Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 279 – 294
© TÜBİTAK
doi:10.3906/elk-2011-41

Temporal bagging: a new method for time-based ensemble learning

Göksu TÜYSÜZOĞLU*, **Derya BİRANT**, **Volkan KIRANOĞLU**
Department of Computer Engineering, Faculty of Engineering, Dokuz Eylül University, İzmir, Turkey

Received: 09.11.2020 • **Accepted/Published Online:** 03.05.2021 • **Final Version:** 19.01.2022

Abstract: One of the main problems associated with the bagging technique in ensemble learning is its random sample selection in which all samples are treated with the same chance of being selected. However, in time-varying dynamic systems, the samples in the training set have not equal importance, where the recent samples contain more useful and accurate information than the former ones. To overcome this problem, this paper proposes a new time-based ensemble learning method, called *temporal bagging (T-Bagging)*. The significant advantage of our method is that it assigns larger weights to more recent samples with respect to older ones, so it reduces the selection chances of former samples, and, thus, it addresses the adaptation to changes in dynamic systems. The experiments show that the proposed T-Bagging method improves the prediction accuracy of the model compared to the standard bagging method on temporal data.

Key words: Machine learning, ensemble learning, bagging, temporal data, support vector machines

1. Introduction

Ensemble learning is a machine learning strategy based on the common ideas of multiple learners to overcome divergence and bias issues faced by a single learner. The frequently used ensemble learning techniques are bagging, boosting, voting, and stacking. *Bagging* is a powerful method that aims to train a model for each resampled training data where sampling is done randomly with replacement by obtaining the same number of instances with the original training set in each ensemble iteration.

The standard bagging method [1] has a random sample selection phase, which is called bootstrapping. When a purely random sample selection is performed, all samples are treated with the same chance of being selected. However, in time-varying dynamic systems, the recent samples consist of more up-to-date and eligible information compared to older ones because changes are observed over time. Several examples can be given as follows: climate changes due to global warming, changes in the traffic due to urbanization, changes in manufacturing equipment caused by aging, and the subtle changes in health due to changes in the environment. The presence of many past samples in the training set may mislead the algorithm to produce up-to-date patterns, leading to prediction errors. To deal with this problem, this paper proposes a new time-based ensemble learning method, called *temporal bagging (T-Bagging)*. The proposed method constructs a set of classifiers by using a weighted random sample selection strategy. Our method reduces the chances of former (less concerned) sample selection. The main advantage of our method is that it can adapt accurately to recent changes in dynamic systems since the recent samples are weighted more heavily.

The main contributions of this study to the literature can be summarized as follows. (i) It proposes a novel ensemble learning strategy, *T-Bagging*, which is a modified version of bagging. (ii) This study is also

*Correspondence: goksutuysuzoglu@gmail.com

This work is licensed under a Creative Commons Attribution 4.0 International License.

original in that it investigates the impact of weight method. (iii) This is the first study that analyzes the type and size of the feature selection method used for bagging on temporal data.

In this study, we conducted several experiments on 12 publicly available temporal datasets to illustrate the efficiency and validity of the proposed method. The experimental results showed that the proposed T-Bagging method obtained more accurate values than the standard bagging method and some other methods that have been successfully applied on temporal data, including ARIMAX (autoregressive integrated moving average with exogenous variable) [2] and temporal classification method proposed in [3].

The organization of the paper is as follows. Section 2 expresses the recent studies on updated versions of bagging and ensemble learning strategies considering time-varying dynamic systems. Section 3 comprehensively talks about the proposed *temporal bagging (T-Bagging)* method. The dataset definition and experimental studies are stated in Section 4. Lastly, the concluding remarks and future works are given in Section 5.

2. Related work

Bagging has been widely used in many fields such as health [4], finance [5], social network analysis [6], and environment [7]. However, the standard bagging procedure [1] is not effective in handling temporal data since it builds a prediction model based on randomly chosen samples. To overcome this limitation, in this paper, we correspondingly changed the "random instance sampling" strategy in the standard bagging method to the "weight-based random instance sampling" strategy.

In order to improve the capability of the original bagging, various modifications have been made by presenting a new method such as A-Bagging (an adaptive version of bagging) [8], PCA-DC-Bagging (bagging algorithm based on principal component analysis and decision-making committee) [9], MTSBag (the integration of Mahalanobis–Taguchi system and bagging) [10], BagGMM (bagging of multiple Gaussian mixture models) [11], iBagging (incremental bagging) [12], FS-HB (feature selection-based hybrid bagging) [13], and OEBag (optimization embedded bagging) [14]. Table 1 shows the comparison of our study with the existing studies. Our method differs from them in filling the gap of prediction when the observed data is temporal and related to a time-based domain.

Temporal data is one of the most common forms of data which contains feature values of the events occurring over the continuous timeline. Temporal prediction is a task to estimate future values based on the previously known observations by considering the inherent sequence of occurrence. To correctly predict future events over temporal data, the existing temporal value should not be ignored. Several ensemble learning techniques are extended to consider the temporal value during machine learning such as temporal sampling forest [15], Bagged.ETS.MBB [16], and time series forest [17]. Table 2 points out their details. While some of the previous studies were implemented using the ensembles of exponential smoothing methods [16], some of them were implemented by developing a temporal sampling module in the attribute level [15]. New methods based on the specialties of random forest were introduced in some of the studies [15, 17]. Unlike previous studies, we propose a new method based on the rationale behind bagging by modifying the sample selection stage by giving weights to instances in the training set according to their up-to-dateness. All operations are implemented in instance-level instead of attribute level.

Table 1. Comparison of our study with the existing studies.

