

Research Article

EBOC: Ensemble-Based Ordinal Classification in Transportation

Pelin Yıldırım,¹ Ulaş K. Birant,² and Derya Birant ²

¹Department of Software Engineering, Manisa Celal Bayar University, 45400, Manisa, Turkey

²Department of Computer Engineering, Dokuz Eylul University, 35390, Izmir, Turkey

Correspondence should be addressed to Derya Birant; derya@cs.deu.edu.tr

Received 3 October 2018; Revised 20 January 2019; Accepted 5 March 2019; Published 24 March 2019

Guest Editor: Mohammad H. Y. Moghaddam

Copyright © 2019 Pelin Yıldırım et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Learning the latent patterns of historical data in an efficient way to model the behaviour of a system is a major need for making right decisions. For this purpose, machine learning solution has already begun its promising marks in transportation as well as in many areas such as marketing, finance, education, and health. However, many classification algorithms in the literature assume that the target attribute values in the datasets are unordered, so they lose inherent order between the class values. To overcome the problem, this study proposes a novel ensemble-based ordinal classification (EBOC) approach which suggests bagging and boosting (AdaBoost algorithm) methods as a solution for ordinal classification problem in transportation sector. This article also compares the proposed EBOC approach with ordinal class classifier and traditional tree-based classification algorithms (i.e., C4.5 decision tree, RandomTree, and REPTree) in terms of accuracy. The results indicate that the proposed EBOC approach achieves better classification performance than the conventional solutions.

1. Introduction

Machine learning plays an important role in many prediction problems by constructing a model from explored dataset. The most common task in learning process is classification. *Classification* is the process of assigning an input item in a collection to predefined classes by discovering relationships among instances in the training set. Classification has a wide range of applications, such as document categorization, medical diagnosis, fraud detection, pattern recognition, sentiment analysis, risk assessment, and signal processing.

Transportation is a sector which focuses on replacement of humans, animals, and stuff from one position to another. Developments in the field of transportation reveal a need to discover associations and obtain complex and nonlinear relations underlying in a vast amount of transportation data. Because of this necessity, in recent years, machine learning techniques, especially classification algorithms, commenced to be used in transportation sector as an interdisciplinary approach. The underlying goals for these solutions are to predict traffic flow [1], classify vehicle images [2], identify different transportation modes [3], analyse traffic incident's severity [4], mitigate unfavourable environmental impacts

(i.e., to optimize energy usage [5]), develop autonomous driving system [6], and improve the productivity and efficiency of transportation systems.

The relationship between the class labels in the dataset to which the machine learning algorithms are applied influences the classification performance. In the literature, for the data of which class attribute values involve some order or a sort of ranking system, ordinal classification approach is proposed. *Ordinal classification* predicts the label of a new ordinal sample by taking ranking relation among the classes into consideration [7]. An example of an ordinal class attribute is one that has the values “large,” “medium,” and “small” for a size attribute. It is clear that there is an order among those values and that we can write $\text{large} > \text{medium} > \text{small}$. Although there are many classification studies [2–6] performed in transportation sector, to the best of our knowledge, there has been no prior detailed investigation for ordinal classification in transportation sector. Considering this drawback, the study presented in this article focuses on the application of ordinal classification algorithms on real-world transportation datasets. Some transportation studies contain an ordinal response variable; for example, the injury severity of traffic accidents can be categorized as fatal >

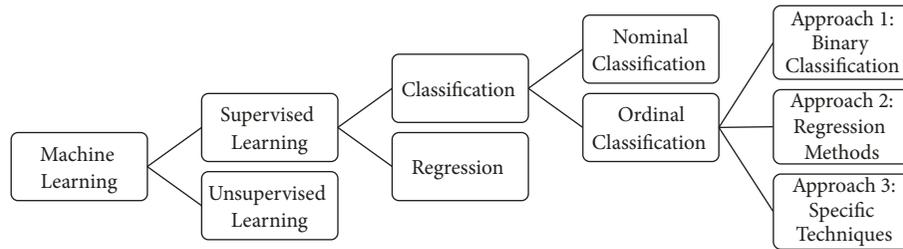


FIGURE 1: The classification methods of machine learning.

serious > slight; similarly, traffic volume can be classified as high > medium > low.

Meanwhile, ensemble learning has been recently preferred in machine learning for the classification task because of the high prediction ability it provides. *Ensemble learning* is a machine learning technique which combines a set of base learning models to get a single final prediction [8]. These learning models can be any classification algorithms, such as neural network (NN), Naive Bayes (NB), decision tree, support vector machine (SVM), regression, and k-nearest neighbour (KNN). Many studies in the literature have stated that ensemble learners improve prediction performance of individual learning algorithms. There exist various ensemble-learning methods: bagging, boosting, stacking, and voting. In this study, bagging and boosting (AdaBoost algorithm) methods are selected, due to their popularity, for the solution of ordinal classification problem in transportation sector.

The novelty and main contributions of this article are as follows: (i) it provides a brief survey of ordinal classification and ensemble learning, which has been revealed to improve prediction performance of the traditional classification algorithms, (ii) it is the first study in which the ordinal classification methods have been implemented in transportation sector, (iii) it proposes a novel ensemble-based ordinal classification (EBOC) approach for transportation, and (iv) it presents experimental studies conducted on twelve different real-world transportation datasets to demonstrate that the proposed EBOC approach shows better classification results than both ordinal class classifier and traditional tree-based classification algorithms in terms of accuracy.

The remainder of this article is structured as follows: in the following section, related literature and previous works on the subject are summarized. Section 3 gives background information about ordinal classification and ensemble learning. This section also explains utilized tree-based algorithms such as C4.5 decision tree, RandomTree, and REPTree in detail. In Section 4, the proposed ensemble-based ordinal classification (EBOC) approach for transportation sector is defined. Section 5 gives the description of transportation datasets used in this study. The application of traditional algorithms and proposed method on the transportation datasets and the experimental results of them with discussions are also presented in this section. Moreover, all obtained results were validated by three statistical methods to ensure the significance of differences among the classifiers on the datasets, including multiple comparisons (Friedman test and

Quade test) and pairwise comparisons (Wilcoxon signed rank test). Finally, the last section gives some concluding remarks and future directions.

2. Related Work

In machine learning, there are two main types of tasks: supervised learning and unsupervised learning. The classification process, which is one of the supervised learning techniques, is divided into two categories: *nominal classification* (where no order is assumed between the classes) and *ordinal classification* (where the ordinal relationship between different class labels should be taken into account). The difference between ordinal and nominal classification is not remarkable in the case of binary classification, owing to the fact that there is always an implicit order in “positive class” and “negative class.” In multiclass classification problems, standard classification algorithms for nominal classes can be applied to ordinal prediction problems by discarding the ordering information in the class attribute. However, this approach does not take advantage of the inner structure of the data and some information that can potentially improve the predictive performance of a classifier is lost since it ignores the existing natural order of the classes. The literature presents three different approaches for the ordinal classification problem: binary classification, regression methods, and specific techniques. The ordinal classification paradigm is summarized in the graph in Figure 1. In this article, a novel ordinal classification study was performed for the transportation sector.

The first approach for the ordinal classification process is to convert an ordinal classification problem into several binary classification problems. In this type of studies [9, 10], two-class classification algorithms are applied on ordinal valued datasets after transforming k class problem into a set of $k-1$ binary subproblems. For example, in the study presented in [10], the researchers proposed a novel approach which reduces the problem of classifying ordered classes to standard two-class problem. They introduced a data replication method which is then mapped into neural networks and support vector machines. In the experiments, they applied the proposed method on both artificial and real datasets for gene expression analysis. Li and Lin [11] developed a reduction framework for ordinal classification that consists of three steps: (i) extracting extended examples from training examples by using weights, (ii) training a classifier on the

extended examples with any binary classification algorithm, and (iii) constructing a ranking rule from the binary classifier.

As a second approach, regression methods [12, 13] can be used to deal with ordinal classification problem, since regression models have been thought for continuous data. In this method, categorical ordinal data is converted into continuous data scale and then regression algorithms are applied on this transformed data as a postprocessing step. However, a disadvantage of this approach is that the natural order of the class values is discarded and the inner structure of the samples is lost. In [12], two most commonly used ordinal logistic models were applied on medical ordinal data: proportional odds (PO) form of an ordinal logistic model and the forward continuation ratio (CR) ordinal logistic model. Rennie and Srebro [13] applied two general threshold-based constructions (the logistic and hinge loss) on the one million MovieLens dataset. The experimental results stated that their proposed approach shows more accurate results than traditional classification and regression models.

In the last approach, problem-specific techniques [14–18] were developed for ordinal data classification by modifying present classification algorithms. The main advantage of this approach is to retain the order among the class labels. However, some of them present some complexities in terms of implementation and training. These are complex and require nontrivial changes in the training methods such as modification of the objective function or using a threshold-based model. Keith and Meneses [14] proposed a novel technique called Barycentric Coordinates for Ordinal Classification (BCOC) which uses barycentric coordinates to represent ordinal classes geometrically. They applied their proposed method on the field of sentiment analysis and presented effective results for complex datasets. Researchers in another study [17] presented a novel heuristic rule learning approach with monotonicity constraints including two novel justifiability measures for ordinal classification. The experiments were performed to test the proposed approach and the results indicated that the novel method showed high prediction performance by guaranteeing monotone classification with low rule set increase.

