

Received February 7, 2019, accepted March 3, 2019, date of publication March 6, 2019, date of current version April 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2903319

Traffic Accident's Severity Prediction: A Deep-Learning Approach-Based CNN Network

MING ZHENG¹, TONG LI², RUI ZHU^{2,3}, JING CHEN^{1,4}, ZIFEI MA³,
MINGJING TANG³, ZHONGQIANG CUI³, AND ZHAN WANG³

¹School of Information Science and Engineering, Yunnan University, Kunming 650091, China

²Key Laboratory in Software Engineering of Yunnan Province, Kunming 650091, China

³School of Software, Yunnan University, Kunming 650091, China

⁴Yunnan Science Research Institute of Communication and transportation, Kunming 650091, China

Corresponding authors: Rui Zhu (rzhu@ynu.edu.cn) and Tong Li (tli@ynu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61662085, in part by the Academic Newcomer Award for Doctoral Candidates in Yunnan Province, China, in part by the Project of the Yunnan Provincial Department of Education Science Research Fund under Grant 2017ZZX227, in part by the Key Science Research Project of Yunnan Education under Grant 2015Z018, in part by the Yunnan University Data-Driven Software Engineering Provincial Science and Technology Innovation Team Project under Grant 2017HC012, in part by the Yunnan University's Research Innovation Fund for Graduate Students under Grant YDY17018, and in part by the Ali Living Water Plan.

ABSTRACT In traffic accident, an accurate and timely severity prediction method is necessary for the successful deployment of an intelligent transportation system to provide corresponding levels of medical aid and transportation in a timely manner. The existing traffic accident's severity prediction methods mainly use shallow severity prediction models and statistical models. To promote the prediction accuracy, a novel traffic accident's severity prediction-convolutional neural network (TASP-CNN) model for traffic accident's severity prediction is proposed that considers combination relationships among traffic accident's features. Based on the weights of traffic accident's features, the feature matrix to gray image (FM2GI) algorithm is proposed to convert a single feature relationship of traffic accident's data into gray images containing combination relationships in parallel as the input variables for the model. Moreover, experiments demonstrated that the proposed model for traffic accident's severity prediction has a better performance.

INDEX TERMS Traffic accident, severity prediction, convolutional neural network, gray image.

I. INTRODUCTION

Every year the lives of more than 1.25 million people are cut short as a result of road traffic accident. Between 20 and 50 million more people suffer non-fatal injuries, with many incurring a disability as a result of their injury. Road traffic injuries cause considerable economic losses to individuals, their families, and nations as a whole. These losses arise from the cost of treatment, as well as lost productivity for those killed or disabled by their injuries and for family members who need to take time off work or school to care for the injured. Road traffic accident cost most countries 3% of their gross domestic product [1]. Traffic accidents severity prediction is an important step in accident management. It provides important information for emergency personnel to

assess the severity of accidents, assess the potential impact of the accident, and implement effective accident management procedures [2].

Nevertheless, for the analysis of the severity of traffic accident, there still exist two problems. The first one is unable to confirm the severity of traffic accident by traditional methods in the field of transportation, the second one is the performance of current traffic accident severity prediction methods is generally not high, and these two problems will lead to untimely rescue or even inadequate rescue facilities, resulting in casualties. Therefore, a correct prediction of traffic accident's severity provides extremely important help to save the lives of those in accidents, the problem of predicting the severity of traffic accident is a major challenge in the field of traffic safety.

At present, deep learning has attracted great attention from researchers. Among them, deep learning theory explains text,

The associate editor coordinating the review of this manuscript and approving it for publication was Victor Sanchez.

image, and voice and has been widely used in the fields of text, image, and voice recognition [3]–[9]. The deep learning method, convolutional neural network (CNN) has become a heated research topic in many scientific fields. CNN is a fast and effective feedforward neural network that is widely used in the computer vision, image recognition, and speech recognition fields and has achieved remarkable results [10]–[15]. CNN has the following feature extraction characteristics: 1) the convolution layer in CNN is locally connected but not fully connected, which means that the output neurons are only connected to locally adjacent input neurons. 2) Another layer structure in CNN, the pool layer, only selectively selects significant features from the receiving area, which greatly reduces the parameter scale of the model. 3) The full connection layer is only used in the final stage of CNN [16]–[18].

This paper aims to solve the above two problems using proposed TASP-CNN model based on CNN which can consider and excavate in detail the combination relationships among the traffic accident's features that affect the severity of traffic accident. In order to extract the combination relationships between traffic accident's severity features and since the model TASP-CNN's input can only be an image, the method for measure the weight of traffic accident's features and an algorithm, FM2GI, to convert a single feature relationship of traffic accident's data into a gray image containing combination relationships in parallel based the weights of traffic accident's features have been proposed at the beginning of this paper, then, a TASP-CNN model is constructed to predict traffic accident's severity.

The main contributions of this paper are summarized as follows:

- 1) An algorithm, FM2GI, is proposed to convert a single feature relationship of traffic accident's data into a gray image containing combination relationships based on the weights of traffic accident's features in parallel.
- 2) The combination relationships among traffic accident's features are considered and applied simultaneously in traffic accident severity prediction using the proposed algorithm, FM2GI, and the TASP-CNN model.
- 3) The performance of the proposed TASP-CNN model was compared with 9 competitive models and the results show that the proposed TASP-CNN model was better to those models.

The remainder of this article is organized as follows. Section 2 discusses related work. Section 3 presents the method for measure the weights of traffic accident's features, the methodology of converting a single feature relationship of traffic accident's data into a gray image containing combination relationships in parallel based on the weights of traffic accident's features and the TASP-CNN model to predict traffic accident's severity. Section 4 gives the experimental results of the TASP-CNN model and analysis of these results. Finally, the conclusion and future work are described in Section 5.

II. RELATED WORK

According to the primary literature on traffic accident's severity prediction in recent years, prediction methods for traffic accident's severity can be divided into two categories: (1) the statistical-learning methods and (2) the machine-learning methods [19].

A. THE STATISTICAL METHODS

The statistical methods have been widely used in traffic accident's severity prediction. For example, various researchers have proposed the logistic regression model, the ordered probit model, and the mixed logit model, which have been used to analyze relevant data from traffic accident and to analyze the influence of multiple variables on the severity of traffic accident [2], [20]–[26], the purpose of these prediction models is to predict the severity of traffic accident. References [2], [27], and [28] have shown that most regression models have assumptions and predefined basic relationships between independent and dependent variables (i.e., a linear relationship between variables). When these assumptions are violated, the severity of traffic accident will be inaccurately predicted by a large probability. Therefore, De Oña *et al.* [28] presented an analysis of 1,536 accidents on rural highways in Spain, where 18 variables representing the aforementioned contributing factors were used to build three different BNs (Bayesian networks) that classified the severity of accidents into slightly injured, killed, or severely injured. The variables that most accurately identified the factors that are associated with accidents in which someone was a killed seriously injured (accident type, driver age, lighting, and number of injuries) were identified by inference. In particular, Zong *et al.* [2] presented a comparison between two modeling techniques, Bayesian networks and regression models, by applying them in accident severity analysis. Three severity indicators, including the number of fatalities, the number of injuries, and property damage, are investigated using the two methods, and the major contributing factors and their effects are identified. The results indicated that the goodness-of-fit of the Bayesian network was higher than that of the regression models in accident severity modeling. Abellán *et al.* [29] demonstrated that rule extraction is unlimited to the structure of the decision tree (DT), and some important relationships between variables cannot be extracted when only one DT is used. Therefore, a more effective method to extract rules from DTs is proposed, but this method can only be applied to a specific traffic accident's data set. Chang and Chien [30] collected the 2005–2006 truck-involved accident data from national freeways in Taiwan and developed a non-parametric classification and regression tree (CART) model to establish the empirical relationship between injury severity outcomes and driver/vehicle characteristics, highway geometric variables, environmental characteristics, and accident variables. The results showed that drinking and driving, seat-belt use, vehicle type, collision type, contributing circumstances leading to driver/vehicle action, number of vehicles involved in the accident, and accident location were the key

determinants of injury severity outcomes in truck accidents. Li *et al.* [31] compared the performance of the support vector machine (SVM) model and the ordered probit (OP) model. It was found that the SVM model produced better prediction performance outcomes for crash injury severity than the OP model did. The percentage of correct predictions from the SVM model was found to be 48.8%, which was higher than that produced by the OP model (44.0%). Even though the SVM model may suffer from a multi-class classification problem, it still provided better prediction results for small proportion injury severities than the OP model did. Chen *et al.* [32] utilized a classification and regression tree (CART) model to identify significant variables, and then SVM models with polynomial and Gaussian radius basis function (RBF) kernels were used for model performance evaluation. It has been shown that SVM models have a reasonable prediction performance, and the polynomial kernel has outperformed the Gaussian RBF kernel. Hashmienejad [33] proposed a novel rule-based method to predict traffic accident's severity according to user's preferences instead of conventional DTs. In their method, they customized a multi-objective genetic algorithm, the non-dominated sorting genetic algorithm (NSGA-II), to optimize and identify rules according to support, confidence, and comprehensibility metrics. The evaluation results revealed that the proposed method outperformed classification methods, such as ANN, SVM, and conventional DTs, according to classification metrics, such as accuracy (88.2%) and performance metrics of rules, such as support and confidence (0.79 and 0.74, respectively).