Ref	Year	Method	Applied technique	Application area
[8]	2020	A-Bagging	Clustering and then weighted majority voting	Information security
[9]	2020	PCA-DC-Bagging	Discriminant analysis based on the decision committee model	Manufacturing
[10]	2020	MTSBag	Combination of the Mahalanobis–Taguchi System (MTS) and bagging	Health care
[11]	2019	BagGMM	Bagging of multiple Gaussian mixture models (GMM)	Biomedicine
[12]	2019	iBagging	Incremental bagging with enhanced semi-random subspace selection	Information security
[14]	2018	OEBag	Employing a decision tree to learn the complex distributions of the minority class	Various domains
[13]	2017	FS-HB	Hybrid bagging using chi-square test and principal component analysis (PCA) for feature selection	Banking
Proposed method		T-Bagging	Bagging with a weighted random sample selection strategy for temporal data	Time-based domains

Table 2. Previous studies on temporal ensemble models.

Ref	Year	Method	Applied Technique	Application Data
[15]	2017	Temporal Sampling Forest	The random forest algorithm with resampling all features in a temporal fashion	Various domains
[16]	2016	Bagged.ETS. MBB	Bagging with exponential smoothing and moving block bootstrapping methods	Finance and industry
[17]	2013	Time Series Forest	Temporal ensemble model by using a splitting criterion and randomly sampling features	Various domains

3. Material and methods

3.1. T-Bagging: the proposed method

In the standard bagging, the samples of training sets in different iterations are chosen randomly by assuming them with equal importance. However, in time-varying dynamic systems, this could be problematic since the significant samples will not have enough chances to be selected in the first place when the number of available samples is huge such as millions. To overcome this problem, this paper proposes a new method, *temporal bagging* (*T-Bagging*) that takes more account of current instances by assigning them more weight values in the selection stage of bootstrapping. In this way, the recent samples are more likely to influence future decisions.

Motivated by the success of bagging, this study presents how the temporalization concept can be embedded in it with the purpose of extending its usage in temporal prediction. In our method, we assume that the informational relevance of samples decays exponentially over time.

Figure 1 presents an overview of the proposed T-Bagging method. In the first step, the proposed method computes a weight for each sample I_i, where i is ranged from 1 to n for n instances, with respect to their temporal values. A weight can be regarded as the probability of the sample to be selected to construct a prediction model. In the second step, the method randomly selects a subset of samples with a weight-based strategy, whereby the recent samples possess higher chances of selection. In the meantime, the randomness in the weight-based sampling ensures that each base learner in the ensemble will be trained on different instance subsets to provide diversity. Many training sets are generated from a single data. After that, each model is built on a bootstrapped sample of the original data. In the next step, test instances are given to each trained model to produce individual outputs of different iterations. Finally, the estimation of previously unseen data is made by majority voting on the predictions of individual models in the ensemble.

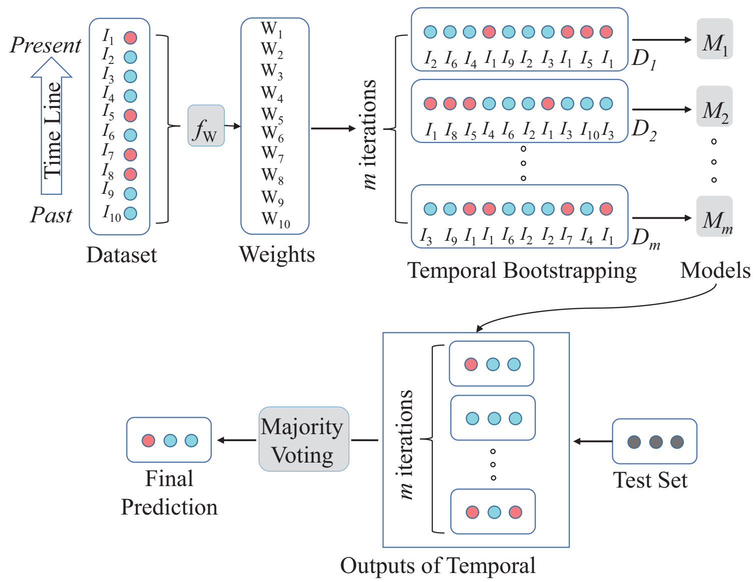

Figure 1. A general overview of the proposed T-Bagging method.

3.2. Formal definition

Given a dataset D with n samples $D = \{(t_1, x_1, y_1), (t_2, x_2, y_2), ..., (t_n, x_n, y_n)\}$ where each component (t, x, y) is composed of timestamp t with the ordering $t_1 < t_2 < ... < t_n$, a d-dimensional input vector $x = [A_1, A_2, ..., A_d]$, and a categorical label y. The weights $W = \{W_1, W_2, ..., W_n\}$ are computed for every sample in the dataset according to their timestamps. These weights are then utilized in the selection of the samples which will be used to build the models in the ensemble. Using temporal bootstrapping strategy, a number of training sets $\{D_1, D_2, ..., D_m\}$ are generated, and then each dataset D_i is used to construct a new classification model M_i. In other words, the method separately builds a number of classification models $M^* = \{M_1, M_2, ..., M_m\}$ on temporal bootstrapped samples. The final prediction is determined by applying majority voting on the label outputs of all models on the unseen input x, denoted by $M^*(x)$.

To add a temporal learning methodology into the bagging procedure, we developed a temporal bootstrapping scheme as defined in Definition 1.

Definition 1 ***(Temporal bootstrapping)*** *Temporal bootstrapping is a special type of bootstrapping, whereby a subset of n samples (D_i) is randomly selected with replacement from the original temporal dataset D with n samples by considering their weights (W).*

In the due course of adoption of temporal bootstrapping, we expect the models will be built by using recent samples; thus, an important prediction performance improvement could be attained through the ensemble of all the learners. We propose a new type of bagging to encourage the algorithm to give more credits to the more recent samples as defined in Definition 2.