Under favour of its high prediction performance, ensemble-learning techniques commenced to be preferred in ordinal classification [19–22] as well as nominal classification problems. Hechenbichler and Schliep [20] proposed an extended weighted k-nearest neighbor (*wkNN*) method for the ordinal class structure. In their study, weighted majority vote mechanism was used for the aggregation process. In the other study [21], an enhanced ensemble of support vector machines method was developed for ordinal regression. The proposed approach was implemented on the benchmark synthetic datasets and was compared with a kernel based ranking method in the experiments. Lin [23] introduced a novel threshold ensemble model and developed a reduction framework to reduce ordinal ranking to weighted binary classification by extending SVM and AdaBoost algorithms. The results of all these studies show that the ensemble methods perform well on the datasets and provide better performance than the individual methods.

Differently from existing studies, our work proposes a novel ensemble-based ordinal classification approach including bagging and boosting (AdaBoost algorithm) methods. Also, the present study is the first study in which an ordinal classification paradigm is implemented on real-world transportation datasets to model the behaviour of transportation systems.

3. Background Information

In this section, background information about ordinal classification, classification algorithms, and ensemble learning is presented to provide the context for this research.

3.1. Ordinal Classification. In most of classification problems, target attribute values which will be predicted usually assumed that they have no ordering relation between them. However, the class labels of some datasets have inherent order. For example, when predicting the price of an object, ordered class labels such as “expensive,” “normal,” and “cheap” can be used. The order among these class labels is clearly understood and denoted as “expensive” > “normal” > “cheap.” Because of this reason, classification techniques differ according to nominal and ordinal data types.

Nominal Data. The data which have no quantitative value is named nominal data. To label variables, nominal scales are utilized such as vehicle types (i.e., car, train, and bus) or document topics (i.e., science, business, and sports).

Ordinal Data. In ordinal data, the values have a natural order. In this data type, the order of values is significant such as data obtained from the use of a Likert scale.

Ordinal classification which is proposed for the prediction of ordinal target values is one of the most important classification problems in machine learning. This paradigm aims to predict the unknown class values of an attribute y that have a natural order. In the ordinal classification problem, the ordinal dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ has a set of m items with input feature space X and class attribute $y = \{c_1, c_2, \dots, c_k\}$ has k class labels with an order $c_k > c_{k-1} > \dots > c_1$, where $>$ denotes the relation of ordering. In other words, an example (x, y) is composed of an input vector $x \in X$ and an ordinal label (i.e., rank) $y \in Y = \{1, 2, \dots, K\}$. The problem is to predict the example (x, y) as rank k , where $k = 1, 2, \dots, K$.

In a decision tree-based ordinal classification, data is transformed from a k -class ordinal problem to $k-1$ binary class problems that encode the ordering of the class labels, as the first step. The ordinal dataset with classes c_1, c_2, \dots, c_k is converted into binary datasets by discriminating c_1, \dots, c_i against c_{i+1}, \dots, c_k that represents the test $c_x > i$. In other words, the upward unions of classes are considered progressively in each stage of binary datasets construction. Then a standard tree-based learning algorithm is employed on the derived binary datasets to construct $k-1$ models. As a result, each model predicts the cumulative probability of an instance of belonging to a certain class. To predict the class

label of an unseen instance, the probability for each ordinal class $P(c_x)$ is estimated by using $k-1$ models. The estimation of the probability for the first ordinal class label depends on a single classifier $1-P(\text{target} > c_1)$. The probability of the last ordinal class is given by $P(\text{target} > c_{k-1})$. In the middle of the range, the probability is calculated by a pair of classifiers $P(\text{target} > c_{i-1}) - P(\text{target} > c_i)$, where $1 < i < k$. Finally, we choose the class label with the highest probability.

The core ideas of using decision tree algorithms for ordinal classification problems can be described in three folds.

First, Frank and Hall [7] reported that decision tree-based ordinal classification algorithm resulted in a significant improvement over the standard version on 29 UCI benchmark datasets. They also experimentally confirmed that the performance gap increases with the number of classes. Most importantly, their results showed that decision tree-based ordinal classification was able to generally produce a much simpler model with better ordinal prediction accuracy.

Second, to consider the inherent order of class labels, other standard classification algorithms such as KNN [20], SVM [21], and NN require modification (nontrivial changes) in the training methods. Traditional regression techniques can also be applied on ordinal valued data due to their ability to classify interval or ratio quantity values. However, their application to truly ordinal problems is necessarily *ad hoc* [7]. In contrast, the key feature of using decision tree for ordinal classification is that there is no need to make any modification on the underlying learning algorithm. The core idea is simply to transform the ordinal classification problem into a series of binary-class problems.

Third, owing to the advantages of fast speed, high precision, and ease of understanding, decision tree algorithms are widely used in classification, as well as ordinal classification. Ordinal classification is sensitive to noise in data. Even a few noisy samples exist in the ordinal dataset; they can change the classification results of the overall system. However, in the presence of noisy and missing data, a pruned decision tree can reflect the order structure information very well with good generalization ability. The benefits of implementing decision tree-based ordinal classification are not limited to these. It builds white box models, so the trees constructed from the algorithm can be visualized and the classification results can be easily explained by Boolean logic. In addition, the most important features, which are near to the root node, are emphasized via the construction of the tree for the ordinal prediction task.

3.2. Tree-Based Classification Algorithms. In this work, three tree-based classification algorithms (C4.5 decision tree, RandomTree, and REPTree) are used as base learners to implement ordinal classification process on transportation datasets that consists of ordered class values.

3.2.1. C4.5 Decision Tree. Decision tree is one of the most successful classification algorithms, which predicts unknown class attributes using a tree structure grown with depth-first strategy. The structure of decision tree consists of

nodes, branches, and leaves that represent attributes, attribute values, and class labels, respectively. In the literature, there are several decision tree algorithms such as C4.5, ID3 (iterative dichotomiser), CART (classification and regression trees), and CHAID (chi-squared automatic interaction detector). In this study, C4.5 decision tree algorithm was used, due to its popularity, for the ordinal classification problem in transportation sector.

The first step in C4.5 decision tree algorithm is to specify root of the tree. The attribute which gives the most determinant information for the prediction process is selected for the root node. To determine the order of features in the decision tree, information gain formula is evaluated for each attribute as defined in

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \quad (1)$$

where S_v is subset of states S for attribute A with value v . The entropy indicates the impurity of a particular attribute in the dataset as defined in

$$\text{Entropy}(S) = - \sum_{i=1}^n p_i \log_2 p_i \quad (2)$$

where i is a state, p_i is the possibility of outcome being in state i for the set S , and n is the number of possible outcomes. The attribute that has the maximum information gain value is selected for the root of the tree. Likewise, all attributes in the dataset are placed to the tree according to their information gain values.

3.2.2. RandomTree. Assume a training set T with z attributes and m instances, and RandomTree is an algorithm for constructing a tree that considers x randomly chosen features ($x \leq z$) with m instances at each node [24]. While, in a standard decision tree, each node is split using the best split among all features, in a random tree, each node is split using the best among the subset of attributes randomly chosen at that node. The tree is built to its maximum depth; in other words, no pruning procedure is applied after the tree has been fully built. The algorithm can deal with both classification and regression problems. In classification task, the predicted class for a sample is determined by traversing the tree from the root node to a leaf according to the question posed about an indicator value at that node. When a leaf node is reached, its label determines the classification decision. The RandomTree algorithm does not need any accuracy estimation metric different from the other classification algorithms.

3.2.3. REPTree. The reduced error pruning tree (REPTree) algorithm builds a decision / regression tree using information gain / variance and prunes it using a simple and fast technique [25]. The pruning technique starts from the bottom level of the tree (from leaf nodes) and replaces the node with most famous class. This change is accepted only if the prediction accuracy is good. By this way, it helps to minimize

the size of decision trees by removing sections of the tree that gives little capacity to classify instances. The values of numeric attributes are sorted once. Missing values are also dealt with by splitting the corresponding instances into parts.

3.3. Ensemble Learning. Ensemble learning is a machine learning technique that combines a set of individual learners and predicts a final output [26]. First, each learner in the ensemble structure is trained separately and multiple classification models are constructed. Then, the obtained outputs from each model are compounded by a voting mechanism. The commonly used voting method available for categorical target values is major class labelling and the voting methods for numerical target values are average, weighted average, median, minimum, and maximum.

Ensemble-learning approach constructs a strong classifier from multiple individual learners and aims to improve classification performance by reducing the risk of an unfortunate selection of these learners. Many ensemble-based studies [27, 28] proved that ensemble learners give more successful results than classical individual learners.

In the literature, the ensemble methods are generally categorized under four techniques: bagging, boosting, stacking, and voting. In one of our studies, we implemented bagging approach by combining multiple neural networks which includes different parameter values for the prediction task in textile sector [26]. In the current research, bagging and boosting (AdaBoost algorithm) methods with ordinal class classifier were applied on transportation datasets. For each method (bagging and boosting), tree-based classification algorithms (C4.5 decision tree, RandomTree, and REPTree algorithms) were used as base learners separately.