B. MACHINE LEARNING METHODS

In recent years, machine learning methods are efficient technologies that have been widely used in traffic prediction problems because of their ability to process multi-dimensional data, flexibility in implementation, versatility, and strong predictive capabilities [34]–[37]. In terms of traffic accident's severity prediction, Kunt *et al.* [38] used twelve accident-related parameters in a genetic algorithm (GA) pattern search and multi-layer perceptron (MLP) structural modeling methods in an artificial neural network (ANN) to predict the severity of freeway traffic accident. The models were constructed based on 1,000 crashes that occurred during 2007 on the Tehran–Ghom Freeway. The best-fit model was selected according to the R-value, root mean square errors (RMSE), mean absolute errors (MAE), and the sum of square error (SSE). The highest R-value was obtained for the ANN, approximately 0.87, demonstrating that ANN provided the best prediction. Zeng and Huang [39] proposed a convex combination (CC) algorithm to rapidly and stably train a neural network (NN) model for crash injury severity prediction, and a modified NN pruning function approximation (N2PFA) algorithm to optimize the network structure. According to the results, the CC algorithm outperformed the BP algorithm both in convergence ability and training speed. Compared with a fully connected NN, the optimized NN contained much

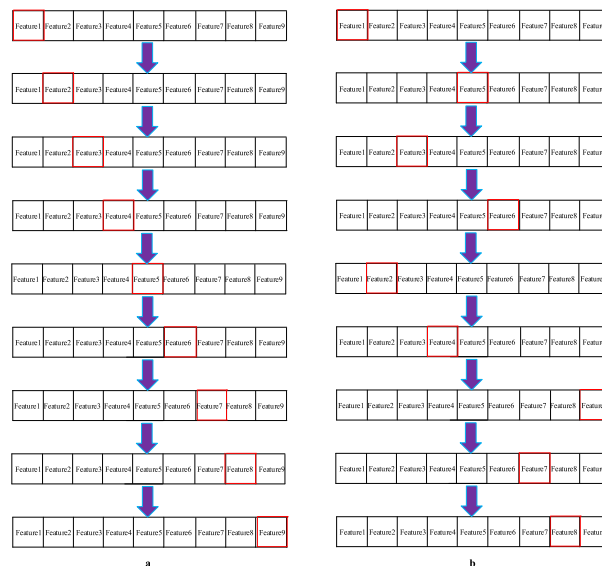


FIGURE 1. Considering the data's features of traditional models (a) ordered, (b) disordered.

fewer network nodes and achieved comparable classification accuracy. Both of them had better fitting and predictive performance than the OL model, which again demonstrates the NN's superiority over statistical models for predicting crash injury severity. Sameen and Pradhan [40] developed a machine-learning model using a recurrent neural network (RNN) and employed to predict the injury severity of traffic accident based on 1,130 accident records that have occurred on the North–South Expressway (NSE), Malaysia over a six-year period from 2009 to 2015. The proposed RNN model was compared with the multilayer perceptron (MLP) and Bayesian logistic regression (BLR) models to understand its advantages and limitations. The results of the comparative analyses showed that the RNN model outperformed the MLP and BLR models. The validation accuracy of the RNN model was 71.77%, whereas the MLP and BLR models achieved 65.48% and 58.30%, respectively.

The majority of traditional statistical and machine learning models consider that data's features are ordered or disordered one by one, as shown in Figure 1.

From the Figure 1, the traditional models only consider the single feature relationship of the data, and don't consider the combination relationships among the features. For this reason, according to the characteristics of CNN, this paper proposes a method to combine the features of data to obtain the combination relationships of data, as shown in Figure 2: the red rectangle is no longer extract a single feature relationship of the data, but a combination relationship among four features. Correspondingly, the size of the red rectangle can be flexibly changed according to different situations to obtain different combination relationships, specifically, it is achieved by changing the kernel size of the convolution operation in CNN. But the key to this method which is also the key to affecting the performance of the model is how to

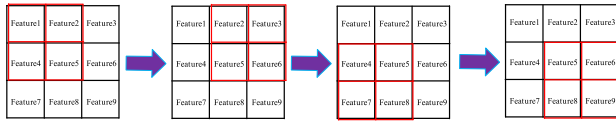


FIGURE 2. Considering the data’s combination relationships in this paper.

arrange the features of the data into the corresponding positions in the matrix, this requires the features’ weights, measuring the weights of features will be explained in the next chapter.

Although feature-to-image conversion practice is not a something new [41], [42], the method we proposed in converting features into images is obviously different from those papers, because our inspiration comes from the inherent attributes and characteristics of CNN, by combining the measuring of feature weights, the features are filled into the all-zero matrix, and finally converted into images as input of CNN, let’s use Figure 2 to illustrate why we did this. As we all know, the core of CNN is convolution operation, so suppose the kernel size is 2 and the stride is 1, we can see that the number of convolutions at the center of the matrix is the largest while the number of convolutions at the edge of the matrix is the smallest when all convolution operations are completed, where feature 5 participated in convolution operation four times, while feature 1, feature 3, feature 7 and feature 9 only participated in convolution operation once, therefore, we may conclude that feature 5 contributes more to extracting feature information from CNN than feature in edge position like feature 1, feature 3, feature 7 and so on, this is why we need to measure the weights of features, generally speaking, the greater the weights of features, the greater the impact on the prediction results, so, we fill the features with larger weights into the center position of the all-zero matrix, accordingly, the features with smaller weights will be filled into the edge position of the all-zero matrix. In this way, we can give full play to the inherent attributes and characteristics of CNN to improve the performance of the model.

III. METHODOLOGY

Traffic accident’s data with feature information must be considered comprehensively to predict the severity of traffic accident. In accordance with Chang and Wang [27] and Kopelias *et al.* [43], the factors affecting the severity of traffic accident mainly include the following five parent features: roadway, accident, vehicle, casualty, and environmental features. However, the most of existing literature did not effectively use and excavate the combination relationships among these features. Based on the above-mentioned five parent features that affect the severity of traffic accident, the method for measure the weights of traffic accident’s features, an algorithm, FM2GI, to convert a single feature relationship of traffic accident’s data into a gray image containing combination relationships in parallel and TASP-CNN architecture are proposed and will be explained in this section.

A. MEASURING THE WEIGHT OF TRAFFIC ACCIDENT’S FEATURES

In order to analyze the combination relationships of traffic accident’s features and its contribution to the traffic accident, it is necessary to measure the weights of the child and parent features of traffic accident. Principle of measuring the weights of traffic accident’s features is based on Gradient Boosting Decision Tree (GBDT) [44], [45]. Let there is a single decision tree T , the weight of each feature X_ℓ can be measured by using the equation (1) [46].

$$\mathfrak{S}_\ell^2(T) = \sum_{t=1}^{J-1} \hat{l}_t^2 I(v(t) = \ell) \tag{1}$$

where the sum is over the $J - 1$ internal nodes of the tree. At each node t , one of the input features $X_{v(t)}$ used to partition the region associated with that node into two sub-regions, within each a separate constant is fit to the severity values. The particular feature selected is the one that gives maximal estimated improvement \hat{l}_t^2 in squared error risk over that for a constant fit over the entire region. The squared weight of feature X_ℓ is the sum of such squared improvements over all internal nodes for which it was selected as the splitting feature.

This weight measure is easily generalized to additive tree expansions and it’s simply averaged over the trees as the equation (2) [45].

$$\mathfrak{S}_\ell^2 = \frac{1}{M} \sum_{m=1}^M \mathfrak{S}_\ell^2(T_m) \tag{2}$$

This measure turns out to be more reliable than equation (1), due to the stabilizing effect of averaging. Please note that equation (1) and (2) refer to squared weight, the actual weights are their respective square roots. For K -class classification, K separate models $f_k(x)$, $k = 1, 2, \dots, K$ are induced, and each consisting of a sum of trees as shown in equation (3) [45].

$$f_k(x) = \sum_{m=1}^M T_{km}(x) \tag{3}$$

In this case, equation (3) will be generalizes to equation (4) [45].

$$\mathfrak{S}_{\ell k}^2 = \frac{1}{M} \sum_{m=1}^M \mathfrak{S}_{\ell}^2(T_{km}) \tag{4}$$

where $\mathfrak{S}_{\ell k}$ is the weight of X_ℓ in separating the class k observations from the other classes. The overall weight of X_ℓ is obtained by averaging over all of the classes by equation (5) [45].

$$\mathfrak{S}_\ell^2 = \frac{1}{K} \sum_{k=1}^K \mathfrak{S}_{\ell k}^2 \tag{5}$$

B. FM2GI: CONVERTING TRAFFIC ACCIDENTS' FEATURE MATRIX TO GRAY IMAGE

According to the feature weight and the characteristics of CNN, this paper proposes an algorithm for converting a single feature relationship of data into a gray image containing combination relationships of data features and will be explained below.

