainly based on the use of Random trees, although the number of the available labeled instances is significantly reduced.

Another approach which is also based on an ensemble methodology is the *Tri-training* algorithm [40] which constitutes an improved single-view extension of the Co-training algorithm. This algorithm uses a labeled dataset to initially train three base learners which are used to make predictions on the instances of the unlabeled dataset. Then, if two base learners agree on labeling an example, then this is labeled for the third base learner too. The "majority teach minority strategy" has the advantage of avoiding the explicitly measuring the confidence of labeling, since such measuring is sometimes a quite complicated and time-consuming process; therefore, the training process is efficient [21].

3.2. Ensemble Self-Labeled Methods

The development of an ensemble of classifiers consists of two main steps: selection and combination. The selection of the component classifiers is considered essential for the efficiency of the ensemble while the key point for its efficacy is based on the diversity and the accuracy of the component classifiers [41]. Furthermore, the combination of the individual predictions of the classifier takes place through several techniques and methodologies with different philosophy and classification performance [15,42].

By taking these into consideration, Kostopoulos et al. [43] and Livieris et al. [18,20] proposed two ensemble SSL algorithms, called CST-Voting and EnSSL, respectively. Both ensemble SSL algorithms exploit the individual predictions of three self-labeled algorithms i.e., Self-training, Co-training and Tri-training using a difference combination technique and mechanism.

CST-Voting [20,43] is based on the idea of combining the predictions of the self-labeled algorithms which constitute the ensemble using a simple majority voting methodology. Initially, the classical self-labeled algorithms are trained using the same labeled L and unlabeled U sets. Next, the final hypothesis on an instance from the testing set combines the individual predictions of the self-labeled algorithms, using a simple majority voting. Hence, the output of the ensemble is the one made by more than half of them. A high-level description of the proposed CST-Voting is presented in Algorithm 1.

EnSSL [18,44] combines the individual prediction of the same self-labeled algorithms using a maximum probability-based voting scheme. More specifically, the self-labeled algorithm which exhibits the most confident prediction over an unlabeled example of the test set is selected. In case the confidence of the prediction of the selected classifier meets a predefined threshold then the classifier labels the example otherwise the prediction is not considered reliable enough. It is worth mentioning that the way in which the confidence predictions are measured depends on the type of used base learner (see [45–47] and the references there in). In this case, the output of the ensemble is defined

as the combined predictions of three self-labeled learning algorithms via a simple majority voting. A high-level description of the En-SSL algorithm is presented in Algorithm 2.

Algorithm 1: CST-Voting

Input: L − Set of labeled training instances.
U − Set of unlabeled training instances.
Output: The labels of instances in the testing set.

/* Phase I: Training */
1: Self-training(L, U)
2: Co-training(L, U)
3: Tri-training(L, U)

/* Phase II: Voting-Fusion */
4: **for each** $x \in T$ **do**
5: Apply Self-training, Co-training, and Tri-training on x.
6: Use majority vote to predict the label y^* of x.
7: **end for**

Algorithm 2: EnSSL

Input: L − Set of labeled training instances.
U − Set of unlabeled training instances.
ThresLev − Threshold level.
Output: The labels of instances in the testing set.

/* Phase I: Training */
1: Self-training(L, U)
2: Co-training(L, U)
3: Tri-training(L, U)

/* Phase II: Voting-Fusion */
4: **for each** $x \in T$ **do**
5: Apply Self-training, Co-training, and Tri-training on x.
6: Find the classifier C^* with the highest confidence prediction on x.
7: **if** (Confidence of $C^* \geq$ *ThresLev*) **then**
8: C^* predicts the label y^* of x.
9: **else**
10: Use majority vote to predict the label y^* of x.
11: **end if**
12: **end for**

4. Experimental Methodology

In this section, we conducted a series of experiments to evaluate the performance of CST-Voting and EnSSL algorithms against the most popular and frequently used self-labeled algorithms.

The implementation code was written in JAVA, making use of the WEKA 3.9 Machine Learning Toolkit [46]. To study the influence of the amount of labeled data, three different ratios (R) of the training data were used, i.e., 10%, 20% and 30% and all self-labeled algorithms were evaluated using the stratified 10-fold cross-validation.

The experiments in our study took place in two distinct phases. In the first phase, we evaluated the classification performance of CST-Voting and EnSSL against the most popular SSL algorithms namely Self-training, Co-training, and Tri-training; while in the second phase, we compared their performance against some state-of-the-art self-labeled algorithms, namely SETRED, Demo-Co and Co-Forest. Table 1 reports the configuration parameters of all evaluated self-labeled algorithms while

all base learners were used with their default parameter settings included in the WEKA 3.9 Machine Learning Toolkit (University of Waikato, Hamilton, New Zealand) for minimizing the effect of any expert bias.

Table 1. Parameter specification for all SSL methods.

SSL algorithm	Parameters
Self-training	MaxIter = 40. $c = 0.95$.
Co-training	MaxIter = 40. Initial unlabeled pool = 75.
Tri-training	No parameters specified.
Democratic-Co	Classifiers = *k*NN, C4.5, NB.
SETRED	MaxIter = 40. Threshold = 0.1.
Co-Forest	Number of Random Forest classifiers = 6. Threshold = 0.75.

All algorithms were evaluated using three different benchmark datasets: Australian credit, Japanest credit and German credit which are publicly available in UCI Machine Learning Repository [48], concerning approved or rejected credit card applications. The first has 690 cases, with 14 explanatory variables (6 continuous and 8 categorical); the second one has 653 instances, with 14 explanatory variables (3 continuous, 3 integer and 9 categorical); while the third one has 1000 instances, with 20 explanatory variables (7 continuous and 13 categorical). With regards to the cardinality of each class, in Australian dataset there is a small imbalance of rejected and accepted instances, namely 383 and 307, respectively. In Japanese dataset there is a small imbalance of rejected and accepted instances, namely 357 and 296, respectively; while in German dataset a sharper imbalance is observed, with 300 negative decisions and 700 positive.