3.3.1. Bagging. Bagging (bootstrap aggregating) is a commonly applied ensemble technique which constitutes training subsets by selecting random instances from original dataset using bootstrap method. Each classifier in the ensemble structure is trained by different training sets and so multiple classification models are produced. Then, a new sample is given to each model and these models predict an output. The obtained outputs are aggregated, and a single final output is achieved.

General Process of Bagging. Let $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ be a set of m items and let $y_i \in Y = \{c_1, c_2, \dots, c_k\}$ be a set of k class labels; C denotes classification algorithm and n is number of learners.

- (1) Draw m items randomly with replacement from the dataset D , so generate bootstrap samples D_1, D_2, \dots, D_n .
- (2) Each dataset D_i is trained by a base learner and multiple classification models are constructed, $M_i = C(D_i)$.
- (3) Consensus of classification models is tested to calculate out-of-bag error.
- (4) New sample x is given to classifiers as input and the outputs y_i are obtained from each model, $y_i = M_i(x)$.

- (5) The outputs of models $\{M_1, M_2, \dots, M_t\}$ are combined as in

$$M^*(x) = \arg \max_{y \in Y} \sum_{i: M_i(x)=y} 1 \quad (3)$$

3.3.2. Boosting. In boosting method, classifiers are trained consecutively to convert weak learners to strong ones. A weight value is assigned to each instance in the training set. Then, in each iteration, while the weights of misclassified samples are increased, correctly classified ones are decreased. In this way, the misclassified samples' chances of being in the training set increase. In this study, AdaBoost algorithm was utilized to implement boosting method.

AdaBoost. AdaBoost, also known as adaptive boosting, is the most popular boosting algorithm which trains learners by reweighting instances in the training set iteratively. Then, the outputs produced by each learning model are aggregated using a weighted voting mechanism.

4. Proposed Method: Ensemble-Based Ordinal Classification (EBOC)

The proposed method, named ensemble-based ordinal classification (EBOC), combines ensemble-learning paradigm including its popular methods such as AdaBoost and bagging with the traditional ordinal class classifier algorithm to improve prediction performance. The ordinal class classifier algorithm which was proposed in [7] is used as a base learner for the ensemble structure of the proposed approach. In addition, tree-based algorithms such as C4.5 decision tree, RandomTree, and REPTree are also used as a classifier in ordinal class classifier algorithm.

The first step of the ordinal class classifier algorithm is to reduce multiclass ordinal classification problem to a binary classification problem. To realize this approach, the ordinal classification problem with k different class values is converted to $k-1$ two-class classification problems. Assume that there is a dataset with four classes ($k=4$); here, the task is to find two sides: (i) class C1 against classes C2, C3, and C4; (ii) classes C1 and C2 against classes C3 and C4; and finally (iii) classes C1, C2, and C3 against class C4.

For example, suppose we have car evaluation dataset [29] which evaluates vehicles according to buying price, maintenance price, and technical characteristics such as comfort, number of doors, person capacity, luggage boot size, and safety of the car. This dataset has an ordinal class attribute with four values: unacc, acc, good, and vgood. First, four different class values of the original dataset are converted to binary values according to these rules: Classes > "unacc", Classes > "acc" and Classes > "good" class valued attribute. If we consider "Classes > 'unacc'" rule, class values higher than "unacc" are labelled as 1 and the others are labelled as 0. In this way, three different transformed datasets that contain binary class values are obtained. In the next stage, a classification algorithm (i.e., C4.5, RandomTree, or REPTree) is applied on each obtained dataset separately. Figure 2 shows the

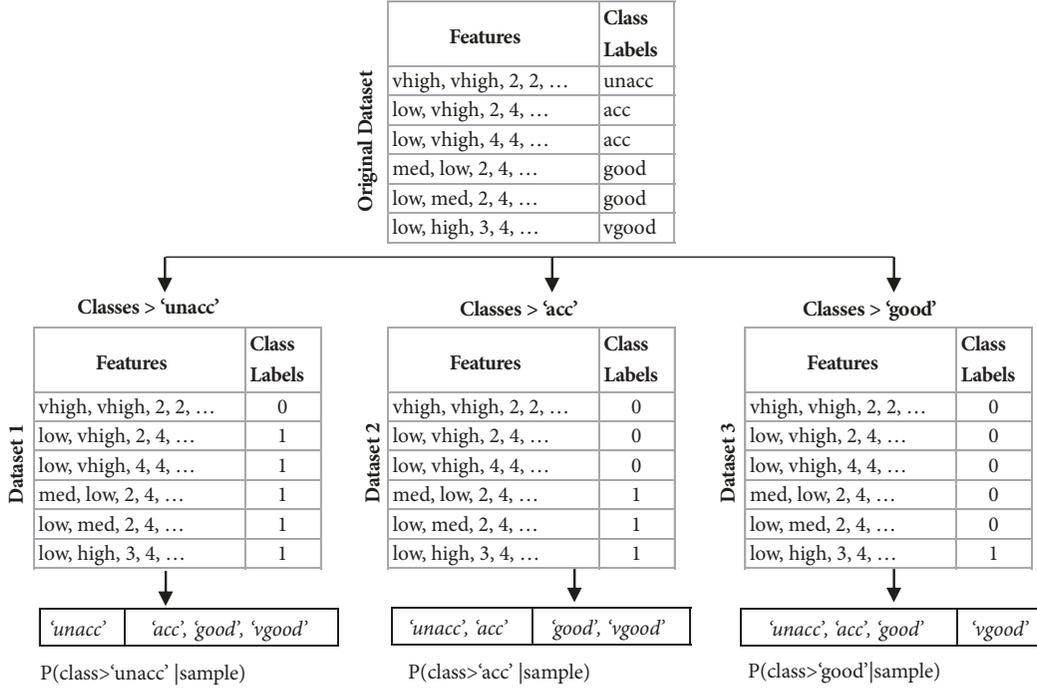


FIGURE 2: Application of ordinal class classifier algorithm on “car evaluation” dataset.

demonstration of how the ordinal class classifier algorithm works on the “car evaluation” dataset.

When predicting a new sample, the probabilities are computed for each k class value using $k-1$ binary classification models. For example, the probability of the “unacc” class value $P(\text{'unacc'} \mid \text{sample})$ in the sample car dataset is evaluated by $1 - P(\text{class} > \text{'unacc'} \mid \text{sample})$. Similarly, the other three ordinal class value probabilities are computed. And, finally, the class label which has the maximum probability value is assigned to the sample. In general, the probabilities of the class attribute values depend on “car evaluation” dataset which are calculated as follows:

$$\begin{aligned}
 P(\text{'unacc'} \mid \text{sample}) &= 1 - P(\text{class} > \text{'unacc'} \mid \text{sample}) \\
 P(\text{'acc'} \mid \text{sample}) &= P(\text{class} > \text{'unacc'} \mid \text{sample}) \\
 &\quad - P(\text{class} > \text{'acc'} \mid \text{sample}) \\
 P(\text{'good'} \mid \text{sample}) &= P(\text{class} > \text{'acc'} \mid \text{sample}) \\
 &\quad - P(\text{class} > \text{'good'} \mid \text{sample}) \\
 P(\text{'vgood'} \mid \text{sample}) &= P(\text{class} > \text{'good'} \mid \text{sample})
 \end{aligned} \tag{4}$$

The proposed approach (EBOC) utilizes the ordinal classification algorithm as base learner for bagging and boosting ensemble methods. This novel ensemble-based approach aims to obtain successful classification results under favour of high prediction ability of ensemble-learning paradigm. The general structure of the proposed approach is presented in Figure 3. When bagging is considered, random samples are selected from the original dataset to produce multiple training sets, or when boosting is utilized, samples are selected with specified probabilities based on their weights. After that, the EBOC derives new datasets from the original dataset, each one with new binary class attribute. Each dataset is given to ordinal class classifier algorithm as an input. Then, a classification algorithm (i.e., C4.5, RandomTree, or REPTree) is applied on the datasets and multiple ensemble-based ordinal classification models are produced. Each classification model in this system gives an output label and the majority vote of these outputs is selected as a final class value.

