algorithms

MDPI

Article

A Soft-Voting Ensemble Based Co-Training Scheme Using Static Selection for Binary Classification Problems

Stamatis Karlos *, Georgios Kostopoulos and Sotiris Kotsiantis

Educational Software Development Laboratory (ESDLab), Department of Mathematics, University of Patras, 26504 Patras, Greece; kostg@sch.gr (G.K.); sotos@math.upatras.gr (S.K.)

* Correspondence: stkarlos@upatras.gr

Received: 1 November 2019; Accepted: 13 January 2020; Published: 16 January 2020

Abstract: In recent years, a forward-looking subfield of machine learning has emerged with important applications in a variety of scientific fields. Semi-supervised learning is increasingly being recognized as a burgeoning area embracing a plethora of efficient methods and algorithms seeking to exploit a small pool of labeled examples together with a large pool of unlabeled ones in the most efficient way. Co-training is a representative semi-supervised classification algorithm originally based on the assumption that each example can be described by two distinct feature sets, usually referred to as views. Since such an assumption can hardly be met in real world problems, several variants of the co-training algorithm have been proposed dealing with the absence or existence of a naturally two-view feature split. In this context, a Static Selection Ensemble-based co-training scheme operating under a random feature split strategy is outlined regarding binary classification problems, where the type of the base ensemble learner is a soft-Voting one composed of two participants. Ensemble methods are commonly used to boost the predictive performance of learning models by using a set of different classifiers, while the Static Ensemble Selection approach seeks to find the most suitable structure of ensemble classifier based on a specific criterion through a pool of candidate classifiers. The efficacy of the proposed scheme is verified through several experiments on a plethora of benchmark datasets as statistically confirmed by the Friedman Aligned Ranks non-parametric test over the behavior of classification accuracy, F_1-score, and Area Under Curve metrics.

Keywords: binary classification; co-training; ensemble methods; feature views; dynamic ensemble selection; Soft-Voting

1. Introduction

In recent years, the latest research on machine learning (ML) which has placed much emphasis on learning from both labeled and unlabeled examples is mainly expressed by semi-supervised learning (SSL) [1]. SSL is increasingly being recognized as a burgeoning area embracing a plethora of efficient methods and algorithms seeking to exploit a small pool of labeled examples together with a large pool of unlabeled ones in the most efficient way. Since in most real-world applications there is an abundance of unlabeled examples, while labeled examples are either difficult or expensive to obtain, SSL has emerged as a promising domain with important applications in a variety of scientific fields with substantial results [2,3].

In general, SSL methods are commonly divided into two key tasks, as follows: semi-supervised classification (SSC) for discrete-value output variables and semi-supervised regression (SSR) for real-value ones [4]. The classification task, usually referred to as pattern recognition in engineering or discriminant analysis in statistics [5], has been widely studied under the semi-supervised framework

for classifying any given example to one out of the included class labels into a predetermined set. Depending on the number of class labels, classification problems may be either binary (with two class labels) or multi-class (with more than two class labels). For that purpose, a number of semi-supervised algorithms have been developed and successfully implemented, such as self-training [6], co-training [7], and tri-training [8], as well as approaches that are based on semi-supervised support vector machines and transductive learning or SSL graph-based methods, to name just a few [9].

Co-training is a representative multi-view SSC algorithm originally based on the assumption that each example can be described by two distinct feature sets, usually referred to as views [7]. Then, two classification algorithms are trained separately on each view and the most confident predictions of each one on the unlabeled data are used to augment the training set of the other. Let L_D denote a small set of labeled examples and U_D a large set of unlabeled ones. The two separate classifiers C_1, C_2 are retrained on the enlarged set L_D and the process is repeated for a predefined number of iterations or until a stopping criterion is satisfied, such as the U_D pool to be empty. However, since such a two-view assumption can hardly be met in real world problems, several variants of the co-training algorithm have been proposed dealing with the absence or existence of a naturally two-view feature split with notable results.

In addition to these, ensemble learning or committee-based learning or learning multiple classifier systems has emerged recently and is considered as one of the most adequate solutions for building powerful and accurate classification models [10]. Instead of using one algorithm for building a learning model, ensemble methods are commonly used to construct and combine a set of classifiers, either weak or strong, generally called base learners. The fundamental points for the effectiveness of an ensemble method concern careful selection of both base learners [11] and the combination method for producing the final hypothesis [12]. Averaging (simple or weighted [13]) and voting (majority, unanimity, plurality, or even weighted votes) are popular and commonly used combination methods [14] depending on the problem which needs to be resolved [10]. Moreover, approaches using committees of base learners into the core of their learning process have also been demonstrated recently, presenting encouraging results [15].

The necessity of accurate and robust decisions inside a semi-supervised scheme play a cardinal role, especially in cases where the number of initially labeled instances is quite small and no decision correcting or editing mechanisms have been placed inside the learning kernel. Therefore, strategies trying to build an ensemble optimizing well-defined criteria could be a useful asset of a compact semi-supervised algorithm for selecting the most suitable structure per task. Although this concept has been highly exploited under the supervised mode, only few works have been detected in the related literature that apply similar approaches [16,17]. Furthermore, there are some works in this field that apply mechanisms suitably combining the decisions of selected base learners, failing, however, to state the reasons of their choice, apart from some generic properties, such as the combination of one generative and one discriminative approach that favors the diversity of the applied co-training algorithm [18].

In this context, a soft-Voting ensemble-based co-training scheme using static selection strategy, regarding binary classification problems, is proposed. Since co-training is primarily relying on the multi-view assumption, a heuristic scheme is adopted for manually generating the two views. Although this random split may act towards injecting diversity into a multi-view SSL approach, the main asset of the proposed algorithm is the construction of an ensemble learner choosing among five different classifiers, based on a novel objective function that measures the efficacy of any examined pair of classifiers per dataset, operating under a soft-Voting scheme. The efficacy of the proposed mechanism for selecting the ensemble's participants is verified through several experiments against both single-view SSL variants—through the well-known self-training scheme—and the co-training scheme, applying all of the 10 different pairs of algorithms into the same soft-Voting learner and the five individual classifiers, on a plethora of benchmark datasets over five separate labeled ratio values.

The obtained results regarding two well-known classification metrics are statistically confirmed by the applied Friedman Aligned Ranks non-parametric test.

The rest of this paper is organized as follows: the co-training framework is presented in Section 2, while reviewing recent studies concerning both the application of co-training in real world applications and Ensemble Selection strategies. In Section 3, we propose and describe in detail the proposed co-training scheme operating under a random feature split using internally a static selection strategy regarding a soft-Voting algorithm. Section 4 includes the experiments carried out along with the relevant results. Finally, in Section 5 we comment on the results considering some thoughts for future work.

2. Related Works

This section consists of two different parts, which highlight the two main points related to our proposed work. After having mentioned some of the most important works towards these directions, we can summarize our main contributions into the next section, facilitating the structure of this work.

2.1. Co-Training Studies

The first part of this section is dedicated to Co-training scheme and some of the numerous variants that have been demonstrated. Hence, Co-training is deemed to be a representative multi-view SSL method established by Blum and Mitchell [7] for binary classification problems and, in particular, for categorizing web pages either as course or as non-course. It is based on the premise that each example can be naturally divided into two separate set of features usually referred to as views, which is clearly an assumption of great importance for the implementation of the particular method. Co-training is identified as a "two-view weakly supervised algorithm" [6] since it incorporates the self-training approach to separately teach each one of the two supervised classifiers in the corresponding feature view and boost the classification performance exploiting the unlabeled examples in the most efficient manner [19]. Moreover, Zhu and Goldberg consider co-training as a wrapper method which is not affected by the two supervised classifiers employed in the relevant procedure, provided that they produce good predictions on unlabeled data [20]. Several modifications have been implemented since then, including mutual-learning and co-EM—bringing together the co-training and Expectation-Maximization (EM) approaches—exploiting mainly simple classifiers like naive Bayes (NB) [7,21].

In addition to the "two view" assumption, the effectiveness of the particular method depends largely on two other key assumptions: the first one is that each view is adequate for classifying the unlabeled data using a small set of labeled examples for training, while the second one is that each view is conditionally independent given the class label. When either of these assumptions is not met, different co-training variants have been proposed with comparable results. In the case where the "two view" assumption is not fulfilled, a random feature partition could take place to facilitate the application of the method as proposed by Zhu and Goldberg [20]. In such cases, the feature set is partitioned into two subsets of almost equal size, which henceforth form the two feature views, while different classifiers C_1, C_2 are employed. In addition, the same classifiers may be used under different configuration parameters, thus ensuring the diversity between them [22].