Definition 2 ***(Temporal bagging)*** *The temporal bagging (T-Bagging) is a time-based ensemble learning technique that builds multiple models, denoted by* $M^* = \{M_1, M_2, \ldots, M_m\}$*, on temporal bootstrapped samples, and each model* M_i *casts a unit vote at given input x, and the most occurring class label among the ensemble outputs is selected to make the final decision.*

To allow a reasonable differentiation in a dynamic system, this paper introduces a novel weighting procedure that renders recent events more important than the past ones. A weight can be regarded as the probability of the sample to be selected to build a classification model. Equation (1) to Equation (7) present the formulas used throughout the experiments in this study for weight assignment, where d is a constant referring to the total time for data collection and t is the timestamp for a specific instance. Zhao [18] proposed nine different temporal weighting strategies. Our study differs from her study in several respects. First, the impact of some weighting strategies given in [18] is strong, i.e. the weights drop quickly; whereas, our weighting strategies show smooth variations in time. Second, when the time period is increased, the strategies given in [18] show abnormal trends. On the other hand, we designed the formulas by considering longer time periods. Third, Zhao [18] presented a solution for a specific problem (pharmacovigilance); whereas, we demonstrated the generalization ability of our weighting strategies on real-world datasets obtained from various domains. Our weighting strategies follow a k-degree polynomial function of t, where k = 1.5, 2.5, and 4.

$$\mathrm{WF}_1 = (d - t)/d \tag{1}$$

$$\mathrm{WF}_2 = (d - t)^{1.5}/d^{1.5} \tag{2}$$

$$\mathrm{WF}_3 = (d - t)^{2.5}/d^{2.5} \tag{3}$$

$$\mathrm{WF}_4 = (d - t)^{4}/d^{4} \tag{4}$$

$$\mathrm{WF}_5 = 1 - (t^{1.5}/d^{1.5}) \tag{5}$$

$$\mathrm{WF}_6 = 1 - (t^{2.5}/d^{2.5}) \tag{6}$$

$$\mathrm{WF}_7 = 1 - (t^{4}/d^{4}) \tag{7}$$

The applied weighting strategies are displayed in Figure 2. The lines express the influence of the weighting strategy on a new sample. Weights decrease from present to past records. They are ordered by the left to right in terms of their strength in decreasing order. The lines under diagonal (i.e., WF_2 to WF_4) demonstrates a quicker drop over time compared to the upper side (i.e., WF_5 to WF_7), which means the recent samples have a more significant impact on the class label prediction of a test case compared to earlier samples.

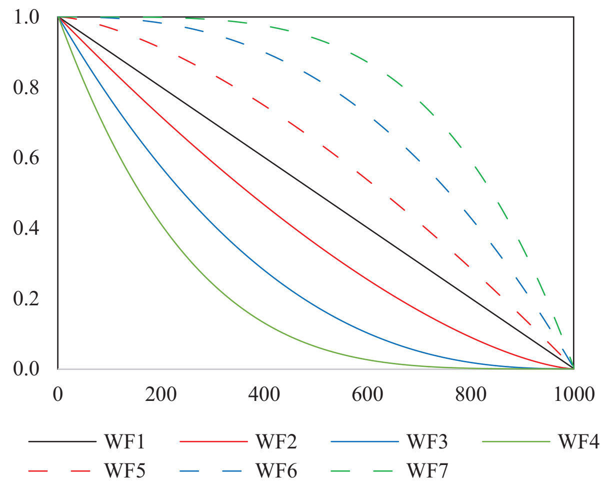

Figure 2. Alternative weighting strategies.

3.3. An example of the proposed method

Table 3 shows a simple example to demonstrate the temporal bootstrapping phase of the T-Bagging. To illustrate, we applied the method for one of the weighting strategies (WF_2) to a small dataset containing 10 instances. Other weighting strategies are also implemented in the same way. The cumulative weight value close to zero means that the corresponding sample is meaningless for prediction; thus, eliminating this sample increases the accuracy of the estimation. The recorded timestamps of the samples are given as $t_1 > t_2 > ... > t_{10}$, where t_1 represents the former and t_{10} represents the recent timestamp. According to Equation (2), the weight values of the samples are calculated as shown in Table 3. The next step is to obtain the cumulative weight of each sample by taking the sum of its own weight and the weight values of all previous samples. After that, the bootstrapping iterations are performed to randomly select 10 instances with replacement by taking cumulative weights into consideration. Table 4 demonstrates the temporal randomly sampling (*temporal bootstrapping*). A new random number is generated in each selection step. The first cumulative weight larger than the generated random number is looked for to select the respective sample. For example, when the random number is 0.61, the first cumulative weight greater than 0.61 is 0.8919 which is the weight of sample 5 in the timestamp t_5. As a result, the 5^{th} sample is selected in this step. It is also apparent in the given example that the possibility of selection is higher for recent samples compared to former ones because cumulative weight ranges get larger from past to present.

3.4. T-Bagging algorithm

Algorithm 1 presents the pseudo-code of the Temporal Bagging method. In the first loop, a selected weighting strategy is applied, and the cumulative total is assigned to each sample. In this way, the most recent sample is given by the highest weight. A cumulative weighting list is the output of this loop. In the next step of the algorithm, a random number is generated for n times to select samples for the creation of a new training set for each ensemble iteration. Since the recent samples have the largest range in terms of weight, their chances of selection are more compared to former ones. Hence, the more recent the sample is relevant and non-redundant, the higher it is used in the model construction. In the main loop, the algorithm builds m models on temporal bootstrapped samples. At the i^{th} iteration, the dataset D_i, which contains temporal bootstrapped samples, is

Table 3. A sample weight calculation step of the proposed method.