The pseudocode of the proposed approach (EBOC) is presented in Algorithm 1. First, multiple training sets are generated according to the preferred ensemble method (bagging or boosting). Second, the algorithm converts ordinal data to binary data. After that, a tree-based classification algorithm (i.e., C4.5, RandomTree, or REPTree) is applied on the binary training sets and, by this way, various classifiers are produced. After this training phase, if boosting is chosen as ensemble method, the weight of each sample in the dataset is updated dynamically according to the performance (error rate) of the classifiers in the current iteration. To predict the class label of a new input sample, the probability for each ordinal class is estimated by using the constructed models. The algorithm chooses the class label with the highest

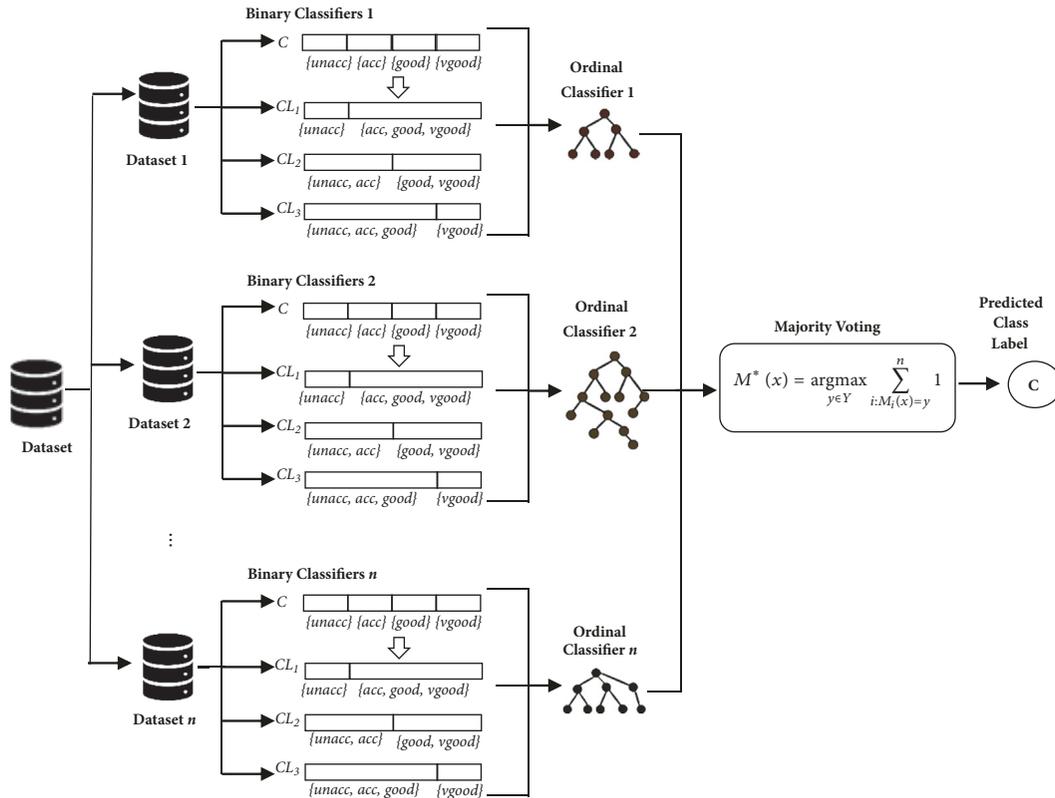


FIGURE 3: The general structure of the proposed approach (EBOC).

probability. Lastly, the majority vote of the outputs obtained from each ordinal classification vote model is selected as final output.

5. Experimental Studies

In the experimental studies, the proposed approach (EBOC) was implemented on twelve different benchmark transportation datasets to demonstrate its advantages over the standard nominal classification methods. The application was developed by using Weka open source data mining library [30]. Individual tree-based classification algorithms (C4.5, RandomTree, and REPTree), ordinal classification algorithm [7], and the EBOC approach were applied on the transportation datasets separately and they were compared in terms of accuracy, precision, recall, F-measure, and ROC area. In the EBOC approach, C4.5, RandomTree and REPTree algorithms were also used as a base learner in ordinal class classifiers separately. The obtained experimental results in this study are presented with the help of tables and graphs.

5.1. Dataset Description. In this study, 12 different transportation datasets which are available in several repositories for public use were selected to demonstrate the capabilities of the proposed EBOC method. The datasets were obtained from the following data archives: UCI Machine Learning Repository [31], Kaggle [32], data mill north [33], and NYC

open data [34]. The detailed descriptions about the datasets (i.e., what are they and how to use them) are given as follows.

Auto MPG. Auto MPG dataset, which was used in the American Statistical Association Exposition, is presented by the StatLib library which is maintained at Carnegie Mellon University. This dataset is utilized for the prediction of the city-cycle fuel consumption in miles per gallon of the cars according to their characteristics such as model year, the number of cylinders, horsepower, engine size (displacement), weight, and acceleration.

Automobile. The dataset was obtained from 1985 Ward's Automotive Yearbook. It consists of various automobile characteristics such as fuel type, body style, number of doors, engine size, engine location, horsepower, length, width, height, price, and insurance score. These features are used for the classification of automobiles by predicting their risk factors, six different risk ranking ranging from risky (+3) to pretty safe (-3).

Bike Sharing. The data includes the hourly count of rental bikes between years 2011 and 2012. It was collected by Capital bike share system from Washington D.C., where membership, rental, and bike return are automated via a network of kiosk locations. The dataset is presented for the prediction of hourly bike rental counts based on the environmental and seasonal settings such as weather situation (i.e., clear, cloudy, rainy, and

Algorithm EBOC: Ensemble-Based Ordinal Classification**Inputs:**

D : the ordinal dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$

m : the number of instances in the dataset D

X : input feature space, an input vector $x \in X$

Y : class attribute, an ordinal class label $y \in \{c_1, c_2, \dots, c_k\} \in Y$, with an order $c_k > c_{k-1} > \dots > c_1$

k : the number of class labels

n : ensemble size

Outputs:

M^* : an ensemble classification model

$M^*(x)$: class label of a new sample x

Begin:

for $i = 1$ **to** n **do**

if (bagging)

D_i = bootstrap samples from D

else if (boosting)

D_i = samples from D according to their weights

 // Construction of binary training sets, D_{ij}

for $j = 1$ **to** $k-1$ **do**

for $s = 1$ **to** m **do**

if ($y_s \leq c_j$)

D_{ij} .Add($x_s, 0$)

else

D_{ij} .Add($x_s, 1$)

end for

end for

 // Construction of binary classifiers, BC_{ij}

for $j = 1$ **to** $k-1$ **do**

BC_{ij} = ClassificationAlgorithm(D_{ij})

end for

if (boosting)

 update weight values

end for

 // Classification of a new sample x

for $i = 1$ **to** n **do**

 // Construction of ordinal classification models, M_i

$P(c_1) = 1 - P(y > c_1)$

for $j = 2$ **to** $k-1$ **do**

$P(c_j) = P(y > c_{j-1}) - P(y > c_j)$

end for

$P(c_k) = P(y > c_{k-1})$

$M_i = \max(P)$

end for

 // Majority voting

if (bagging)

$$M^*(x) = \arg \max_{y \in Y} \sum_{i: M_i(x)=y}^n 1$$

else if (boosting)

$$M^*(x) = \arg \max_{y \in Y} \sum_{i: M_i(x)=y}^n weight_i$$

End Algorithm

ALGORITHM 1: The pseudocode of the proposed EBOC approach.

snowy), temperature, humidity, wind speed, month, season, and the day of the week and year.

Car Evaluation. This dataset is useful to evaluate the quality of the cars according to various characteristics such as buying

price, maintenance price, number of doors, person capacity, size of luggage boot, and estimated safety of the car. The target attribute in the dataset indicates the overall scores of the cars as unacceptable, acceptable, good, and very good.

Car Sale Advertisements. The data was collected from private car sale advertisements in Ukraine in 2016. It was proposed for the prediction of seller's price (denominated in USD) in the advertisement according to well-known car features such as model, body type, mileage, engine volume, engine type, and drive type.

NYS Air Passenger Traffic. The data was collected monthly by the Port Authority of New York State between years 1977 and 2015. The aim is to predict the total number of domestic and international passengers for five airports (ACY, EWR, JFK, LGA, and SWF).

Road Traffic Accidents (2017). The dataset contains the records of traffic accidents across the City of Leeds, UK, reported in 2017 with the information of location, number of people and vehicles involved, the type of vehicle, road surface, weather situations, lighting conditions, and the age and sex of casualty. The aim is to predict the severity of casualty as slight, serious, or fatal.

SF Air Traffic Landings Statistics and SF Air Traffic Passenger Statistics. These two separate datasets include aircraft landings and passenger statistics of San Francisco International Airport recorded during the period from July 2005 to March 2018. These datasets are used to predict monthly landing and passenger counts, respectively.

Smart City Traffic Patterns. This dataset includes the traffic patterns of the four junctions of a city collected between 2015 and 2017. The aim is to manage the traffic of the city better and to provide input on infrastructure planning for the future to improve the efficiency of services for the citizens. To serve this purpose, the number of vehicles in the traffic is predicted to implement a robust traffic system for the city by being prepared for traffic peaks.

Statlog (Vehicle Silhouettes). The Statlog dataset contains the features of vehicle silhouettes extracted by Hierarchical Image Processing System (HIPS). The vehicle may be viewed from one of many different angles. The aim is to classify a given silhouette using a set of features extracted from the image.

Traffic Volume Counts (2012-2013). The dataset includes the hourly traffic volume counts collected by New York Metropolitan Transportation Council between years 2012 and 2013. The traffic counts were measured on various roads from one intersection to another with a specified direction (NB, SB, WB, and EB). The attribute that will be predicted in this dataset is traffic volume at 11:00 - 12:00 AM.