The number of studies that propose co-training as an effective SSL method is really restricted. One of these is presented in [22], where sentiment analysis is the main focus, while in [23] the authors have tackled a health care issue. Although the popularity of this type of problem is widespread and even though any shortcomings that may be associated with a large amount of labeled data can be efficiently leveraged by other SSL methods, yet co-training seems to not have been delved into thoroughly enough. In this study, three different sources of text data were examined: news articles, online reviews, and blogs. A number of co-training variants were designed, focusing on the way the split of the feature space takes place, fitting appropriately the specific properties that characterize text data, such as the creation of one view by unigrams and the rest by bigrams or by

adopting character-based language models and bag-of-words models, respectively. The produced results demonstrate the effectiveness of the co-training algorithm.

Another task that has been efficiently tackled by using the co-training method is that of drug discovery, where classification methods need to be applied so as to predict the suitability of some molecules considering treatments of diseases and their possibly induced side-effects during the initial steps of tedious experiments [24]. Accurate predictions may save both time and money, since fewer combinations would be investigated and the final results could be acquired much faster. In this work, two different views were available, stemming from chemistry and biology, and had to be mixed to reach the final conclusion. The approaches that were examined may be summed up as follows: (i) access separately each view either with a base classifier or the partial least squares (PLS) regression method [25], (ii) fuse the different views, either by joining the heterogeneous data without any preprocess or after having applied the PLS method, also used for dimensionality reduction, and (iii) a modification of the co-training method (co-FTF). Ensemble tree-based learners were preferred in this last approach, handling imbalanced datasets appropriately and leading to promising results, while examining two labeled ratio scenarios. In addition, a random forest of predictive clustering trees was incorporated in a self-training scheme for multi-target regression, thus improving the performance of the employed SSL approach [26].

An expansion of the co-training algorithm, which includes an ensemble of tree-based learners as base learner, has been proposed in [27]. Under the assumptions that are presented there, the necessity of two sufficient and redundant views has been eliminated for the proper operation of Co-Forest. Furthermore, the bootstrap method that is exploited during the creation of the included decision trees provides the required diversity and, at the same time, reduces the chance of exporting biased decisions, leading to an efficient operation of the SSL scheme. Adaptive Data Editing based Co-Forest (ADE-Co-Forest) [28] constitutes a variant of the original Co-Forest algorithm, introducing an internal mechanism in order to tackle the mislabeled instances, thus improving the total predictive behavior, since both false negative/positive error rates are further reduced, compared to its ancestor. A boosted co-training algorithm has also been proposed for a real-task problem—to be more specific, it concerns the human action recognition—which is based on the mutual information and the consistency between labeled and unlabeled data. Two metrics, named inter-view and intra-view confidence, are introduced and exploited dynamically so as to select the most appropriate subset of the unlabeled pool with the corresponding pseudo-labels [29].

Recently, a quite effective co-training method was introduced in [30] for early prognosis of undergraduate students' performance in the final examinations of a distance learning course based on attributes which are naturally divided into two separate and independent views. The first one concerns students' characteristics and academic achievements which are manually filled out by tutors, while the second one refers to attributes tracking students' online activity in the course learning management system and which are automatically recorded by the system. It should be mentioned that semi-supervised multi-view learning has also been successfully applied for gene network reconstruction combining the interactions predicted by a number of different inference methods [19]. In a similar work, an ensemble-based SSL approach has been proposed for the computational discovery of miRNA regulatory networks from large-scale predictions produced by different algorithms [31].

2.2. Ensemble Selection Strategies

The second part is oriented towards reporting briefly some of the most important points related with Ensemble Selection concept [32,33]. To be more specific, some usual keywords in this field are Multiple Classification Systems (MCSs), Static Ensemble Classifier (SEC), and Dynamic Ensemble Classifier (DEC), as well as classifiers' competence and diversification. The way that all these terms are connected is the fact that when a new ensemble learner is designed, the main ambitions are the employment of complementary and diverse participants, following the main asset of MCSs regarding the continuous increase of the predictive rate. The main difference between the remaining two

terms is the fact that SEC strategies examine a global solution regarding the total set of unknown instances, while the DES approaches provide a separate solution per test instance using mainly local restrictions. Despite their distinct roles, they can be combined under hybrid mechanisms sharing similar measurement metrics or ML techniques for converging to their decisions [34,35].

Ensemble Selection has been inserted as a new stage into the original chain of constructing an ensemble learner, taking into consideration both the importance of computational needs that arise when we trust ensembles with too many participants and the fact of discarding less accurate models or models that reduce the internal diversity. This tactic is usually referred to as ensemble pruning or selective ensemble. A taxonomy of these techniques has been proposed in [36], assigning them to four different categories: (i) ranking-based, (ii) clustering-based, (iii) optimization-based, and (iv) others, including the remaining techniques that cannot be strictly categorized to any of the previous three subsets. Another taxonomy was demonstrated in 2014, concerning mainly the actual need of DES in practice and the relation between the inherent complexity of classification problem, measured by appropriate metrics, and the contribution of the examined Dynamic Selection approaches [16]. Prototype selection techniques have also been examined in the abovementioned framework, acting beneficially towards both reducing computational resources and boosting the classification accuracy [37]. Furthermore, one related work on the field of SSL has been proposed using the competence of selected classifiers that stems from an affinity graph, achieving smoothness of the decisions for neighboring data [17].

3. The Proposed Co-Training Scheme

Motivated by the above studies, in the present paper we make an attempt to put forward an ensemble-based co-training scheme for binary classification problems adopting a strategy of choosing the base classifiers of the ensemble from an available pool of candidate classification algorithms per dataset. The most important points concerning our contribution are outlined below:

- We propose a multi-view SSL algorithm that handles efficiently both labeled (L) and unlabeled (U) data in the case of binary output variables.
- Instead of demanding two sufficient and redundant views, a random feature split is applied, thereby increasing the applicability and improving the performance of the finally formatted algorithm [38].
- We introduce a simple mechanism concerning the cardinality of unlabeled examples per different class that is mined for avoiding overfitting phenomena in cases where imbalanced datasets must be assessed.
- We insert a preprocess stage, where a pool of single learners is mined by a Static Ensemble Selection algorithm to extract a powerful soft-Voting ensemble per different classification problem, seeking to produce a more accurate and robust semi-supervised algorithm operating under small labeled ratio values.

Let the whole dataset (X) consist of n instances and k features, apart from the class variable (Y) that, in the context of this work, is restricted to be a binary one. Thus, without loss of generality, we assume that $y_i \in \{0,1\}$ for each labeled instance $\{l_i, 1 \leq i \leq n_i\}$, while each unlabeled instance $\{u_i, 1 \leq i \leq n_u\}$ is characterized by the absence of the corresponding y_i value. The parameters n_l and n_u represent the cardinality of L and U subsets, respectively. After having removed all missing values—leading to a new cardinality of total instances (n′)—it is evident that the following equation holds:

$$n' = n_i + n_u \tag{1}$$

Besides holding both numeric and categorical features, all the features of the latter form are converted into binary ones, increasing the initial number of k features into k′, in case X contains at least one of them. Otherwise, since no augmentation of the initial features has been applied, the next two quantities coincide: $k \equiv k'$. Under this generic approach, classification algorithms that cannot handle categorical data are not rejected by the total proposed process.

This choice seems safe enough, since it does not reject the adoption of any learner—this mainly refers to learning algorithms that cannot handle efficiently the existence of both numerical and categorical data—although the manipulation of heterogeneous features is an open issue [39]. Afterwards, without introducing any specific assumption about the relationship or the origination of any included feature, the available feature vector F: $<f_1, f_2, \ldots, f_{k'}>$ is split into two newly formatted subsets F_1 and F_2, where $F = F_1 \cup F_2$. Hence, two different datasets X_1, X_2 are generated, respectively, both including disjoint feature sets, but sharing the same class variable Y. Therefore, the final hypothesis space could be summarized as follows: F_{view}: $X_{view} \rightarrow [0, 1]$, where view $= 1, 2$.

Through the above described methodology, the following two choices are enabled: either to apply a common learning strategy for both views, such as adopting the same learner, or tackling each view separately, depending on underlying properties, such as the views' cardinalities, independence or correlation assumptions that affect the views' internal structure or other kind of relationships that specify the nature of each view, since two distinct tasks have been raised. Following the majority of the existing approaches found in the literature and taking into consideration that a random feature split operates as an agnostic factor regarding the structure of the constructed views, the first approach was adopted in the present study [40].

Under this strategy, and before the common base learner is built per view, a preprocess stage is inserted. This aims to measure the rate of the imbalanced instances found in the provided training set and to define the number of the instances that have to be mined from each class per iteration ($Mined_{class0}$, $Mined_{class1}$). Due to the SSL concept, the quota of L and U subsets is defined by a labeled ratio value (R). Given this setting, the amount of the initial training set (L_{view}) is computed according to the following formula:

$$\text{InitSize} = R \times \text{size}(X), \ \forall \text{view} = 1, 2 \quad (2)$$

The cardinalities of both classes are then computed (C_{max}, C_{min}) regarding the available L^0_{view}. The minimum of them is set equal to 1 ($Mined_{class0}$), while the other one is equal to $\lfloor C_{max}/C_{min} \rfloor$ ($Mined_{class1}$). In this way, the provided class distribution of the labeled instances is assumed to be representative of the total problem defined also by the unknown instances that must be assessed. Finally, these two variables are exploited during the learning stage to retrieve a suitable number of unlabeled instances per class during each iteration.

