Transportation Research Part C 133 (2021) 103414

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

An improved learning-based LSTM approach for lane change intention prediction subject to imbalanced data ☆

Qian Shi [a], Hui Zhang [a,b,*]

[a] School of Transportation Science and Engineering, Beihang University, Beijing 100191, China
[b] Ningbo Institute of Technology, Beihang University, Ningbo 315323, China

ARTICLE INFO

Keywords:
Autonomous Vehicles
Lane change intention prediction
Long Short Term Memory (LSTM) network
Data imbalance
Over-sampling method
Ensemble learning

ABSTRACT

Lane change intention prediction is an essential component for motion planning of Autonomous Vehicles (AVs). In this work, we aim to achieve this task by using Long-Short Term Memory (LSTM) network. One critical challenge on this task is that the dataset used for training such a network is usually highly imbalanced due to the fact that the size of left/right lane change data is much smaller than that of the lane keeping data. The imbalanced dateset would lead to trivial output of LSTM model. To deal with this problem, we propose a hierarchical over-sampling bagging method based on Grey Wolf Optimizer (GWO) algorithm and Synthetic Minority Over-sampling Technique (SMOTE). With the proposed method, more diverse and informative instances of minority classes can be generated for training LSTM model. Furthermore, we also propose a sampling technique to keep the temporal information and make the proposed method applicable to sequential data. Moreover, to further improve the prediction performance, we also take the interactions between neighboring vehicles into account by concatenating their trajectories when constructing features. We evaluate our method against several baseline algorithms over two benchmark datasets and the empirical results validate the effectiveness and efficiency of our method in terms of the indexes of prediction time, F1, and G-mean.

1. Introduction

Autonomous Vehicles (AVs) which remove the human element in driving tasks have been extensively studied by transportation researchers and practitioners with the advancements in sensing, software modules, and vehicle technologies (Cremean et al., 2006; Falcone et al., 2007; Turk et al., 1988). In general, an AV consists of three components: perception, motion planning, and control (Lefevre et al., 2016). Predicting possible lane change behaviors of other vehicles is an essential capability for AVs to safely plan motions (Ding et al., 2019; Lee et al., 2017). This paper aims to predict lane change intention of vehicles for ego AVs (Bakhit et al., 2017).

Data-driven approaches formulate lane change intention detection as a classification problem. Several classification methods in the field of machine learning have been applied to lane change intention prediction, which include the Support Vector Machine (SVM) (Mandalia and Salvucci, 2005; Kumar et al., 2013; Morris et al., 2011), the Hidden Markov Model (HMM) (Hou et al., 2011; Zheng and Hansen, 2017; Pentland and Liu, 1999), the Multi-Layer Perception network (MLP) (Benterki et al., 2020; Peng et al., 2015), the Convolution Neural Network (CNN) (Lee et al., 2017; Fernández-Llorca et al., 2020), Decision Trees (DT) (Hou

☆ This work was supported in part by the National Natural Science Foundation of China under Grant U1864201 and Defense Industrial Technology Development Program.

* Corresponding author at: School of Transportation Science and Engineering, Beihang University, Beijing 100191, China.
E-mail address: huizhang285@gmail.com (H. Zhang).

https://doi.org/10.1016/j.trc.2021.103414
Received 13 December 2020; Received in revised form 19 August 2021; Accepted 21 September 2021
Available online 3 November 2021

et al., 2014), and the Long Short Term Memory (LSTM) network (Mahajan et al., 2020; Xing et al., 2020; Zyner et al., 2017). For instance, Hiren et al. used SVM method to infer driver intentions to change lanes by classifying the features in the proposed feature sets (Mandalia and Salvucci, 2005). Yang introduced a combined HMM method after adopting the pre-processing, pre-filtering, and spectral time–frequency analysis segmentation approach (Zheng and Hansen, 2017). Abdelmoudjib used MLP network to predict lane change intentions of the surrounding vehicles, and integrated it with the trajectory prediction (Benterki et al., 2020). David adopted CNN approach to predict lane change intentions, which relied on visual cues computed from regions of interest (Fernández-Llorca et al., 2020). SVM method was used to automatically label the lane change data clusters, and then the features and labels were input to LSTM model to predict maneuver classes in Mahajan et al. (2020). Bayes classifier and DT methods were also adopted to model lane changes using vehicle trajectory data (Hou et al., 2014). Among these methods, the HMM method and the LSTM network can handle time series input directly and capture the temporal information in the raw data collected in CV environment. Compared with the HMM approach, LSTM network has the advantage of storing information over extended time intervals, and avoids the shortage of HMM method which depends only on the current states (Hochreiter and Schmidhuber, 1997; Graves, 2012; Bhahramani, 2001).

When dealing with the trajectory data collected from a CV environment, one critical issue is that the dataset is usually highly imbalanced, since the lane change data (minority class) is much less than the lane keeping data (majority class) (Benterki et al., 2020). Conventional LSTM methods usually have difficulty learning from imbalanced class problems since their objective is to minimize the overall error rate. This makes LSTM methods implicitly treat all misclassification costs equally (Elreedy and Atiya, 2019; De Morais and Vasconcelos, 2019; Wang and Pineau, 2016). Consequently, the LSTM network trained on such a dataset will produce trivial outputs (Wu et al., 2018; Petmezas et al., 2021) which classify all the instances as the lane keeping class. More importantly, it is often the case that the minority class is the one of greater interest: the expected cost of missing a lane change instance may be much higher than the cost of mis-identifying a lane keeping instance. One popular remedy for this problem is to use re-sampling techniques to form a balanced dataset. Re-sampling can be accomplished either by over-sampling the minority class or under-sampling the majority class. Under-sampling removing examples from the majority class may cause the classifier to miss important information (Tang et al., 2009; Haibo and Edwardo, 2009). Different from under-sampling methods, over-sampling approaches generate minority examples and information loss is avoided. Among the over-sampling approaches, Synthetic Minority Over-Sampling Technique (SMOTE) is probably the most popular over-sampling algorithm in the literature Chawla et al. (2002). It works by selecting two neighboring examples in the minority class and randomly creating new examples from the convex combination of the chosen examples. One issue with SMOTE technique is that the synthetic instances only lie within the line between two minority instances, which leads to the lack of diversity in the new examples. To address this problem, we propose to use Grey Wolf Optimizer (GWO) algorithm to modify SMOTE technique. In particular, the synthetic samples are generated from the computation of GWO algorithm (Mirjalili et al., 2014) which obtains the new samples from the combination of four examples. After the computations, more diverse examples beyond the range of line segments in SMOTE technique can be generated.

For lane change intention prediction, the relationship between the features collected at different sampling instants can reflect the motion of the vehicles. This relationship is represented as the temporal information. Traditional SMOTE method is designed for feature vectors, where vector is regarded as the minimal manipulation unit. In traditional SMOTE, feature vectors sampled at different instants compose the data pool. Vectors in data pool are equally handled to generate the synthetic vectors, that loses the time stamps. However, the input for LSTM is in the form of sequence. The typical approach which generates sequence from these vectors will lead to the loss of temporal information. In this research, we propose a modified over-sampling technique which regards the sequence as the minimal manipulation unit and handles the vectors sampled at different time differently. The synthetic over-sampling sequence is generated following the order of sampling instant. In this way, the temporal structure of the trajectory sequence is kept to a great extent. Ensemble learning approaches combining with SMOTE techniques have been shown to be particularly effective when dealing with imbalanced classification problem (Wang and Yao, 2009). In this paper, we will therefore combine ensemble method with the proposed GWO-SMOTE approach to further improve the classification accuracy for minority data.

To further reduce the imbalance ratio and avoid the ineffectiveness of LSTM classification, a hierarchical classification structure is proposed. The lane change and lane keeping classifier training is firstly accomplished, and then with all the lane change data as the training data, the classifier for left lane change/right lane change is acquired via LSTM algorithm. In this way, the original 3-label classification of lane keeping, left lane change, and right lane change is transformed into hierarchical binary classifications of lane keeping/lane change, and left lane changing/right lane change. The highly imbalanced classifications including lane keeping/left lane change and lane keeping/right lane change are avoided.

The contributions and highlights of this work are summarized as follows:

(1) We take the data imbalance into consideration when training the lane change intention predictor and modify the SMOTE method by combining the GWO algorithm.
(2) We propose a modified over-sampling method which can generate imbalanced trajectory sequence and keep the temporal information in the newly formed sequence.
(3) We use a hierarchical structure to predict the lane change intentions, where the lane change/lane keeping intention is firstly classified. Then, the left lane change/right lane change prediction is made.