First, the data features and their correlation were formally defined.

Definition 1 (Feature Vector): The feature vector is the expression form of a data's features which is a 3-tuple, $FV = \{FP, FC, \mathbf{wc}\}$, where:

- 1) FP represents all the parent features of a data. $FP = \{FP_1, \dots, FP_m\}$, m represents the number of parent features of a data, in this paper, $m=5$;
- 2) FC represents all the child features of a data. $FC = \{FC_{1,1}, \dots, FC_{m,n}\}$, n represents the number of child features of a data. In this paper, $n = 12$. $\forall FC_{i,j} \in FP_i$, where $i \in [1, m]$, $j \in [1, n]$ represents the j th child feature, and $FC_{i,j}$'s parent feature is FP_i of a data. Satisfy: $\bigcup_{i=1}^m FP_i = FC$, $\forall FP_i \cap FP_j = \emptyset$, $i \neq j$, and the number of child features included in the i th parent feature is recorded as $NP_i = |FP_i|$;
- 3) \mathbf{wc} represents the weights vector of all child features of a data, $\mathbf{wc} = (w_{1,1}, \dots, w_{i,j})$, where $i \in [1, m]$, $j \in [1, n]$, $\forall w_{i,j}$ represents the weight of the j th child feature of a data, and it belongs to the parent feature FP_i .

According to definition 1, the feature vector of a data in this traffic accidents' data sets can be formally described as: $FV = (FP, FC, \mathbf{wc})$, where:

- 1) $FP = \{\text{Accident Features}_1, \text{Roadway Features}_2, \text{Environmental Features}_3, \text{Vehicle Features}_4, \text{Casualty Features}_5\}$;
- 2) $FC = \{\text{Easting}_{1,1}, \text{Northing}_{1,2}, \text{1st Road Class}_{1,3}, \text{Accident Time}_{1,4}, \text{Number of Vehicles}_{1,5}, \text{Road Surface}_{2,6}, \text{Lighting Conditions}_{3,7}, \text{Weather Conditions}_{3,8}, \text{Type of Vehicle}_{4,9}, \text{Casualty Class}_{5,10}, \text{Sex of Casualty}_{5,11}, \text{Age of Casualty}_{5,12}\}$;
- 3) $\mathbf{wc} = (0.165774538_{1,1}, 0.171530785_{1,2}, 0.082228259_{1,3}, 0.047771472_{1,4}, 0.060763375_{1,5}, 0.048847406_{2,6}, 0.041826936_{3,7}, 0.04354843_{3,8}, 0.126314657_{4,9}, 0.067057589_{5,10}, 0.049116389_{5,11}, 0.095220163_{5,12})$

Definition 2 (Feature Matrix): The feature matrix is the expression of all data features in the data sets. The feature matrix is a set of feature vectors whose expression is:

$$FM = \{FV_1, \dots, FV_k\}, \quad \text{and } FM \in R^{k \times n}$$

where k represents the size of data sets, and n represents the number of child features of each data in the data sets. Then, the algorithm 1 of converting a single feature relationship into a gray image containing combination relationships of data features is as follow.

Algorithm 1 classifies the features in data sets, the n child features classified into the corresponding m parent features according to the characteristics of data sets, and each of

Algorithm 1 (FV2GI): Converting the Feature Vector of a Piece of Data Sets Into a Gray Image

Input: Feature vector of a piece of data sets: FV

Output: Gray image form of a piece of data sets: $grayImage$

Steps:

1. Begin
2. for $i = 1 \rightarrow m$
3. $max_dim = \max(NP_i, m)$;
4. end for
5. Initialize an all-zero matrix namely $Mat^{max_dim \times max_dim}$;
6. for $i = 1 \rightarrow m$
7. for $j = 1 \rightarrow n$
8. $wp_i = \sum w_{i,j}$;
9. end for
10. end for
11. Define $\mathbf{wp} = (wp_1, \dots, wp_m)$;
12. $Sorted_FP = \text{sort}(FP, \mathbf{wp}, \text{'Descend'})$;
13. for $i = 1 \rightarrow m$
14. $Mat = \text{fill_row}(Mat, Sorted_FP)$;
15. for $j = 1 \rightarrow NP_i$
16. $Sorted_FC = \text{sort}(FC, \mathbf{wc}, \text{'Descend'})$;
17. $Mat = \text{fill_column}(Mat, Sorted_FC)$;
18. end for
19. end for
20. $grayImage = \text{reshape}(Mat)$;
21. Return $grayImage$;
22. End

child feature belongs to one and only one parent feature. The number of child features is contained in the i th parent feature is expressed as NP_i , and i belong to $[1, m]$. In addition, the weights vector of all child features is initialized and recorded as \mathbf{wc} . Algorithm 1 first searches for the maximum number of child features are contained in the parent feature, this means that it finds the parent feature containing the maximum number of child features and returns the size of parent feature at this time, then it compares the value with m , the maximum value is defined as max_dim , and then initializes an all-zero matrix $Mat^{max_dim \times max_dim}$ as the final storage unit of data sets.

After obtaining the all-zero matrix, the features of the data set filled into the all-zero matrix according to the following steps and rules: first, all the parent features are arranged in descending order, and the descending order standard is based on the weight of each of parent feature (the weight of each parent feature is the sum of the weights of all child features belong to it). All parent features will be filled by the center of the all-zero matrix along the longitudinal axis according to the rule that the maximum weight is in the center, and the upper weight is greater than the lower weight (implementation by function $fill_row$). Second, since the child features are contained in parent feature are currently unordered, after

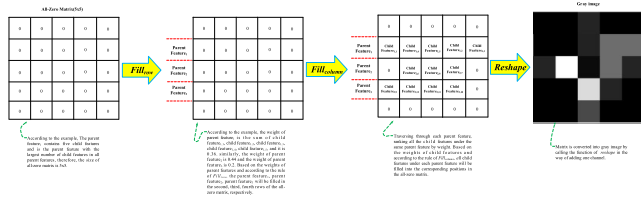


FIGURE 3. An example to illustrate the process of algorithm.

Algorithm 2 (FM2GI): The Feature Matrix of the Data Sets Is Converted Into Gray Images in Parallel

Input: Feature matrix of all data sets: FM
Output: Gray images linked list of all data sets: $grayImageList$
Steps:

1. Begin
2. $k = getNum(FV, FM);$
3. for $i = 1 \rightarrow k$
4. $FV2GI(FM[i]);$
5. Return $grayImage;$
6. end for
7. $grayImageList = grayImageList \cup \{grayImage\}$
8. ;
9. Return $grayImageList;$
10. End

the first step was completed, the child features under each of parent feature need to be arranged in descending order according to the weights of the child features, then all child features under each parent feature are filled by the center of the all-zero matrix along the horizontal axis according to the rule that the maximum weight is in the center and the left weight is greater than the right weight (implementation by function $fill_column$). Third, after the above two-level nesting loop is completed, the final result matrix will be obtained. Finally, the matrix is converted into gray image by calling the function of graphic processing which named $reshape$ in the way of expanding the shape of matrix..

In order to clearly understand the process of algorithm 1, we give an example to illustrate it, as shown in the following Figure 3. Suppose there are three parent features in the dataset: parent feature₁, parent feature₂ and parent feature₃. Among them, parent feature₁ contains five child features: child feature_{1,1}, child feature_{1,2}, child feature_{1,3}, child feature_{1,4}, child feature_{1,5} and their weights are 0.03, 0.06, 0.01, 0.11, 0.15, respectively. Parent feature₂ contains three child features: child feature_{2,6}, child feature_{2,7}, child feature_{2,8} and their weights are 0.1, 0.04, 0.3, respectively. There are four child features in parent feature₃: child feature_{3,9}, child feature_{3,10}, child feature_{3,11}, child feature_{3,12} and their weights are 0.02, 0.03, 0.08, 0.07, respectively.