Now, as it regards the choice of the base learners, we selected five representative algorithms from different learning families, capturing a wide spectrum of properties, concerning both assets and defects, which should be combined and avoided, respectively, in order to construct appropriately an accurate and robust enough ensemble learner per dataset so as to initialize the co-training process [41]. For this purpose, our pool of classifiers (C) consists of support vector machines (SVMs) [42], k-nearest-neighbors (kNN) [43], a simple tree inducer (DT) from family of decision trees [44], naive Bayes (NB) [45], and logistic regression (LR) [46]. In order to keep the computational needs of the exported ensemble, we restrict the cardinality of classifier participants under our Voting scheme, setting this number equal to 2. Thus, we had to employ a soft variant of Voting scheme which takes into account the class-probabilities of each algorithm and combines these decisions through averaging process, instead of hard voting through on-off decisions [29], where the occurrence of ties with the even number of base learners would appear too frequent. Furthermore, the stage of averaging the decisions of each individual participant generally leads to the reduction of the ensemble's variance and helps to surpass the structure sensitivity that is usually detected in more unstable methods, considering the input data

To be more specific, if we assume that we tackle with a binary classification problem containing a set of labels $Y = \{0, 1\}$ and a feature space $X \in \mathbb{R}^k$, such that for any probabilistic classifier F holds the next function: $F : X \rightarrow Y$, then for each instance m the decision profile of learner j is a pair of class probabilities $\left[P_{j0}, P_{j1}\right]$ which sum up to 1. Consequently, the mechanism of a simple, without

weighting factors, soft-Voting classifier, given an instance x_m, combines the decisions of all the candidate classification algorithms searching the most probable class (ω) as follows:

$$\hat{y}_m = arg \max_{\omega} \sum_{j=1}^{|p|} P_j(=\omega|x_m),\ y_m \in Y,\ m \in \{1, 2, \ldots, n'\}, \tag{3}$$

The class with the largest average probability is exported as the prevalent one through this pipeline, where $\hat{y}_m \in Y$ and the notation of $\left|p\right|$ depicts the number of the combined classifiers.

Trying to uncover the function of our preprocess stage which constructs the base learner of the proposed co-training scheme, we had to refer that the ambition of any Static Ensemble Selection strategy is to construct a subset C^*, such that $C^* \subset C$ and $|C^*| = 2$, which satisfies better the chosen criteria for obtaining the most desired performance over test instances. In our case, we investigate the most compatible pair of learners that maximizes our proposed criterion under an unweighted soft-Voting scheme. Through this, we measure the number of instances for which the decision of the soft-Voting scheme remains correct when the two candidate participants disagree ($q_{corrected}$), normalized by the total amount of disagreements based on the label of the examined instances ($q_{disaggre}$), as well as the rate of non-common errors ($q_{common\ errors}/v$). To this end, we introduce the objective function of Equation (4), which is defined as a linear combination of the mentioned quantities:

$$\begin{gathered} Q^a_{soft}(i,j) = a * \frac{q^{i,j}_{corrected}}{q^{i,j}_{disagree}} + (1-a) * (1 - \frac{q^{i,j}_{common\ errors}}{v}), \\ 0 \le a \le 1,\ i, j \in \{0, 1, 2 \ldots, |C^*|\}\ with\ i \neq j, \end{gathered} \tag{4}$$

where a is a parameter to balance the importance between the included terms. Actually, the first one rewards the pair of classifiers that managed to act complementary, since the more times the confidence of the classifier that guessed correctly the corresponding class label overpowered against the erroneous one, the larger values this term records. On the other hand, the second term penalizes the pair of classifiers whose common decisions coincide with mislabeling cases by reducing its value when such behavior occurs. The parameter v symbolizes the cardinality of the validation set over which the rest of quantities are calculated. Giacinto and Roli called this diversity measure as "the double-fault measure" [47].

Although an analysis of the selected a value could raise the interest of further research, we selected the value of 0.5 for equal importance. Thus, for each examined dataset D, which contains both labeled and unlabeled data, we split the labeled set into train and validation set, in a same manner as the default k-fold-cross-validation strategy, applying the previously referred Static Ensemble Selection strategy so as to detect the most favorable pair of classifiers for our soft-Voting ensemble learner. In case that $q_{disaggre} = 0$, then a is set equal to 0, holding only the second term.

Exploiting the exported soft-Voting ensemble learner as the base learner of our co-training variant, each L^0_{view} is fitted with $Co(Votesoft(C^*_i, C^*_j)) \equiv Co(Vote^{SEC}_{soft})$—we use the notation C^*_i and C^*_j for the selected learners which are included into C^*—and the corresponding class probabilities for each unlabeled instance per view (u^i_{view}) are computed per iteration. Next, only the top-class0 and top-class1 instances per class are selected, based on the estimated confidence measure. Subsequently, these instances are exported by the current U subset (since both views share the common unlabeled set, it does not need to use the view index when referring to the U subset). Then, they are added to the training set of the opposite view along with the most prominent class label based on base learner's decision. Therefore, if the target variable of the m-th instance of U is categorized as class0 by the F_1 classifier (x_m: <f_1, f_2, ... , $f_{k'/2}$ with $probclass0_{first} > 0.5$), then the L^{iter}_2 subset during the iter-th iteration has to be augmented with the same instance, using the corresponding features of the second view and the estimated class variable (x_m: <$f_{k'/2+1}$, $f_{k'/2+2}$, ... , $f_{k'}$|class0>).

According to this learning scheme whose main ambition is to teach two different learners of the same classification algorithm through mutual disagreement concept, each learner injects into the other the information that is retrieved by the supplied view per iteration. A more theoretical analysis of the error bounds that can be achieved through the disagreement-based concept in case of Co-training could be found in [48]. Since our strategy of constructing the base learner of co-training through a static ensemble selection mechanism $Vote^{SEC}_{soft}$, we assume that we provide an accurate enough algorithm whose both competence's performance and diversity's behavior have been verified through a validation set so as to avoid overfitting phenomena or heavy mislabeling learning behaviors.

To sum up, the pseudo-code of the introduced SEC strategy (SSoftEC) as well as the proposed co-training variant are presented in Algorithms 1 and 2, respectively.

Algorithm 1. SSoftEC *strategy*

Input:
L—labeled set
f—number of folds to split the L
C—pool of classification algorithms exporting class probabilities
α—value of balancing parameter
Main Procedure:
For each $i, j \in \{0, 1, \ldots, |C|\}$ and $i \neq j$ **do**
Set $iter = 0$, $Q^{a}_{soft}(i, j) = 0$
Split L to f separate folds: $\left\{L^{(1)}, L^{(2)}, \ldots, L^{(f)}\right\}$
While $iter \leq$ f **do**
    Train C_i, C_j on $L \setminus L^{(iter)}$
    Apply C_i, C_j on $L^{(iter)}$
    Update $Q^{a}_{soft}(i, j)$ according to Equation (4)
    $iter = iter + 1$
Output:
Return pair of indices i, j such that: $(i, j)^{*} : arg \max_{i,j} Q^{a}_{soft}(i, j)$.

Algorithm 2. *Ensemble based co-training variant*

Mode:
Pool-based scenario over a provided dataset $D = X_{n \times k} \bigcup Y_{n \times 1}$
x_i—vector with k features $<f_1, f_2, \ldots f_k> \forall\ 1 \leq i \leq n$
y_i—scalar class variable with $y_i \in \{0, 1\}\ \forall\ 1 \leq i \leq n$
$\{x_i, y_i\}$—i-th labeled instance (l^i) with $1 \leq i \leq n_l$
$\{x_i\}$—i-th unlabeled instance (u^i) with $1 \leq i \leq n_u$
F_{view}—separate feature sets with view $\in$ [1,2]
$learner_{view}$—build of selected learner on corresponding View, $\forall view = 1, 2$
Input:
L^{iter}—labeled instances during iter-th iteration, $L^{iter} \subset D$
U^{iter}—unlabeled instances during iter-th iteration, $U^{iter} \subset D$
iter—number of combined executed iterations
MaxIter—maximum number of iterations
C—pool of classifiers $\equiv$ {SVM, kNN, DT, NB, LR}
(f, α)—number of folds to split the validation set during SEC and value of Equation (4)
Preprocess:
k′—number of features after having converted each categorical feature into binary
n′—number of instances after having removed instances with at least one missing value
C_j—instance cardinalities of both existing classes with j $\in$ {min, max}
$Mined_c$—define number of mined instances per class, where c $\in$ {$class_0$, $class_1$}

Main Procedure:
Apply SSoftEC(L^0, f, C, α) and obtain C_i^*, C_j^*
Construct $Vote_{soft}(C_i^*, C_j^*)$
Set *iter* = 0
While *iter* < MaxIter **do**
  For each *view*
    Train learner$_{\text{view}}$ on L_{view}^{iter}
    Assign class probabilities for each $u_i \in U^{iter}$
    For each *class*
      Detect the top Mined$_{\text{class}} \equiv$ Ind$_{\text{view}}$
      Update:
      $L_{view}^{iter+1} \leftarrow L_{view}^{iter} \cup \left\{x_j,\ arg\ \max_{class} P(\ Y = class|X_{view}\ )\ \forall\ j \in Ind_{\sim view})\right\}$
      (The sign $\sim$ view means the opposite view from the current.
      $U_{view}^{iter+1} \leftarrow U_{view}^{iter} \setminus \left\{x_j\right\} \forall j \in Ind_{\sim view}$
iter = *iter* + 1
Output:
Use $Vote_{soft}(C_i^*, C_j^*)$ trained on L^{MaxIter} to predict class labels of test data.