Notations. In this paper, the following notations are adopted. Superscript $\top$ represents matrix transposition. $A(i)$ stands for the ith element of vector A. D stands for training dataset consisting of N sequence records $\mathbb{X}_1$, $\mathbb{X}_2$, …, $\mathbb{X}_N$ and the related labels y_1, y_2, …, y_N. For each sequence record $\mathbb{X}_k$, the elements include the vectors X_k^1, X_k^2, …, $X_k^{T_{\text{win}}}$, which represent the features sampled from $t = 1$ to $t = T_{\text{win}}$. D^- and D^+ denote the minority dataset and the majority dataset, respectively.

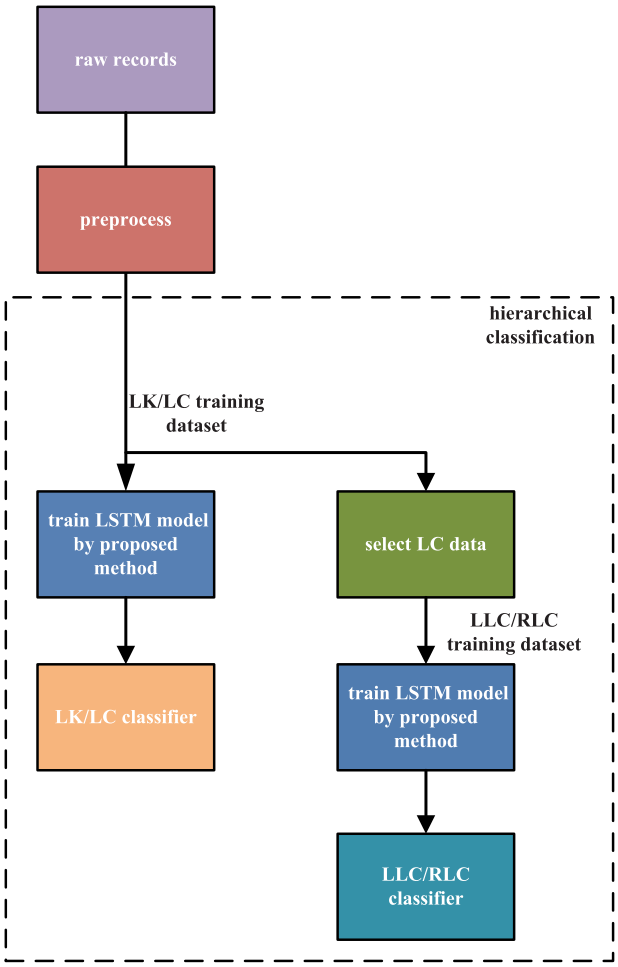

Fig. 1. Framework of lane change intention prediction (Different block colors indicate different procedure types) .

2. Background of lane change intention prediction based on LSTM method

2.1. Framework of lane change intention prediction

In this paper, the prediction of subject vehicle's lane change intention is studied which includes the following 3 categories:

- intention 1: LLC (left lane change),
- intention 2: RLC (right lane change),
- intention 3: LK (lane keeping).

The proposed framework of lane change intention prediction can be found in Fig. 1. After preprocessing the raw data, the training and testing data is generated.

For the obtained training dataset, the size of left lane change data or right lane change data is largely smaller than that of the lane keeping data. In this study, with the consideration of the following two facts: (1) the lane changing data which includes the left lane change data and the right lane change data is less than the lane keeping class; (2) the pattern difference associated with the lane change process is more significant between left lane change data and right lane change data, we propose to use a hierarchical classification structure. In the hierarchical classification structure, we firstly train the lane change and lane keeping classifier by using the LSTM method. Then, with all the lane change data as the training data, the classifier for left lane change/right lane change is acquired via LSTM method. In this way, the highly imbalanced classification between lane keeping and left lane change or lane keeping and right lane change is avoided. The procedures are depicted in Fig. 1.

Data imbalance may still exist in the lane change/lane keeping classifier training and left lane change/right lane change classifier training. To further reduce the effect of imbalance, we propose a GWO-SMOTE with bagging method. Specifically, after over-sampling for minority data via GWO-SMOTE method, several training datasets are sampled with replacement. Then bagging strategy is applied which trains M LSTM weak classifiers from the training datasets as Fig. 2 shows.

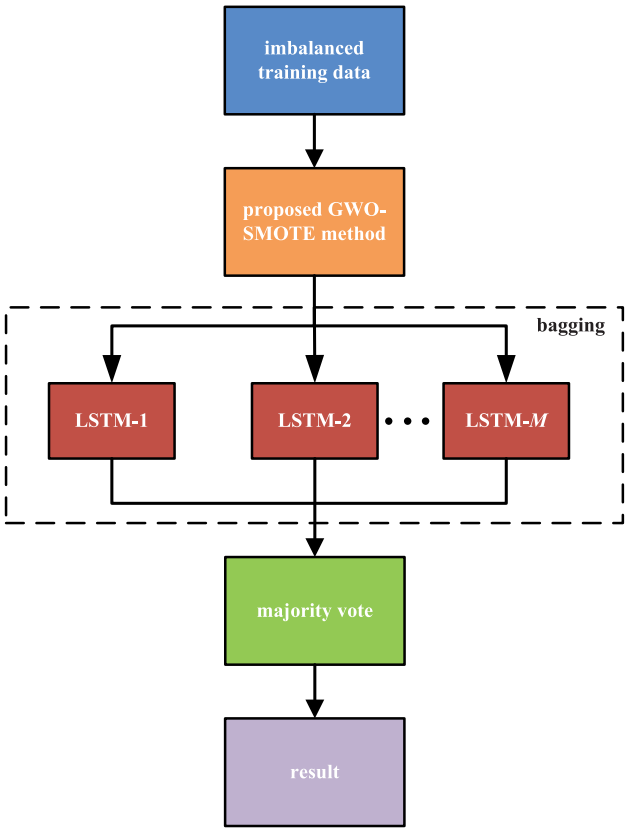

Fig. 2. Flowchart of proposed classifier training method (The colors of blocks distinguish the procedure types).

2.2. Dataset description and data preprocess

Lane change intention prediction of the target vehicle should consider interactions between surrounding vehicles and target vehicle. The target vehicle and surrounding vehicles are plotted in Fig. 3. The surrounding vehicles plotted in Fig. 3 are selected which include the preceding vehicle and the following vehicle in the same lane with the target vehicle, and the preceding and following vehicles in two adjacent lanes. This information cannot be provided by onboard sensors of ego vehicle since the sensors are egocentric.

One possible solution to provide the required information is vehicle-to-infrastructure technique. In the vehicle-to-infrastructure structure, roadside sensors are mounted on the infrastructure. Roadside sensors such as laser scanner, camera system, radio frequency identification system, and wireless sensor network, can provide full observability and measure the position, speed, and ID information of road users (Ojala et al., 2020). What is more, these measurements can be transmitted to ego AVs utilizing a wireless vehicle-to-infrastructure communication. Lane change intention prediction is based on the traffic condition at present and in the immediate past. In vehicle-to-infrastructure environment, ego AVs use received measurements which reflect the traffic condition at present and in the past for predicting the lane change intention of subject vehicles.

However, the vehicle-to-infrastructure technique has not been fully implemented. In this paper, the Next Generation Simulation (NGSIM) public dataset gathered in America is used instead. We adopt the dataset of the detailed vehicle trajectory collected on southbound segment of US Highway 101 (US 101) in Los Angeles, CA, which is approximately 640 meters long and consists of five mainline lanes, and a eastbound segment of Interstate 80 (I-80) in Emeryville, CA, which is approximately 500 meters long and consists of six freeway lanes. The dataset contains 0.1s time series about the vehicle unique identifier, precise local and global lateral and longitudinal locations, and the lane ID for every vehicle traveling over the study segment. The lateral and longitudinal local position sequences of the subject vehicle and surrounding vehicles are selected as the classification features after balancing the classification performance and the efficiency. For the situation where there are no surrounding vehicles, the features are set as zero.