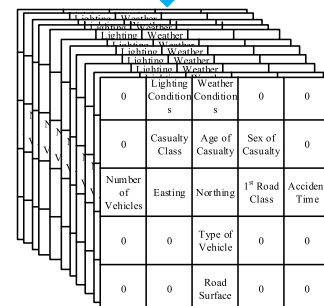
Algorithm 2 converting the feature matrix of the data sets into gray images in parallel as follows:



Matrix Representation

Easting	Northing	1 st Road Class	Road Accident Number of Vehicles	Road Surface	Lighting Conditions	Weather Conditions	Type of Vehicle	Casualty Class	Sex of Casualty	Age of Casualty	
429093	436238	6	2	1	1	3	1	7	3	1	3
434723	435534	6	2	1	1	3	1	7	1	2	2
441173	433047	6	1	1	1	3	1	7	3	2	1
428487	431364	3	1	1	1	3	1	7	3	1	1
425928	435480	6	1	2	1	1	1	7	1	2	3
...
425642	434240	3	2	2	1	3	1	7	1	1	3
429610	439798	6	2	1	1	3	1	7	2	1	1
426535	433642	6	2	2	2	3	1	7	2	2	2
437658	445061	3	1	1	2	2	1	7	2	1	2
431508	435120	6	1	2	1	3	1	7	1	2	1

Matrix Representation of Combination relationships



Gray Images Generation

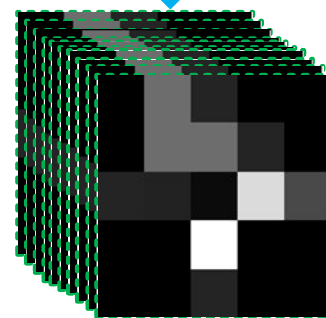


FIGURE 4. A schematic diagram of how to convert feature vector of traffic accidents' data sets into gray images.

Algorithm 2 (FM2GI) uses a parallel method to convert the feature matrix of the data sets into gray images in parallel. The algorithm first obtains the size of data sets of the feature matrix FM , and records it as k , what is, allocates k threads, then traverses the total number of threads, k , and calls the Algorithm 1, FV2GI, for each thread to perform gray image conversion on the corresponding feature vector in the feature matrix. Finally, the result of converting the feature vectors obtained by each thread into gray images is stored in $grayImageList$, which returns the result.

Figure 4 shows how to convert the feature vector of traffic accidents' data sets into gray images.

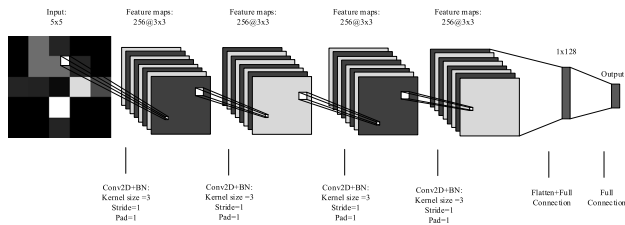


FIGURE 5. The structure of TASP-CNN.

C. TASP-CNN ARCHITECTURE

As shown in Figure 5, the structure of TASP-CNN includes four main parts: model input, convolution layer, full connection layer, and model output layer. Each section will be described in detail below.

First, the input of TASP-CNN is the gray image of the converted traffic accidents' data sets, which includes 5 parent features and 12 child features of the traffic accident. Accordingly, the input mathematical form of the model is expressed as follow:

$$x_i = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1M} \\ P_{21} & P_{22} & \dots & P_{2M} \\ \dots & \dots & \dots & \dots \\ P_{M1} & P_{M2} & \dots & P_{MM} \end{bmatrix}, \quad i \in [1, N],$$

$$M = \max(PC, CC) \quad (6)$$

where i represents the index of the traffic accidents' data sets x ; N represents the size of traffic accidents' data sets; PC is the number of parent features of the traffic accidents' data sets; and CC is the largest number of child features under all parent features.

Then, the core layer of TASP – CNN is a convolution layer, and its purpose is to extract the abstract feature in the traffic accidents' data sets. To clearly describe the calculation process of the convolution layer, each pixel of a traffic accident's image is first numbered, $P_{c,i,j}$, and it represents the pixel elements in row i and column j of the c channel image. Then, each weight of the filter is numbered, and the weight of row m and column n of the channel c filter is represented by $w_{c,m,n}$, finally, the convolution is calculated using the following equation:

$$a_{i,j} = f\left(\sum_{c=1}^C \sum_{m,n=1}^F w_{c,m,n} P_{c,i+m,j+n} + w_b\right) \quad (7)$$

where $a_{i,j}$ represents column j elements in row i of the feature map; C is the number of channels; F is the size of the filter (width or height, both the same); w_b represents the offset term of the filter, f represents the activation function.

The activation function used in this article is the rectified linear unit (ReLU) [47], and the calculation equation is as follow:

$$f(x) = \max(0, x) \quad (8)$$

where x represents the input of neurons. Each convolution layer may have multiple filters. After convolving each filter

with the original traffic accident's image, feature map can be obtained. Therefore, the channel (number) of the convolved feature map is the same as the filter number of the convolution layer.

Next, the following setup of the full connection layer is to convert the final and highest-level features that were extracted and learned by the last convolution layer, flatten them into a one-dimensional vector, and calculate the flatten full connection layer using the following equation:

$$a^{flatten} = \text{flatten}([a_1, a_2, \dots, a_c]), \quad c \in [1, C] \quad (9)$$

Finally, the output of the upper full connection layer is taken as the input of the next full connection layer and finally output to the output layer. The output layer uses the softmax activation function [48] to classify the severity of traffic accident. The output of the model is the corresponding traffic accident's severity level, including slight traffic accident, serious traffic accident, and fatal traffic accident. The calculation equation for the full connection layer is as follow:

$$\hat{y} = w_f a_D^{flatten} + b_f \quad (10)$$

where w_f represents the weight of the full connection layer; and b_f represents the offset term of the full connection layer.

In addition, batch normalization [49] is used between the convolution layers, between the convolution layer and the full connection layer, and between the full connection layer and the full connection layer to train the acceleration model and prevent over-fitting.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The TASP-CNN model proposed in this study was implemented in Python using Google's open source deep-learning framework TensorFlow [50]. TensorFlow was used because it has the advantages of availability, flexibility, and high efficiency. It can easily define and execute various deep-learning networks. It was specifically configured as Intel Xeon E5-2682v4 (Broad Well) processor, 2.5 GHz clock speed, NVIDIA p100 GPU, 12 GiB display memory, GPU server with 9.3 TFLOPS single-precision floating point and 4.7 TFLOPS double-precision floating point computing capability, all the experiments in this paper were completed under this experimental environment, the average result of ten repeated of experiments is the final result of the experiment and the CSP-CNN model was used to train 100 epochs in each experiment.

A. DATA DESCRIPTION

The traffic accidents' data for an 8-year period (2009–2016) from the Leeds City Council, United Kingdom were used in this study. The total number of accident records obtained for this period was 21,436. The severity was categorized into three levels: slight, serious, and fatal. Twelve different child features were collected from each accident record of the traffic accident that included Easting, Northing, 1st road class, Accident time, Number of vehicles, Road surface, Lighting

TABLE 1. Twelve child features and five parent features of the traffic accidents' data sets and their corresponding descriptions.

Parent features	Child features	Feature description
Accident feature	Easting	Location in the Easting of the roadside where the accident occurred
	Northing	Location in the Northing of the roadside where the accident occurred
	1st road class	Motorway=1, A (M)=2, A=3, B=4, C=5, Unclassified=6
	Accident time	Day [6:00,18:00)=1, Night [18:00,6:00)=2
	Number of vehicles	Vehicle involved (positive integer)
Roadway feature	Road surface	Dry=1, Wet/Damp=2, Snow=3, Frost/Ice=4, Flood=5
Environmental feature	Lighting conditions	Daylight: street lights present=1, Darkness: no street lighting=2, Darkness: street lights present and lit=3, Darkness: street lights present but unlit=4
	Weather conditions	Fine without high winds=1, Raining without high winds=2, Snowing without high winds=3, Fine with high winds=4, Raining with high winds=5, Snowing with high winds=6, Fog or mist=7, Other=8
Vehicle feature	Type of vehicle	Pedal cycle=1, M/cycle 50cc and under=2, Motorcycle over 50cc and up to 125cc=3, Motorcycle over 125cc and up to 500cc=4, Motorcycle over 500cc=5, Taxi/Private hire car=6, Car=7, Minibus (8 ~16 passenger seats) =8, Bus or coach (17 or more passenger seats) =9, Ridden horse=10, Agricultural vehicle (includes diggers etc.)=11, Tram / Light rail=12, Goods vehicle 3.5 tonnes mgw and under=13, Goods vehicle over 3.5 tonnes and under 7.5 tonnes mgw=14, Goods vehicle 7.5 tonnes mgw and over=15, Mobility Scooter=16, Other Vehicle =17, Motorcycle-Unknown CC=18
Casualty feature	Casualty class	Driver or rider=1, Vehicle or pillion passenger=2, Pedestrian=3
	Sex of casualty	Male=1, Female=2
	Age of casualty	Age 18 and under=1, Age over 18 and up to 25=2, Age over 25 and up to 65=3, Age over 65=4

conditions, Weather conditions, Type of vehicle, Casualty class, Sex of casualty and Age of casualty, and they are belong to one of the five parent features as shown in Table 1, among them, the child features Easting, Northing, 1st road class, Accident time and Number of vehicles belong to parent feature Accident feature, the child feature Road surface belongs to parent feature Roadway feature, the child features Lighting conditions and Weather conditions are belong to parent feature Environmental feature, the child feature Type of vehicle belongs to parent feature Vehicle feature, the child features Casualty class, Sex of casualty and Age of casualty are belong to parent feature Casualty feature.