4. Experimental Procedure and Results

For the purpose of our study a number of experiments were carried out using 27 benchmark datasets from UCI Machine Learning Repository [49] regarding binary classification problems (Table 1), where the sign # depicts the cardinality of the corresponding quantity. Note that the columns entitled # Features in Table 1, counts all the features apart from the class variable. These datasets have been partitioned into 10 equal-sized folds using the stratified 10-fold-CV resampling procedure so that each fold should have the same distribution as the entire dataset [50]. This process was repeated 10 times until all folds were used as the testing set and the results were averaged. Moreover, each fold was divided into two subsets, one labeled and the other one unlabeled, in accordance with a selected labeled ratio value (R) which is defined as follows:

$$R = |L_D|/(|L_D|+|U_D|). \tag{5}$$

Table 1. Description of datasets used from the UCI repository.

Dataset	# Instances	# Features	Dataset	# Instances	# Features
bands	365	19	monk-2	432	6
breast	277	48	pima	768	8
bupa	345	6	saheart	468	9
chess	3196	38	sick	3772	33
colic.orig	368	471	tic-tac-toe	958	27
diabetes	768	8	vote	435	16
heart-statlog	270	13	wdbc	569	30
kr-vs-kp	3196	40	wisconsin	683	9
mammographic	830	5			

#: the cardinality of the corresponding quantity.

In order to study the influence of the amount of labeled data in the training set, three different ratios were used, and in particular: 10%, 20%, and 30%. In general, the *R* (%) values over which researchers are interested are the smaller ones ($R < 50\%$), so as to be consistent with the practical aspect of SSL scenario. The effectiveness of the proposed co-training scheme was compared to several co-training and self-training variants. For verifying the supremacy of the $Vote_{soft}^{SEC}$ as base classifier, we built the soft-Voting versions based on all pairs of the inserted pool of classifiers (C). Furthermore, the version that exploits the decisions of all the participants of C pool was implemented, as well as the

individual variants without voting. Thus, 16 different supervised classifiers were exhibited as base learners, all imported by the scikit-learn Python library [51] and in particular:

The SVMs using Radial Basis Function as kernel inside its implementation, representing one universal learner that tries to separate instances using hyper-planes and 'Kernel-trick' [52],

- The k-Nearest Neighbor (kNN) instance-based learner [53] with k equal to 5, a very effective method for classification problems, using the Euclidean metric as a similarity measure to determine the distance between two instances,
- A simple Decision Tree (DT) algorithm, a variant of tree induction algorithms with large depth that split the feature space using 'gini' criterion [44],
- The NB probabilistic classifier, a simple and quite efficient classification algorithm based on the assumption that features are independent of each other given the class label [54],
- The Logistic Regression (LR), a well-known discriminative algorithm that assumes the log likelihood ratio of class distributions is linear in the provided examples. Its main function supports the binomial case of the target variable, exporting posterior probabilities in a direct way. In our implementation, L2-norm during penalization stage was chosen [55].

For simplicity, we made use of the following notation in the experiments, while the parameters' configuration for all applied classification methods is presented in Table 2:

- C ≡ {SVM, kNN, DT, NB, LR}, the list of participant classification algorithms,
- $Self(learner)$, where $learner \in$ C,
- $Self(Vote(learner_i, learner_j))$, where $learner_i$, $learner_j \in$ C with $i \neq j$,
- $Self(Vote(all))$, where all participants of C are exploited under the Voting scheme,
- $Co(learner)$, where this kind of approach corresponds to the case that learner1 ≡ learner2 ≡ learner, with $learner \in$ C,
- $Co(Vote(learner_i, learner_j))$, where $learner_i$, $learner_j \in$ C with $i \neq j$, and the ensemble Voting learner is the same for both views, similar with the previous scenario,
- $Co(Vote(all))$, where all participants of C are exploited under the Voting scheme for each view, and finally,
- $Co(Vote^{SEC}_{soft})$, which coincides with the proposed semi-supervised algorithm.

Table 2. Configuration of exploited algorithms' parameters.

Algorithm	Parameters
k-NN	Number of neighbors: 5
	Distance function: Euclidean distance
SVM	Kernel function: RBF
DT	Splitting criterion: gini
	Min instances per leaf = 2
LR	Norm: L2
NB	Gaussian distribution
Self-training	MaxIter = 20
Co-training	MaxIter = 20

As mentioned before, there are 10 different pairs of algorithms that can be formatted with a pool of five candidate classifiers. In addition, the case of applying each one individually takes also place, as well as the case that all participants of pool C are exploited under the same Voting stage. Thus, 16 self-training variants and 16 co-training variants are examined against the proposed co-training algorithm, which selects through a static selection strategy the soft-Voting ensemble base learner into its operation per different task. As it concerns the parameter f, it has been set equal to 10, leading to a 10-fold-cross-validation procedure per examined dataset. The next tables depict only one out of three

different labeled ratio scenarios concerning the top five algorithms, based on total Friedman Ranking statistical process along with a smaller statistical comparison concerning only the top five algorithms. For a deeper analysis, the total results can be found in http://mL.math.upatras.gr/wp-content/uploads/2019/12/Official_results_co_training_ssoftec_voting.7z. Moreover, a pie chart has been provided in Figure 1, depicting the participation, into per centage style, of the pair of classifiers that were employed into the proposed strategy as base learner during all the experiments.

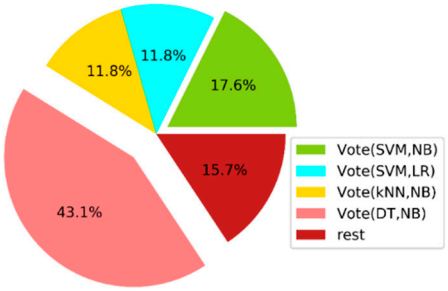

Figure 1. Pie chart depicting the participation of each combination inside our Static Ensemble strategy.

For evaluating the predictive performance of the proposed algorithm, three representative and widely used evaluation measures were adopted for measuring the obtained performance over the test set: classification accuracy, F_1-score, and Area Under the ROC Curve (AUC). Accuracy corresponds to the percentage of correctly classified instances, while F_1-score is an appropriate metric for imbalanced datasets and is defined as the harmonic mean of recall (r) and precision (p). In the case of a binary classification problem, they are defined as:

$$Accuracy = (tp + tn)/n \tag{6}$$

$$F_1 score = 2 \times tp/(2 \times tp + fp + fn). \tag{7}$$

where tp, tn, fp, fn, and n correspond to the number of true positive, true negative, false positive, false negative, and total number of instances, respectively. Finally, the latter one is related to the quality of the examined classifier ranking of any randomly chosen instance and is computed by aggregating the corresponding performance across all possible classification thresholds. The most favorable manner to visualize this metric is through plots of *TPR* vs. *FPR* or Sensitivity vs. (1-Specificity) relationship at different classification thresholds, where *TPR* stands for True Positive Rate, while *FPR* stands for False Positive Rate. Their analytical formulas are provided here:

$$TPR = tp/(tp + fn), \tag{8}$$

$$FPR = fp/(fp + tn). \tag{9}$$

The experimental results using 10% labeled ratio are summarized in Tables 3–5, where the best value per dataset is bold highlighted. Overall, it appears that the co-training Vote performs better than the corresponding self-training variants. Moreover, among the co-training variants employed, the proposed algorithm takes precedence over the rest on most of the datasets. In addition, we applied a familiar statistical tool to confirm the observed results. Hence, the Friedman Aligned Ranks [56] non-parametric test (significance level $\alpha = 0.05$) was used to compare all the employed SSL methods (Table 6). According to the calculated results, the algorithms are sorted from the best performer (lowest ranking) to the worst one (higher ranking). Therefore, it is statistically confirmed the supremacy of the $Co(Vote^{SEC}_{soft})$ algorithm, while the null hypothesis H_0 (i.e., the means of the results of two or more algorithms are the same) is rejected. Furthermore, the Nemenyi post-hoc test [57] ($\alpha = 0.05$) was applied to detect the specific differences between the algorithms, which is a commonly used non-parametric test for pairwise multiple comparisons. Table 6 includes the computed Critical

Difference (CD) which is the same for all the cases of this R-based scenario (CD = 2.27). It is statistically confirmed that the difference between the $Co(Vote^{SEC}_{soft})$ algorithm and the majority of the other methods is statistically significant in all examined metrics, thus verifying the predominance of the proposed co-training scheme. The fact also that the proposed algorithm outperforms the Vote (all) variants means that the implemented time-efficient SEC strategy provides a more accurate base learner for the field of SSL. Towards this direction, we visualize the performance of the proposed algorithm against $Co(Vote(all))$ for the examined metrics and the case of R = 90% via a violin plot which favors the comparison of the distribution of the achieved values per algorithm, including also some important statistical quantities: median, interquartile range, and 1.5× interquartile range (Figure 2). Therefore, we can deduce experimentally the success of the proposed approach, especially when generic binary datasets constitute the main issue to be tackled when the collected labeled instances are highly numerically restricted.