The feature vector X_k^t at sampling instant t is formed by concatenating the position vectors of subject vehicle and surrounding vehicles collected at time instant t. In the following, this manipulation is denoted as the mathematical integration of trajectories of surrounding vehicles and subject vehicle. With the mathematical integration manipulation, X_k^t is in the form of:

$$X_k^t = [p_{k,1}^t \; q_{k,1}^t \; p_{k,2}^t \; q_{k,2}^t \; p_{k,3}^t \; q_{k,3}^t \; p_{k,4}^t \; q_{k,4}^t \; p_{k,5}^t \; q_{k,5}^t \; p_{k,6}^t \; q_{k,6}^t \; p_{k,7}^t \; q_{k,7}^t]^\top \tag{1}$$

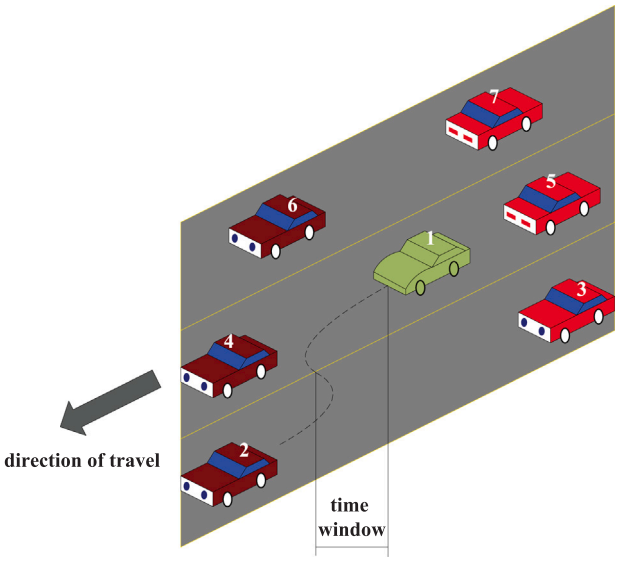

Fig. 3. Data preparation for interacted traffic (The green car is the target vehicle and other colors indicate the surrounding vehicles) .

where p and q represent the lateral local position and longitudinal local position, respectively, the superscript denotes the time instant, the subscript k stands for the number of the sequence or the number of subject vehicle, and the subscripts 1, 2, 3, 4, 5, 6, 7 as Fig. 3 shows are the numbers for the subject vehicle and the corresponding surrounding vehicles, respectively. In this way, the interactions between the surrounding vehicles and the subject vehicle are considered.

Note that, in Fig. 3, the dotted line represents the future trajectory for subject vehicle. We firstly define a time window for collecting data as shown in Fig. 3. The feature vectors collected in the time window are used as one vector of time series to be classified. Since the last sample in the window offers the latest information about the driving state, the features are labeled based on the lane change state of the last sample in the time window. Lane change is defined for the case when the vehicle moves across the lane boundary and never returns. As for NGSIM dataset, since the lane ID is documented, the ground truth is determined by analyzing the changes of vehicle lane ID. The last sample in the time window is collected at the instant when the lane ID changes.

2.3. LSTM algorithm for lane change intention prediction

When the dataset is prepared well, the lane change intention prediction problem can be formulated into a classification problem. The classification problem aims to classify the collected trajectory time series to the classes of left lane change, right lane change, and lane keeping. The LSTM algorithm is adopted in this research to train the classifier. In the training dataset, we have N vectors of time series $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_N$ where $\mathbb{X}_k$ with $k = 1, 2, \ldots, N$ represents the trajectory sequence which is mathematically integrated from kth subject vehicle and the related surrounding vehicles, and the corresponding lane change labels $y_1, y_2, \ldots, y_N$. The computation for a single LSTM unit is realized via

$$\begin{aligned} z_k &= \tanh(W \cdot [X_k^t; h_k^{t-1}; 1]), \\ z_k^i &= \delta(W^i \cdot [X_k^t; h_k^{t-1}; 1]), \\ z_k^f &= \delta(W^f \cdot [X_k^t; h_k^{t-1}; 1]), \\ z_k^o &= \delta(W^o \cdot [X_k^t; h_k^{t-1}; 1]), \end{aligned} \tag{2}$$

where tanh stands for tanh function, δ represents sigmoid function, W, W^i, W^f, W^o are row vectors which stand for the weights combined with bias parameters for the LSTM cell, input gate, forget gate, and output gate, respectively, z_k represents the scalar output of the LSTM cell for kth sequence, z_k^i, z_k^f, z_k^o represent the scalar outputs of input gate, forget gate, and output gate for kth sequence, respectively, $[X_k^t; h_k^{t-1}; 1]$ is a column vector combined by column vector X_k^t, h_k^{t-1}, and 1. Column vector X_k^t is the feature vector sampled at time instant t in the sequence $\mathbb{X}_k$, scalar h_k^{t-1} is the hidden state of LSTM cell for kth sequence at $(t-1)$th time instant (Hochreiter and Schmidhuber, 1997).

The scalar hidden state h_k^t for LSTM unit is calculated as:

$$h_k^t = z^o \cdot \tanh(c^t), \tag{3}$$

where c^t denotes the cell state and

$$c^t = z_k^f \cdot c^{t-1} + z_k^i \cdot z_k. \tag{4}$$

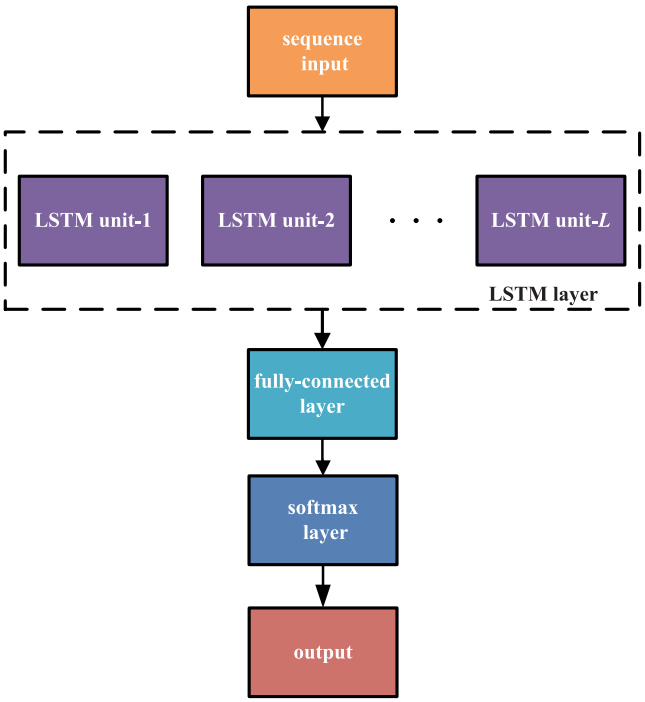

Fig. 4. The fundamental framework of LSTM networks (The block color represents the layer type) .

Denoting the lane change moment when the vehicle lane ID changes as T_{win}, the equations in Eq. (2) are recursively calculated for input sequence with T_{win} vectors by starting at $t = 1$ while incrementing t. All states and activations are initialized to zero at $t = 0$.

With the single unit computation procedure in Eq. (2), the LSTM layer consisting of L LSTM units calculates L outputs and transmits them to fully-connected layer, softmax layer sequentially, as illustrated in Fig. 4. The fully-connected layer adopts the following computation:

$$o_k = \omega \cdot H_k^{T_{\text{win}}}, \tag{5}$$

where ω is the fully-connected layer parameter matrix of dimension $2 \times L$, o_k is the output vector of the output layer, and $H_k^{T_{\text{win}}}$ is a L-dimensional vector consists of the hidden state of the L LSTM units at $t = T_{\text{win}}$. Then, the output is conveyed to softmax layer, which can be calculated as:

$$a_k(i) = \frac{e^{o_k(i)}}{\sum_{i=1}^{2} e^{o_k(i)}}, i = 1, 2 \tag{6}$$

where $o_k(i)$ is the ith element of o_k. The goal function for LSTM network classification is cross-entropy which is defined as:

$$L_o = -\sum_k \sum_i y_k(i) \cdot \ln(a_k(i)). \tag{7}$$

The classifier is obtained after optimizing the goal function for the parameters W, W^i, W^f, W^o, and ω for training dataset. Lane change intention prediction can be achieved by applying the obtained classifier to the test data. However, the data is imbalanced for the reason that the lane keeping records are highly more than the lane change records. For the reason that the goal function in Eq. (7) considers the classification of minority data and majority data equally (Wozniak et al., 2013; Sun et al., 2007), the traditional LSTM network does not work.

3. Improved LSTM approach for imbalanced data

The hierarchical structure is proposed to transform the multi-label classification to binary-label classification and reduce the impact of data imbalance. However, data imbalance still exists within the binary-label classification which would lead to trivial results of classifiers. Over-sampling method and bagging strategy are techniques used widely to deal with this. In this section, after introduction of bagging and SMOTE with bagging method, a method called GWO-SMOTE with bagging method is proposed.

3.1. Introduction of bagging and SMOTE with bagging method

Ensemble learning can improve both the accuracy and generalization of the final classifier (Breiman, 1996). In ensemble learning, several base weak classifiers are firstly trained from different datasets. The classification results are generated via vote from the base weak classifiers (Yang et al., 2020). Various types of ensemble methods have been researched, such as the bagging method (Breiman, 1996), Adaboost method (Freund and Schapire, 1995), negative correlation learning (Liu and Yao, 1999), and multistage ensemble learning model (Yu et al., 2008). To sum up, differences between ensemble methods lie in the approach to generate the datasets and to average the results of base classifiers. Among these methods, the bagging technique is discussed and adopted in this paper for its simplicity.

In the bagging technique, several bootstrap sample sets are firstly formed by randomly selecting samples from the dataset with replacement. Then, the algorithm is applied to these bootstrap sample sets. The algorithm for bagging technique is shown in Algorithm 1 where S_m is the bootstrap sample set. Note that the diversity among the classifiers is key to the effectiveness. In Algorithm 1, diversity is introduced by independently constructing different subsets S_m. x is the formal parameter for feature, y represents the formal parameter for label, and Y stands for the label set.