Sample ID	Time	Weight Calculation	Weight	Cumulative Weight
1	t_1 (former)	$(10\text{-}9)^{1.5}$ / $10^{1.5}$	0.0316	0.0316
2	t_2	$(10\text{-}8)^{1.5}$ / $10^{1.5}$	0.0894	0.1210
3	t_3	$(10\text{-}7)^{1.5}$ / $10^{1.5}$	0.1643	0.2853
4	t_4	$(10\text{-}6)^{1.5}$ / $10^{1.5}$	0.2530	0.5383
5	t_5	$(10\text{-}5)^{1.5}$ / $10^{1.5}$	0.3536	0.8919
6	t_6	$(10\text{-}4)^{1.5}$ / $10^{1.5}$	0.4648	1.3567
7	t_7	$(10\text{-}3)^{1.5}$ / $10^{1.5}$	0.5857	1.9424
8	t_8	$(10\text{-}2)^{1.5}$ / $10^{1.5}$	0.7155	2.6579
9	t_9	$(10\text{-}1)^{1.5}$ / $10^{1.5}$	0.8538	3.5117
10	t_{10} (recent)	$(10\text{-}0)^{1.5}$ / $10^{1.5}$	1	4.5117

Table 4. Temporal randomly sampling (*temporal bootstrapping*).

Random number	Comparison	Selected sample	Selected timestamp									
			former									*recent*
0.61	0.61 < 0.8919	5					■					
0.15	0.15 < 0.2853	3			■							
1.84	1.84 < 1.9424	7							■			
2.50	2.50 < 2.6579	8								■		
2.26	2.26 < 2.6579	8								■		
1.68	1.68 < 1.9424	7							■			
1.21	1.21 < 1.3567	6						■				
3.10	3.10 < 3.5117	9									■	
4.20	4.20 < 4.5117	10										■
2.46	2.46 < 2.6579	8								■		

used to build the i^{th} model M_i, which is added to the ensemble M^*. Finally, in order to classify an unseen instance x, the constructed models under M^* are used and each one estimates a class label for the specified sample. The final class label is then assigned by taking the joint decision on the outputs of M^* via majority voting.

The time complexity of the proposed T-Bagging algorithm is $O(m\ .\ L(n) + T)$, where m is the number of learners (ensemble size), n is the number of instances in the dataset, L is the time required for the execution of a learning algorithm on n instances, and T represents the time needed for the temporal sampling process.

3.5. The advantages of t-bagging

A number of advantages are valid for both bagging and the proposed method, called T-Bagging. First, both of them can be used with the combination of any base learner such as support vector machines, k-nearest neighbors, and neural networks. They are entirely unaware of the classification method. In this way, T-Bagging

Algorithm 1: Temporal Bagging (T-Bagging)

Inputs :

D: The temporal dataset $D = \{(t_1, x_1, y_1), (t_2, x_2, y_2), ..., (t_n, x_n, y_n)\}$
D_{train}: Training set
D_{test}: Test set
WF: Weight assignment formula
m: The number of models (ensemble size)
n: The number of instances in the training set

Output:

M^*: A collection of models in the ensemble
$M^*(x)$: Predicted class label in a new sample x

```
begin
    cumulative = 0
    M* = ∅
    for i = 1 to n do
        weight = WF(i)
        cumulative = cumulative + weight
        W_i = cumulative
    end
    for i = 1 to m do
        for j = 1 to n do
            rnd = GenerateRandNum(0, W_n)
            for k = 1 to n do
                if rnd <= W_k then
                    D_i(j) = D_train(k)
                    break
                end
            end
        end
        M_i = Training(D_i)
        M* = M* ∪ M_i
    end
    M*(x) = Voting(M_1(x), M_2(x), ..., M_m(x))
    = argmax_y Σ_{i:y=M_i(x)}^{m} 1
end
```

simply learns from temporal data. Furthermore, their implementation simplicity makes them favorable to be applied, for instance, the temporal bootstrapping step of T-Bagging is easy to implement. Both of them have advantages over single models since multiple predictors are combined to improve generalizability, accuracy, and robustness. Besides, they can handle input features with different types (i.e., categorical, numerical).

In addition to these similarities, the advantages of the proposed method that differ from the traditional bagging method are as follows:

- It takes into account the timeliness of the data while making a prediction for a future sample.
- It extends the classical bagging method with the ability to handle temporal data. In this way, it expands the usage of bagging.
- It can be applied to any temporal data without any prior information about the dataset. It does not make

any specific assumptions and specific knowledge for the given data.

- It is easy to deploy temporal sampling since its parameters can be set easily.
- It is easily parallelized to speed up the training phase when temporal bootstrapping is applied. Data can be distributed over many nodes in a time-based manner.

3.6. Feature selection

Feature selection is a very critical component in machine learning studies by reducing the dimensions without much loss of the total information and decreasing the training time and the risk of overfitting resulting from the increasing number of features.In this study, four feature selection methods, namely information gain, gain ratio, correlation-based feature selection, and ReliefF, were used to investigate the most informative attributes when T-Bagging was implemented.

Information Gain is one of the feature selection techniques that evaluates the gain of each feature in the context of a target variable. It is performed by taking the mutual information (i.e. the determination of the statistical dependence) between two random variables. It measures the reduction in entropy before and after splitting a dataset according to a given value of a random variable as expressed in Equation (8) where S is a dataset with a set of instances, A is an attribute, v is all the possible values of attribute A, and S_v is the subset of S where attribute A has the value v. Entropy expresses the amount of information in a random variable. Lower probability events yield more information with high entropy and higher probability events yield less information with low entropy.

$$InfoGain(S,A) = Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|}.Entropy(S_v) \quad (8)$$

Gain Ratio is another feature selection technique. In case there are attributes with the large number of values, Information Gain does not perform well, and results can be obtained as biased. Gain Ratio can address this flaw and attributes with many values are penalized. When selecting an attribute, the number and the size of the branches are considered. Information Gain is normalized with the split information as given in Equation (9).

$$GainRatio(S,A) = \frac{Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|}.Entropy(S_v)}{-\sum_{i=1}^{v} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}} \quad (9)$$