The performance of the classification algorithms was evaluated using the following four performance metrics: Sensitivity (*Sen*), Specificity (*Spe*), F_1 and Accuracy (*Acc*) which are respectively defined by

$$Sen = \frac{T_P}{T_P + F_N}, \qquad Spe = \frac{T_N}{T_N + F_P}, \qquad F_1 = \frac{2T_P}{2T_P + F_N + F_P} \qquad Acc = \frac{T_P + T_N}{T_P + T_N + F_P + F_N},$$

where T_P stands for the number of instances which have been correctly classified as positive, T_N stands for the number of instances which have been correctly classified as negative, F_P (type I error) stands for the number of instances which have been wrongly classified as positive, F_N (type II error) stands for the number of instances which have been wrongly classified as negative.

It is worth mentioning that Sensitivity of classification is the proportion of actual positives which are predicted as positive; Specificity represents the proportion of actual negatives which are predicted as negative, F_1 consists of a harmonic mean of precision and recall while Accuracy is the ratio of correct predictions of a classification model.

4.1. First Phase of Experiments

In the sequel, we focus our interest on the experimental analysis for evaluating the classification performance of CST-Voting and EnSSL algorithms against its component self-labeled methods, i.e., Self-training, Co-training, and Tri-training. All SSL algorithms were evaluated by deploying as base learners the Naive Bayes (NB) [49], the Sequential Minimum Optimization (SMO) [50], the Multilayer Perceptron (MLP) [51] and the *k*NN algorithm [52]. These algorithms probably constitute the most effective and popular machine learning algorithms for classification problems [53]. Moreover, similar to Blum and Mitchell [37], a limit to the number of iterations of all self-labeled algorithms is established. This strategy has also been adopted by many researchers [18–22,25,47,54,55].

Tables 2–4 present the performance of each compared SSL algorithms for Australian, Japanese, German datasets, respectively, relative to all performance metrics. Notice that the highest classification accuracy is highlighted in bold for each base learner. Firstly, it is worth mentioning that the ensemble SSL methods, CST-Voting and EnSSL, exhibited the best performance, regarding all datasets and improved their performance metric as the labeled ratio increased. In more detail:

- CST-Voting exhibited the best performance in 10, 8 and 8 cases for Australian dataset, Japanese dataset and German dataset, respectively, while EnSSL exhibited the highest accuracy in 6, 8 and 8 cases in the same situations.
- Depending upon the base classifier, CST-Voting is the most effective method using NB or SMO as base learner, while EnSSL reported the highest performance using MLP as base learner.

In machine learning, the statistical comparison of several evaluation algorithms over multiple datasets is fundamental and it is frequently performed by means of a statistical test [20,21,56]. Since our motivation stems from the fact that we are interested in evaluating the rejection of the hypothesis that all the algorithms perform equally well for a given level based on their classification accuracy and highlighting the existence of significant differences between our proposed algorithm and the classical self-labeled algorithms, we used the non-parametric Friedman Aligned Ranking (FAR) [57] test. Moreover, the Finner test [58] is applied as a post-hoc procedure to find out which algorithms present significant differences.

Table 5 presents the information of the statistical analysis performed by nonparametric multiple comparison procedures for Self-training, Co-training, Tri-Training, CST-Voting and EnSSL algorithms. Notice that the control algorithm for the post-hoc test is determined by the best (e.g., lowest) ranking obtained in each FAR test. Moreover, the adjusted p-value with Finner's test (p_F) was presented based on the corresponding control algorithm at the $\alpha = 0.05$ level of significance. The post-hoc test rejects the hypothesis of equality when the value of p_F is less than the value of a.

Clearly, CST-Voting and EnSSL demonstrate the best overall performance, as they outperform the rest self-labeled algorithms. This is because it reports the highest probability-based ranking by statistically presenting better results, relative to all used base learners. CST-Voting exhibited the best performance using NB and SMO as base learners while EnSSL presented the best performance using MLP and *k*NN as base learners. Furthermore, the FAR test but mostly the Finner post-hoc test revealed that CST-Voting and EnSSL perform equally well.

Table 2. Performance evaluation of Self-training, Co-training, Tri-training, CST-Voting and EnSSL on the Australian credit dataset.