Based on the above, through the interface provided by XGboost since it's an improved algorithm based on the

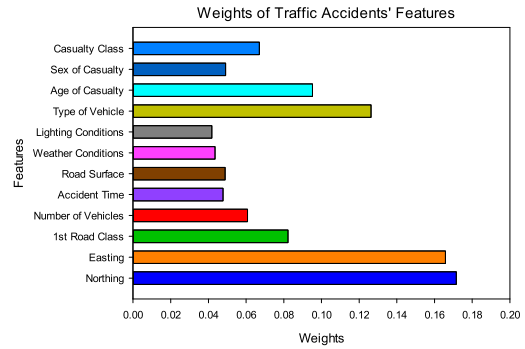


FIGURE 6. Weights distribution of child features of the traffic accident's data sets.

TABLE 2. Weights of the traffic accidents' features.

Parent features	Weights	Child features	Weights
Accident feature	0.528068429	Northing	0.171530785
		Easting	0.165774538
		1st road class	0.082228259
		Number of vehicles	0.060763375
		Accident time	0.047771472
Roadway feature	0.048847406	Road surface	0.048847406
Environmental feature	0.085375366	Weather conditions	0.04354843
		Lighting conditions	0.041826936
Vehicle feature	0.126314657	Type of vehicle	0.126314657
Casualty feature	0.211394141	Age of casualty	0.095220163
		Sex of casualty	0.049116389
		Casualty class	0.067057589

GBDT principle, which is efficient and can be parallelized [51], the weights distribution results obtained by 1,000 iterations of 12 child features of traffic accident are shown in Figure 6 and Table 2 below.

According to the Table 2, we can find that the weight of Accident feature is the highest among the five of parent features from the perspective of parent features, which indicates that the Accident feature has the greatest impact on the traffic accident. From the perspective of child features, from the perspective of child features, the weights of Northing, Easting and Type of vehicle are the highest among the twelve child features, which indicates that there are some traffic accident's black spots with frequent traffic accident and the possibility of traffic accident of different vehicle types are different.

B. DATA PREPROCESSING AND EVALUATION METRICS

Before applying the traffic accidents' data sets as input to TASP-CNN, the data sets needed to be preprocessed. The preliminary processing of the data included deleting incomplete, erroneous, and repeated traffic accidents' data, normalizing traffic accidents' data sets and imbalanced processing of traffic accidents' data. There were 18,727 pieces of data in the entire data sets that were available to be trained after incomplete, erroneous, and repeated data were deleted.

TABLE 3. Distribution of traffic accidents' data sets in ten experiments.

	Training set (before over-sampling)			Training set (after over-sampling)			Testing set		
	Slight	Serious	Fatal	Slight	Serious	Fatal	Slight	Serious	Fatal
Experiment1	13117	1750	114	13117	13117	13117	3302	426	18
Experiment2	13114	1753	114	13114	13114	13114	3305	423	18
Experiment3	13127	1742	112	13127	13127	13127	3292	434	20
Experiment4	13118	1757	106	13118	13118	13118	3301	419	26
Experiment5	13124	1747	110	13124	13124	13124	3295	429	22
Experiment6	13118	1759	104	13118	13118	13118	3301	417	28
Experiment7	13113	1759	109	13113	13113	13113	3306	417	23
Experiment8	13135	1733	113	13135	13135	13135	3284	443	19
Experiment9	13130	1744	107	13130	13130	13130	3289	432	25
Experiment10	13154	1721	106	13154	13154	13154	3265	455	26

Since the dimensions of each of the 12 child features of traffic accident are different, it was necessary to normalize the data under each feature, remove the unit limitation of the data, and convert it into dimensionless pure values so that the features of the different units could be compared. In addition, the normalization of traffic accidents' data sets can also improve the convergence speed and accuracy of the model [49], [52]. By using the standardization method in statistics: z-score normalization, also called zero-mean normalization, to normalize the traffic accidents' data sets, the data obtained after normalizing the traffic accidents' data sets conforms to the standard normal distribution. Specifically, the average value is 0, the standard deviation is 1, and the transformation function is:

$$x^* = \frac{x - u}{\sigma} \quad (11)$$

where x^* represents a single feature under a data; u is the average value of all data under a single feature; and σ is the standard deviation of all data under a single feature.

If the data sets of traffic accident are not balanced, the training of the model will focus on the data categories that account for a large proportion of the total data, while ignoring the data categories that account for a small proportion of the total data. This will eventually result in the training model over-fitting the sample categories that account for a large proportion, while under-fitting the sample categories that account for a small proportion [53], [54]. Generally speaking, there are two ways to deal with imbalanced data using sampling methods, under-sampling and over-sampling. Due to the loss of some data sets due to under-sampling, data sets cannot be fully utilized, what's more, the traffic accidents' data with categories fatal and serious is much less than slight. To make full use of traffic accidents' data sets, the improved Borderline-SMOTE2 algorithm [55] based on the Synthetic Minority Oversampling Technique (SMOTE) [56] was used to solve the problem of data imbalance, and it proved to be effective for the processing of imbalanced data [57]–[60].

To prevent testing set from being affected by synthetic data (through Borderline-SMOTE2), the testing set needs to be isolated away from the training set in the preprocessing and we only oversampled the training set. First, we randomly extract 20% of the data from the data set as the testing set, then over-sampling the remaining 80% of the data as the training set. The data distribution of ten experiments is shown in the table 3.

Since the testing set contains only the real data itself and will eventually be imbalanced, total accuracy is not

appropriate for it, Micro_F1score is used because some classes are much larger (more instances) than others in traffic accident's data set [61], moreover, in order to consider the actual application situation of the model, precision and recall are introduced to analyze the testing data set of traffic accident [62]. The calculation equation of them as follows:

$$Micro_Precision = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i} \quad (12)$$

$$Micro_Recall = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FN_i} \quad (13)$$

$$Micro_F1score = \frac{2 \times Micro_Precision \times Micro_Recall}{Micro_Precision + Micro_Recall} \quad (14)$$

where TP_i (true positive) represents a real case of $class_i$; that is, the real category is a true positive case, and the prediction category is a true positive case; and FP_i (false positive) represents a false positive case of $class_i$; that is, the real category is a negative case, and the prediction category is a positive case; FN_i (false negative) represents a false negative case of $class_i$; that is, the real category is a positive case, and the predicted category is a negative case.

C. DETERMINATION OF THE TASP-CNN NETWORK

Through the interface provided by scikit-learn [63] and a combination of GridSearchCV and RandomizeSearchCV algorithms, the parameter combination of TASP-CNN was searched for 100 epochs, and the optimal TASP-CNN hyperparameter combination was determined. Using only GridSearchCV requires high computational cost, while using only RandomizeSearchCV will find the locally optimal combination of hyperparameters. To make better use of them, RandomizeSearchCV was used to search for the globally optimal combination of hyperparameters, and GridSearchCV was used to search for the locally optimal combination of hyperparameters. Thus, the computational cost required was reduced, and it was not easy to fall into a situation of the locally optimal combination of hyperparameters. More accurate results can be obtained by adjusting the combination of hyperparameters using this cross-combination method. By establishing models with various hyperparameter combination and using 5-fold cross-validation to evaluate each model, the hyperparameter combination with the highest Micro_F1score was finally obtained. Table 4 shows the hyperparameter combination used in TASP-CNN after searching using this hybrid method.

In general, in the deep-learning model, multiple modules and multiple layers can be stacked together. Therefore, it is very important to analyze the depth of the network to understand the network behavior. Generally speaking, the depth of CNN should not be too large or too small [35], so CNN

TABLE 4. Hyperparameter combination of the TASP-CNN model.

Hyper-parameter	Best value	Description
Batch size	128	Number of training cases over which the Adagrad update is computed.
Loss function	categorical cross-entropy	The objective function or optimization score function is also known as the multiclass log loss, which is appropriate for categorical targets.
Optimizer	Adam	The gradient descent optimization.
Learning rate	0.1	The learning rate used by the Adam optimizer.
Epsilon	1e-06	Small floating-point numbers greater than 0 prevent division 0 errors.
Kernel initializer	Uniform	Uniform distribution initialization method.