One of the measures used for feature selection is dependency measures. In order to determine how close two variables are to having a linear relationship with each other, a *correlation* metric is used. In this context, *correlation-based feature selection* is another way of finding the association between the continuous features and the target variable. Pearson's correlation is one of the most used correlation measures that is calculated for a pair of variables (x, y), where n is the number of pairs of data, as given in Equation (10) by resulting with a correlation coefficient R.

$$R = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (10)$$

Relief is another feature selection method that evaluates the worth of an attribute according to how well their values distinguish among instances that are near each other. If an attribute has the same value for samples in the same class but it differentiates between samples from a different class, it is assumed as a "good" attribute. Its extension *ReliefF* can deal with noisy and incomplete datasets and effectively handles multi-class problems. It randomly selects l instances. Assuming that x_k and x_i are instances, to predict the class feature C, where $|C|$ is the number of classes, it searches for k-nearest neighbors of the same class (called *nearest hit* and denoted by H_k), and then the same procedure is applied for each of the other classes (called *nearest miss* and denoted by M_k) by computing the distances d and probabilities p. The feature rank is calculated by Equation (11).

$$RF = \frac{1}{|C|}\sum_{k=1}^{l}\left(-\frac{1}{|M_k|}\sum_{x_i\in M_k} d(x_k, x_i) + \sum_{c\neq c(x_k)}\frac{p(c)}{|H_k|(1-p(c))}\sum_{x_i\in H_k} d(x_k, x_i)\right) \tag{11}$$

4. Experimental studies

In order to demonstrate the effectiveness of the proposed T-Bagging method, the experiments were carried out on 12 temporal datasets. In this study, we preferred to use support vector machines (SVM) as the base classifier because of its advantages such as nonlinear learning ability and good generalization capability. However, other classification algorithms such as decision tree, k nearest neighbors, artificial neural networks can also be used. The proposed method T-Bagging$_{\text{SVM}}$ was compared with two existing methods: a single version of SVM [19] and its ensemble version Bagged$_{\text{SVM}}$ [1] and and two temporal methods: a classification method based on ARIMAX [2] and temporal SVM (TSVM) implemented as in [3].

Except for ARIMAX, the implementation was developed in the C# programming language using the WEKA open-source machine learning library [20]. In order to apply SVM, the sequential minimal optimization (SMO) algorithm under WEKA was used with its default parameters where the complexity parameter is 1 and the applied kernel is the polynomial kernel.ARIMAX, which is one of the multivariate time series methods, was implemented on Spyder using Python. The parameters of ARIMAX were set as default values in the "statsmodels" library such that the number of auto-regressive lags (p), the degree of differencing (the number of times the dependent variable has to be differenced to produce the stationary series) (d), and the number of moving average lags (q) are (1, 0, 0). ARIMAX produces numerical predictions for the class attribute for multivariate time series datasets. In our study, it was performed to produce a numerical output for each instance, then these predictions were transformed into categorical labels that represent the real classes of the respective dataset. Assume that there are two classes of a dataset with the labels of "0" and "1". If the result from ARIMAX is less than 0.5, the class label 0 is assigned to the corresponding instance, otherwise, 1 is assigned. Conversions from numeric to categoric predictions were done in this way for all datasets and results were obtained.

The temporal classification method presented in [3] was implemented using SVM as the base classifier for the model. In this study, it is mentioned as the temporal SVM method (TSVM). The parameter d of the model represents the number of timestamps that any particular instance depends on before its timestamp. Given that $d = 2$ and an instance is recorded at the timestamp t_i, then this instance is concatenated with the features of the instances at the timestamps of t_{i-1} and t_{i-2}. In this way, the temporal dimension is added to the dataset. After updating the data format, classical SVM is applied for classification. In our study, $d = 2$ was used to implement the TSVM model.

The classification accuracies of the aforementioned methods were obtained using all features. Furthermore, in order to select the most informative attributes, feature selection was applied to all datasets. "*Information gain*", "*Gain ratio*", "*ReliefF*" and "*Correlation-based feature subset selection*" methods were performed to evaluate the worth of the features when T-Bagging was implemented. Ranker was set as the search method. In *ReliefF*, the number of nearest neighbors for attribute selection was 10. The number of selected features was analyzed from 10% to all of the features with a 10% increase in order to determine the optimum number of attributes in model creation for each weight formula.

4.1. Dataset description

In this study, the experiments were carried out on 12 univariate time series datasets available in the repository[1]. The records in the datasets are ordered according to their timestamps. Table 5 presents the main properties of the datasets, including their train/test sizes, the number of attributes, and the number of classes.

Table 5. Characteristics of the datasets used in the experiments.

Dataset	Num of classes	Num of features	Train size	Test size
ArrowHead	3	251	36	175
Earthquakes	2	512	322	139
ECG200	2	96	100	100
ECG5000	5	140	500	4500
ElectricDeviceDetection	2	256	623	3767
FreezerRegularTrain	2	301	150	2850
FreezerSmallTrain	2	301	28	2850
GunPoint	2	150	50	150
Ham	2	431	109	105
MiddlePhalanxOutlineCorrect	2	80	600	291
PhalangesOutlinesCorrect	2	80	1800	858
Wafer	2	152	1000	6164

4.2. Experimental results

Table 6 shows the accuracy results of the applied methods ARIMAX, TSVM, SVM, $\text{Bagged}_{\text{SVM}}$, and $\text{T-Bagging}_{\text{SVM}}$ for the balanced and imbalanced datasets, separately. The ensemble size of $\text{Bagged}_{\text{SVM}}$ and $\text{T-Bagging}_{\text{SVM}}$ is 10. The outputs of different weight functions of T-Bagging are indicated in different columns, separately. The best results for each dataset are displayed in bold font.