Algorithm 1 bagging algorithm (Breiman, 1996)

Input: D: dataset, M: number of base learners
Output: $H(x)$: final classifier

for $m = 1, 2, \dots, M$ **do**
  Sample with replacement from D and form dataset S_m;
  Train a base learner h_m using S_m;
end for
return $H(x) = \text{argmax}_{y \in Y} \sum_{m=1}^{M} I(h_m(x) = y)$

Combination of SMOTE algorithm with bagging technique is proposed in Wang and Yao (2009), which is shown to have the ability to reduce the data imbalance effect. In the SMOTE with bagging technique, after generating the synthetic points in the minority class, datasets with different imbalance ratios are formed. Then, the datasets are used as the training data to generate the bootstrap samples for bagging which is shown in Algorithm 2. The diversity is further boosted by varying $\frac{m}{M}$ in Algorithm 2. Note that the SMOTE method generates synthetic examples by interpolating the minority examples as shown in Algorithm 2, and this leads to the loss of diversity in the synthetic examples.

Algorithm 2 SMOTE with bagging algorithm (Wang and Yao, 2009)

Input: D: dataset, M: number of base learners, $C > 1$: sampling rate, k: number of nearest neighbors, ρ: over-sampling rate
Output: $H(x)$: final classifier

function SMOTE(D, ρ, k)
  Count the size of D as N;
  for $i = 1, 2, \dots, N$ **do**
    for $j = 1, 2, \dots, \rho$ **do**
      Choose randomly one point x_n from the nearest neighbors of x_i;
      Calculate the distance between x_n and x_i;
      Generate random number γ within the range of [0, 1];
      Generate the new data point $x_{n,g}^{j} = x_n + \gamma(x_n - x_i)$;
    end for
  end for
  Combine all the $x_{n,g}^{j}$, $n = 1, 2, \dots, N$, $j = 1, 2, \dots, \rho$ as X_g;
return X_g
end function
for $m = 1, 2, \dots, M$ **do**
  Sample the majority data with replacement and form D^+;
  Sample $\frac{m}{M}C$ instances from the minority data with replacement and form D_r^-;
  Generate $C(1 - \frac{m}{M})$ instances from minority data set and form D_m^-;
  $D_g^- \leftarrow$ SMOTE(D_m^-, ρ, k);
  Combine D^+, D_r^-, and D_g^- as S_m;
  Train a base learner h_m using S_m;
end for
return $H(x) = \text{argmax}_{y \in Y} \sum_{m=1}^{M} I(h_m(x) = y)$

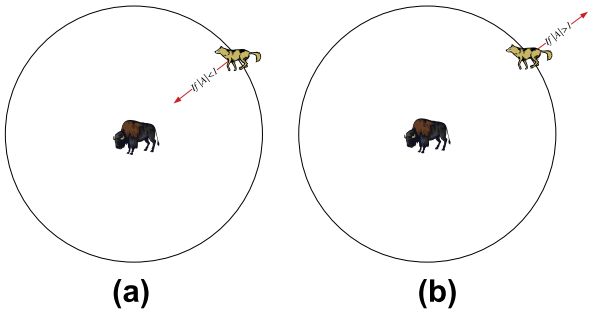

Fig. 5. Searching and attacking the prey of GWO algorithm (Mirjalili et al., 2014).

3.2. GWO-SMOTE with bagging method

GWO algorithm is proposed as a heuristic swarm intelligence optimization method where optimal solution is the position of the prey of grey wolves. GWO algorithm gets the optimal position by randomly adjusting the distances between the best wolves of which the cost function value is lowest and rest wolves. The calculation details can be found in the work Mirjalili et al. (2014). With the analogy between the position of the wolves and the minority class samples, the main manipulations are similar in GWO algorithm and SMOTE technique. Both of them choose one subject sample, select the candidate samples for the subject sample, and generate the new data points between candidate samples and present sample.

However, there are some advantages in the GWO algorithm compared to the SMOTE method. Firstly, the GWO algorithm has the superiority of meta-heuristics which include: simplicity, and flexibility (Mirjalili et al., 2014). Secondly, the GWO algorithm has the exploration ability which means that the GWO algorithm can get solutions or data samples which jump out of ranges of the initial solution or data samples. This feature is lacked in the present SMOTE algorithms which can only generate data points in the line segment between the minority samples.

Inspired by the GWO algorithm, the proposed GWO-SMOTE method is formulated as:

$$\begin{aligned} D_\alpha &= P_\alpha - P_\omega \\ D_\beta &= P_\beta - P_\omega \\ D_\delta &= P_\delta - P_\omega \\ P_{\mathrm{GWO1}} &= P_\alpha - A_1 * D_\alpha \\ P_{\mathrm{GWO2}} &= P_\beta - A_2 * D_\beta \\ P_{\mathrm{GWO3}} &= P_\delta - A_3 * D_\delta, \end{aligned} \tag{8}$$

where A_1, A_2, A_3 are coefficient vectors; P_ω is the candidate sample in the minority class; P_α, P_β, and P_δ represent the three selected data points in the minority class for P_ω respectively; D_α, D_β and D_δ denote the distance vectors between the selected minority sample P_ω and the samples P_α, P_β, P_δ, respectively. In Eq. (8), '$*$' is the element-wise product symbol. The first three equations in Eq. (8) calculate the distance between the minority point P_ω and the minority points P_α, P_β, P_δ, respectively. Three minority samples are generated from the calculation of last three equations in Eq. (8). It should be noted that in procedures of GWO-SMOTE method, the swarm size is set by users.

The parameter vectors A_i in the GWO algorithm are calculated as Mirjalili et al. (2014):

$$A_i = 2a \cdot r_i - a, \tag{9}$$

where i can be 1, 2, and 3; a is a vector of the same dimension with optimizing variable or minority data points, components of a are linearly decreased from 2 to 0 over the course of iterations, and r_i are random vectors in the range of (0, 1). With the parameter a varying within the limit of [0, 2], A_i is in the range of [0, 2]. In Fig. 5, the prey can be regarded as P_α, P_β, and P_δ, and the wolf can be seen as P_ω. As Fig. 5 shows, candidate solutions P_ω tend to leave the minority samples P_α, P_β, and P_δ when $A_j > 1$ and approach towards the them when $A_j < 1$. When $A_j > 1$, the deviation of P_ω from the minority samples P_α, P_β, and P_δ further boosts the diversity of GWO-SMOTE method. Note that the value of parameter a can be set by users to seek a balance between classification performance and data diversity.

Synthetic minority sequences should be meaningful as a trajectory and the temporal structure of the minority sequence should be kept. When applying the GWO-SMOTE method, the sequence $\mathbb{X}_k$ is randomly selected from the training dataset which consists of vectors X_k^1, X_k^2, …, $X_k^{T_{\mathrm{win}}}$, then the sequence is over-sampled vector by vector from $t = 1$ to $t = T_{\mathrm{win}}$. The subject vector is adopted as P_ω in GWO-SMOTE method and the three nearest vectors at the same sampling instant in other minority sequences are searched by K-Nearest Neighbor (KNN) method (Cover and Hart, 1953), which act as X_α, X_β, X_δ, respectively in Eq. (8). The synthetic minority vector is the average of X_α, X_β, and X_δ. Taking X_k^1 as an example, after calculation of Eq. (8), the generated vector $X_{k,g}^1$ is solved out by

$$\begin{aligned} &X_{k,g}^1 = rd_1 \cdot P_{\mathrm{GWO1}} + rd_2 \cdot P_{\mathrm{GWO2}} + rd_3 \cdot P_{\mathrm{GWO3}}, \\ &rd_1 + rd_2 + rd_3 = 1, \\ &rd_1 > 0, rd_2 > 0, rd_3 > 0, \end{aligned} \tag{10}$$

where rd_1, rd_2, and rd_3 denote three random numbers. $X_{k,g}^2$, $X_{k,g}^3$, …, $X_{k,g}^{T_{\text{win}}}$ are generated from X_k^2, X_k^3, …, $X_k^{T_{\text{win}}}$ following the same procedure in Eq. (10). A new sequence X_g is generated combining the vectors $X_{k,g}^1$, $X_{k,g}^2$, …, $X_{k,g}^{T_{\text{win}}}$. The synthetic sequence is generated by integrating the vectors of different time instants in time order such that the hidden temporal structure in the time sequence for lane keeping, left lane changing, and right lane changing are kept.

After labeling the generated minority sequences with the minority class label, the new training dataset $\mathcal{D}_g^-$ forms and the GWO-SMOTE procedure ends. The algorithm for applying the GWO-SMOTE method with bagging in the training procedure is summarized in Algorithm 3, where the structure is similar with that of SMOTE with bagging method in Algorithm 2.