Before the application of the nominal and ordinal classification algorithms, generally the datasets have passed through data preprocessing steps. In this study, the ID attributes in the transportation datasets were eliminated in the data reduction step. The date attributes were split into three triplets: day, month, and year, because they provide more useful information for the prediction task. In addition, continuous class values were discretized into 3 bins using equal

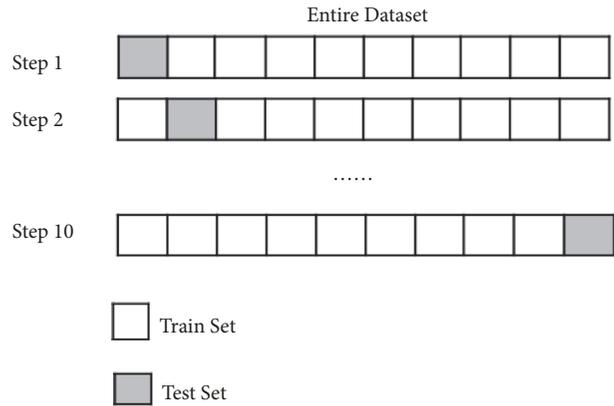


FIGURE 4: The 10-fold cross-validation process.

frequency technique, since ordinal classification algorithms require categorical ordinal data.

Basic characteristics of the investigated transportation datasets are given in Table 1. These are the number of instances, attributes and classes, target attribute, data preprocessing operations, and class distributions in each bin.

5.2. Experimental Work. In each experiment, four methods were compared on the transportation datasets: (i) individual tree-based classification algorithms, (ii) ordinal class classifier, (iii) boosting-based ordinal classification, and (iv) bagging-based ordinal classification. They were compared by using n -fold cross-validation technique selecting n as 10. In this validation technique, the entire dataset is divided into n equal size parts, $n-1$ of them is chosen for training phase of the classification model and the last part is used in testing phase. This process is repeated n times with changing parts for training and testing phase. When the cross-validation process is terminated, the accuracy values obtained in each step are averaged and a single accuracy value is produced as a success sign of the algorithm. Figure 4 represents 10-fold cross-validation process.

In this study, alternative tree-based ordinal classification algorithms were compared according to accuracy, precision, recall, F-measure, and ROC area measures. The metrics are explained with their abbreviations, formulas, and definitions in Table 2.

5.3. Experimental Results. In this study, three different experiments with three different tree-based classification algorithms (C4.5, RandomTree, and REPTree) were performed to compare the classification success of the proposed EBOC approach with the existing individual algorithms and ordinal class classifier algorithm. To evaluate the classification performances of the algorithms on a specific transportation dataset, accuracy rate values were calculated using n -fold cross-validation technique selecting n as 10.

In each experiment, one of the tree-based classification algorithms was used as base learners. For example, in the first experiment, four methods were applied and compared on 12 different transportation datasets: (i) C4.5: the individual

TABLE 1: The basic characteristics of the transportation datasets. (I: number of instances, A: number of attributes, C: number of classes).

Dataset	I	A	C	Class Distribution	Target Attribute	Data Preprocessing
Auto MPG	398	8	3	131-134-133	mpg	Removing the columns with unique values (i.e., car name)
Automobile	205	26	7	0-3-22-67-54-32-27	symboling	-
Bike Sharing	17379	13	3	5797-5783-5799	cnt	Removing the columns "instant," "dteday," "casual," and "registered"
Car Evaluation	1728	7	4	1210-384-69-65	class	-
Car Sale Advertisements	9309	10	3	3112-3080-3117	price	Removing rows with zero price value
NYS Air Passenger Traffic	1584	4	3	528-528-528	total passengers	Removing the columns "Domestic" and "International Passengers"
Road Traffic Accidents (2017)	2203	13	3	1879-309-15	casualty severity	(i) Removing columns that hold reference numbers (ii) Splitting date column into day and month
SF Air Traffic Landings Statistics	21105	14	3	6941-7103-7061	landing count	-
SF Air Traffic Passenger Statistics	18398	12	3	6132-6133-6133	passenger count	-
Smart City Traffic Patterns	48120	6	3	15687-16436-15997	vehicles	(i) Removing ID column (ii) Splitting date column into day, month, and year
Statlog (Vehicle Silhouettes)	846	19	4	212-217-218-199	class	-
Traffic Volume Counts (2012-2013)	5945	31	3	1983-1989-1973	traffic volume at 11:00 -12:00AM	(i) Removing the columns with unique values (i.e., ID, Segment ID) (ii) Splitting date column into day, month, and year

TABLE 2: The detailed information about classifier performance measures. (TP: true positives, FP: false positives, TN: true negatives, and FN: false negatives).

Abbreviation	Measure	Formula	Description
Acc	Accuracy	$Acc = \frac{TP + TN}{TP + TN + FP + FN}$	The ratio of the number of correctly classified instances to the total number of records.
Prec	Precision	$Prec = \frac{TP}{TP + FP}$	The ratio of correctly classified positive instances against all positive results.
Rec	Recall	$Rec = \frac{TP}{TP + FN}$	The ratio of correct answers for a class over all the answers correctly belonging to this class.
F	F-measure	$F = \frac{2 * precision * recall}{precision + recall}$	The harmonic mean of precision and recall.
ROC Area	Receiver Operating Characteristic Area	The area under the curve which is generated to compare correctly and incorrectly classified instances.	

decision tree algorithm; (ii) *Ord.C4.5*: ordinal class classifier with C4.5; (iii) *Ada.Ord.C4.5*: AdaBoost based ordinal classification with C4.5 as base learners; and finally (iv) *Bag.Ord.C4.5*: bagging based ordinal classification with C4.5 as base learners. In the second and third experiments, *RandomTree* and *REPTree* algorithms were utilized as base learners, respectively. The default parameters of the

algorithms in Weka were used in all experiments, except ensemble size (the number of members) parameter which was set to 100. As a result of the experiments, the accuracy rates of each algorithm were evaluated. Tables 3, 4, and 5 show the comparative results of implemented techniques on twelve benchmark datasets in terms of the accuracy rates. The accuracy rates which are higher than individual classification

TABLE 3: The comparison of C4.5 based ordinal and ensemble learning methods in terms of classification accuracy.

Dataset	C4.5 (%)	Ord.C4.5 (%)	Ada.Ord.C4.5 (%)	Bag.Ord.C4.5 (%)
Auto MPG	80.40	80.90 *	83.67*	81.91 *
Automobile	81.95	66.34	84.39 *	73.66
Bike Sharing	87.75	87.72	89.29 *	89.79 *
Car Evaluation	92.36	92.19	98.73*	94.27 *
Car Sale Advertisements	83.65	81.09	83.82 *	85.00 *
NYS Air Passenger Traffic	84.91	85.42 *	86.05 *	86.81 *
Road Traffic Accidents (2017)	85.29	85.29 *	81.03	84.38
SF Air Traffic Landings Statistics	98.74	98.31	99.37*	98.69
SF Air Traffic Passenger Statistics	90.68	90.82 *	90.41	91.43 *
Smart City Traffic Patterns	85.66	85.98 *	85.07	86.94 *
Statlog (Vehicle Silhouettes)	72.46	68.91	78.13*	74.82 *
Traffic Volume Counts (2012-2013)	88.36	88.12	89.77 *	89.10 *
<i>Average</i>	86.02	84.26	87.48	86.40

TABLE 4: The comparison of RandomTree based ordinal and ensemble learning methods in terms of classification accuracy.

Dataset	Random Tree (%)	Ord.Random Tree (%)	Ada.Ord. RandomTree (%)	Bag.Ord. RandomTree (%)
Auto MPG	77.64	81.66 *	82.66 *	81.41 *
Automobile	76.59	74.63	83.41 *	84.88 *
Bike Sharing	80.07	80.56 *	86.98 *	87.73 *
Car Evaluation	83.16	85.94 *	97.22 *	95.83 *
Car Sale Advertisements	82.38	82.12	83.90 *	86.34 *
NYS Air Passenger Traffic	86.11	86.17 *	85.61	86.49 *
Road Traffic Accidents (2017)	78.76	78.44	82.21 *	84.48 *
SF Air Traffic Landings Statistics	97.38	97.43 *	98.93 *	98.83 *
SF Air Traffic Passenger Statistics	89.22	89.27 *	89.05	89.17
Smart City Traffic Patterns	82.26	82.79 *	84.72 *	85.91 *
Statlog (Vehicle Silhouettes)	70.92	65.60	76.71 *	76.00 *
Traffic Volume Counts (2012-2013)	82.96	83.45 *	89.77 *	89.69 *
<i>Average</i>	82.29	82.34	86.76	87.23

algorithm's accuracy rate in that row are marked with *. In addition, the accuracy rates which have the maximum value in each row are made bold. The experimental results show that the EBOC approaches (Ada.Ord.C4.5, Bag.Ord.C4.5, Ada.Ord.RandomTree, Bag.Ord.RandomTree, Ada.Ord.REPTree, and Bag.Ord.REPTree) generally provide higher accuracy values than individual tree-based classification algorithms. For example, on 12 datasets, Bag.Ord.RandomTree (87.23%) is significantly more accurate than RandomTree (82.29%). Similarly, both Ada.Ord.REPTree and Bag.Ord.REPTree win against plain REPTree on 11 datasets and lose on only one. The results also show that the performance gap generally increases with the number of classes. When the average accuracy values are considered in general, it is possible to say that the proposed EBOC approach has the best accuracy score in all three

experiments: Ada.Ord.C4.5 is 87.48%, Bag.Ord.RandomTree is 87.23%, and Bag.Ord.REPTree is 85.67%.