Base	**Alg.**	**Ratio = 10%**				**Ratio = 20%**				**Ratio = 30%**				**Ratio = 40%**			
Learner		**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**
NB	Self	73.9%	88.3%	78.4%	81.9%	78.2%	91.4%	82.8%	85.5%	78.2%	90.3%	82.2%	84.9%	78.5%	90.6%	82.5%	85.2%
	Co	**78.2%**	83.6%	78.7%	81.2%	77.5%	**92.7%**	83.1%	**85.9%**	78.8%	91.4%	83.2%	85.8%	79.2%	91.4%	83.4%	85.9%
	Tri	61.2%	**91.6%**	71.3%	78.1%	76.2%	86.7%	79.1%	82.0%	**79.8%**	86.7%	81.3%	83.6%	80.1%	87.2%	81.7%	84.1%
	CST	75.6%	90.3%	**80.6%**	**83.8%**	78.2%	91.9%	**83.0%**	85.8%	79.2%	**91.6%**	**83.5%**	**86.1%**	79.8%	**92.2%**	**84.2%**	**86.7%**
	EnSSL	74.6%	90.6%	80.1%	83.5%	**78.5%**	90.6%	82.5%	85.2%	**79.8%**	90.9%	**83.5%**	85.9%	**80.5%**	91.4%	**84.2%**	86.5%
SMO	Self	88.9%	79.1%	82.7%	83.5%	85.7%	83.3%	83.0%	84.3%	88.9%	81.7%	84.0%	84.9%	88.9%	82.0%	84.1%	85.1%
	Co	**92.2%**	79.1%	84.5%	84.9%	**94.1%**	79.1%	**85.5%**	85.8%	**94.1%**	79.1%	85.5%	85.8%	**94.1%**	79.1%	85.5%	85.8%
	Tri	77.5%	**86.7%**	79.9%	82.6%	89.3%	83.0%	84.8%	85.8%	89.3%	80.9%	83.8%	84.6%	89.6%	81.2%	84.1%	84.9%
	CST	89.9%	84.9%	**86.1%**	**87.1%**	90.6%	80.4%	84.2%	84.9%	93.8%	82.0%	**86.7%**	**87.2%**	**94.1%**	82.2%	**87.0%**	**87.5%**
	EnSSL	88.9%	83.6%	84.9%	85.9%	89.3%	**83.3%**	85.0%	**85.9%**	88.9%	**84.3%**	85.3%	86.4%	90.9%	**84.9%**	86.6%	**87.5%**
MLP	Self	82.1%	87.7%	83.2%	85.2%	80.1%	88.0%	82.1%	84.5%	82.7%	86.9%	83.1%	85.1%	82.7%	87.2%	83.3%	85.2%
	Co	80.8%	87.7%	82.4%	84.6%	79.8%	**91.4%**	83.8%	86.2%	79.5%	**91.1%**	83.4%	85.9%	80.5%	**91.4%**	84.2%	86.5%
	Tri	71.3%	89.0%	77.1%	81.2%	83.1%	82.2%	81.0%	82.6%	**89.3%**	83.0%	84.8%	85.8%	**89.6%**	83.6%	85.3%	86.2%
	CST	82.4%	88.0%	83.5%	85.5%	82.4%	88.0%	83.5%	85.5%	85.0%	87.2%	84.6%	86.2%	87.9%	88.0%	86.7%	88.0%
	EnSSL	**82.7%**	**90.3%**	**84.9%**	**87.0%**	**85.0%**	88.0%	**85.0%**	**86.7%**	87.9%	87.5%	**86.4%**	**87.7%**	89.3%	88.8%	**87.8%**	**89.0%**
*k*NN	Self	73.9%	88.3%	78.4%	81.9%	73.3%	88.3%	78.0%	81.6%	73.3%	91.4%	79.6%	83.3%	73.6%	91.4%	79.9%	83.5%
	Co	**78.2%**	83.6%	78.7%	81.2%	**77.5%**	84.6%	78.8%	81.4%	**78.8%**	87.5%	81.1%	83.6%	79.2%	86.9%	81.0%	83.5%
	Tri	61.2%	**91.6%**	71.3%	78.1%	67.8%	**91.6%**	76.1%	81.0%	74.9%	89.3%	79.6%	82.9%	75.9%	90.1%	80.6%	83.8%
	CST	75.6%	90.3%	**80.6%**	**83.8%**	74.6%	90.9%	**80.2%**	**83.6%**	78.5%	**92.2%**	**83.4%**	**86.1%**	**79.8%**	**92.4%**	**84.3%**	**86.8%**
	EnSSL	74.6%	90.9%	80.2%	83.6%	73.9%	89.8%	79.2%	82.8%	78.5%	90.9%	82.7%	85.4%	79.2%	91.4%	83.4%	85.9%

Notice that the highest classification accuracy is highlighted in bold for each base learner.

Table 3. Performance evaluation of Self-training, Co-training, Tri-training, CST-Voting and EnSSL on the Japanese credit dataset.

Base	**Alg.**	**Ratio = 10%**				**Ratio = 20%**				**Ratio = 30%**				**Ratio = 40%**			
Learner		**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**
NB	Self	75.3%	88.2%	79.5%	82.4%	**79.1%**	90.2%	82.8%	85.1%	79.1%	90.8%	83.1%	85.5%	79.1%	91.0%	83.3%	85.6%
	Co	**83.1%**	86.8%	**83.5%**	85.1%	78.7%	90.8%	**82.9%**	**85.3%**	**79.7%**	91.6%	84.0%	86.2%	**80.1%**	91.6%	84.2%	86.4%
	Tri	74.3%	88.2%	78.9%	81.9%	73.6%	**91.6%**	80.1%	83.5%	73.0%	90.5%	79.1%	82.5%	73.6%	91.6%	80.1%	83.5%
	CST	79.4%	**90.8%**	83.3%	**85.6%**	78.0%	91.3%	82.8%	**85.3%**	79.1%	**92.2%**	83.9%	86.2%	79.7%	**92.4%**	84.4%	86.7%
	EnSSL	79.4%	89.1%	82.5%	84.7%	76.7%	**91.6%**	82.1%	84.8%	**79.7%**	**92.2%**	**84.3%**	**86.5%**	**80.1%**	**92.4%**	**84.6%**	**86.8%**
SMO	Self	92.2%	81.0%	85.7%	86.1%	91.9%	81.0%	85.5%	85.9%	92.9%	80.7%	85.9%	86.2%	92.6%	81.0%	85.9%	86.2%
	Co	**93.9%**	79.8%	86.1%	86.2%	**93.9%**	79.8%	86.1%	86.2%	**93.9%**	80.1%	86.2%	86.4%	**93.9%**	80.1%	86.2%	86.4%
	Tri	86.5%	**86.3%**	85.2%	86.4%	79.7%	**86.0%**	81.1%	83.2%	74.7%	84.0%	77.0%	79.8%	76.0%	85.7%	78.7%	81.3%
	CST	93.6%	80.7%	86.3%	86.5%	93.2%	80.1%	85.8%	86.1%	93.2%	**86.6%**	**89.0%**	**89.6%**	93.6%	**87.1%**	**89.5%**	**90.0%**
	EnSSL	93.2%	81.2%	**86.4%**	**86.7%**	93.2%	81.2%	**86.4%**	**86.7%**	93.2%	85.4%	88.5%	89.0%	93.6%	86.0%	88.9%	89.4%
MLP	Self	**84.1%**	87.4%	84.4%	85.9%	**86.1%**	85.7%	84.7%	85.9%	**86.1%**	87.7%	85.7%	87.0%	**86.5%**	88.0%	86.1%	87.3%
	Co	81.1%	88.8%	83.3%	85.3%	82.8%	89.9%	84.9%	86.7%	82.8%	89.6%	84.8%	86.5%	83.4%	89.6%	85.2%	86.8%
	Tri	69.3%	88.0%	75.4%	79.5%	65.2%	**91.6%**	74.4%	79.6%	65.2%	**93.3%**	75.2%	80.6%	65.5%	**93.8%**	75.8%	81.0%
	CST	80.7%	90.2%	83.9%	85.9%	84.1%	89.9%	85.7%	87.3%	84.1%	90.2%	85.9%	87.4%	84.5%	90.5%	86.2%	87.7%
	EnSSL	81.4%	**90.8%**	**84.6%**	**86.5%**	85.1%	90.8%	**86.7%**	**88.2%**	85.1%	92.2%	**87.5%**	**89.0%**	85.5%	92.7%	**88.0%**	**89.4%**
*k*NN	Self	75.7%	88.2%	79.7%	82.5%	76.4%	88.5%	80.3%	83.0%	75.3%	89.6%	80.2%	83.2%	75.7%	88.2%	79.7%	82.5%
	Co	**79.4%**	85.4%	80.6%	82.7%	**79.1%**	86.6%	81.0%	83.2%	**83.1%**	85.4%	82.8%	84.4%	**83.4%**	85.7%	83.2%	84.7%
	Tri	56.1%	88.2%	65.9%	73.7%	59.8%	**93.0%**	71.1%	77.9%	74.3%	**95.2%**	82.6%	85.8%	76.0%	**94.4%**	83.2%	86.1%
	CST	76.0%	**90.8%**	**81.2%**	**84.1%**	76.7%	90.2%	**81.4%**	**84.1%**	79.4%	92.2%	**84.1%**	**86.4%**	79.7%	92.4%	**84.4%**	**86.7%**
	EnSSL	75.0%	89.6%	80.0%	83.0%	75.7%	90.2%	80.7%	83.6%	78.4%	92.4%	83.6%	86.1%	79.1%	92.4%	84.0%	86.4%