TABLE 5. TASP-CNN model at different depths.

Depth	Structures of TASP-CNN	Micro F1score
depth-4	conv2d→flatten→dense→dense	0.86
depth-5	conv2d→conv2d→flatten→dense→dense	0.87
depth-6	conv2d→conv2d→conv2d→flatten→dense→dense	0.78
depth-7	conv2d→conv2d→conv2d→conv2d→flatten→dense→dense	0.76

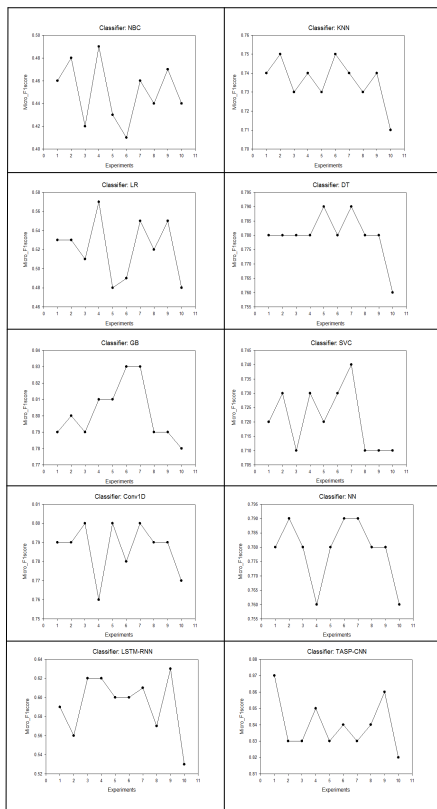


FIGURE 7. The Micro_F1score under different model in ten experiments.

can learn more complicated relationships, while maintaining the convergence of the model. Different depth values were assigned to the TASP-CNN model for testing from small to large. Table 5 lists the network structures of TASP-CNN at different depths. Experiments based on the TASP-CNN

TABLE 6. AVERAGE Micro_F1 score (Ten Experiments) under the different models.

Model	NBC	KNN	LR	DT	GB	SVC	Conv1D	NN	LSTM-RNN	TASP-CNN
Micro_F1score	0.46	0.74	0.53	0.78	0.79	0.72	0.79	0.78	0.59	0.87
Experiment 1	0.48	0.75	0.53	0.78	0.8	0.73	0.79	0.79	0.56	0.83
Experiment 2	0.42	0.73	0.51	0.78	0.79	0.71	0.8	0.78	0.62	0.83
Experiment 3	0.49	0.74	0.57	0.78	0.81	0.73	0.76	0.76	0.62	0.85
Experiment 4	0.43	0.73	0.48	0.79	0.81	0.72	0.8	0.78	0.6	0.83
Experiment 5	0.41	0.75	0.49	0.78	0.83	0.73	0.78	0.79	0.6	0.84
Experiment 6	0.46	0.74	0.55	0.79	0.83	0.74	0.8	0.79	0.61	0.83
Experiment 7	0.44	0.73	0.52	0.78	0.79	0.71	0.79	0.78	0.57	0.84
Experiment 8	0.47	0.74	0.55	0.78	0.79	0.71	0.79	0.78	0.63	0.86
Experiment 9	0.44	0.71	0.48	0.76	0.78	0.71	0.77	0.76	0.53	0.82
Experiment 10	0.45	0.736	0.521	0.78	0.802	0.721	0.787	0.779	0.593	0.84
Average	0.45	0.736	0.521	0.78	0.802	0.721	0.787	0.779	0.593	0.84

TABLE 7. Average precision and average recall predicted by different models under different traffic accident's severities.

Severity of traffic accident	Model	Average Precision	Average Recall
Slight	NBC	0.934	0.430
	KNN	0.909	0.789
	LR	0.937	0.519
	DT	0.901	0.854
	GB	0.911	0.868
	SVC	0.927	0.755
	Conv1D	0.900	0.856
	NN	0.900	0.851
	LSTM-RNN	0.924	0.613
	TASP-CNN	0.893	0.932
Serious	NBC	0.151	0.618
	KNN	0.202	0.387
	LR	0.186	0.522
	DT	0.210	0.289
	GB	0.266	0.329
	SVC	0.227	0.525
	Conv1D	0.212	0.296
	NN	0.208	0.292
	LSTM-RNN	0.190	0.429
	TASP-CNN	0.248	0.167
Fatal	NBC	0.015	0.078
	KNN	0.055	0.078
	LR	0.013	0.081
	DT	0.050	0.397
	GB	0.040	0.119
	SVC	0.031	0.313
	Conv1D	0.058	0.061
	NN	0.057	0.071
	LSTM-RNN	0.016	0.389
	TASP-CNN	0.063	0.063

network structures show the Micro_F1score of the testing set and are listed in Table 5. When the depth of TASP-CNN was 4, the Micro_F1score of the testing set was 0.86. The Micro_F1score of the testing set reached its highest at 0.87 when the depth was 5. When the depth of the TASP-CNN model was more than 5, the Micro_F1score of the testing set gradually decreased. The best Micro_F1score was achieved by using two convolution layers with 256 filters: 1 flatten layer, 1 full connection layer with 128 hidden units, and 1 softmax full connection layer with 3 hidden units. The Micro_F1score of the testing set reached 0.87. Therefore, the TASP-CNN model with a depth of 5 was used in this experiment.

D. EXPERIMENT RESULTS

To illustrate the effectiveness of the TASP-CNN model, this experiment compared the model with six statistical models and three machine learning models.

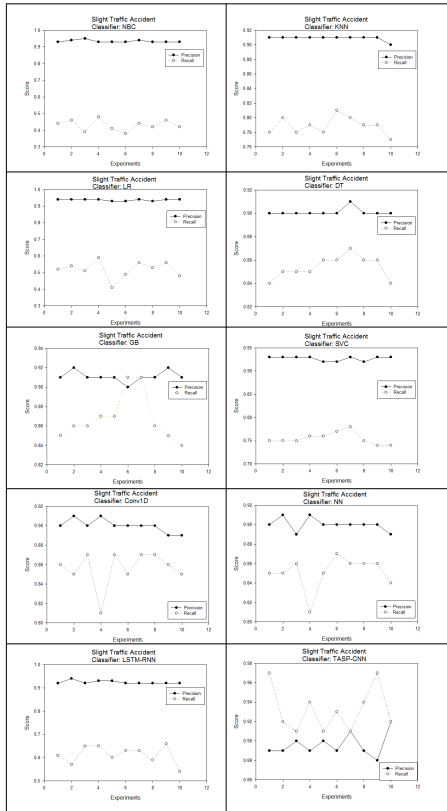


FIGURE 8. Precision and recall line chart predicted using different models under slight traffic accident.

The six of statistical models were: K-nearest neighbor algorithm (K-NN) [64], DT [65]. Naive Bayes’ classifiers (NBC) [66], Logistic regression (LR) [67], Gradient boosting (GB) [68] and Support vector machines (SVMs, also known as support vector networks) [69]. Correspondingly, the three machine learning algorithms are: neural networks (NNs) or connectionist systems [70], Long Short-Term Memory-Recurrent Neural Network (LSTM-RNN) [71]–[73] and 1D convolution (a convolutional form of convolutional neural networks).

Moreover, the above six statistical models were implemented using an interface provided by scikit-learn [63], and the parameters are optimized. The NN model was tuned to form up to five hidden layers with 128 hidden units in each layer and one softmax fully connected layer. The activation function was ReLU, and the optimizer was SGD (Stochastic Gradient Descent). In addition, the kernel initializer was uniform. LSTM-RNN was optimized to contain one LSTM layer with 128 hidden units and three hidden layers with 64, 128, and 256 hidden units in the three layers, respectively, and the last one is the softmax fully connected layer. The optimizer was SGD with a learning rate=0.001, decay=0.9, and momentum=0.8. The Conv1D’s configurations were set up to include three hidden layers with 256 hidden units in each 1D convolution layer and one softmax fully connected layer was added. Finally, the activation function was ReLU, and the optimizer was Adam [74].

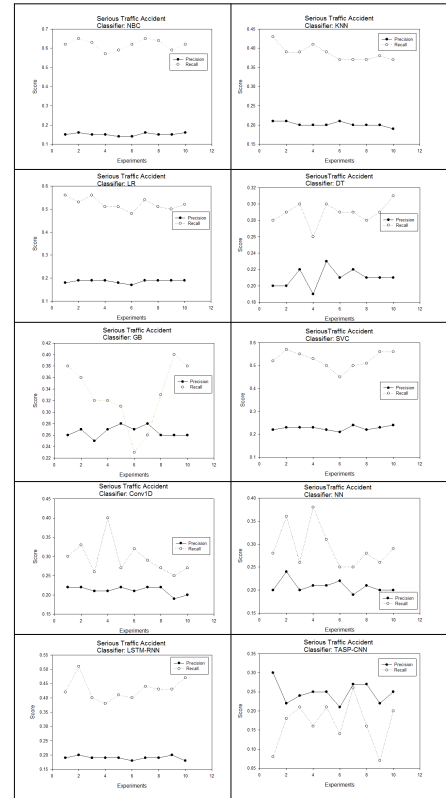


FIGURE 9. Precision and recall line chart predicted using different models under serious traffic accident.