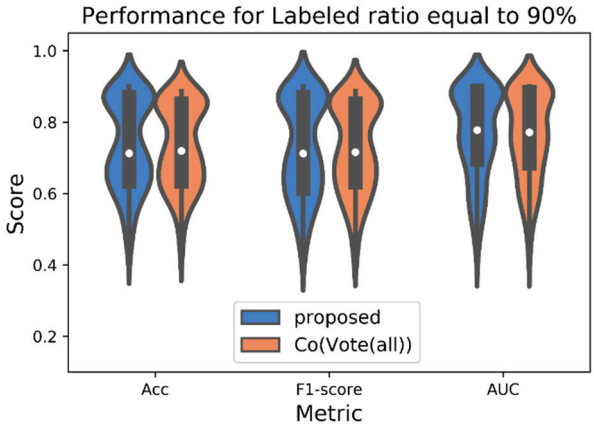

Figure 2. Violin plots of the proposed algorithm against $Co(Vote(all))$ approach over the three examined metrics.

Table 3. Classification accuracy (±stdev) values for the best five variants (labeled ratio 10%).

Dataset	Algorithms				
	$Co(Vote^{SEC}_{soft})$	$Co(Vote(all))$	$Co(Self(all))$	$Co(LR)$	$Co(Vote(DT,LR))$
bands	**0.668 ± 0.082**	0.668 ± 0.051	0.649 ± 0.036	0.627 ± 0.036	0.657 ± 0.084
breast	0.679 ± 0.065	0.696 ± 0.056	**0.7 ± 0.038**	0.693 ± 0.056	0.686 ± 0.114
bupa	0.571 ± 0.052	0.603 ± 0.085	0.529 ± 0.041	0.537 ± 0.042	**0.591 ± 0.057**
chess	**0.966 ± 0.013**	0.948 ± 0.007	0.955 ± 0.008	0.947 ± 0.011	0.962 ± 0.01
colic.ORIG	0.759 ± 0.055	0.757 ± 0.044	0.724 ± 0.077	0.73 ± 0.053	**0.768 ± 0.036**
diabetes	0.773 ± 0.038	**0.777 ± 0.032**	0.738 ± 0.048	0.765 ± 0.03	0.681 ± 0.059
h-statlog	0.719 ± 0.036	0.7 ± 0.027	0.707 ± 0.051	**0.77 ± 0.034**	0.656 ± 0.08
kr-vs-kp	**0.966 ± 0.019**	0.947 ± 0.008	0.953 ± 0.008	0.946 ± 0.012	0.963 ± 0.012
mammographic	0.788 ± 0.019	**0.796 ± 0.023**	0.793 ± 0.024	0.787 ± 0.025	0.78 ± 0.039
monk-2	**1 ± 0**	0.918 ± 0.019	0.88 ± 0.047	0.784 ± 0.029	**1 ± 0**
pima	0.691 ± 0.02	0.683 ± 0.025	0.686 ± 0.038	**0.697 ± 0.032**	0.652 ± 0.033
saheart	0.715 ± 0.021	0.713 ± 0.067	0.702 ± 0.036	**0.728 ± 0.02**	0.672 ± 0.038
sick	0.976 ± 0.003	0.967 ± 0.005	0.963 ± 0.006	0.95 ± 0.004	**0.978 ± 0.004**
tic-tac-toe	0.804 ± 0.07	0.805 ± 0.038	0.735 ± 0.037	**0.824 ± 0.036**	0.796 ± 0.057
vote	0.914 ± 0.026	0.891 ± 0.026	**0.916 ± 0.024**	0.889 ± 0.02	0.889 ± 0.033
wdbc	0.958 ± 0.021	0.963 ± 0.01	**0.972 ± 0.012**	0.956 ± 0.021	0.942 ± 0.031
wisconsin	0.974 ± 0.006	**0.98 ± 0.01**	0.972 ± 0.011	0.97 ± 0.016	0.936 ± 0.039

Bold highlighted means the best value per dataset.

Table 4. F_1-score (±stdev) values for the best five variants (labeled ratio 10%).

Dataset	Algorithms				
	$Co(Vote^{SEC}_{soft})$	$Co(Vote(all))$	$Co(Vote(DT,LR))$	$Co(Vote(all))$	$Co(LR)$
bands	**0.646 ± 0.096**	0.632 ± 0.068	0.643 ± 0.082	0.603 ± 0.049	0.594 ± 0.028
breast	0.652 ± 0.07	0.664 ± 0.076	**0.67 ± 0.118**	0.669 ± 0.032	0.649 ± 0.072
bupa	0.554 ± 0.049	**0.588 ± 0.085**	0.586 ± 0.059	0.502 ± 0.042	0.512 ± 0.032
chess	**0.966 ± 0.013**	0.947 ± 0.007	0.962 ± 0.01	0.955 ± 0.008	0.947 ± 0.011
colic.ORIG	0.745 ± 0.056	0.761 ± 0.043	**0.762 ± 0.039**	0.708 ± 0.083	0.735 ± 0.049
diabetes	0.765 ± 0.05	**0.769 ± 0.034**	0.68 ± 0.054	0.723 ± 0.055	0.75 ± 0.042
h-statlog	0.717 ± 0.036	0.7 ± 0.027	0.653 ± 0.082	0.706 ± 0.052	**0.77 ± 0.034**
kr-vs-kp	**0.966 ± 0.019**	0.947 ± 0.008	0.963 ± 0.012	0.953 ± 0.008	0.946 ± 0.012
mammographic	0.786 ± 0.019	**0.795 ± 0.022**	0.779 ± 0.039	0.791 ± 0.025	0.785 ± 0.026
monk-2	**1 ± 0**	0.918 ± 0.019	**1 ± 0**	0.879 ± 0.047	0.78 ± 0.03
pima	0.673 ± 0.018	0.664 ± 0.027	0.652 ± 0.034	0.664 ± 0.033	**0.685 ± 0.03**
saheart	0.65 ± 0.034	0.7 ± 0.073	0.675 ± 0.038	0.688 ± 0.039	**0.716 ± 0.019**
sick	0.974 ± 0.003	0.962 ± 0.007	**0.976 ± 0.004**	0.959 ± 0.008	0.939 ± 0.006
tic-tac-toe	0.807 ± 0.069	0.805 ± 0.037	0.799 ± 0.056	0.733 ± 0.038	**0.822 ± 0.036**
vote	0.914 ± 0.026	0.891 ± 0.026	0.888 ± 0.033	**0.916 ± 0.024**	0.888 ± 0.02
wdbc	0.958 ± 0.021	0.963 ± 0.01	0.943 ± 0.031	**0.972 ± 0.012**	0.956 ± 0.02
wisconsin	0.974 ± 0.006	**0.98 ± 0.01**	0.935 ± 0.04	0.972 ± 0.011	0.969 ± 0.016

Bold highlighted means the best value per dataset.

Table 5. Classification accuracy (±stdev) values for the best five variants (labeled ratio 10%).