Notice that the highest classification accuracy is highlighted in bold for each base learner.

Table 4. Performance evaluation of Self-training, Co-training, Tri-training, CST-Voting and EnSSL on the German credit dataset.

Base	**Alg.**	**Ratio = 10%**				**Ratio = 20%**				**Ratio = 30%**				**Ratio = 40%**			
Learner		**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**
NB	Self	80.7%	45.3%	79.1%	70.1%	84.6%	**48.7%**	81.9%	73.8%	84.6%	50.3%	82.2%	74.3%	84.7%	50.0%	82.2%	74.3%
	Co	80.0%	45.0%	78.6%	69.5%	85.7%	46.7%	82.2%	74.0%	86.4%	51.7%	83.4%	76.0%	86.6%	**51.7%**	83.5%	76.1%
	Tri	81.6%	45.7%	79.6%	70.8%	86.0%	46.0%	82.2%	74.0%	87.1%	51.0%	83.7%	76.3%	87.4%	50.7%	83.8%	76.4%
	CST	81.7%	**46.0%**	**79.8%**	**71.0%**	**86.4%**	47.7%	**82.8%**	**74.8%**	**87.9%**	51.7%	**84.2%**	**77.0%**	**88.0%**	**51.7%**	**84.3%**	**77.1%**
	EnSSL	**81.9%**	45.3%	79.7%	70.9%	86.1%	46.7%	82.4%	74.3%	87.4%	**52.0%**	84.1%	76.8%	87.6%	**51.7%**	84.1%	76.8%
SMO	Self	84.6%	44.7%	81.2%	72.6%	86.4%	45.0%	82.3%	74.0%	87.1%	46.0%	82.9%	74.8%	87.3%	46.7%	83.1%	75.1%
	Co	84.3%	45.0%	81.1%	72.5%	86.0%	**47.3%**	82.5%	74.4%	87.0%	**48.3%**	83.2%	75.4%	87.1%	48.3%	83.3%	75.5%
	Tri	84.4%	45.7%	81.3%	72.8%	86.7%	46.7%	82.8%	74.7%	**87.4%**	47.3%	83.3%	75.4%	87.6%	47.7%	83.4%	75.6%
	CST	**86.4%**	**46.0%**	**82.5%**	**74.3%**	**87.0%**	47.0%	**83.0%**	**75.0%**	**87.4%**	48.0%	**83.4%**	**75.6%**	**88.0%**	**49.0%**	**83.9%**	**76.3%**
	EnSSL	86.3%	**46.0%**	82.4%	74.2%	86.4%	46.7%	82.6%	74.5%	87.3%	47.7%	83.2%	75.4%	87.4%	48.3%	83.4%	75.7%
MLP	Self	84.6%	**47.0%**	81.6%	73.3%	86.4%	**47.3%**	82.7%	74.7%	87.1%	**48.3%**	83.3%	75.5%	87.3%	**48.3%**	83.4%	75.6%
	Co	85.4%	43.3%	81.5%	72.8%	86.0%	44.0%	81.9%	73.4%	87.4%	44.3%	82.8%	74.5%	87.9%	44.3%	83.0%	74.8%
	Tri	**87.4%**	45.0%	82.9%	74.7%	86.7%	44.0%	82.3%	73.9%	87.9%	45.0%	83.1%	75.0%	88.0%	46.0%	83.4%	75.4%
	CST	87.0%	46.0%	82.8%	74.7%	87.1%	45.0%	82.7%	74.5%	**88.3%**	47.0%	**83.7%**	**75.9%**	88.3%	47.3%	83.7%	76.0%
	EnSSL	**87.4%**	**47.0%**	**83.1%**	**75.2%**	**87.6%**	47.0%	**83.3%**	**75.4%**	88.1%	47.3%	**83.7%**	**75.9%**	**88.6%**	**48.3%**	**84.1%**	**76.5%**
*k*NN	Self	84.6%	40.0%	80.4%	71.2%	86.4%	42.3%	81.9%	73.2%	86.1%	43.3%	81.9%	73.3%	86.4%	43.7%	82.1%	73.6%
	Co	85.4%	40.7%	81.0%	72.0%	86.0%	41.7%	81.5%	72.7%	87.4%	43.7%	82.6%	74.3%	87.7%	43.7%	82.8%	74.5%
	Tri	**87.4%**	40.7%	82.1%	73.4%	86.7%	42.7%	82.1%	73.5%	87.9%	44.0%	82.9%	74.7%	88.0%	44.3%	83.1%	74.9%
	CST	85.0%	47.7%	82.0%	73.8%	87.0%	46.7%	82.9%	74.9%	86.4%	46.0%	82.5%	74.3%	86.7%	46.3%	82.7%	74.6%
	EnSSL	**87.4%**	**48.3%**	**83.4%**	**75.7%**	**87.3%**	**47.0%**	**83.1%**	**75.2%**	**88.1%**	**46.7%**	**83.5%**	**75.7%**	**88.1%**	**47.0%**	**83.6%**	**75.8%**