Algorithm 3 GWO-SMOTE with bagging algorithm for sequence data

Input: $\mathcal{D}$: dataset, M: number of base learners, N_{GWO}: size of the grey wolf swarm, $N_{GWO} \le N_{minority}$; $C > 1$: sampling rate
Output: $H(x)$: final classifier

function GWO-SMOTE($\mathcal{D}^-$, N_{GWO})
  Initialize the grey wolf population with the minority class sequence;
  Initialize $a(t)$;
  $t \leftarrow 1$;
  $i \leftarrow 1$;
  while $i \le N_{GWO}$ **do**
    Randomly choose one sequence X_k from minority class;
    while $t \le T_{\text{win}}$ **do**
      $P_\omega \leftarrow X_k^t$;
      Find the nearest three tth vectors P_α, P_β, P_δ in the minority sequences;
      Update parameter A_i by Eq. (9);
      Update $P_{\text{GWO1}}, P_{\text{GWO2}}, P_{\text{GWO3}}$ from Eq. (8);
      Generate random number $rd_1 > 0$, $rd_2 > 0$, $rd_3 > 0$;
      Generate $X_{k,g}^t = rd_1 * P_{\text{GWO1}} + rd_2 * P_{\text{GWO2}} + rd_3 * P_{\text{GWO3}}$;
      $t \leftarrow t + 1$
    end while
    Save $[X_{k,g}^1, X_{k,g}^2, \ldots, X_{k,g}^{T_{\text{win}}}]$ as sequence $\mathbb{X}_{i,g}$;
    $i \leftarrow i + 1$
  end while
  Combine $\mathbb{X}_{1,g}$, $\mathbb{X}_{2,g}$, …, $\mathbb{X}_{N_{GWO},g}$ as $\mathcal{D}_g^-$;
return $\mathcal{D}_g^-$
end function
for $m = 1, 2, \ldots, M$ **do**
  Sample the majority data with replacement and form $\mathcal{D}^+$;
  Sample $\frac{m}{M}C$ instances from the minority data with replacement and form $\mathcal{D}_r^-$;
  Generate $C(1 - \frac{m}{M})$ instances from minority data and form $\mathcal{D}_m^-$;
  $\mathcal{D}_g^- \leftarrow$ GWO-SMOTE($\mathcal{D}_m^-$,N_{GWO});
  Combine $\mathcal{D}^+$, $\mathcal{D}_r^-$, and $\mathcal{D}_g^-$ as S_m;
  Train a base learner h_m using S_m;
end for
return $H(x) = \text{argmax}_{y \in Y} \sum_{m=1}^{M} I(h_m(x) = y)$

3.3. Evaluation of prediction accuracy

The algorithm accuracy which is an important evaluation criterion is calculated as Eq. (11). The accuracy may be misleading when used as the performance evaluation criterion for imbalanced classification, since a high value of accuracy may be generated with the misclassification of minority class. In this paper, the typical criterions G-mean and F1 for imbalanced dataset are adopted, which are calculated based on the fusion matrix in Table 1. In Table 1, TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively. G-mean and F1 criterions are calculated by Eqs. (14) and (17) (Shi and Zhang, 2020). G-mean and F1 values are in the range of [0, 1], and for both of them a value closer to 1 represents a better performance.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{11}$$

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{13}$$

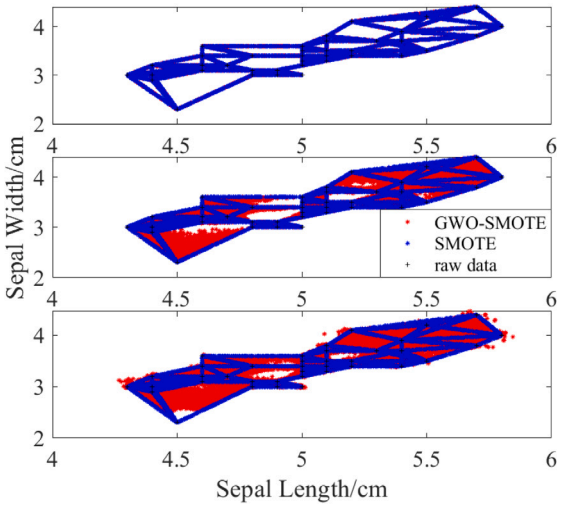

Fig. 6. Synthetic Iris data generated by SMOTE and GWO-SMOTE with different parameter settings.

$$\text{G-mean} = \sqrt{\text{sensitivity} \cdot \text{specificity}} \tag{14}$$

$$\text{P} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{15}$$

$$\text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

$$\text{F1} = \frac{2\text{P} \cdot \text{R}}{\text{P} + \text{R}} \tag{17}$$

Remark. The proposed method adopts the GWO algorithm which is an optimization algorithm as an approach to generate more diverse synthetic samples in over-sampling technique. This can be used as an alternative to generate minority data in imbalanced learning tasks. What is more, we propose a sequence-handling approach which takes the temporal information in the sequence data into consideration when applying the over-sampling technique. The sequence-handling technique can be used in the future sequential data learning tasks when adopting over-sampling technique.

4. Experiment results

To show the difference between SMOTE technique and proposed GWO-SMOTE technique, 50 data points of class setosa in public Iris data are input to SMOTE and the proposed GWO-SMOTE. The sepal length and sepal width features are selected from the four features in order to clearly show the difference in 2-dimensional figures. We analyze three types of parameter setting for GWO-SMOTE and generate 50,000 data points by SMOTE and GWO-SMOTE. The data points are plotted in Fig. 6. The upper figure in Fig. 6 is for the case A1 = 0, A2 = 0, A3 = 0, rd1 = rand, rd2 = 1 − rd1, rd3 = 0 (rand means random number in the range of [0, 1]). With this setting, it can be seen that the generated data by SMOTE and GWO-SMOTE is the same. The SMOTE method is a special case of GWO-SMOTE method. In the middle figure in Fig. 6, A1 = rand, A2 = rand, A3 = rand, rd1 = rand, rd2 = rad1, and rd3 = 1 − rd1 − rd2. In this case, the data points which are not in the line segment between the raw data are generated by GWO-SMOTE method. This cannot be realized by SMOTE. The data diversity is improved by GWO-SMOTE method. However, the generated data by GWO-SMOTE is still bounded by range formed by the raw data. When A1 = rand-0.3, A2 = rand-0.3, A3 = rand − 0.3, rd1 = rand/3, rd2 = rand/3, rd3 = 1 − rd1 − rd2, the data points out the range formed by raw data are generated in the last figure in Fig. 6.

Then, we use the public traffic data I-80 and US101 to validate the effectiveness of proposed method when predicting lane change intentions. We refer to I-80 dataset as Dataset 1 and US101 dataset as Dataset 2. Then, 2/3 of dataset is selected randomly as training data and the rest data is used for testing. The training set characteristics are shown in Tables 2 and 3, respectively. Note that each vector of time series denotes a feature set of one subject vehicle. In Table 2, 11.67 is data imbalance ratio between left lane change and lane keeping, while 2.75 is for right lane change and lane keeping. It can be seen from Table 2 that by adopting

Table 1
Confusion matrix.

	Positive prediction	Negative prediction
Real positive sample	TP	FN
Real negative sample	FP	TN

Table 2
Characteristics of Dataset 1.

Dataset	Size	Dimension	Imbalance ratio
LK/LLC	1280	14	11.67
LK/RLC	1608	14	2.75
LK/LC	1709	14	2.22
RLC/LLC	530	14	4.25

Table 3
Characteristics of Dataset 2.

Dataset	Size	Dimension	Imbalance ratio
LK/LLC	1677	14	8.02
LK/RLC	1808	14	4.70
LK/LC	1994	14	2.96
RLC/LLC	353	14	4.80

Table 4
Parameters for algorithm.

Parameter	Value
k	5
M	11
ρ	1
C for LK/LC	1
C for LLC/RLC	3
N_{GWO}	$C(1-\frac{m}{M})$
L	100
lr	0.001
a	1

Table 5
Training data and test data performance.

Dataset	Method	Data type	F1	G-mean
1	LK/LC	Training	0.77 ± 0.03	0.80 ± 0.03
1	LK/LC	Test	0.68 ± 0.03	0.78 ± 0.03
1	LLC/RLC	Training	0.50 ± 0.39	0.60 ± 0.33
1	LLC/RLC	Test	0.50 ± 0.39	0.60 ± 0.33
2	LK/LC	Training	0.90 ± 0.00	0.92 ± 0.00
2	LK/LC	Test	0.78 ± 0.03	0.88 ± 0.03
2	LLC/RLC	Training	0.58 ± 0.30	0.67 ± 0.30
2	LLC/RLC	Test	0.59 ± 0.24	0.68 ± 0.21

the hierarchical structure which transforms the 3-label classification to binary classifications, the imbalance ratio is reduced and the same conclusion can be obtained from the data in Table 3.