It is clearly seen that the accuracy values of SVM and $\text{Bagged}_{\text{SVM}}$ were enhanced for all datasets by considering temporal effects. When the average accuracy values are examined, the best one was obtained by $\text{T-Bagging}_{\text{SVM}}$ when WF_5 was used. While SVM and $\text{Bagged}_{\text{SVM}}$ performed 79.10% and 79.65% accuracy respectively, $\text{T-Bagging}_{\text{SVM}}$ with WF_5 achieved more accurate classification as 81.19%. Our method clearly increased the classification performance of each model in the ensemble since the models were built by using more informative samples. For example, the accuracies of T-Bagging and the standard bagging methods are (88.67%)

[1]Bagnall A, Keogh E, Lines J, Bostrom A, Large J. (2020). Time Series Classification Repository [online]. Website http://timeseriesclassification.com [accessed 1 July 2020]

and (78%), respectively, on the GunPoint dataset. In particular, the biggest accuracy difference of T-Bagging and Bagging was observed on the Ham dataset, where T-Bagging increased the accuracy by over 12%. Therefore, it can be concluded that the proposed method yields better results than the standard bagging method when dealing with temporal data. Furthermore, when temporal methods were examined, T-Bagging performed more accurate classification in most of the datasets (9 out of 12). TSVM achieved the best performance on only 3 datasets with an overall accuracy of 78.09%, while the ARIMAX model could not manage to classify well with the overall accuracy of 68.61% compared to its counterparts.

Table 6. Comparison of the proposed method with the existing methods in terms of classification accuracy (%).

B/I	Dataset	ARIMAX	TSVM	$Bagged_{SVM}$	SVM	$T\text{-}Bagging_{SVM}$						
						WF1	WF2	WF3	WF4	WF5	WF6	WF7
Balanced	Arrow head	61.71	32.37	73.71	73.14	77.14	70.29	73.71	69.71	76.57	**78.86**	72.00
	ECG200	51.00	78.35	82.00	81.00	83.00	83.00	82.00	83.00	**84.00**	81.00	77.00
	Freezer Regular Train	90.98	96.98	97.30	97.02	96.67	96.88	95.44	92.49	**97.37**	94.81	97.16
	Freezer Small Train	68.21	89.15	88.67	90.39	80.49	91.30	90.95	**94.84**	82.11	80.74	90.49
	GunPoint	78.67	76.87	78.00	80.00	**88.67**	75.33	80.00	84.67	87.33	73.33	78.00
	Ham	52.38	**82.52**	60.00	60.00	65.71	64.76	62.86	67.62	72.38	69.52	70.48
	Middle Phalanx Outline Correct	58.42	**69.20**	63.57	63.23	66.67	65.64	63.23	65.64	64.26	67.01	65.29
	Phalanges Outlines Correct	66.08	**69.40**	65.15	64.69	65.15	64.92	64.92	65.85	65.85	66.43	65.38
	Average	*65.93*	*74.36*	*76.05*	*76.18*	*77.94*	*76.52*	*76.64*	*77.98*	***78.73***	76.46	76.98
Imbalanced	Earthquakes	64.03	70.80	71.94	64.03	72.66	72.66	**74.82**	73.38	69.78	72.66	73.38
	ECG5000	57.07	91.22	94.07	93.84	93.96	93.91	**94.33**	93.11	93.58	93.58	93.64
	Electric Device Detection	84.29	84.73	85.56	85.91	85.11	86.31	85.3	85.56	85.14	85.27	**86.68**
	Wafer	90.48	95.49	95.86	95.98	94.87	95.28	**96.04**	93.19	95.93	96.03	94.32
	Average	*73.97*	*85.56*	*86.86*	84.94	86.65	*87.04*	***87.62***	*86.31*	*86.11*	*86.89*	87.01
General Average		*68.61*	*78.09*	*79.65*	*79.10*	*80.84*	*80.02*	80.30	*80.76*	***81.19***	*79.94*	*80.32*

A key feature of our T-Bagging method is that it can be applied to any temporal data without the knowledge of its structure. It is clear from Table 6 that the accuracy values for WF_5 ranged between 64.26% and 97.37%, changing from dataset to dataset. For example, the T-Bagging method achieved the best accuracy (97.37%) on the FreezerRegularTrain dataset. Hence, it experimentally confirmed that the characteristics of the data have a significant impact on the performance of the algorithm. When the datasets are analyzed considering the class values in the most recent time interval, four of them show imbalanced data characteristics. When we elaborate further by taking this into account, the proposed method also achieved successful results in the imbalanced datasets. The highest accuracies were obtained using the $T - Bagging_{SVM}$ with WF_3 with the average of 87.62%. On the other hand, the best results in the balanced datasets were attained using the

$T - Bagging_{SVM}$ with WF_5 as in the overall results with the average of 78.73%. It is clear that the best results for both balanced and imbalanced data were obtained when the proposed method was performed.

As expected, a single SVM method showed the worst performance (79.10%) on average compared to ensemble methods. This is because of the fact that ensemble methods usually improve the generalization and robustness by combining the decisions of multiple classifiers. In this way, they utilize the strengths of a group of predictors while avoiding the weaknesses of a single predictor.

The ranking procedure applied to all data was started by assigning rank 1 to the most accurate method, rank 2 to the second-best one, and was continued until giving rank 9 to the worst one. In the case of a tie, the average rank value was assigned to each method. According to the results in Figure 3, the proposed T-Bagging method exhibited the best performance compared to the existing methods because it had the lowest rank value with WF_5 (4.75). The T-Bagging with WF_3 method followed it with a rank value of 5.13. Therefore, it can be concluded that T-Bagging performed well on the average ranks.