Table 6 and Figure 7 show the average Micro_F1score and the Micro_F1score of all ten experiments after applying six statistical models, three machine learning models, and TASP-CNN to the traffic accident’s data sets. The results show that the TASP-CNN model proposed in this study is better to other statistical models and machine learning models in the Micro_F1score of the testing set. This is the evidence that TASP-CNN can be generalized to the new traffic accident’s data sets. One possible reason is that when the statistical model treated the traffic accident’s data sets, it thought that there was no local correlation between the features of the traffic accident’s data sets, and it ignored the combination relationships among the features of the traffic accident’s data sets. Similarly, these machine learning models cannot analyze the combination relationships among traffic accident’s data sets features from the perspective of model structure, and these traffic accident’s data sets features have strong combination relationships and correlation relationships, while the TASP-CNN model proposed in this study was locally perceived and fully extracted the combination relationships among the features of traffic accident’s data sets which had explained above.

The essential purpose of predicting the severity of traffic accident is to provide corresponding medical assistance to the personnel involved in the traffic accident in time, reduce casualties in the accident, inform the corresponding emergency decision-making department in time, and avoid greater property losses. Therefore, the predicted severity of traffic

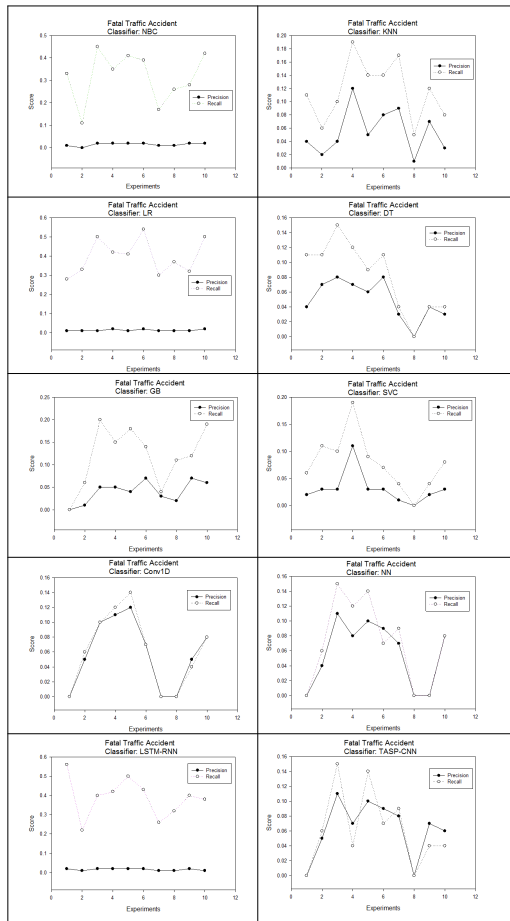


FIGURE 10. Precision and recall line chart predicted using different models under fatal traffic accident.

accident was further divided into three levels of analysis: slight traffic accident, serious traffic accident, and fatal traffic accident.

Table 7 and Figures 8-10 show the average Precision, average Recall and the Precision, Recall of all ten experiments by different models under slight, serious and fatal traffic accident.

From Table 7 and Figures 8-10, the results from the slight traffic accident’s testing set show that the precision of the NBC model is the highest compared with other models, while the recall is the highest for the TASP-CNN. The results from the serious traffic accident’s testing show that the TASP-CNN and GB have the relatively high precision compared with the other eight of different models. The results of the fatal traffic accident’s testing set show that TASP-CNN has the highest precision with other models. But by considering with the actual situation analysis, we can allow certain errors in the precision of the prediction of slight traffic accident, because slight traffic accident has high probability of not causing significant casualties and major property losses. However, for serious and fatal traffic accident, especially for fatal accident, the precision requirement of the prediction must be relatively high, because as long as the prediction is slightly inaccurate, the corresponding emergency medical

support and the decision of the corresponding emergency department may not be provided, resulting in significant casualties and properties losses. Therefore, the performance of the TASP-CNN model is better than other models when analyzed from the perspective of this combination of specific situations.

To sum up, the performance of the TASP-CNN model proposed in this study was better than that of other nine models, when it was based on the analysis of the Micro_F1score of model prediction and considering the specific application scenarios to analyze traffic accident with different severity levels.

V. CONCLUSION AND FUTURE WORK

In this paper, a deep-learning approach with a TASP-CNN model was proposed for traffic accident severity prediction. Unlike the previous methods that only consider the shallow structure of traffic accident, the proposed method successfully fund the latent traffic accident’s severity feature representation, such as the feature combination, and deeper feature correlations from traffic accident’s data. The performance of the proposed TASP-CNN model was evaluated using traffic accident’s data for an 8-year period (2009–2016) from the Leeds City Council, and it was compared with the NBC, the KNN, the LR, the DT, the GB, the SVC the Conv1D, the NN, and the LSTM-RNN models. The results show that the proposed TASP-CNN model was better to the competitive models.

For future work, it would be significant to investigate other machine learning algorithms for traffic accident’s severity prediction and apply these algorithms on different public traffic data sets to examine their effectiveness. Furthermore, the TASP-CNN model in this study is novel, especially the FM2GI algorithm for converting traffic accident’s data into gray images. Therefore, there is a need to analyze the generalization ability of the TASP-CNN model proposed in this study. More specifically, there is a need to understand whether the TASP-CNN model can be applied to other areas, and whether it is more accurate than the corresponding competitive model.

REFERENCES

- [1] *Road Traffic Injuries*. Accessed: Jul. 18, 2018. [Online]. Available: <http://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>
- [2] F. Zong, H. Xu, and H. Zhang, “Prediction for traffic accident severity: Comparing the Bayesian network and regression models,” *Math. Problems Eng.*, vol. 2013, nos. 2–3, 2013, Art. no. 475194.
- [3] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in *Proc. ACM 25th Int. Conf. Mach. Learn.*, 2008, pp. 160–167.
- [4] J. Dean et al., “Large scale distributed deep networks,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1223–1231.
- [5] L. Deng, G. Hinton, and B. Kingsbury, “New types of deep neural network learning for speech recognition and related applications: An overview,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 8599–8603.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Jun. 2016, pp. 770–778.
- [7] N. Majumder et al., “Deep learning-based document modeling for personality detection from text,” *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 74–79, Mar. 2017.