Notice that the highest classification accuracy is highlighted in bold for each base learner.

Table 5. Friedman Aligned Ranking (FAR) test and Finner post-hoc test for Self-training, Co-training, Tri-training, CST-Voting and EnSSL.

Algorithm	**FAR**	**Finner Post-Hoc Test**	
		p_F**-Value**	**Null Hypothesis**
CST-Voting	12.7917	-	-
EnSSL	19.8333	0.323326	accepted
Co-training	29.0833	0.029637	rejected
Self-training	42.3333	0.000068	rejected
Tri-training	48.4583	0.000002	rejected

(a) using NB as base learner

Algorithm	**FAR**	**Finner Post-Hoc Test**	
		p_F**-Value**	**Null Hypothesis**
CST-Voting	14.375	-	-
EnSSL	15.9583	0.824256	accepted
Co-training	32.75	0.013257	rejected
Tri-training	44.1667	0.000060	rejected
Self-training	45.25	0.000060	rejected

(b) using SMO as base learner

Algorithm	**FAR**	**Finner Post-Hoc Test**	
		p_F**-Value**	**Null Hypothesis**
EnSSL	9.9167	-	-
CST-Voting	22.2917	0.082620	accepted
Self-training	34.7083	0.000675	rejected
Co-training	36.6667	0.000351	rejected
Tri-training	48.9167	0.000000	rejected

(c) using MLP as base learner

Algorithm	**FAR**	**Finner Post-Hoc Test**	
		p_F**-Value**	**Null Hypothesis**
CST-Voting	14.5417	-	-
EnSSL	14.7083	0.98135	accepted
Co-training	38.7083	0.000933	rejected
Tri-training	41.8333	0.000312	rejected
Self-training	42.7083	0.000312	rejected

(d) using kNN as base learner

4.2. Second Phase of Experiments

Next, we evaluated the classification performance of the presented ensemble algorithms, CST-Voting and EnSSL, against some other state-of-the-art self-labeled algorithms such as SETRED, Co-Forest and Democratic-Co learning. Notice that CST-Voting and EnSSL uses SMO and MLP as base learners, respectively which exhibited the best performance, relative to all performance metrics.

Tables 6–8 report the performance of each tested self-labeled algorithm on Australian dataset, Japanese dataset and German credit dataset, respectively. As above mentioned, the accuracy measure of the best performing algorithm is highlighted in bold. Clearly, the presented ensemble SSL algorithms illustrate the best performance, independent of the used labeled ratio. Furthermore, it is worth noticing that EnSSL exhibits slightly better average performance than CST-Voting.

Table 6. Performance evaluation of SETRED, Co-Forest, Democratic-Co learning, CST-Voting and EnSSL on the Australian credit dataset.

Algorithm	**Ratio = 10%**				**Ratio = 20%**				**Ratio = 30%**				**Ratio = 40%**			
	Sen	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**
SETRED	87.9%	78.3%	81.8%	82.6%	87.6%	82.2%	83.5%	84.6%	91.2%	82.8%	85.8%	86.5%	91.5%	82.8%	85.9%	86.7%
Co-Forest	81.4%	87.5%	82.6%	84.8%	80.5%	**89.0%**	82.9%	85.2%	81.4%	**91.4%**	84.7%	87.0%	81.8%	**91.4%**	84.9%	87.1%
Demo-Co	82.7%	82.0%	80.6%	82.3%	83.1%	85.4%	82.5%	84.3%	84.0%	86.9%	83.9%	85.7%	84.0%	87.2%	84.0%	85.8%
CST	**89.9%**	84.9%	**86.1%**	**87.1%**	**90.6%**	80.4%	84.2%	84.9%	**93.8%**	82.0%	**86.7%**	87.2%	**94.1%**	82.2%	87.0%	87.5%
EnSSL	82.7%	**90.3%**	84.9%	87.0%	85.0%	88.0%	**85.0%**	**86.7%**	87.9%	87.5%	86.4%	**87.7%**	89.3%	88.8%	**87.8%**	**89.0%**

The accuracy measure of the best performing algorithm is highlighted in bold.

Table 7. Performance evaluation of SETRED, Co-Forest, Democratic-Co learning, CST-Voting and EnSSL on the Japanese credit dataset.