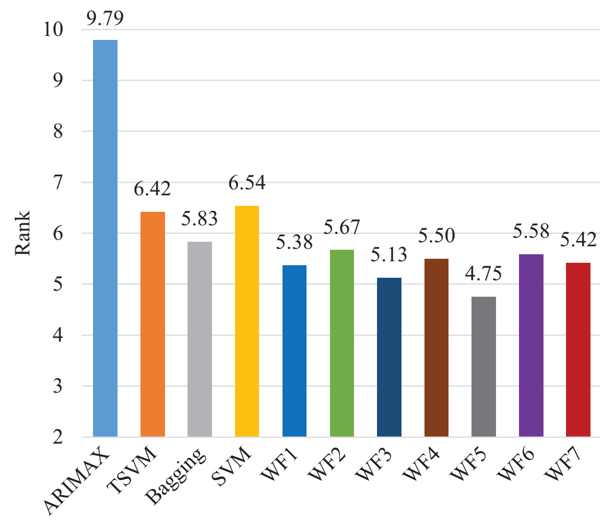

Figure 3. Comparison of the methods in terms of average rank.

In addition to accuracy evaluation, other metrics such as precision, recall, and F-measure are also important to consider in model selection in favor of the bias-variance tradeoff. Figure 4 shows the variations in the average values of the precision, recall, and F-measure for all datasets when SVM, $\mathrm{Bagged}_{\mathrm{SVM}}$, ARIMAX, TSVM, and $\mathrm{T\text{-}Bagging}_{\mathrm{SVM}}$ using different weights were applied. As in the case of accuracy, WF_5 attained the best results compared to its counterparts in the criteria of Recall and F-measure with the values of 81.19% and 79.74%, respectively. WF_1 achieved the best precision as 80.72%. SVM and $\mathrm{Bagged}_{\mathrm{SVM}}$ did not manage to defeat their T-Bagging counterparts here either with the precision, recall, and F-measure values as 79.22%, 79.10%, 77.46% for SVM and 79.24%, 79.65%, and 77.82% for $\mathrm{Bagged}_{\mathrm{SVM}}$. The lowest values were obtained by the ARIMAX method as 68.62% precision, 70.30% recall, and 68.43% F-Measure. TSVM followed ARIMAX with the values of 76.63% precision, 78.10% recall, and 76.68% F-Measure. The existing temporal methods could not reach the values obtained by the proposed T-Bagging method.

In order to determine the number of the most informative attributes explaining the datasets, various feature selection methods were applied when T-Bagging was implemented. At first, 60% of the total number of features was selected using gain ratio, information gain, ReliefF feature selection, and correlation-based feature

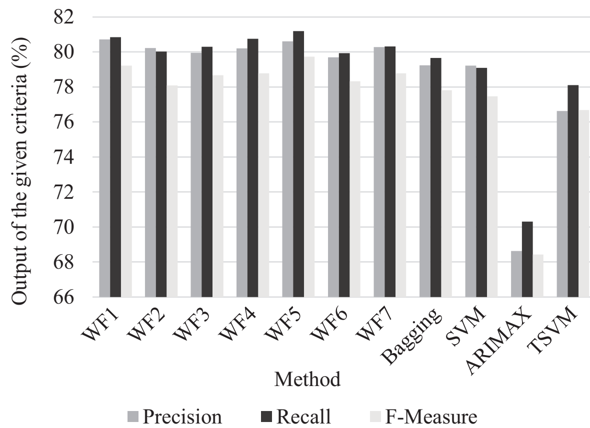

Figure 4. The results of precision, recall, and F-measure performance metrics when T-Bagging, Bagging, ARIMAX, TSVM, and SVM were applied.

subset selection with Pearson's correlation to select the best method among them. Table 7 displays the average accuracy results obtained for all datasets using different weight formulas. It is obvious that the Correlation feature selection method provided the best accuracy compared to others in general. Therefore, for further analysis, it was chosen among others.

Table 7. Comparison of the feature selection methods under different weights when 60% of features were chosen.

Weight	Gain Ratio	Info Gain	Correlation	ReliefF
WF_1	81.37	79.36	81.77	**81.79**
WF_2	79.66	77.42	**80.29**	79.84
WF_3	80.54	79.59	**80.77**	80.56
WF_4	80.62	78.90	**81.96**	80.89
WF_5	82.00	80.39	**82.20**	81.77
WF_6	80.85	79.75	**81.54**	80.40
WF_7	80.13	78.92	**80.82**	80.48

Figure 5 demonstrates the trend in the classification accuracy as the selected percentage of features were changed from 10% to whole attributes when Pearson's correlation feature selection method was used. It is evident that the correctly classified samples mostly increased as the number of selected features was increased. It is also shown that after a specific percentage (generally 50% – 70%) the classification accuracy started to decrease. This is expected because the whole feature set can include some redundant attributes which do not affect the general pattern explaining the data. Feature selection facilitates the determination of the most informative attributes to construct the model for optimal classification. The scenario in Figure 5 proves this inference by obtaining the eliminated feature set resulting in the most accurate results. For example, in the case of WF_4, a general increase is clearly seen in the accuracy until 70% of the selected features and the peek accuracy is obtained here as 81.97%. From here on, classification accuracy begins to decline due to the inclusion of redundant features in model creation.

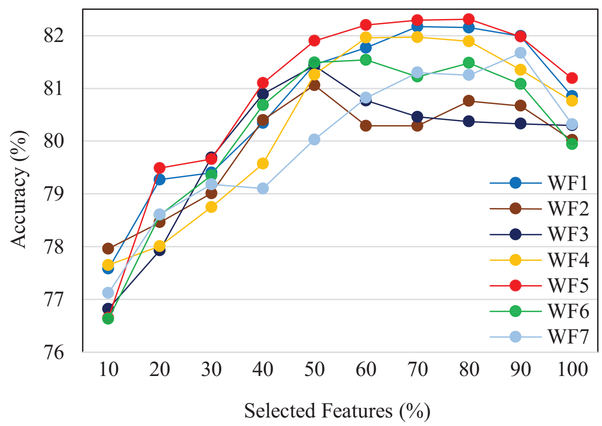

Figure 5. The impacts of different selection ratios when Correlation feature selection was applied.