- [8] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2015, pp. 373–382.
- [9] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5005–5013.
- [10] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2015.
- [11] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, and G. Penn, "Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2012, pp. 4277–4280.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [13] Q. Mao, M. Dong, Z. Huang, and Y. Zhan, "Learning salient features for speech emotion recognition using convolutional neural networks," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2203–2213, Dec. 2014.
- [14] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1717–1724.
- [15] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1120–1124, Sep. 2014.
- [16] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
- [17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
- [18] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.
- [19] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," *Transp. Res. C*, vol. 19, no. 3, pp. 387–399, Jun. 2011.
- [20] A. S. Al-Ghamdi, "Using logistic regression to estimate the influence of accident factors on accident severity," *Accident Anal. Prevention*, vol. 34, no. 6, pp. 729–741, 2002.
- [21] M. Bédard, G. H. Guyatt, M. J. Stones, and J. P. Hirdes, "The independent contribution of driver, crash, and vehicle characteristics to driver fatalities," *Accid Anal. Prevention*, vol. 34, no. 6, pp. 717–727, 2002.
- [22] K. M. Kockelman and Y. J. Kweon, "Driver injury severity: An application of ordered probit models," *Accident Anal. Prevention*, vol. 34, no. 3, pp. 313–321, 2002.
- [23] D. W. Kononen, C. A. C. Flannagan, and S. C. Wang, "Identification and validation of a logistic regression model for predicting serious injuries associated with motor vehicle crashes," *Accident Anal. Prevention*, vol. 43, no. 1, pp. 112–122, 2011.
- [24] J. C. Milton, V. N. Shankar, and F. L. Mannering, "Highway accident severities and the mixed logit model: An exploratory empirical analysis," *Accident Anal. Prevention*, vol. 40, no. 1, pp. 260–266, 2008.
- [25] T. Yamamoto and V. N. Shankar, "Bivariate ordered-response probit model of driver's and passenger's injury severities in collisions with fixed objects," *Accident Anal. Prevention*, vol. 36, no. 5, pp. 869–876, 2004.
- [26] K. K. W. Yau, H. P. Lo, and S. H. H. Fung, "Multiple-vehicle traffic accidents in Hong Kong," *Accident Anal. Prevention*, vol. 38, no. 6, pp. 1157–1161, 2006.
- [27] L.-Y. Chang and H.-W. Wang, "Analysis of traffic injury severity: An application of non-parametric classification tree techniques," *Accident Anal. Prevention*, vol. 38, no. 5, pp. 1019–1027, Sep. 2006.
- [28] J. de Oña, R. O. Mujalli, and F. J. Calvo, "Analysis of traffic accident injury severity on Spanish rural highways using Bayesian networks," *Accident Anal. Prevention*, vol. 43, no. 1, pp. 402–411, 2011.
- [29] J. Abellán, G. López, and J. de Oña, "Analysis of traffic accident severity using decision rules via decision trees," *Expert Syst. Appl.*, vol. 40, no. 15, pp. 6047–6054, 2013.
- [30] L.-Y. Chang and J.-T. Chien, "Analysis of driver injury severity in truck-involved accidents using a non-parametric classification tree model," *Saf. Sci.*, vol. 51, no. 1, pp. 17–22, 2013.
- [31] Z. Li, P. Liu, W. Wang, and C. Xu, "Using support vector machine models for crash injury severity analysis," *Accident Anal. Prevention*, vol. 45, no. 2, pp. 478–486, 2012.
- [32] C. Chen, G. Zhang, Z. Qian, R. A. Tarefder, and Z. Tian, "Investigating driver injury severity patterns in rollover crashes using support vector machine models," *Accident Anal. Prevention*, vol. 90, pp. 128–139, May 2016.
- [33] S. H. A. Hashmiejad and S. M. H. Hasheminejad, "Traffic accident severity prediction using a novel multi-objective genetic algorithm," *Int. J. Crashworthiness*, vol. 22, no. 4, pp. 425–440, 2017.
- [34] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.
- [35] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.
- [36] Z. Xie and J. Yan, "Detecting traffic accident clusters with network kernel density estimation and local spatial statistics: An integrated approach," *J. Transp. Geogr.*, vol. 31, no. 5, pp. 64–71, Jul. 2013.
- [37] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: A GRU based deep learning approach," *IET Intell. Transp. Syst.*, vol. 12, no. 7, pp. 578–585, 2018.
- [38] M. M. Kunt, I. Aghayan, and N. Noii, "Prediction for traffic accident severity: Comparing the artificial neural network, genetic algorithm, combined genetic algorithm and pattern search methods," *Transport*, vol. 26, no. 4, pp. 353–366, 2011.
- [39] Q. Zeng and H. Huang, "A stable and optimized neural network model for crash injury severity prediction," *Accident Anal. Prevention*, vol. 73, pp. 351–358, Dec. 2014.
- [40] M. I. Sameen and B. Pradhan, "Severity prediction of traffic accidents with recurrent neural networks," *Appl. Sci.*, vol. 7, no. 6, p. 476, 2017.
- [41] Z. Wang and T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks," in *Proc. Workshops 29th AAAI Conf. Artif. Intell.*, vol. 1, 2015, pp. 40–46.
- [42] J. Yang, M. N. Nguyen, P. P. San, L. L. Xiao, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. Int. Conf. Artif. Intell.*, 2015, pp. 3995–4001.
- [43] P. Kopelias, F. Papadimitriou, K. Papandreou, and P. Prevedouras, "Urban freeway crash analysis: Geometric, operational, and weather effects on crash number and severity," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2015, no. 1, pp. 123–131, 2007.
- [44] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [45] H. Trevor, T. Robert, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009.
- [46] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees." Belmont, CA, USA: Mahwah, 1984.
- [47] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.
- [48] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006, pp. 140–155.
- [49] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: <https://arxiv.org/abs/1502.03167>
- [50] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.
- [51] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, pp. 785–794.
- [52] J. Hrus, *Data Science from Scratch: First Principles With Python*. Newton, MA, USA: O'Reilly Media, 2015.
- [53] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special issue on learning from imbalanced data sets," *ACM Sigkdd Explor. Newslett.*, vol. 6, no. 1, pp. 1–6, 2004.
- [54] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
- [55] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing (Lecture Notes in Computer Science)*, vol. 3644, no. 5. Berlin, Germany: Springer, 2005, pp. 878–887.
- [56] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.
- [57] Z. Hu and P. Jiang, "An imbalance modified deep neural network with dynamical incremental learning for chemical fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 540–550, Jan. 2019.

- [58] W. Liu, Z. Luo, and S. Li, "Improving deep ensemble vehicle classification by using selected adversarial samples," *Knowl.-Based Syst.*, vol. 160, pp. 167–175, Nov. 2018.
- [59] M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo, and J. Santos, "Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 4, pp. 59–76, Nov. 2018.
- [60] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Inf. Sci.*, vol. 465, pp. 1–20, Oct. 2018.
- [61] Y. Yang, "An evaluation of statistical approaches to text categorization," *Inf. Retr.*, vol. 1, nos. 1–2, pp. 69–90, 1999.
- [62] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.
- [63] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
- [64] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statist.*, vol. 46, no. 3, pp. 175–185, 1992.
- [65] D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man Cybern.*, vol. 21, no. 3, pp. 660–674, May 1991.
- [66] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. London, U.K.: Pearson, 2016.
- [67] G. Rodríguez. (2007). *Lecture Notes on Generalized Linear Models*. Accessed: Nov. 14, 2018. [Online]. Available: <http://data.princeton.edu/wws509/notes/>
- [68] L. Breiman, "Arcing the edge," Dept. Statist., Univ. California Berkeley, Berkeley, CA, USA, Tech. Rep. 486, 1997.
- [69] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [70] M. van Gerven and S. Bohte, "Editorial: Artificial neural networks as models of neural information processing," *Frontiers Comput. Neurosci.* vol. 11, p. 114, Feb. 2017.
- [71] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2342–2350.
- [72] J. Koutník, K. Greff, F. Gomez, and J. Schmidhuber. (2014). "A clockwork RNN." [Online]. Available: <https://arxiv.org/abs/1402.3511>
- [73] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.
- [74] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: <https://arxiv.org/abs/1412.6980>



MING ZHENG was born in Anqing, Anhui, China, in 1992. He received the B.S. degree in information and computing science from the School of Mathematics and Statistics, Yunnan University, Kunming, China, where he is currently pursuing the Ph.D. degree in information and communication engineering with the School of Information Science and Engineering. His research interests include data mining, and machine learning and its application.



and the Director of the Yunnan Key Laboratory of Software Engineering. His research interest includes the principal of psychology in Yunnan cloud computing engineering research.

TONG LI was born in Kunming, Yunnan, China, in 1963. He received the Ph.D. degree in software engineering from the School of Computer Science and Engineering, De Montfort University, U.K. He is currently a Professor, a Doctoral Tutor, a young and middle-aged Academic and Technical Leader in Yunnan Province, a famous Teacher in Yunnan Province, and a Leader of the teaching team in Yunnan Province, the Director of the National Linux Technology Training and Promotion Center,



Ph.D. Scholar Newcomer Award, and a Secretary of the Yunnan Provincial Software Engineering Teaching Committee.

RUI ZHU was born in Kaifeng, Henan, China, in 1987. He received the Ph.D. degree in software engineering from Yunnan University, Kunming, China, in 2016, where he is currently a Lecturer with the School of Software. His research interest includes data mining. He is an important member of the Science and Technology Innovation Team, Data-Driven Software Engineering Department, Yunnan University, a scholar of the Ali Living Water Project, a winner of the Yunnan Province



JING CHEN was born in Qujing, Yunnan, China, in 1988. She received the master's degree from Yunnan University. She is currently a Senior Engineer with the Yunnan Science Research Institute of Communication and transportation. Her research interests include data mining and deep learning.



ZIFEI MA was born in Kunming, Yunnan, China, in 1990. He is currently pursuing the Ph.D. degree with Yunnan University with a focus on system analysis and integration. His current research interests include machine learning and data mining.



MINGJING TANG was born in Yuanling, Hunan, China, in 1978. He is currently pursuing the Ph.D. degree with Yunnan University, Kunming, Yunnan, China. He is also an Associate Professor with the President Office, Yunnan Normal University, Kunming. His research interests include data mining and machine learning.



ZHONGQIANG CUI was born in Liaocheng, Shandong, China, in 1993. He received the bachelor's degree in network engineering from the School of Software, Yunnan University, Kunming, where he is currently pursuing the master's degree in systems analysis and integration. His research interests include deep learning and machine learning.



ZHAN WANG was born in Zhaotong, Yunnan, China, in 1996. He received the bachelor's degree from the School of Software, Yunnan University, Kunming. His current research interest includes data mining.

...