The matrices presented in Tables 6, 7, and 8 give all pairwise combinations of the tree-based algorithms with ordinal and ensemble-learning versions. Each cell in the matrix represents the number of wins, losses, and ties between the approach in that row and the approach in that column. For example, in the pairwise of C4.5 and Bag.Ord.C4.5 algorithms, 2-10-0 indicates that standard C4.5 algorithm is better than Bag.Ord.C4.5 only on 2 datasets, while Bag.Ord.C4.5 is better than the other on 10 datasets. Their accuracies are not equal in any dataset. When all matrices are examined, it is clearly seen that our proposed approach EBOC outperforms all other methods.

The graphs given in Figures 5, 6, and 7 show the average ranks of the tree-based algorithms (C4.5, RandomTree, and

TABLE 5: The comparison of REPTree based ordinal and ensemble learning methods in terms of classification accuracy.

Dataset	REPTree (%)	Ord.REPTree (%)	Ada.Ord. REPTree (%)	Bag.Ord. REPTree (%)
Auto MPG	75.38	76.63 *	81.91 *	81.16 *
Automobile	63.41	66.83 *	83.90 *	72.20 *
Bike Sharing	84.96	85.27 *	88.75 *	88.12 *
Car Evaluation	87.67	90.91 *	98.67 *	93.63 *
Car Sale Advertisements	80.70	80.42	82.52 *	82.30 *
NYS Air Passenger Traffic	84.34	84.91 *	86.74 *	87.31 *
Road Traffic Accidents (2017)	84.66	85.02 *	80.07	84.57
SF Air Traffic Landings Statistics	98.04	98.10 *	99.12 *	98.57 *
SF Air Traffic Passenger Statistics	90.39	90.49 *	90.97 *	91.25 *
Smart City Traffic Patterns	84.66	85.06 *	85.33 *	86.38 *
Statlog (Vehicle Silhouettes)	72.34	69.62	77.54 *	73.17 *
Traffic Volume Counts (2012-2013)	80.57	88.75 *	86.39 *	89.37 *
<i>Average</i>	82.26	83.50	86.83	85.67

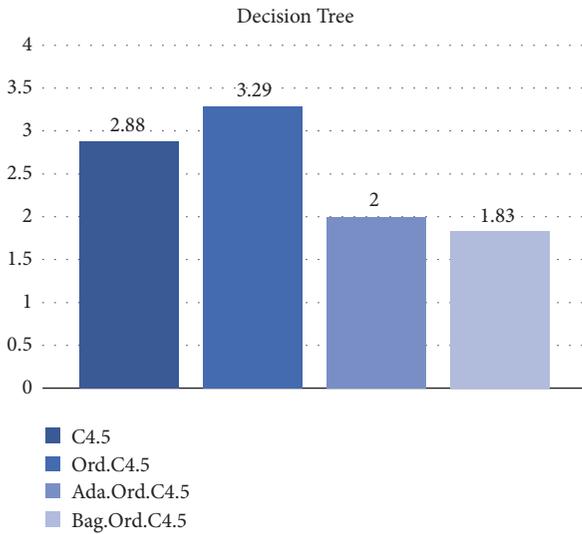


FIGURE 5: The average ranks of C4.5 algorithm with ordinal and ensemble-learning versions.

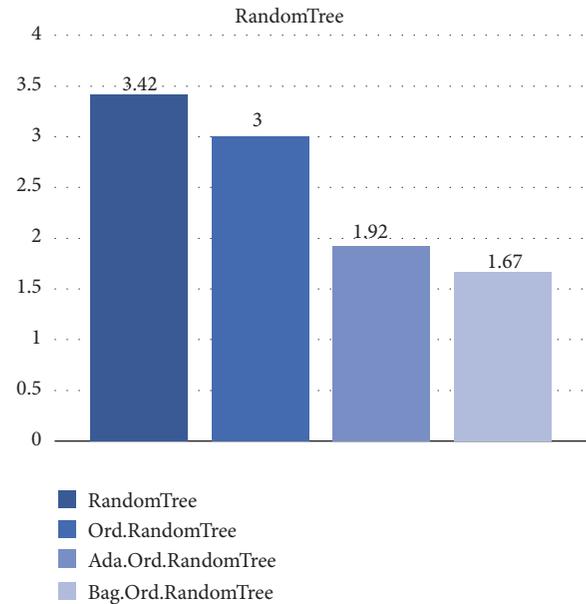


FIGURE 6: The average ranks of RandomTree algorithm with ordinal and ensemble-learning versions.

REPTree) with ordinal and ensemble-learning (AdaBoost and bagging) versions, respectively. In the ranking method, each approach used in this study is rated according to its accuracy score on the dataset. This process is performed by starting with giving rank 1 to the classifier with the highest classification accuracy and continues to increase the rank value of the classifiers until assigning rank m to the worst one of the m classifiers. In the case of tie, average of the classifiers' rankings is assigned to each classifier. Then, mean values of the ranks per classifiers on each datasets are computed as average ranks. According to the comparative results, EBOC approach with bagging method has the best performance among the others in Figures 5 and 6, because it gives the lowest rank value.

The proposed algorithms (Ada.Ord.* and Bag.Ord.*) were applied on the transportation datasets and were compared with ordinal and nominal (standard) classification algorithms in terms of accuracy, precision, recall, F-measure, and ROC area metrics. Additional measures, besides accuracy, are required to evaluate all aspects of the classifiers and they are useful for providing additional insight into their performance evaluations. Table 9 shows the average results obtained from 12 transportation datasets at each experiment. Except accuracy, the metrics listed in Table 9 give a degree ranging from 0 to 1. The algorithm with higher rate means that it is more successful than others. Among C4.5 decision

TABLE 6: The pairwise combinations of the C4.5 algorithm with ordinal and ensemble learning versions.

	Bag.Ord.C4.5	Ada.Ord.C4.5	Ord.C4.5
C4.5	3-9-0	3-9-0	7-4-1
Ord.C4.5	1-11-0	3-9-0	
Ada.Ord.C4.5	6-6-0		
Bag.Ord.C4.5			

TABLE 7: The pairwise combinations of the RandomTree algorithm with ordinal and ensemble learning versions.

	Bag.Ord. RandomTree	Ada.Ord. RandomTree	Ord.RandomTree
RandomTree	1-11-0	2-10-0	4-8-0
Ord.RandomTree	2-10-0	2-10-0	
Ada.Ord.RandomTree	5-7-0		
Bag.Ord.RandomTree			

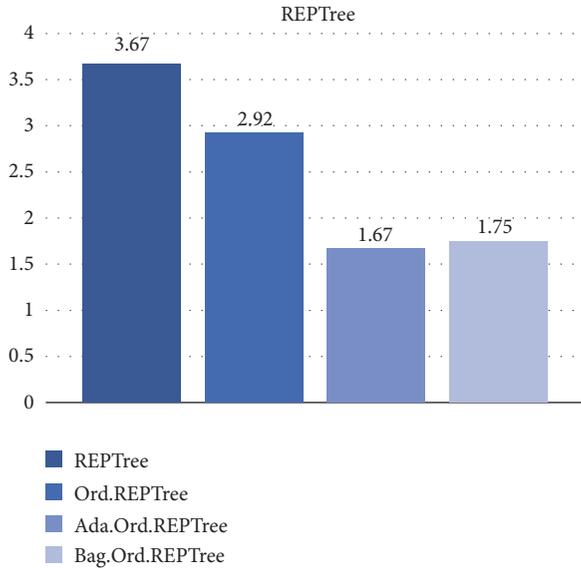


FIGURE 7: The average ranks of REPTree algorithm with ordinal and ensemble-learning versions.

tree-based algorithms, it is clearly seen that Ada.Ord.C4.5 algorithm has the best accuracy (87.467), precision (0.874), recall (0.875), and F-measure (0.874) results. While bagging has the highest values for the RandomTree algorithm, boosting is the best among REPTree-based algorithms. When the ROC area is considered, Bag.Ord.* algorithms have the highest scores according to the experimental results. As a result, it is possible to say that ensemble-based ordinal classification (EBOC) approach provides more accurate and robust classification results than traditional methods.

5.4. Statistical Significance Tests. The field of inferential statistics offers special procedures for testing the significance of differences between multiple classifiers. Although the

ensemble-based ordinal classification (EBOC) algorithms (Ada.Ord.* and Bag.Ord.*) have lower ranking value, we should statistically verify that these algorithms are significantly different from the others. For verification, we used three well-known statistical methods [35]: multiple comparisons (Friedman test and Quade test) and pairwise comparisons (Wilcoxon signed rank test).

The null hypothesis (H_0) for a statistical test is one in which provided classification models are equivalent; otherwise, the alternative hypothesis (H_1) is present when not all classifiers are equivalent.