Dataset	Algorithms				
	$Co(Vote^{SEC}_{soft})$	$Co(Vote(DT,LR))$	$Co(Vote(all))$	$Co(Vote(5NN,LR))$	$Self(Vote(all))$
bands	0.716 ± 0.083	0.652 ± 0.079	**0.764 ± 0.063**	0.716 ± 0.052	0.69 ± 0.06
breast	**0.701 ± 0.077**	0.671 ± 0.105	0.652 ± 0.062	0.661 ± 0.071	0.621 ± 0.081
bupa	0.584 ± 0.09	0.567 ± 0.065	**0.602 ± 0.08**	**0.602 ± 0.08**	0.54 ± 0.082
chess	0.99 ± 0.004	**0.992 ± 0.004**	0.985 ± 0.007	0.981 ± 0.006	**0.992 ± 0.005**
colic.ORIG	**0.835 ± 0.044**	0.817 ± 0.029	0.722 ± 0.079	0.813 ± 0.026	0.762 ± 0.097
diabetes	0.835 ± 0.037	0.772 ± 0.028	**0.854 ± 0.011**	0.824 ± 0.035	0.801 ± 0.039
h-statlog	0.765 ± 0.05	0.779 ± 0.053	**0.802 ± 0.024**	0.73 ± 0.051	0.769 ± 0.033
kr-vs-kp	0.99 ± 0.004	0.991 ± 0.004	0.985 ± 0.007	0.979 ± 0.006	**0.992 ± 0.005**
mammographic	0.857 ± 0.016	**0.876 ± 0.021**	0.856 ± 0.015	0.87 ± 0.021	0.874 ± 0.016
monk-2	0.996 ± 0.005	**1 ± 0**	**1 ± 0**	0.956 ± 0.018	0.958 ± 0.025
pima	0.655 ± 0.041	**0.681 ± 0.031**	0.68 ± 0.02	0.647 ± 0.041	0.651 ± 0.041
saheart	0.784 ± 0.057	0.764 ± 0.033	0.772 ± 0.019	**0.791 ± 0.028**	0.767 ± 0.046
sick	0.956 ± 0.011	0.961 ± 0.013	0.904 ± 0.024	0.917 ± 0.011	**0.963 ± 0.025**
tic-tac-toe	0.883 ± 0.031	0.903 ± 0.036	0.852 ± 0.054	**0.914 ± 0.035**	0.804 ± 0.041
vote	0.963 ± 0.005	**0.964 ± 0.006**	0.953 ± 0.018	0.958 ± 0.007	0.963 ± 0.007
wdbc	**0.995 ± 0.003**	0.992 ± 0.006	0.993 ± 0.004	**0.995 ± 0.003**	0.993 ± 0.006
wisconsin	**0.998 ± 0.002**	0.995 ± 0.007	0.996 ± 0.003	0.997 ± 0.003	0.997 ± 0.002

Bold highlighted means the best value per dataset.

Table 6. Friedman Rankings for all examined algorithms and statistical importance based on Nemenyi post-hoc test.

Friedman Ranking					
Acc		**F_1-Score**		**AUC**	
Algorithm	**Rank**	**Algorithm**	**Rank**	**Algorithm**	**Rank**
$Co(Vote^{SEC}_{soft})$	9.96	$Co(Vote^{SEC}_{soft})$	10.43	$Co(Vote^{SEC}_{soft})$	10.43
Co(Vote(all))	11.59	*Co(Vote(all))*	11.29	*Co(Vote(DT,LR)*	11.33
Self(Vote(all))	13.19	*Co(Vote(DT,LR))*	12.81	*Co(Vote(all))*	12.88
Co(LR)	13.36	*Self(Vote(all))*	13.28	*Co(Vote(kNN,LR)*	13.17
Co(Vote(DT,LR))	13.89	*Co(LR)*	13.34	*Self(Vote(all))*	13.66
Co(Vote(SVM,LR))	14.13	*Co(DT)*	13.44	*Co(Vote(DT,GNB))*	14.17
Co(DT)	14.41	*Co(Vote(SVM,DT))*	13.54	*Self(Vote(DT,LR)*	14.28
Co(Vote(SVM,DT))	14.58	*Co(Vote(DT,GNB))*	13.64	*Co(Vote(SVM,LR)*	14.98
Co(Vote(DT,GNB))	14.58	*Self(Vote(DT,GNB))*	14.31	*Co(SVM)*	14.98
Co(Vote(kNN,LR))	14.92	*Co(Vote(kNN,DT))*	14.54	*Co(LR)*	14.98
Self(Vote(DT,GNB))	15.33	*Co(Vote(kNN,LR))*	15.06	*Self(Vote(DT,GNB))*	15.24
Self(LR)	15.44	*Self(DT)*	15.12	*Co(Vote(GNB,LR))*	15.29
Co(Vote(kNN,DT))	15.64	*Self(LR)*	15.31	*Self(LR)*	16.01
Self(Vote(SVM,LR))	16.02	*Self(Vote(kNN,DT))*	15.46	*Self(Vote(kNN,LR)*	16.59
Self(DT)	16.04	*Self(Vote(DT,LR))*	15.78	*Co(Vote(kNN,DT)*	16.74
Self(Vote(kNN,DT))	16.61	*Self(Vote(SVM,DT))*	16.33	*Co(Vote(kNN,GNB))*	16.81
Self(Vote(DT,LR))	16.91	*Co(Vote(SVM,LR))*	16.33	*Co(Vote(SVM,DT)*	16.84
Self(Vote(kNN,LR))	16.96	*Co(Vote(kNN,GNB))*	17.05	*Co(DT)*	16.84
Co(Vote(SVM,kNN))	17.49	*Self(Vote(kNN,LR))*	17.21	*Co(Vote(SVM,GNB))*	17.06
Self(Vote(SVM,DT))	17.56	*Self(Vote(SVM,LR))*	17.38	*Co(GNB)*	17.06
Co(Vote(kNN,GNB))	18.06	*Self(Vote(kNN,GNB))*	17.69	*Self(Vote(GNB,LR))*	17.59
Co(kNN)	18.18	*Co(kNN)*	17.95	*Self(Vote(SVM,LR)*	18.09
Self(Vote(kNN,GNB))	18.34	*Co(Vote(SVM,kNN))*	19.20	*Self(SVM)*	18.09
Co(Vote(SVM,GNB))	19.45	*Co(Vote(SVM,GNB))*	19.59	*Self(Vote(kNN,GNB))*	18.42
Self(Vote(SVM,kNN))	19.46	*Self(Vote(SVM,GNB))*	19.83	*Self(Vote(kNN,DT)*	18.54
Self(Vote(SVM,GNB))	20.01	*Self(Vote(GNB,LR))*	19.88	*Self(Vote(SVM,GNB))*	18.62
Self(Vote(GNB,LR))	20.29	*Co(Vote(GNB,LR))*	20.02	*Co(Vote(SVM,kNN)*	19.41
Co(Vote(GNB,LR))	20.43	*Self(kNN)*	21.06	*Co(kNN)*	19.41
Co(SVM)	20.74	*Self(GNB)*	21.15	*Self(GNB)*	19.52
Self(kNN)	20.85	*Self(Vote(SVM,kNN))*	21.61	*Self(Vote(SVM,DT)*	21.03
Self(SVM)	21.89	*Co(GNB)*	21.84	*Self(Vote(SVM,kNN))*	22.62
Self(GNB)	22.14	*Co(SVM)*	24.21	*Self(kNN)*	24.67
Co(GNB)	22.56	*Self(SVM)*	25.33	*Self(DT)*	25.67

Adoption of more dedicated preprocessing stages oriented towards more specific problems should be applied, in order to boost the performance of the SSoftEC strategy and provide the Co-training scheme a more appropriate base learner [58–60]. However, in our generic experimental stage, which covers various applications, the proposed algorithm recorded a both robust and accurate enough performance, especially in the case of the F_1-score metric which is critical for real problems with class distribution different from the optimal, under a computational inexpensive manner, in contrast with DEC strategies that employ a new classifier search per test instance. The smoothing of the decisions that are produced through the proposed soft-Voting ensemble seems to favor the exported decision profile, since a large number of decisions that were initially misclassified based on individual predictions were reverted towards the ground truth label. While at the same time, numerous cases where the two participants disagree over the binary label were not affected. This happens because a large correct confidence value combined with a smaller incorrect one remains untouched under such a voting scheme, according to Equation (3).

5. Conclusions

In the present study, a soft-Voting ensemble-based co-training scheme through a Static Selection strategy operating under a random feature split, called $Co(Vote^{SEC}_{soft})$, was presented regarding binary classification problems. The proposed algorithm harnesses the benefits of the SSL approach and the ensemble methods that are built using heterogeneous approaches [61–63]. The experimental results using 27 benchmark datasets demonstrate the prevalence of the proposed algorithm in terms of classification accuracy and F_1-score, compared to several co-training and self-training variants, while using five different labeled ratios. Thus, the employment of a static selection classifier that tries to find a suitable combination among five provided classifiers for feeding appropriately an ensemble scheme seems that has favored the final predictive ability, without consuming much computational resources during the preprocess step per different task. This was successfully provoked by using soft-Voting strategy: for each class, the maximum averaged confidence is exported as the most prominent, taking into consideration the corresponding confidence values of both exploited learners.