For implementation of the proposed method, we use LSTM toolbox in MATLAB R2018b to train our models. The 'adam' optimization algorithm is adopted in the toolbox for training (Kingma and Ba, 2015). The training stops when training epoch reaches 120 for ensuring model performance and preventing data over-fitting. The minibatch size is 27. In Table 4, the training process parameters, and the parameters of the proposed algorithm and the comparison algorithms are given. Note that lr represents learning rate in Table 4. After calculation of the algorithms, the criterions in Section 3.3 are calculated. The training and test data performance are shown in Table 5. It can be concluded that the over-fitting phenomenon is avoided. Note that the confidence level of the statistical results in this paper is 95%.

In this research, the length of the time window T_{win} which adequately captures the lane change patterns is a free parameter and therefore left to experimentation. Window sizes varying between 2 s to 4 s are analyzed. The results for various window sizes are given in Table 6 where we use 'LC' to represent lane change data which includes left lane change data and right lane change data.

Then, we use 2 s as the window size for validating the performance of the proposed method. The accurate prediction of the lane change intention is important. In this paper, comparisons between the proposed hierarchical GWO-SMOTE with bagging LSTM

Table 6
Test data results of various window sizes.

Dataset	Method	T_{win} (s)	F1	G-mean
1	LK/LC	2	0.68 ± 0.03	0.78 ± 0.03
1	LK/LC	3	0.69 ± 0.03	0.78 ± 0.00
1	LK/LC	4	0.68 ± 0.00	0.78 ± 0.03
1	LLC/RLC	2	0.50 ± 0.13	0.60 ± 0.11
1	LLC/RLC	3	0.42 ± 0.21	0.54 ± 0.21
1	LLC/RLC	4	0.45 ± 0.30	0.56 ± 0.30
2	LK/LC	2	0.78 ± 0.01	0.88 ± 0.01
2	LK/LC	3	0.78 ± 0.00	0.87 ± 0.00
2	LK/LC	4	0.76 ± 0.03	0.87 ± 0.03
2	LLC/RLC	2	0.59 ± 0.24	0.68 ± 0.21
2	LLC/RLC	3	0.61 ± 0.09	0.70 ± 0.06
2	LLC/RLC	4	0.62 ± 0.09	0.71 ± 0.06

Table 7
Test data result comparisons for Dataset 1.

Method	P	R	F1	Sensitivity	Specificity	G-mean
LK/LC by NS	0.63 ± 0.03	0.75 ± 0.03	0.68 ± 0.03	0.75 ± 0.03	0.81 ± 0.03	0.78 ± 0.00
LK/LC without interactions	0.58 ± 0.15	0.75 ± 0.09	0.65 ± 0.06	0.75 ± 0.09	0.76 ± 0.18	0.75 ± 0.06
LK/LC	0.62 ± 0.06	0.77 ± 0.06	0.68 ± 0.03	0.77 ± 0.06	0.79 ± 0.09	0.78 ± 0.03
LK/LC by bagging	0.63 ± 0.03	0.76 ± 0.03	0.68 ± 0.00	0.76 ± 0.03	0.81 ± 0.03	0.78 ± 0.00
LK/LC by C-S with bagging	0.64 ± 0.03	0.74 ± 0.00	0.62 ± 0.12	0.74 ± 0.00	0.70 ± 0.09	0.70 ± 0.09
LK/LC by S with bagging	0.62 ± 0.03	0.75 ± 0.00	0.68 ± 0.00	0.75 ± 0.00	0.80 ± 0.03	0.78 ± 0.00
LK/LC by RUS with bagging	0.58 ± 0.06	0.78 ± 0.03	0.67 ± 0.03	0.78 ± 0.03	0.76 ± 0.03	0.77 ± 0.03
LK/LC by G-S with bagging	**0.60 ± 0.09**	**0.77 ± 0.06**	**0.68 ± 0.03**	**0.77 ± 0.06**	**0.78 ± 0.00**	**0.78 ± 0.03**
LLC/RLC by NS	1.00 ± 0.00	0.06 ± 0.00	0.12 ± 0.06	0.06 ± 0.00	1.00 ± 0.00	0.25 ± 0.06
LLC/RLC without interactions	0.62 ± 0.62	0.23 ± 0.23	0.31 ± 0.18	0.23 ± 0.23	0.95 ± 0.12	0.46 ± 0.24
LLC/RLC	0.87 ± 0.36	0.41 ± 0.41	0.51 ± 0.51	0.41 ± 0.41	0.98 ± 0.06	0.61 ± 0.61
LLC/RLC by bagging	0.84 ± 0.12	0.43 ± 0.21	0.56 ± 0.21	0.43 ± 0.21	0.98 ± 0.02	0.65 ± 0.15
LLC/RLC by C-S with bagging	0.80 ± 0.06	0.79 ± 0.12	0.80 ± 0.12	0.79 ± 0.12	0.96 ± 0.03	0.87 ± 0.12
LLC/RLC by S with bagging	0.85 ± 0.09	0.94 ± 0.06	0.89 ± 0.09	0.94 ± 0.06	0.94 ± 0.06	0.94 ± 0.06
LLC/RLC by RUS with bagging	0.55 ± 0.15	0.92 ± 0.12	0.69 ± 0.12	0.92 ± 0.12	0.84 ± 0.09	0.87 ± 0.06
LLC/RLC by G-S with bagging	**0.86 ± 0.21**	**0.94 ± 0.03**	**0.93 ± 0.01**	**0.93 ± 0.03**	**0.96 ± 0.00**	**0.96 ± 0.00**

method, the hierarchical LSTM method (Graves, 2012), and the hierarchical SMOTE with bagging LSTM method (Wang and Yao, 2009) are made to show the proposed method's effectiveness. This is realized by the computations in Algorithms 1, 2, and 3. We design the algorithm considering vehicle interactions, temporal information, and data imbalance. A better performance is expected for the model calculated from the proposed algorithm compared with the methods without consideration of traffic interactions, temporal information, and/or data imbalance. To verify the presumption, we test the model on the test data from Dataset 1 and Dataset 2. The statistic results for Dataset 1 and Dataset 2 are given in Tables 7 and 8, respectively. In Tables 7 and 8, 'S' represents the SMOTE method, 'C-S' denotes conventional SMOTE method without considering the sequence structure, 'RUS' represents random under-sampling approach, and 'G-S' stands for the GWO-SMOTE method. In 'C-S' method, the synthetic vectors are firstly generated by SMOTE using vectors from data pool which consists of the vectors collected at different sampling instants. Then the sequence is formed by contabulating the vectors regardless of the sampling instants. The time step of synthetic data point for C-S approach follows the original time order of the real sequence data that generates the synthetic data. The rows 'LK/LC' and 'LLC/RLC' represent the LSTM model trained by original imbalanced dataset without any imbalance-handling technique. In Table 8, 'NaN' means not a number. It should be noted that all of the methods in Tables 7 and 8 adopt the LSTM classifier. All of the methods in Tables 7 and 8 adopt the interaction-aware technique unless stated otherwise.

In Tables 7 and 8, the results with the best performance are bold. The proposed method shows highest values of F1 and G-mean for both datasets. For dataset with high imbalance ratio (taking the LLC/RLC in Dataset 1 as an example), the improvement of the F1 and G-mean values is obvious comparing the results of imbalance-handling method and the LLC/RLC method. This proves the necessity of imbalance-handling technique. Additionally, the 'G-S' method performs better than 'S' method, which shows that the proposed GWO modified SMOTE in contribution (1) is effective. The classification performance of the 'C-S with bagging' method is poor. Therefore, it is necessary to keep the temporal information in the sequence when over-sampling the minority data. Contribution (2) is one useful approach to keep the temporal information. Moreover, it can be seen from Tables 7 and 8 that the methods with the consideration of surrounding vehicle interactions produce a higher value for both F1 and G-mean compared with that without considering the interactions, which indicates the influence of surrounding vehicles should be considered in the lane change intention prediction. The F1 and G-mean values of the proposed method are compared with the normal structure (NS). NS, different from the hierarchical structure, directly classifies the sequence into the intentions of LK, LLC, and RLC. From the comparison results between NS with its related hierarchical structure, it can be seen that for both Dataset 1 and Dataset 2 the improvement for LLC/RLC is significant.

Table 8
Testing-data result comparisons for Dataset 2.