Algorithm	**Ratio = 10%**				**Ratio = 20%**				**Ratio = 30%**				**Ratio = 40%**			
	Sen	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**
SETRED	91.2%	81.2%	85.3%	85.8%	92.2%	81.2%	85.8%	86.2%	92.9%	81.5%	86.3%	86.7%	92.9%	81.8%	86.5%	86.8%
Co-Forest	84.5%	88.5%	85.2%	**86.7%**	85.1%	89.4%	86.0%	87.4%	85.1%	89.9%	86.3%	87.7%	85.1%	90.5%	86.6%	88.1%
Demo-Co	85.5%	84.6%	83.8%	85.0%	84.5%	85.7%	83.8%	85.1%	84.8%	85.7%	83.9%	85.3%	86.1%	86.0%	84.9%	86.1%
CST	**93.6%**	80.7%	**86.3%**	86.5%	**93.2%**	80.1%	85.8%	86.1%	**93.2%**	86.6%	**89.0%**	**89.6%**	**93.6%**	87.1%	**89.5%**	**90.0%**
EnSSL	81.4%	**90.8%**	84.6%	86.5%	85.1%	**90.8%**	**86.7%**	**88.2%**	85.1%	**92.2%**	87.5%	89.0%	85.5%	**92.7%**	88.0%	89.4%

The accuracy measure of the best performing algorithm is highlighted in bold.

Table 8. Performance evaluation of SETRED, Co-Forest, Democratic-Co learning, CST-Voting and EnSSL on the German credit dataset.

Algorithm	**Ratio = 10%**				**Ratio = 20%**				**Ratio = 30%**				**Ratio = 40%**			
	Sen	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**	**Sen**	**Spe**	**F_1**	**Acc**
SETRED	84.3%	44.7%	81.0%	72.4%	86.7%	45.0%	82.5%	74.2%	87.4%	46.7%	83.2%	75.2%	87.6%	47.0%	83.3%	75.4%
Co-Forest	85.7%	45.0%	81.9%	73.5%	87.1%	45.0%	82.7%	74.5%	87.3%	46.7%	83.1%	75.1%	87.4%	47.3%	83.3%	75.4%
Demo-Co	83.6%	43.7%	80.5%	71.6%	86.0%	45.3%	82.1%	73.8%	87.0%	**48.0%**	83.1%	75.3%	87.1%	48.3%	83.3%	75.5%
CST	86.4%	46.0%	82.5%	74.3%	87.0%	**47.0%**	83.0%	75.0%	87.4%	**48.0%**	83.4%	75.6%	88.0%	**49.0%**	83.9%	76.3%
EnSSL	**87.3%**	**47.0%**	**83.1%**	**75.2%**	**87.6%**	**47.0%**	**83.3%**	**75.4%**	**88.1%**	47.3%	**83.7%**	**75.9%**	**88.6%**	48.3%	**84.1%**	**76.5%**

The accuracy measure of the best performing algorithm is highlighted in bold.

Table 9 presents the statistical analysis for SETRED, Co-Forest, Democratic-Co learning, CST-Voting and EnSSL, performed by nonparametric multiple comparison procedures. As mentioned above, the control algorithm for the post-hoc test is determined by the best (e.g., lowest) ranking obtained in each FAR test while the adjusted p-value with Finner's test (p_F) was presented based on the corresponding control algorithm at the $\alpha = 0.05$ level of significance. The interpretation of Table 9 illustrates that CST-Voting and EnSSL exhibit the highest probability-based ranking by statistically presenting better results. Moreover, it is worth noticing that CST-Voting and EnSSL perform similarly with EnSSL presenting slightly better performance according to the FAR test.

Table 9. Friedman Aligned Ranking (FAR) test and Finner post-hoc test for SETRED, Co-Forest, Democratic-Co learning, CST-Voting and EnSSL.

Algorithm	FAR	Finner Post-Hoc Test	
		p_F-Value	Null Hypothesis
EnSSL	10.375	-	-
CST-Voting	18.7917	0.237802	accepted
Co-Forest	28.2083	0.016466	rejected
SETRED	44.2917	0.000004	rejected
Democratic-Co	50.8333	0.000000	rejected

5. Conclusions

In this work, we evaluated the performance of two ensemble SSL algorithms, entitled CST-Voting and EnSSL, for the credit scoring problem. The proposed ensemble algorithms combine the individual predictions of three of the most efficient and popular self-labeled algorithms, i.e., Co-training, Self-training, and Tri-training, using two different voting methodologies. The numerical experiments presented the efficacy of the ensemble SSL algorithms on three well-known credit score datasets, illustrating that reliable and robust prediction models could be developed by the adaptation of ensemble techniques in the semi-supervised learning framework.

It is worth noticing that we understand the limitations imposed on the generalizability of the presented results due to the use of the only three free available data sets as compared to other works [26–29]. Furthermore, since we do not know whether the values of the key parameters of the base learners within WEKA 3.9 are randomly initialized, optimized, or adapted, one may generally consider this approach as a limitation when comparisons of algorithms are conducted using only three datasets. We certainly intend to investigate this further in the near future.

Additionally, another interesting aspect for future research could be the development of a decision-support tool based on an ensemble SSL algorithm, concerning the credit risk scoring problem. The use of a predictive tool could assist financial institutions to decide whether to grant credit to consumers who apply. Since our numerical experiments are quite encouraging, our future work is concentrated on evaluating the proposed algorithms versus relevant methodologies and frameworks addressing the credit score problem such as [27–32] and versus recently proposed advanced SSL algorithms such as [59–61].

Author Contributions: I.E.L., N.K., A.K., V.T. and P.P. conceived of the idea, designed and performed the experiments, analyzed the results, drafted the initial manuscript and revised the final manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mays, E. *Handbook of Credit Scoring*; Global Professional Publishing: London, UK, 2001.
2. Altman, E. *Bankruptcy, Credit Risk, and High Yield Junk Bonds*; Wiley-Blackwell: Hoboken, NJ, USA, 2002.