H_0 : There are no performance differences among the classifiers on the datasets

H_1 : There are performance differences among the classifiers on the datasets

The p -value is defined as the probability, which is a decimal number between 0 and 1, and it can be expressed as a percentage (i.e., 0.1 = 10%). With a small p -value, we reject the null hypothesis (H_0), so the relationship between the classification results is significantly different.

5.4.1. Statistical Tests for Comparing Multiple Classifiers.

Multiple comparison tests (MCT) are used to establish a statistical comparison of the results reported among various classification algorithms. We performed two well-known MCT to determine whether the classifiers have significant differences or not: Friedman test and Quade test.

Friedman Test. The *Friedman test* is a nonparametric statistical test that aims to detect significant differences between the behaviors of two or more algorithms. Initially, it ranks the algorithms for each dataset separately between 1 (smallest) and 4 (largest). In case of ties, the average rank is assigned to them. After that, the Friedman test statistics (χ_r^2) is calculated according to the following:

$$\chi_r^2 = \frac{12}{d * t * (t + 1)} * \sum_{i=1}^t R_i^2 - 3 * d * (t + 1) \quad (5)$$

TABLE 8: The pairwise combinations of the REPTree algorithm with ordinal and ensemble learning versions.

	Bag.Ord.REPTree	Ada.Ord.REPTree	Ord.REPTree
REPTree	1-11-0	1-11-0	2-10-0
Ord.REPTree	1-11-0	2-10-0	
Ada.Ord.REPTree	7-5-0		
Bag.Ord.REPTree			

TABLE 9: The comparison of algorithms in terms of accuracy, precision, recall, F-measure, and ROC area.

Compared Algorithms	Accuracy (%)	Precision	Recall	F-measure	ROC Area
C4.5	86.02	0.861	0.860	0.866	0.897
Ord.C4.5	84.26	0.843	0.842	0.848	0.884
Ada.Ord.C4.5	87.48	0.874	0.875	0.874	0.933
Bag.Ord.C4.5	86.40	0.859	0.864	0.860	0.947
RandomTree	82.29	0.823	0.823	0.823	0.860
Ord.RandomTree	82.34	0.825	0.823	0.824	0.861
Ada.Ord.RandomTree	86.76	0.866	0.868	0.866	0.928
Bag.Ord.RandomTree	87.23	0.868	0.872	0.869	0.947
REPTree	82.26	0.817	0.823	0.817	0.903
Ord.REPTree	83.50	0.831	0.835	0.829	0.908
Ada.Ord.REPTree	86.83	0.868	0.868	0.867	0.925
Bag.Ord.REPTree	85.67	0.852	0.857	0.852	0.939

where d is the number of datasets, t is the number of classifiers, and R_i is the total of the ranks for the i th classifier among all datasets. The Friedman test is approximately chi-square (χ^2) distributed with $t-1$ degrees of freedom (df) and the null hypothesis is rejected, if $x_r^2 > x_{t-1,\alpha}^2$ in the corresponding significance α .

The obtained Friedman test value for the four C4.5-based classifiers is 10.525. Since the obtained value is greater than the critical value ($x_{df=3,\alpha=0.05}^2 = 7.81$), null hypothesis (H_0) is rejected, so it has been concluded that the four classifiers are significantly different. The same situation is also valid for RandomTree-based and REPTree-based algorithms, which have 15.3 and 20.1 test values, respectively. The obtained p -values for the C4.5, RandomTree, and REPTree related EBOC results (Tables 3, 4, and 5) are 0.01459, 0.00158, and 0.00016, which are extremely lower than 0.05 level of significance. Thus, it is possible to say that the differences between their performances are unlikely to occur by chance.

Quade Test. Like the Friedman test, the *Quade test* is a nonparametric test, which is used to prove that the differences among classifiers are significant. First, the performance results of each classifier are ranked within each dataset to yield $R_{i,j}$, in the same way as the Friedman test does, where d is the number of datasets and t is the number of classifiers, for $i = \{1, 2, \dots, d\}$ and $j = \{1, 2, \dots, t\}$. Then, the range for each dataset is calculated by finding the difference between the largest and the smallest observations within that dataset. The obtained rank for dataset i is denoted by Q_i . The weighted average adjusted rank for dataset i with classifier j is then

computed as $S_{i,j} = Q_i * (R_{i,j} - (t + 1)/2)$. The Quade test statistic is then given by

$$\hat{F} = \frac{(d-1)(1/d) \sum_{j=1}^t S_j^2}{\sum_{i=1}^d \sum_{j=1}^t S_{i,j}^2 - (1/d) \sum_{j=1}^t S_j^2} \quad (6)$$

where $S_j = \sum_{i=1}^d S_{i,j}$ is the sum of the weighted ranks for each classifier. The \hat{F} value is tested against the F-distribution for a given α with $df_1 = (k-1)$ and $df_2 = (b-1)(k-1)$ degrees of freedom. If $\hat{F} > F_{(k-1),(b-1)(k-1),\alpha}$ then null hypothesis is rejected. Moreover, the p -value could be computed through normal approximations.

The p -values computed through the Quade statistical tests for the C4.5, RandomTree, and REPTree related EBOC algorithms are 0.013398, 0.000018, and 0.000522, respectively. Test results strongly suggest the existence of significant differences among the considered algorithms, at the level of significance $\alpha = 0.05$. Hence, we reject the null hypothesis which indicates that all classification algorithms have the same performance. Thus, it is possible to say that at least one of them behaves differently.

5.4.2. Statistical Tests for Comparing Paired Classifiers. In addition to *multiple comparison tests*, we also conducted *pairwise comparison test* to demonstrate that the proposed algorithms (Ada.Ord.* and Bag.Ord.*) have significant differences from others when individually compared. Wilcoxon signed rank test was applied to see pairwise performance differences.

TABLE 10: Wilcoxon signed ranks test results.

Compared Algorithms	W^+	W^-	W	z -score	p -value	Significance Level
Ada.Ord.C4.5 vs C4.5	63	15	15	-1.882715	0.059739	moderate
Ada.Ord.C4.5 vs Ord.C4.5	65	13	13	-2.039608	0.041389	strong
Bag.Ord.C4.5 vs C4.5	61	17	17	-1.725822	0.084379	moderate
Bag.Ord.C4.5 vs Ord.C4.5	75	3	3	-2.824072	0.004742	very strong
Ada.Ord.RandomTree vs RandomTree	75	3	3	-2.824072	0.004742	very strong
Ada.Ord.RandomTree vs Ord.RandomTree	75	3	3	-2.824072	0.004742	very strong
Bag.Ord.RandomTree vs RandomTree	77	1	1	-2.980965	0.002873	very strong
Bag.Ord.RandomTree vs vs Ord.RandomTree	75	3	3	-2.824072	0.004742	very strong
Ada.Ord. REPTree vs REPTree	71	7	7	-2.510287	0.012063	strong
Ada.Ord. REPTree vs Ord.REPTree	64	14	14	-1.961161	0.049860	strong
Bag.Ord.REPTree vs REPTree	77	1	1	-2.980965	0.002873	very strong
Bag.Ord.REPTree vs Ord.REPTree	77	1	1	-2.980965	0.002873	very strong
<i>Average</i>	71.25	6.75	6.75	-2.602996	0.022918	

Wilcoxon Signed Rank Test. This statistical test consists of the following steps to check the validity of the null hypotheses: (i) calculate performance differences between two algorithms for each dataset, (ii) order the differences according to their absolute values from smallest to largest, (iii) rank the absolute value of differences, starting with the smallest as 1 and assigning average ranks in case of ties, (iv) calculate W^+ and W^- by summing the ranks of the positive and negative differences separately, (v) find W as the smaller of the sums, $W = \min(W^+, W^-)$, (vi) calculate z -score as $z = W/\sigma_w$ and find p -value from the distribution table, and (vii) determine the significance level according to the computed p -value [36] and reject the null hypothesis if p -value is less than a certain risk threshold (i.e., 0.1 = 10%):

Significance level

$$= \begin{cases} p > 0.1 & \text{weak or none} \\ 0.05 < p \leq 0.1 & \text{moderate} \\ 0.01 < p \leq 0.05 & \text{strong} \\ p \leq 0.01 & \text{very strong} \end{cases} \quad (7)$$

Table 10 demonstrates the W , z -score, and p -values computed through Wilcoxon signed rank tests. Based on the reported p -values, most of the tests strongly or very strongly prove the existence of significant differences among the considered algorithms. As the table states, for each test, the ensemble-based ordered classification algorithms (Bag.Ord.* and Ada.Ord.*) always show a significant improvement over the traditional versions, with a level of significance $\alpha=0.1$. In general, it can be observed that the average p -value of proposed Bag.Ord.* variant algorithms (0.0171) is a little less than Ada.Ord.* variants (0.0288).

Based on all statistical tests (Friedman, Quade, and Wilcoxon signed rank test), we can safely reject the null hypothesis (i.e., there are no performance differences among the classifiers on the datasets), since the p -values computed through the statistics are less than the critical value (0.05). Thus, the proposed approach (EBOC) has a statistically

significant effect on the response at the 95.0% confidence level.