Method	P	R	F1	Sensitivity	Specificity	G-mean
LK/LC by NS	0.77 ± 0.06	0.81 ± 0.03	0.79 ± 0.03	0.81 ± 0.03	0.94 ± 0.03	0.87 ± 0.00
LK/LC without interactions	0.48 ± 0.09	0.80 ± 0.12	0.60 ± 0.06	0.80 ± 0.12	0.77 ± 0.09	0.79 ± 0.06
LK/LC	0.74 ± 0.06	0.84 ± 0.03	0.78 ± 0.03	0.84 ± 0.03	0.92 ± 0.03	0.88 ± 0.00
LK/LC by bagging	0.75 ± 0.03	0.84 ± 0.03	0.79 ± 0.03	0.84 ± 0.03	0.93 ± 0.00	0.88 ± 0.00
LK/LC by C-S with bagging	0.91 ± 0.06	0.38 ± 0.00	0.54 ± 0.03	0.38 ± 0.00	0.99 ± 0.00	0.61 ± 0.00
LK/LC by S with bagging	0.73 ± 0.03	0.82 ± 0.03	0.77 ± 0.00	0.82 ± 0.03	0.92 ± 0.00	0.87 ± 0.00
LK/LC by RUS with bagging	0.70 ± 0.03	0.89 ± 0.00	0.78 ± 0.04	0.89 ± 0.00	0.90 ± 0.00	0.89 ± 0.00
LK/LC by G-S with bagging	**0.79 ± 0.03**	**0.88 ± 0.00**	**0.79 ± 0.03**	**0.88 ± 0.00**	**0.88 ± 0.00**	**0.88 ± 0.00**
LLC/RLC by NS	0.56 ± 0.12	0.44 ± 0.30	0.49 ± 0.15	0.44 ± 0.30	0.82 ± 0.21	0.60 ± 0.12
LLC/RLC without interactions	0.60 ± 0.42	0.59 ± 0.39	0.59 ± 0.33	0.59 ± 0.39	0.78 ± 0.42	0.67 ± 0.30
LLC/RLC	0.55 ± 0.18	0.62 ± 0.38	0.58 ± 0.27	0.62 ± 0.38	0.74 ± 0.21	0.67 ± 0.24
LLC/RLC by bagging	0.63 ± 0.04	0.64 ± 0.33	0.63 ± 0.15	0.64 ± 0.33	0.80 ± 0.18	0.71 ± 0.12
LLC/RLC by C-S with bagging	NaN ± NaN	0.00 ± 0.03	NaN ± NaN	0.00 ± 0.03	1.00 ± 0.00	0.03 ± 0.03
LLC/RLC by S with bagging	0.55 ± 0.12	0.80 ± 0.12	0.65 ± 0.12	0.80 ± 0.12	0.66 ± 0.12	0.72 ± 0.12
LLC/RLC by RUS with bagging	0.56 ± 0.06	0.75 ± 0.21	0.64 ± 0.12	0.75 ± 0.21	0.70 ± 0.06	0.72 ± 0.09
LLC/RLC by G-S with bagging	**0.56 ± 0.02**	**0.96 ± 0.04**	**0.70 ± 0.03**	**0.96 ± 0.04**	**0.59 ± 0.06**	**0.75 ± 0.06**

Table 9
Testing-data result comparisons for high-imbalance-ratio Dataset 1.

Method	P	R	F1	Sensitivity	Specificity	G-mean
LK/LC by bagging	0.7165 ± 0.1143	0.5550 ± 0.0807	0.6243 ± 0.0144	0.5550 ± 0.0807	0.9041 ± 0.0708	0.7080 ± 0.0246
LK/LC by S with bagging	0.5724 ± 0.0549	0.8000 ± 0.0876	0.6667 ± 0.0201	0.8000 ± 0.0876	0.7419 ± 0.0846	0.7700 ± 0.0171
LK/LC by RUS with bagging	0.4104 ± 0.0525	0.8040 ± 0.0372	0.5432 ± 0.0477	0.8040 ± 0.0372	0.5019 ± 0.1116	0.6349 ± 0.0711
LK/LC by G-S with bagging	**0.5813 ± 0.0342**	**0.7990 ± 0.0195**	**0.6729 ± 0.0204**	**0.7990 ± 0.0195**	**0.7523 ± 0.0384**	**0.7752 ± 0.0156**

Table 10
Testing-data result comparisons for high-imbalance-ratio dataset 2.

Method	P	R	F1	Sensitivity	Specificity	G-mean
LK/LC by bagging	0.9166 ± 0.0770	0.3889 ± 0.0472	0.5458 ± 0.0427	0.3889 ± 0.0472	0.9908 ± 0.0092	0.6206 ± 0.0360
LK/LC by S with bagging	0.9292 ± 0.0422	0.6481 ± 0.0346	0.7636 ± 0.0377	0.6481 ± 0.0346	0.9873 ± 0.0076	0.7999 ± 0.0242
LK/LC by RUS with bagging	0.5212 ± 0.4342	0.9037 ± 0.1246	0.6495 ± 0.3139	0.9037 ± 0.1246	0.7561 ± 0.3826	0.8231 ± 0.1659
LK/LC by G-S with bagging	**0.7548 ± 0.0208**	**0.7938 ± 0.0941**	**0.7735 ± 0.0443**	**0.7938 ± 0.0941**	**0.9334 ± 0.0114**	**0.8607 ± 0.0469**

Table 11
Prediction time comparisons between proposed method and benchmark methods.

Method	Prediction time (s)
LK/LC by bagging	1.8989 ± 0.0317
LK/LC by C-S with bagging	1.8998 ± 0.0131
LK/LC by S with bagging	1.8962 ± 0.2242
LK/LC by RUS with bagging	1.8966 ± 0.1856
LK/LC by G-S with bagging	1.8947 ± 0.2522
LLC/RLC by bagging	1.8840 ± 0.3197
LLC/RLC by C-S with bagging	1.8578 ± 0.6583
LLC/RLC by S with bagging	1.8714 ± 0.4342
LLC/RLC by RUS with bagging	1.8744 ± 0.3859
LLC/RLC by G-S with bagging	1.8489 ± 0.6953

Moreover, interesting results can be seen from Tables 7 and 8. Comparing the LLC/RLC results by the 'G-S with bagging' method for Dataset 1 and Dataset 2, Dataset 1 shows a higher F1 and G-mean value, this may be related with the different imbalance ratios for the two datasets. The performance of method without consideration of interactions for Dataset 2 is not as good as Dataset 1. This may be explained by the reason that the interaction information is more important for lane change intention prediction of Dataset 2 which was collected in a more crowded traffic (Talebpour et al., 2018). What is more, the improvements for LK/LC results by the proposed method compared with the method without imbalance handling technique are not obvious. This is because the low imbalance ratio between LK and LC data, which is around 2 in Tables 2 and 3. We apply the proposed method and other imbalance-handling approaches to high-imbalance-ratio LK/LC data in Dataset 1 and Dataset 2. The imbalance ratio for data in Table 9 is 23.80 and in Table 10 is 28.43. The results are given in Tables 9 and 10. The results indicate that the proposed method performs better compared with other methods for high-imbalance-ratio dataset.

In Table 7, the F1 score and G-mean value for the row 'LLC/RLC' by NS are extremely low. The reason for this very low F1 score is that R value is too small compared with P value (which equals 1). Since the imbalance ratio between the dataset for LLC/LK by NS is too high (which is 11.67 as shown in Table 2), the predictor could only predict a small number (not zero) of positive samples (LLC

data) correctly. Sensitivity value is the same as R value and a smaller value of sensitivity causes a lower G-mean value. In Table 8, F1 score is not a number for the row 'LLC/RLC by C-S with bagging'. The main reason is that the predictor wrongly classifies all the test samples as negative such that the values for TP and FP are zeros. According to Eq. (15), we can only get a 'NaN'.

If the lane change intention is detected earlier, accident rates will be reduced. We use the same prediction point and prediction time definition as Kumar et al. (2013). Prediction point represents the time instant when the lane change is predicted. Prediction time stands for the time difference between the ground truth and the corresponding prediction point. The prediction time is calculated for the correctly-predicted sequence and the comparison results are given in Table 11. The comparisons are carried out with the methods without sequence-handling or imbalance-handling technique. We can see from Table 11 that, though the prediction time of proposed method does not improve too much, it still has some advantages.

5. Conclusions

To deal with the poor classification performance in LSTM network lane change intention prediction for the imbalanced data, this research proposed a GWO-SMOTE method and combined it with the bagging strategy. The proposed over-sampling technique keeps the temporal information which captures the patterns of lane change in the original minority class. What is more, we take the traffic interactions into consideration when forming the training and test data. To validate the method's classification performance, the proposed method was compared with the method without consideration of traffic interactions and data imbalance. The proposed method performs best with highest F1 and G-mean values among the benchmark methods for US 101 and I-80 data from NGSIM dataset. These results indicate that the data imbalance, data temporal information, and traffic interactions should be considered in the lane change prediction method. The proposed method predicts the lane change intention with a prediction time of around 1.8 s which is comparable to benchmark methods. A larger value of prediction time will provide the AVs more time to react and take actions. For the proposed method, one possible approach to increase the prediction time is to add more information that would influence the lane change decision to the input of the predictor. This research is a basis for motion planning of AVs. Future work will be focused on the application to traffic data from other areas and field test of the proposed method on AVs and vehicle platoons Ju et al. (2021), Luo et al. (2021), Wang et al. (2020). In addition, the proposed method will be applied to other practical problems subject to imbalance dataset.