Regarding our initial ambition, Co-training scheme seems more favorable for exploiting unlabeled instances and augmenting the initially collected labeled instances through the most reliable of the former, even during small labeled-ratio conditions, against Self-training approach. Furthermore, since the source and the structure of our examined datasets vary, application of the proposed method under more well-defined fields/tasks could be benefited by more advanced preprocessing stages, such as feature engineering, or tuning of participant learners, while specific criteria could be defined so as to avoid random split and converge to a more suitable feature split [40,64]. Increasing the cardinality and the diversification of the candidate classifiers should also be examined in following research, and especially the case when more strong classifiers are available, since their decision profile might demand a weighting soft or hard Voting scheme. In any case, the fact that unlabeled examples may boost the contribution of ensemble learners and their Selection strategies under SSL schemes seems to hold our assumption through our experimental stage [65]. Such strategies could also be used for selecting appropriate learners under more sophisticated ensemble structures like Stacking [66].

One interesting point, especially in the case that such a co-training algorithm should be applied to datasets that are characterized by more intense imbalanced classes, is the adoption of either more specified preprocess stages, such as the use of SMOTE algorithm, a well-known oversampling method that generates new instances or any of its descendants [67] or the embedding of similar methods inside the operation of base learner (s). Such an approach has been demonstrated recently in [68], where Rotation Forest, a popular ensemble learner based also on DTs, is combined with an under-sampling method so as to tackle this kind of issue. Regarding both the produced results and the fact that in semi-supervised scenarios the amount of collected data is much more restricted than the default supervised case, more sophisticated approaches for avoiding imbalanced datasets during the initialization of base learner, could be proven really promising for acquiring better learning rates [69]. Another interesting point is the expansion of the proposed scheme on the multi-class semi-supervised classification problem, such as in [70] where a new loss function was applied using gradient descent in functional space.

Moreover, appropriate experiments should be made towards the direction of recognizing the importance of the included features per view or among all the provided features, in case that random feature split is applied instead of merging all the distinct views into a compact but still heterogeneous view, so as to either propose a more detailed strategy for formatting the two separate views or applying feature reduction techniques that may favor the final predictive performance [64]. The "absent levels" problem [39] should also be studied under a SSL scenario, avoiding constructing views or preprocess feature sets that may lead to more implicit approaches, especially in real-life situations that the interpretability of the exported predictive model is of high priority for the business part or the corresponding field that is connected with the examined problem [71,72]. Construction of artificially generated data could be a safe strategy for excluding such conclusion. Otherwise, application to real-world data from totally different domains might infer biased decisions regarding the applicability

to a wider range of datasets. Transfer learning, combined probably with Active Learning framework that permits the knowledge blending of human factors into the learning pipeline, could prove to be a valuable approach for exporting more robust classifiers by enriching the feature vector and/or applying more compatible modifications [73].

Finally, deep neural networks (DNNs) [74] could be employed under a co-training scheme to boost the predictive performance, fed with either raw data or other generic kinds of datasets. More specifically, long short term memory (LSTM) networks have already proven efficient enough when combined with SSL methods for constructing clinical support decision systems [75]. In case DNNs should be exploited, creation of new insights into inserted data could take place, providing either totally new view (s) or augmenting the existing one (s). Thus, several feature engineering approaches should be adopted to enhance the quality of the co-training scheme and possibly violate the assumption about the independent views less.

Author Contributions: Conceptualization, S.K. (Stamatis Karlos); methodology, S.K. (Stamatis Karlos), G.K. and S.K. (Sotiris Kotsiantis); software, S.K. (Stamatis Karlos); validation, S.K. (Stamatis Karlos), G.K. and S.K. (Sotiris Kotsiantis); formal analysis, G.K. and S.K. (Sotiris Kotsiantis); investigation, S.K. (Stamatis Karlos); resources, S.K. (Stamatis Karlos), G.K. and S.K. (Sotiris Kotsiantis); data curation, S.K. (Stamatis Karlos); writing—original draft preparation, S.K. (Stamatis Karlos), G.K.; writing—review and editing, S.K. (Stamatis Karlos), G.K. and S.K. (Sotiris Kotsiantis); visualization, S.K. (Stamatis Karlos); supervision, S.K. (Sotiris Kotsiantis); project administration, S.K. (Stamatis Karlos); funding acquisition, S.K. (Sotiris Kotsiantis). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Schwenker, F.; Trentin, E. Pattern classification and clustering: A review of partially supervised learning approaches. *Pattern Recognit. Lett.* **2014**, *37*, 4–14. [CrossRef]
2. Kim, A.; Cho, S.-B. An ensemble semi-supervised learning method for predicting defaults in social lending. *Eng. Appl. Artif. Intell.* **2019**, *81*, 193–199. [CrossRef]
3. Li, J.; Wu, S.; Liu, C.; Yu, Z.; Wong, H.-S. Semi-Supervised Deep Coupled Ensemble Learning With Classification Landmark Exploration. *IEEE Trans. Image Process.* **2020**, *29*, 538–550. [CrossRef] [PubMed]
4. Kostopoulos, G.; Karlos, S.; Kotsiantis, S.; Ragos, O. Semi-supervised regression: A recent review. *J. Intell. Fuzzy Syst.* **2018**, *35*, 1483–1500. [CrossRef]
5. Alpaydin, E. *Introduction to Machine Learning*; MIT Press: Cambridge, MA, USA, 2010.
6. Ng, V.; Cardie, C. Weakly supervised natural language learning without redundant views. In Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Edmonton, AB, Canada, 27 May–1 June 2003.
7. Blum, A.; Mitchell, T. Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory—COLT' 98, New York, NY, USA, 24–26 July 1998; pp. 92–100.
8. Zhou, Z.-H.; Li, M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 1529–1541. [CrossRef]
9. Zhu, X.; Goldberg, A.B. *Introduction to Semi-Supervised Learning*; Morgan & Claypool Publishers: Williston, VN, USA, 2009.
10. Zhou, Z.-H. *Ensemble Methods: Foundations and Algorithms*; Taylor & Francis: Abingdon, UK, 2012.
11. Zhou, Z.-H. When semi-supervised learning meets ensemble learning. *Front. Electr. Electron. Eng. China* **2011**, *6*, 6–16. [CrossRef]
12. Sinha, A.; Chen, H.; Danu, D.G.; Kirubarajan, T.; Farooq, M. Estimation and decision fusion: A survey. *Neurocomputing* **2008**, *71*, 2650–2656. [CrossRef]
13. Wu, Y.; He, J.; Man, Y.; Arribas, J.I. Neural network fusion strategies for identifying breast masses. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), Budapest, Hungary, 25–29 July 2004; pp. 2437–2442.

14. Wu, Y.; Arribas, J.I. Fusing output information in neural networks: Ensemble performs better. In Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439), Cancun, Mexico, 17–21 September 2003; pp. 2265–2268.
15. Livieris, I.; Kanavos, A.; Tampakas, V.; Pintelas, P. An auto-adjustable semi-supervised self-training algorithm. *Algorithms* **2018**, *11*, 139. [CrossRef]
16. Britto, A.S.; Sabourin, R.; Oliveira, L.E.S. Dynamic selection of classifiers—A comprehensive review. *Pattern Recognit.* **2014**, *47*, 3665–3680. [CrossRef]
17. Hou, C.; Xia, Y.; Xu, Z.; Sun, J. Semi-supervised learning competence of classifiers based on graph for dynamic classifier selection. In Proceedings of the IEEE 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 3650–3654.
18. Jiang, Z.; Zhang, S.; Zeng, J. A hybrid generative/discriminative method for semi-supervised classification. *Knowl. Based Syst.* **2013**, *37*, 137–145. [CrossRef]
19. Ceci, M.; Pio, G.; Kuzmanovski, V.; Džeroski, S. Semi-supervised multi-view learning for gene network reconstruction. *PLoS ONE* **2015**, *10*, e0144031. [CrossRef] [PubMed]
20. Zhu, X.; Goldberg, A.B. Introduction to Semi-Supervised Learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2009**, *3*, 1–130. [CrossRef]
21. Nigam, K.; Ghani, R. Analyzing the effectiveness and applicability of co-training. In Proceedings of the Ninth International Conference on Information and Knowledge Management, New York, NY, USA, 6–11 November 2000; pp. 86–93. [CrossRef]
22. Yu, N. Exploring C o-training strategies for opinion detection. *J. Assoc. Inf. Sci. Technol.* **2014**, *65*, 2098–2110. [CrossRef]
23. Lin, W.-Y.; Lo, C.-F. Co-training and ensemble based duplicate detection in adverse drug event reporting systems. In Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, Shanghai, China, 18–21 December 2013; pp. 7–8.
24. Culp, M.; Michailidis, G. A co-training algorithm for multi-view data with applications in data fusion. *J. Chemom.* **2009**, *23*, 294–303. [CrossRef]
25. Wehrens, R.; Mevik, B.-H. The pls package: Principal component and partial least squares regression in R. *J. Stat. Softw.* **2007**, *18*, 1–23. [CrossRef]
26. 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]
27. Li, M.; Zhou, Z.-H. 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]
28. Deng, C.; Guo, M.Z. A new co-training-style random forest for computer aided diagnosis. *J. Intell. Inf. Syst.* **2011**, *36*, 253–281. [CrossRef]
29. Liu, C.; Yuen, P.C. A Boosted Co-Training Algorithm for Human Action Recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 1203–1213. [CrossRef]
30. Kostopoulos, G.; Karlos, S.; Kotsiantis, S.B. Multi-view Learning for Early Prognosis of Academic Performance: A Case Study. *IEEE Trans. Learn. Technol.* **2019**, *12*, 212–224. [CrossRef]
31. Pio, G.; Malerba, D.; D'Elia, D.; Ceci, M. Integrating microRNA target predictions for the discovery of gene regulatory networks: A semi-supervised ensemble learning approach. *BMC Bioinform.* **2014**, *15*, S4. [CrossRef] [PubMed]
32. Dietterich, T.G. Ensemble Methods in Machine Learning. *Mult. Classif. Syst.* **2000**, *1857*, 1–15. [CrossRef]
33. Bolón-Canedo, V.; Alonso-Betanzos, A. *Recent Advances in Ensembles for Feature Selection*; Springer International Publishing: Cham, Switzerland, 2018.
34. Azizi, N.; Farah, N. From static to dynamic ensemble of classifiers selection: Application to Arabic handwritten recognition. *Int. J. Knowl. Based Intell. Eng. Syst.* **2012**, *16*, 279–288. [CrossRef]
35. Mousavi, R.; Eftekhari, M.; Rahdari, F. Omni-Ensemble Learning (OEL): Utilizing Over-Bagging, Static and Dynamic Ensemble Selection Approaches for Software Defect Prediction. *Int. J. Artif. Intell. Tools* **2018**, *27*, 1850024. [CrossRef]
36. Tsoumakas, G.; Partalas, I.; Vlahavas, I. An Ensemble Pruning Primer. *Appl. Supervised Unsupervised Ensemble Methods* **2009**, *245*, 1–13. [CrossRef]