3. Kramer, J. Clearly Money Has Something to Do with Life—But What Exactly? Reflections on Recent Credit Crunch Fiction (s). In *London Post-2010 in British Literature and Culture*; Koninklijke Brill NV: Leiden, Netherlands, 2017; p. 215.
4. Demyanyk, Y.; Van Hemert, O. Understanding the subprime mortgage crisis. *Rev. Financial Stud.* **2009**, *24*, 1848–1880. [CrossRef]
5. Hand, D.J.; Henley, W.E. Statistical classification methods in consumer credit scoring: A review. *J. R. Stat. Soc. Ser. A (Stat. Soc.)* **1997**, *160*, 523–541. [CrossRef]
6. Venkatraman, S. A Proposed Business Intelligent Framework for Recommender Systems. *Informatics* **2017**, *4*, 40. [CrossRef]
7. Lanza-Cruz, I.; Berlanga, R.; Aramburu, M. Modeling Analytical Streams for Social Business Intelligence. *Informatics* **2018**, *5*, 33. [CrossRef]
8. Stamate, C.; Magoulas, G.; Thomas, M. Transfer learning approach for financial applications. *arXiv* **2015**, arXiv:1509.02807.
9. Pavlidis, N.; Tasoulis, D.; Plagianakos, V.; Vrahatis, M. Computational intelligence methods for financial time series modeling. *Int. J. Bifurc. Chaos* **2006**, *16*, 2053–2062. [CrossRef]
10. Pavlidis, N.; Tasoulis, D.; Vrahatis, M. Financial forecasting through unsupervised clustering and evolutionary trained neural networks. In Proceedings of the Congress on Evolutionary Computation, Canberra, ACT, Australia, 8–12 December 2003; Volume 4, pp. 2314–2321.
11. Pavlidis, N.; Plagianakos, V.; Tasoulis, D.; Vrahatis, M. Financial forecasting through unsupervised clustering and neural networks. *Oper. Res.* **2006**, *6*, 103–127. [CrossRef]
12. Council, N.R. *Building a Workforce for the Information Economy*; National Academies Press: Washington, DC, USA, 2001.
13. Wowczko, I. Skills and vacancy analysis with data mining techniques. *Informatics* **2015**, *2*, 31–49. [CrossRef]
14. Dinh, T.; Kwon, Y. An Empirical Study on Importance of Modeling Parameters and Trading Volume-Based Features in Daily Stock Trading Using Neural Networks. *Informatics* **2018**, *5*, 36. [CrossRef]
15. Rokach, L. *Pattern Classification Using Ensemble Methods*; World Scientific Publishing Company: Singapore, 2010.
16. Zhu, X.; Goldberg, A. Introduction to semi-supervised learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2009**, *3*, 1–130. [CrossRef]
17. Guo, T.; Li, G. Improved tri-training with unlabeled data. In *Software Engineering and Knowledge Engineering: Theory and Practice*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 139–147.
18. Livieris, I.E.; Tampakas, V.; Kiriakidou, N.; Mikropoulos, T.; Pintelas, P. Forecasting students' performance using an ensemble SSL algorithm. In Proceedings of the 8th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Infoexclusion, Thessaloniki, Greece, 20–22 June 2018; ACM: New York, NY, USA, 2018; pp. 1–5.
19. Livieris, I.E.; Kanavos, A.; Vonitsanos, G.; Kiriakidou, N.; Vikatos, A.; Giotopoulos, K.; Tampakas, V. Performance evaluation of a SSL algorithm for forecasting the Dow Jones index. In Proceedings of the 9th International Conference on Information, Intelligence, Systems and Applications, Zakynthos, Greece, 23–25 July 2018; pp. 1–8.
20. Livieris, I.E.; Kanavos, A.; Tampakas, V.; Pintelas, P. An ensemble SSL algorithm for efficient chest X-ray image classification. *J. Imaging* **2018**, *4*, 95. [CrossRef]
21. Livieris, I.E.; Drakopoulou, K.; Tampakas, V.; Mikropoulos, T.; Pintelas, P. Predicting secondary school students' performance utilizing a semi-supervised learning approach. *J. Educ. Comput. Res.* **2018**. [CrossRef]
22. Livieris, I.E.; Drakopoulou, K.; Tampakas, V.; Mikropoulos, T.; Pintelas, P. An ensemble-based semi-supervised approach for predicting students' performance. In *Research on e-Learning and ICT in Education*; Springer: Berlin, Germany, 2018.
23. Levatić, J.; Brbić, M.; Perdih, T.; Kocev, D.; Vidulin, V.; Šmuc, T.; Supek, F.; Džeroski, S. Phenotype prediction with semi-supervised learning. In Proceedings of the New Frontiers in Mining Complex Patterns: Sixth Edition of the International Workshop NFMCP 2017 in Conjunction with ECML-PKDD 2017, Skopje, Macedonia, 18–22 September 2017.
24. Levatić, J.; Dzeroski, S.; Supek, F.; Smuc, T. Semi-supervised learning for quantitative structure-activity modeling. *Informatica* **2013**, *37*, 173.