5. Conclusion and future work

To take into account the fact that the recent past is more significant than the remote past, this study proposes a new time-based ensemble learning method, called temporal bagging (T-Bagging). In the proposed T-Bagging method, the prediction ability of base learners is enhanced through the adoption of a weight-based random sampling on the original dataset for the purpose of building models on training sets having more informative samples.

In the experimental studies, SVM was selected as a base classifier and it was compared with its bagging and T-Bagging versions. In addition, temporal methods, namely ARIMAX and TSVM, were also compared with the proposed method to investigate whether T-Bagging can be an alternative to them for temporal data classification or not. In terms of accuracy, T-Bagging achieved the most accurate classifications compared to its counterparts using different temporal weight functions. Besides, a considerable performance was also observed in precision, recall, and F-measure metrics. When Pearson's correlation feature selection method was applied, the overall performance was improved even more.

This study mainly contributes to the following subjects:

- Bagging is promoted with a time-specific specialty in case the dataset is highly dependent on temporal effects.
- The temporal effects are investigated in detail by considering the impacts of various weights.
- The impact of different feature selection techniques on the proposed method are analyzed comprehensively by varying the percentages of the selected features.

In the future, differently from the applied scenario, the proposed method can also be implemented in regression and clustering problems by changing the base learner and the decision rule instead of using majority voting. Moreover, as an alternative to bagging, other ensemble learning techniques such as boosting, or stacking can be performed by considering temporal impacts. Furthermore, rather than SVM, other base classifiers such as decision trees, neural networks may be implemented according to the performance values on different datasets.

References

[1] Breiman L. Bagging predictors. Machine Learning 1996; 24 (2): 123-140. doi: 10.1007/BF00058655

[2] Williams BM. Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. Transportation Research Record 2001; 1776 (1): 194-200. doi: 10.3141/1776-25

[3] Revesz P, Triplet T. Temporal data classification using linear classifiers. Information Systems 2011; 36 (1): 30-41. doi: 10.1016/j.is.2010.06.006

[4] Sim JA, Kim YA, Kim JH, Lee JM, Kim MS et al. The major effects of health-related quality of life on 5-year survival prediction among lung cancer survivors: applications of machine learning. Scientific Reports 2020; 10 (1): 1-12. doi: 10.1038/s41598-020-67604-3

[5] Nti IK, Adekoya AF, Weyori BA. A comprehensive evaluation of ensemble learning for stock-market prediction. Journal of Big Data 2020; 7 (1): 1-40. doi: 10.1186/s40537-020-00299-5

[6] Makris C, Pispirigos G, Rizos IO. A distributed bagging ensemble methodology for community prediction in social networks. Information 2020; 11 (4): 199. doi: 10.3390/info11040199

[7] Chen W, Hong H, Li S, Shahabi H, Wang Y, Wang X, Ahmad BB. Flood susceptibility modelling using novel hybrid approach of reduced-error pruning trees with bagging and random subspace ensembles. Journal of Hydrology 2019; 575: 864-873. doi: 10.1016/j.jhydrol.2019.05.089

[8] Agarwal S, Chowdary CR. A-Stacking and A-Bagging: Adaptive versions of ensemble learning algorithms for spoof fingerprint detection. Expert Systems with Applications 2020; 146: 113160. doi: 10.1016/j.eswa.2019.113160

[9] Long S, Zhao M, Song J. A novel PCA-DC-Bagging algorithm on yield stress prediction of RAFM steel. Computing 2020; 102 (1); 19-42. doi: 10.1007/s00607-019-00727-2

[10] Hsiao YH, Su CT, Fu PC. Integrating MTS with bagging strategy for class imbalance problems. International Journal of Machine Learning and Cybernetics 2020; 11: 1217-1230. doi: 10.1007/s13042-019-01033-1

[11] Li Y, Zhang J, Yuan X. BagGMM: Calling copy number variation by bagging multiple Gaussian mixture models from tumor and matched normal next-generation sequencing data. Digital Signal Processing 2019, 88: 90-100. doi: 10.1016/j.dsp.2019.01.025

[12] Al-rimy BAS, Maarof MA, Shaid SZM. Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection. Future Generation Computer Systems 2019; 101: 476-491. doi: 10.1016/j.future.2019.06.005

[13] Dahiya S, Handa SS, Singh NP. A feature selection enabled hybrid-bagging algorithm for credit risk evaluation. Expert Systems 2017; 34 (6): e12217. doi: 10.1111/exsy.12217

[14] Guan H, Zhang Y, Cheng H, Tang X. A novel imbalanced classification method based on decision tree and bagging. International Journal of Performability Engineering 2018; 14 (6): 1140-1148. doi: 10.23940/ijpe.18.06.p5.11401148

[15] Ooi SY, Tan SC, Cheah WP. Temporal sampling forest (TS-FTS-F): an ensemble temporal learner. Soft Computing 2017; 21: 7039–7052. doi: 10.1007/s00500-016-2242-7

[16] Bergmeir C, Hyndman RJ, Benítez JM. Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation. International Journal of Forecasting 2016; 32 (2): 303-312. doi: 10.1016/j.ijforecast.2015.07.002

[17] Deng H, Runger G, Tuv E, Vladimir M. A time series forest for classification and feature extraction. Information Sciences 2013; 239: 142–153. doi: 10.1016/j.ins.2013.02.030

[18] Zhao J. Temporal weighting of clinical events in electronic health records for pharmacovigilance. In: IEEE 2015 International Conference on Bioinformatics and Biomedicine; Washington, DC, USA; 2015. pp. 375-381. doi: 10.1109/BIBM.2015.7359710

[19] Cortes C, Vapnik V. Support-vector networks. Machine Learning 1995; 20: 273-297. doi: 10.1007/BF00994018.

[20] Witten IH, Frank E, Hall MA, Pal CJ. Data Mining: Practical Machine Learning Tools and Techniques. Cambridge, MA, USA: Morgan Kaufmann, 2016.