CRediT authorship contribution statement

Qian Shi: Conceptualization, Methodology, Writing. **Hui Zhang:** Supervision, Reviewing, Editing.

References

Bakhit, P.R., Osman, O.A., Ishak, S., 2017. Detecting imminent lane change maneuvers in connected vehicle environments. Transp. Res. Rec. J. Transp. Res. Board (2645), 168–175.

Benterki, A., Boukhnifer, M., Judalet, V., Maaoui, C., 2020. Artificial intelligence for vehicle behavior anticipation: Hybrid approach based on maneuver classification and trajectory prediction. IEEE Access 8, 56992–57002.

Bhahramani, Z., 2001. An introduction to hidden Markov models and Bayesian networks. Int. J. Pattern Recognit. Artif. Intell. 15 (1), 9–42.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: Synthetic minority over-sampling technique. J. Artificial Intelligence Res. 16 (1), 321–357.

Cover, T.M., Hart, P.E., 1953. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory 13 (1), 21–27.

Cremean, L.B., Foote, T.B., Gillula, J.H., Hines, G.H., Kogan, D., Kriechbaum, K.L., Lamb, J.C., Leibs, J., Lindzey, L., Rasmussen, C.E., 2006. Alice: An information-rich autonomous vehicle for high-speed desert navigation. J. Field Robotics 23 (9), 777–810.

De Morais, R.F.A.B., Vasconcelos, G.C., 2019. Boosting the performance of over-sampling algorithms through under-sampling the minority class. Neurocomputing 343, 3–18.

Ding, W., Chen, J., Shen, S., 2019. Predicting vehicle behaviors over an extended horizon using behavior interaction network. In: 2019 International Conference on Robotics and Automation, ICRA. pp. 8634–8640.

Elreedy, D., Atiya, A.F., 2019. A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance. Inform. Sci. 505, 32–64.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., Hrovat, D., 2007. Predictive active steering control for autonomous vehicle systems. IEEE Trans. Control Syst. Technol. 15 (3), 566–580.

Fernández-Llorca, D., Biparva, M., Izquierdo-Gonzalo, R., Tsotsos, J.K., 2020. Two-stream networks for lane-change prediction of surrounding vehicles. In: 2020 International Conference on Intelligent Transportation Systems, ITSC. pp. 1–6.

Freund, Y., Schapire, R.E., 1995. A decision-thoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55 (1), 119–139.

Graves, A., 2012. Long Short-Term Memory. Springer Berlin Heidelberg.

Haibo, H., Edwardo, A.G., 2009. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21 (9), 1263–1284.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Hou, Y., Edara, P., Sun, C., 2014. Modeling mandatory lane changing using Bayes classifier and decision trees. IEEE Trans. Intell. Transp. Syst. 15 (2), 647–655.

Hou, H., Jin, L., Niu, Q., Sun, Y., Lu, M., 2011. Driver intention recognition method using continuous hidden Markov model. Int. J. Comput. Intell. Syst. 386–393.

Ju, Z., Zhang, H., Tan, Y., 2021. Distributed stochastic model predictive control for heterogeneous vehicle platoons subject to modeling uncertainties. IEEE Intell. Transp. Syst. Mag. http://dx.doi.org/10.1109/MITS.2021.3084964.

Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic optimization. In: 2015 International Conference on Learning Representations, ICLR. pp. 1–15.

Kumar, P., Perrollaz, M., Lefèvre, S., Laugier, C., 2013. Learning-based approach for online lane change intention prediction. In: Intelligent Vehicles Symposium. pp. 797–802.

Lee, D., Kwon, Y.P., McMains, S., Hedrick, J.K., 2017. Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC. In: 2017 International Conference on Intelligent Transportation Systems, ITSC. pp. 1–6.

Lefevre, S., Carvalho, A., Borrelli, F., 2016. A learning-based framework for velocity control in autonomous driving. IEEE Trans. Autom. Sci. Eng. 13 (1), 32–42.

Liu, Y., Yao, X., 1999. Ensemble learning via negative correlation. Neural Netw. 12 (10), 1399–1404.
Luo, Q., Nguyen, A., Fleming, J., Zhang, H., 2021. Unknown input observer based approach for distributed tube-based model predictive control of heterogeneous vehicle platoons. IEEE Trans. Veh. Technol. 70 (4), 2930–2944.
Mahajan, V., Katrakazas, C., Antoniou, C., 2020. Prediction of lane-changing maneuvers with automatic labeling and deep learning. Transp. Res. Rec. J. Transp. Res. Board 2674 (7), 336–347.
Mandalia, H.M., Salvucci, M.D.D., 2005. Using Support Vector Machines for Lane-change detection. Proc. Hum. Factors Ergon. Soc. Annu. Meet. 49 (22), 1965–1969.
Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey wolf optimizer. Adv. Eng. Softw. 69, 46–61.
Morris, B., Doshi, A., Trivedi, M., 2011. Lane change intent prediction for driver assistance: On-road design and evaluation. In: 2011 IEEE Intelligent Vehicles Symposium, IV. pp. 895–901.
Ojala, R., Vepsäläinen, J., Hanhirova, J., Hirvisalo, V., Tammi, K., 2020. Novel convolutional neural network-based roadside unit for accurate pedestrian localisation. IEEE Trans. Intell. Transp. Syst. 21 (9), 3756–3765.
Peng, J., Guo, Y., Fu, R., Yuan, W., Wang, C., 2015. Multi-parameter prediction of drivers' lane-changing behaviour with neural network model. Applied Ergon. 50, 207–217.
Pentland, A., Liu, A., 1999. Modeling and prediction of human behavior. Neural Comput. 11, 229–242.
Petmezas, G., Haris, K., Stefanopoulos, L., Kilintzis, V., Maglaveras, N., 2021. Automated atrial fibrillation detection using a hybrid CNN-LSTM network on imbalanced ECG datasets. Biomed. Signal Process. Control 63, 1–9.
Shi, Q., Zhang, H., 2020. Fault diagnosis of an autonomous vehicle with an improved SVM algorithm subject to unbalanced datasets. IEEE Trans. Ind. Electron. 68 (7), 6248–6256.
Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y., 2007. Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit. 40 (12), 3358–3378.
Talebpour, A., Mahmassani, H.S., Mete, F., Hamdar, S.H., 2018. Near-crash identification in a connected vehicle environment. Transp. Res. Rec. J. Transp. Res. Board (2424), 20–28.
Tang, Y., Zhang, Y.Q., Chawla, N.V., Krasser, S., 2009. SVMs modeling for highly imbalanced classification. IEEE Trans. Syst. Man Cybern. B 39 (1), 281–288.
Turk, M.A., Morgenthaler, D.G., Gremban, K.D., Marra, M., 1988. VITS-a vision system for autonomous land vehicle navigation. IEEE Trans. Pattern Anal. Mach. Intell. 10 (3), 342–361.
Wang, H., Huang, Y., Khajepour, A., Cao, D., Lv, C., 2020. Ethical decision-making platform in autonomous vehicles with lexicographic optimization based model predictive controller. IEEE Trans. Veh. Technol. 69 (8), 8164–8175.
Wang, B., Pineau, J., 2016. Online bagging and boosting for imbalanced data streams. IEEE Trans. Knowl. Data Eng. 28 (12), 3353–3366.
Wang, S., Yao, X., 2009. Diversity analysis on imbalanced data sets by using ensemble models. In: IEEE Symposium on Computational Intelligence & Data Mining. pp. 1–8.
Wozniak, M., Krawczyk, B., Schaefer, G., 2013. Cost-sensitive decision tree ensembles for effective imbalanced classification. Appl. Soft Comput. 14 (1), 554–562.
Wu, Z., Guo, Y., Lin, W., Yu, S., Ji, Y., 2018. A weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems. Sensors 18 (4), 1096–1110.
Xing, Y., Lv, C., Cao, D., 2020. An ensemble deep learning approach for driver lane change intention inference. Transp. Res. C 115, 1–19.
Yang, K., Yu, Z., Wen, X., Cao, W., Chen, C.L.P., Wong, H.S., You, J., 2020. Hybrid classifier ensemble for imbalanced data. IEEE Trans. Neural Netw. Learn. Syst. 31 (4), 1387–1400.
Yu, L., Wang, S., Lai, K.K., 2008. Credit risk assessment with a multistage neural network ensemble learning approach. Expert Syst. Appl. 34 (2), 1434–1444.
Zheng, Y., Hansen, J.H.L., 2017. Lane-change detection from steering signal using spectral segmentation and learning-based classification. IEEE Trans. Intell. Veh. 2 (1), 14–24.
Zyner, A., Worrall, S., Ward, J., Nebot, E., 2017. Long short term memory for driver intent prediction. In: 2017 IEEE Intelligent Vehicles Symposium, IV. pp. 1484–1489.