37. Cruz, R.M.O.; Sabourin, R.; Cavalcanti, G.D.C. Analyzing different prototype selection techniques for dynamic classifier and ensemble selection. In Proceedings of the IEEE 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 3959–3966.
38. Zhao, J.; Xie, X.; Xu, X.; Sun, S. Multi-view learning overview: Recent progress and new challenges. *Inf. Fusion* **2017**, *38*, 43–54. [CrossRef]
39. Au, T.C. Random Forests, Decision Trees, and Categorical Predictors: The "Absent Levels" Problem. *J. Mach. Learn. Res.* **2018**, *19*, 1–30.
40. Ling, C.X.; Du, J.; Zhou, Z.-H. When does Co-training Work in Real Data? *Adv. Knowl. Discov. Data Min. Proc.* **2009**, *5476*, 596–603.
41. Ni, Q.; Zhang, L.; Li, L. A Heterogeneous Ensemble Approach for Activity Recognition with Integration of Change Point-Based Data Segmentation. *Appl. Sci.* **2018**, *8*, 1695. [CrossRef]
42. Platt, J.C. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Adv. Large Margin Classif.* **1999**, *10*, 61–74.
43. Garcia, E.K.; Feldman, S.; Gupta, M.R.; Srivastava, S. Completely lazy learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1274–1285. [CrossRef]
44. Loh, W.-Y. Classification and regression trees. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 14–23. [CrossRef]
45. Zheng, F.; Webb, G. A comparative study of semi-naive Bayes methods in classification learning. In Proceedings of the 4th Australas Data Mining Conference AusDM05 2005, Sydney, Australia, 5–6 December 2005; pp. 141–156.
46. Samworth, R.J. Optimal weighted nearest neighbour classifiers. *arXiv* **2011**, arXiv:1101.5783v3. [CrossRef]
47. Giacinto, G.; Roli, F. Design of effective neural network ensembles for image classification purposes. *Image Vis. Comput.* **2001**, *19*, 699–707. [CrossRef]
48. Wang, W.; Zhou, Z.-H. Theoretical Foundation of Co-Training and Disagreement-Based Algorithms. *arXiv* **2017**, arXiv:1708.04403.
49. Dua, D.; Graff, C. UCI Machine Learning Repository. Available online: http://archive.ics.uci.edu/ml (accessed on 1 November 2019).
50. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning: with Applications in R*; Springer: New York, NY, USA, 2013.
51. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
52. Chang, C.; Lin, C. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–39. [CrossRef]
53. Aha, D.W.; Kibler, D.; Albert, M.K. Instance-Based Learning Algorithms. *Mach. Learn.* **1991**, *6*, 37–66. [CrossRef]
54. Rish, I. An empirical study of the naive Bayes classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Seattle, WA, USA, 4–6 August 2001; pp. 41–46.
55. Sperandei, S. Understanding logistic regression analysis. *Biochem. Medica* **2014**, *24*, 12–18. [CrossRef]
56. Hodges, J.L.; Lehmann, E.L. Rank methods for combination of independent experiments in analysis of variance. *Ann. Math. Stat.* **1962**, *33*, 482–497. [CrossRef]
57. Hollander, M.; Wolfe, D.A.; Chicken, E. *Nonparametric Statistical Methods*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2014.
58. Kumar, G.; Kumar, K. The Use of Artificial-Intelligence-Based Ensembles for Intrusion Detection: A Review. *Appl. Comput. Intell. Soft Comput.* **2012**, *2012*, 850160. [CrossRef]
59. Karlos, S.; Kaleris, K.; Fazakis, N. Optimized Active Learning Strategy for Audiovisual Speaker Recognition. In Proceedings of the 20th International Conference on Speech and Computer SPECOM 2018, Leipzig, Germany, 18–22 September 2018; pp. 281–290.
60. Tencer, L.; Reznakova, M.; Cheriet, M. Summit-Training: A hybrid Semi-Supervised technique and its application to classification tasks. *Appl. Soft Comput. J.* **2017**, *50*, 1–20. [CrossRef]
61. Tanha, J.; van Someren, M.; Afsarmanesh, H. Semi-supervised self-training for decision tree classifiers. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 355–370. [CrossRef]

62. Chapelle, O.; Schölkopf, B.; Zien, A. Metric-Based Approaches for Semi-Supervised Regression and Classification. In *Semi-Supervised Learning*; MIT Press: Cambridge, MA, USA, 2006; pp. 420–451.
63. Wainer, J. Comparison of 14 different families of classification algorithms on 115 binary datasets. *arXiv* **2016**, arXiv:1606.00930.
64. Yaslan, Y.; Cataltepe, Z. Co-training with relevant random subspaces. *Neurocomputing* **2010**, *73*, 1652–1661. [CrossRef]
65. Zhang, M.-L.; Zhou, Z.-H. Exploiting unlabeled data to enhance ensemble diversity. *Data Min. Knowl. Discov.* **2013**, *26*, 98–129. [CrossRef]
66. Karlos, S.; Fazakis, N.; Kotsiantis, S.; Sgarbas, K. Self-Trained Stacking Model for Semi-Supervised Learning. *Int. J. Artif. Intell. Tools* **2017**, *26*. [CrossRef]
67. Barua, S.; Islam, M.M.; Yao, X.; Murase, K. MWMOTE–Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 405–425. [CrossRef]
68. Guo, H.; Diao, X.; Liu, H. Embedding Undersampling Rotation Forest for Imbalanced Problem. *Comput. Intell. Neurosci.* **2018**, *2018*, 6798042. [CrossRef]
69. Vluymans, S. Learning from Imbalanced Data. In *Dealing with Imbalanced and Weakly Labelled Data in Machine Learning Using Fuzzy and Rough Set Methods. Studies in Computational Intelligence*; Springer: Cham, Switzerland, 2019; pp. 81–110.
70. Tanha, J. MSSBoost: A new multiclass boosting to semi-supervised learning. *Neurocomputing* **2018**, *314*, 251–266. [CrossRef]
71. Chuang, C.-L. Application of hybrid case-based reasoning for enhanced performance in bankruptcy prediction. *Inf. Sci.* **2013**, *236*, 174–185. [CrossRef]
72. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD'16, San Francisco, CA, USA, 13–17 August 2016; ACM Press: New York, NY, USA, 2016; pp. 1135–1144.
73. Kale, D.; Liu, Y. Accelerating Active Learning with Transfer Learning. In Proceedings of the 2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 7–10 December 2013; pp. 1085–1090. [CrossRef]
74. Nielsen, M.A. Neural Networks and Deep Learning. 2015. Available online: http://neuralnetworksanddeeplearning.com (accessed on 1 November 2019).
75. Chen, D.; Che, N.; Le, J.; Pan, Q. A co-training based entity recognition approach for cross-disease clinical documents. *Concurr. Comput. Pract. Exp.* **2018**, e4505. [CrossRef]