25. Triguero, I.; García, S.; Herrera, F. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowl. Inf. Syst.* **2015**, *42*, 245–284. [CrossRef]
26. Triguero, I.; García, S.; Herrera, F. SEG-SSC: A Framework Based on Synthetic Examples Generation for Self-Labeled Semi-Supervised Classification. *IEEE Trans. Cybern.* **2014**, *45*, 622–634. [CrossRef] [PubMed]
27. Louzada, F.; Ara, A.; Fernandes, G.B. Classification methods applied to credit scoring: Systematic review and overall comparison. *Surv. Oper. Res. Manag. Sci.* **2016**, *21*, 117–134. [CrossRef]
28. Kennedy, K.; Namee, B.M.; Delany, S.J. Using semi-supervised classifiers for credit scoring. *J. Oper. Res. Soc.* **2013**, *64*, 513–529. [CrossRef]
29. Ala'raj, M.; Abbod, M. A new hybrid ensemble credit scoring model based on classifiers consensus system approach. *Expert Syst. Appl.* **2016**, *64*, 36–55. [CrossRef]
30. Abellán, J.; Castellano, J.G. A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Syst. Appl.* **2017**, *73*, 1–10. [CrossRef]
31. Tripathi, D.; Edla, D.R.; Cheruku, R. Hybrid credit scoring model using neighborhood rough set and multi-layer ensemble classification. *J. Intell. Fuzzy Syst.* **2018**, *34*, 1543–1549. [CrossRef]
32. Zhang, H.; He, H.; Zhang, W. Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring. *Neurocomputing* **2018**, *316*, 210–221. [CrossRef]
33. Levatić, J.; Ceci, M.; Kocev, D.; Džeroski, S. Self-training for multi-target regression with tree ensembles. *Knowl.-Based Syst.* **2017**, *123*, 41–60. [CrossRef]
34. Levatić, J.; Kocev, D.; Ceci, M.; Džeroski, S. Semi-supervised trees for multi-target regression. *Inf. Sci.* **2018**, *450*, 109–127. [CrossRef]
35. Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, Cambridge, MA, USA, 26–30 June 1995; pp. 189–196.
36. Li, M.; Zhou, Z. SETRED: Self-training with editing. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin, Germany, 2005; pp. 611–621.
37. Blum, A.; Mitchell, T. Combining labeled and unlabeled data with co-training. In Proceedings of the 11th Annual Conference on Computational Learning Theory, Madison, WI, USA, 24–26 July 1998; pp. 92–100.
38. Zhou, Y.; Goldman, S. Democratic co-learning. In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Boca Raton, FL, USA, 15–17 November 2004; pp. 594–602.
39. Li, M.; Zhou, Z. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2007**, *37*, 1088–1098. [CrossRef]
40. Zhou, Z.; Li, M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 1529–1541. [CrossRef]
41. Zhou, Z. When Semi-supervised Learning Meets Ensemble Learning. In *Frontiers of Electrical and Electronic Engineering in China*; Springer: Berlin, Germany, 2011; Volume 6, pp. 6–16.
42. Dietterich, T. Ensemble methods in machine learning. In *Multiple Classifier Systems*; Kittler, J.; Roli, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 1857, pp. 1–15.
43. Kostopoulos, G.; Livieris, I.; Kotsiantis, S.; Tampakas, V. CST-Voting—A semi-supervised ensemble method for classification problems. *J. Intell. Fuzzy Syst.* **2018**, *35*, 99–109. [CrossRef]
44. Livieris, I.E. A new ensemble self-labeled semi-supervised algorithm. *Informatica* **2019**, to be appeared, pp. 1–14.
45. Baumgartner, D.; Serpen, G. Large Experiment and Evaluation Tool for WEKA Classifiers. In Proceedings of the International Conference on Data Mining, Miami, FL, USA, 6–9 December 2009, Volume 16, pp. 340–346.
46. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. The WEKA data mining software: An update. *SIGKDD Explor. Newslett.* **2009**, *11*, 10–18. [CrossRef]
47. Triguero, I.; Sáez, J.; Luengo, J.; García, S.; Herrera, F. On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. *Neurocomputing* **2014**, *132*, 30–41. [CrossRef]
48. Bache, K.; Lichman, M. *UCI Machine Learning Repository*; University of California, Department of Information and Computer Science: Irvine, CA, USA, 2013.
49. Domingos, P.; Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Mach. Learn.* **1997**, *29*, 103–130. [CrossRef]

50. Platt, J. Using sparseness and analytic QP to speed training of support vector machines. In *Advances in Neural Information Processing Systems*; Kearns, M., Solla, S., Cohn, D., Eds.; MIT Press: Cambridge, MA, USA, 1999; pp. 557–563.
51. Rumelhart, D.; Hinton, G.; Williams, R. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*; Rumelhart, D., McClelland, J., Eds.; MIT Press: Cambridge, MA, USA, 1986; pp. 318–362.
52. Aha, D. *Lazy Learning*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1997.
53. Wu, X.; Kumar, V.; Quinlan, J.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.; Ng, A.; Liu, B.; Yu, P.; et al. Top 10 algorithms in data mining. *Knowl. Inf. Syst.* **2008**, *14*, 1–37. [CrossRef]
54. Liu, C.; Yuen, P. A boosted co-training algorithm for human action recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 1203–1213. [CrossRef]
55. Tanha, J.; van Someren, M.; Afsarmanesh, H. Semi-supervised selftraining for decision tree classifiers. *Int. J. Mach. Learn. Cybern.* **2015**, *8*, 355–370. [CrossRef]
56. Livieris, I.; Kanavos, A.; Tampakas, V.; Pintelas, P. An auto-adjustable semi-supervised self-training algorithm. *Algorithm* **2018**, *11*, 139. [CrossRef]
57. Hodges, J.; Lehmann, E. Rank methods for combination of independent experiments in analysis of variance. *Ann. Math. Stat.* **1962**, *33*, 482–497. [CrossRef]
58. Finner, H. On a monotonicity problem in step-down multiple test procedures. *J. Am. Stat. Assoc.* **1993**, *88*, 920–923. [CrossRef]
59. Levatić, J.; Ceci, M.; Kocev, D.; Džeroski, S. Semi-supervised classification trees. *J. Intell. Inf. Syst.* **2017**, *49*, 461–486. [CrossRef]
60. Jia, X.; Wang, R.; Liu, J.; Powers, D.M. A semi-supervised online sequential extreme learning machine method. *Neurocomputing* **2016**, *174*, 168–178. [CrossRef]
61. Li, K.; Zhang, J.; Xu, H.; Luo, S.; Li, H. A semi-supervised extreme learning machine method based on co-training. *J. Comput. Inf. Syst.* **2013**, *9*, 207–214.