6. Conclusions and Future Works

As a result of developments in the field of transportation, enormous amounts of raw data are generated every day. This situation creates the potential to discover knowledge patterns or rules from it and to model the behaviour of a transportation system using machine learning techniques. To serve the purpose, this study focuses on the application of ordinal classification algorithms on real-world transportation datasets. It proposes a novel ensemble-based ordinal classification (EBOC) approach. This approach converts the original ordinal class problem into a series of binary class problems and use ensemble-learning paradigm (boosting and bagging) at the same time. To the best of our knowledge, this is the first study in which ordinal classification methods and our proposed approach were applied on transportation sector. In the experimental studies, the proposed model with the tree-based learners (C4.5, RandomTree, and REPTree) was implemented on twelve benchmark transportation datasets that are available for public use. The proposed EBOC method was compared with ordinal class classifier and traditional tree-based classification algorithms in terms of accuracy, precision, recall, f-measure, and ROC area. The results indicate that the EBOC approach provides more accurate classification results than them. Moreover, statistical test methods (Friedman, Quade, and Wilcoxon signed rank tests) were used to prove that the classification accuracies obtained from the proposed algorithms (Ada.Ord.* and Bag.Ord.*) are significantly different from the traditional methods. Therefore, the proposed EBOC method can assist in making right decisions in transportation. Our findings demonstrate that the improvement in performance is a result of exploiting ordering information and applying ensemble strategy at the same time.

As future work, different types of ensemble-based ordinal classifiers can be developed using different ensemble methods

such as stacking and voting, and using different classification algorithms such as Naive Bayes, k-nearest neighbour, neural network, and support vector machine. In addition, ensemble clustering models can be improved to cluster transportation data. Furthermore, a comparative study which implements ensemble-learning and deep learning paradigms can be performed in transportation fields.

Data Availability

The “Auto MPG,” “Automobile,” “Bike Sharing,” “Car Evaluation,” and “Statlog (Vehicle Silhouettes)” datasets used to support the findings of this study are freely available at <https://archive.ics.uci.edu/ml/datasets>. The “Car Sale Advertisements,” “NYS Air Passenger Traffic,” “Smart City Traffic Patterns,” “SF Air Traffic Landings Statistics,” and “SF Air Traffic Passenger Statistics” datasets used to support the findings of this study are freely available at <https://www.kaggle.com/datasets>. The “Road Traffic Accidents (2017)” dataset used to support the findings of this study is freely available at <https://datamillnorth.org/dataset>. The “Traffic Volume Counts (2012-2013)” dataset used to support the findings of this study is freely available at <https://opendata.cityofnewyork.us/data>.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

- [1] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic flow prediction with big data: a deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.
- [2] X. Wen, L. Shao, Y. Xue, and W. Fang, “A rapid learning algorithm for vehicle classification,” *Information Sciences*, vol. 295, pp. 395–406, 2015.
- [3] S. Ballı and E. A. Sağbaşı, “Diagnosis of transportation modes on mobile phone using logistic regression classification,” *IET Software*, vol. 12, no. 2, pp. 142–151, 2018.
- [4] H. Nguyen, C. Cai, and F. Chen, “Automatic classification of traffic incident’s severity using machine learning approaches,” *IET Intelligent Transport Systems*, vol. 11, no. 10, pp. 615–623, 2017.
- [5] G. Kedar-Dongarkar and M. Das, “Driver classification for optimization of energy usage in a vehicle,” *Procedia Computer Science*, vol. 8, pp. 388–393, 2012.
- [6] N. Deepika and V. V. Sajith Variyar, “Obstacle classification and detection for vision based navigation for autonomous driving,” in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*, pp. 2092–2097, Udupi, India, September 2017.
- [7] E. Frank and M. Hall, “A simple approach to ordinal classification,” in *Proceedings of the European Conference on Machine Learning (ECML)*, vol. 2167 of *Lecture Notes in Computer Science*, pp. 145–156, Springer, 2001.
- [8] E. Alpaydin, *Introduction to Machine Learning*, The MIT Press, 3rd edition, 2014.
- [9] S. Destercke and G. Yang, “Cautious ordinal classification by binary decomposition,” in *Proceedings of the Machine Learning and Knowledge Discovery in Databases - European Conference ECML/PKDD*, pp. 323–337, Nancy, France, 2014.
- [10] J. S. Cardoso and J. F. Pinto da Costa, “Learning to classify ordinal data: the data replication method,” *Journal of Machine Learning Research (JMLR)*, vol. 8, pp. 1393–1429, 2007.
- [11] L. Li and H. T. Lin, “Ordinal regression by extended binary classification,” in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pp. 865–872, Canada, 2006.
- [12] F. E. Harrell, “Ordinal logistic regression,” in *Regression Modeling Strategies*, Springer Series in Statistics, pp. 311–325, Springer International Publishing, Cham, Switzerland, 2015.
- [13] J. D. Rennie and N. Srebro, “Loss functions for preference levels: regression with discrete ordered labels,” in *Proceedings of the IJCAI Multidisciplinary Workshop Adv. Preference Handling*, pp. 180–186, 2005.
- [14] B. Keith, “Barycentric coordinates for ordinal sentiment classification,” in *Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Halifax, Canada, 2017.
- [15] E. Fernández, J. R. Figueira, J. Navarro, and B. Roy, “ELECTRE TRI-nB: A new multiple criteria ordinal classification method,” *European Journal of Operational Research*, vol. 263, no. 1, pp. 214–224, 2017.
- [16] M. M. Stenina, M. P. Kuznetsov, and V. V. Strijov, “Ordinal classification using Pareto fronts,” *Expert Systems with Applications*, vol. 42, no. 14, pp. 5947–5953, 2015.
- [17] W. Verbeke, D. Martens, and B. Baesens, “RULEM: A novel heuristic rule learning approach for ordinal classification with monotonicity constraints,” *Applied Soft Computing*, vol. 60, pp. 858–873, 2017.
- [18] W. Kotlowski and R. Slowinski, “On nonparametric ordinal classification with monotonicity constraints,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2576–2589, 2013.
- [19] K. Dembczynski, W. Kotlowski, and R. Slowinski, “Ordinal classification with decision rules,” in *Proceedings of the Third International Conference on Mining Complex Data (MCD’07)*, pp. 169–181, Warsaw, Poland, 2007.
- [20] K. Hechenbichler and K. Schliep, “Weighted k-nearest-neighbor techniques and ordinal classification,” *Collaborative Research Center*, vol. 399, pp. 1–16, 2004.
- [21] W. Waegeman and L. Boullart, “An ensemble of weighted support vector machines for ordinal regression,” *International Journal of Computer and Information Engineering*, vol. 1, no. 12, pp. 599–603, 2007.
- [22] K. Dembczynski, W. Kotlowski, and R. Slowinski, “Ensemble of decision rules for ordinal classification with monotonicity constraints,” in *Proceedings of the International Conference on Rough Sets and Knowledge Technology*, vol. 5009 of *Lecture Notes in Computer Science*, pp. 260–267, 2008.
- [23] H. T. Lin, *From Ordinal Ranking to Binary Classification [PhD thesis]*, California Institute of Technology, 2008.
- [24] C. K. A. Cuartas, A. J. P. Anzola, and B. G. M. Tarazona, “Classification methodology of research topics based in decision trees: J48 and randomtree,” *International Journal of Applied Engineering Research*, vol. 10, no. 8, pp. 19413–19424, 2015.
- [25] C. Parimala and R. Porkodi, “Classification algorithms in data mining: a survey,” in *Proceedings of the International Journal of Scientific Research in Computer Science*, vol. 3, pp. 349–355, 2018.

- [26] P. Yildirim, D. Birant, and T. Alpyildiz, "Improving prediction performance using ensemble neural networks in textile sector," in *Proceedings of the 2017 International Conference on Computer Science and Engineering (UBMK)*, pp. 639–644, Antalya, Turkey, October 2017.
- [27] H. Yu and J. Ni, "An improved ensemble learning method for classifying high-dimensional and imbalanced biomedicine data," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 4, pp. 657–666, 2014.
- [28] D. Che, Q. Liu, K. Rasheed, and X. Tao, "Decision tree and ensemble learning algorithms with their applications in bioinformatics," in *Software Tools and Algorithms for Biological Systems*, H. R. Arabnia and Q-N. Tran, Eds., vol. 696, pp. 191–199, Springer, 2011.
- [29] "UCI Machine Learning Repository: Car Evaluation Data Set," Archive.ics.uci.edu, 2018, <https://archive.ics.uci.edu/ml/datasets/car+evaluation>.
- [30] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java," Cs.waikato.ac.nz, 2018, <https://www.cs.waikato.ac.nz/ml/weka/>.
- [31] "UCI Machine Learning Repository," Archive.ics.uci.edu, 2018, <https://archive.ics.uci.edu/ml/index.php>.
- [32] "Datasets — Kaggle," Kaggle.com, 2018, <https://www.kaggle.com/datasets>.
- [33] "Data Mill North," Datamillnorth.org, 2018, <https://datamillnorth.org/dataset>.
- [34] "N.: City of New York, "NYC Open Data"," Open-data.cityofnewyork.us, 2018, <https://opendata.cityofnewyork.us/>.
- [35] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [36] N. A. Weiss, *Introductory Statistics*, Addison-Wesley, 9th edition, 2012.

