Received: 13 September 2017 | Revised: 18 June 2018 | Accepted: 30 June 2018

DOI: 10.1002/for.2542

RESEARCH ARTICLE

WILEY

Forecasting accident frequency of an urban road network: A comparison of four artificial neural network techniques

Hamid Behbahani[1] | Amir Mohamadian Amiri[2] | Reza Imaninasab[3] | Meysam Alizamir[4]

[1] Department of Civil Engineering, Iran University of Science and Technology, Tehran, Iran

[2] Department of Highway Engineering, Iran University of Science and Technology, Tehran, Iran

[3] Department of Highway Engineering, Lyles School of Civil Engineering, Purdue University, West Lafayette, IN, USA

[4] Young Researchers and Elite Club, Islamic Azad University, Hamedan, Iran

Correspondence
Hamid Behbahani, Department of Civil Engineering, Iran University of Science and Technology, Narmak, PO Box 1684613114, Tehran, Iran.
Email: behbahani@iust.ac.ir

Abstract

Considerable effort has been made to determine which of the most common prediction modeling techniques performs best, based on crash-related data. Accordingly, the present study aims to evaluate how crashes in the urban road network are affected by contributing factors. Therefore, in the present paper, a comparison has been done among four artificial neural network (ANN) techniques: extreme learning machine (ELM), probabilistic neural network (PNN), radial basis function (RBF), and multilayer perceptron (MLP). According to the measures used, including Nash–Sutcliffe (NS), mean absolute error (MAE), and root mean square error (RMSE), ELM was found to be the most successful approach in addressing the objectives defined in the present study. Moreover, not only is ELM the fastest algorithm due to its different structure, but it has also led to the most accurate prediction. In the end, the RReliefF algorithm was utilized to find the importance of variables used, including V/C, speed, vehicle kilometer traveled (VKT), roadway width, existence of median, and allowable/not-allowable parking. It was proved that VKT is the most influential variable in accident occurrence, followed by two traffic flow characteristics: V/C and speed.

KEYWORDS

accident frequency, artificial neural network, extreme learning machine, RReliefF algorithm

1 | INTRODUCTION

Traffic accidents are associated with fatality, injuries, financial losses, and delays as direct costs, and with energy waste, missed workdays, and economic and psychological consequences as some of the indirect costs (Nassiri, Najaf, & Amiri, 2014). With this in mind, many efforts have been made to lessen the number of accidents on rural and urban roads.

In recent decades, more researchers have been focusing on predicting accident frequencies in order to identify and rank effective parameters in its occurrence. Researchers have tried to employ more capable approaches and develop models that more accurately estimate the number of accidents and better correlate governing variables to accident frequency.

Generally, accident prediction models can be categorized into two major groups: statistical techniques such as regression on the one hand and artificial intelligence (AI) approaches on the other. Single and multivariate deterministic models, probabilistic models and multiple logistic models are the most well-known statistical approaches. For more than two decades, probabilistic models such as Poisson and negative binomial (NB) have drawn most attention in the field of traffic safety. However, these statistical methods are developed based on strong assumptions and predefined underlying relationships between variables. Once these assumptions are violated, the model can result in an erroneous prediction of accident frequency. Therefore, since AI approaches have been proven to be more efficient and effective in the modeling process, they have been employed frequently to forecast accident frequency.

AI approaches, including conventional intelligence and computational intelligence (CI), are capable of mapping the nonlinear functions of variables (Ivancevic & Ivancevic, 2007). Since these kinds of techniques can be used as an arbitrary approximation procedure, they can learn from observed data without any kind of predefined assumptions. Furthermore, their simple implementation and the existence of mostly local dependencies in their structure make them relatively fast and also beneficial (Mittal, 2016).

Generally, CI techniques can be classified into the several subsets such as artificial neural network (ANN), fuzzy logic (FL), evolutionary computation (EC), and so forth. EC mostly involves metaheuristic optimization algorithms such as genetic algorithms (GA), evolutionary programming (EP), evolution strategy (ES), learning classifier systems (LCS), and swarm intelligence (SI). SI itself includes ant colony optimization (ACO) and particle swarm optimization (PSO) (Bittermann, 2010).

In the present paper, different ANN techniques, as the most popular method for data training and model development, were used to predict accident frequency based on contributing factors; subsequently, their results were compared to find the most efficient approach. The ANN methods used in this research are multilayer perceptron (MLP), radial basis function (RBF) and probabilistic neural network (PNN), and extreme learning machine (ELM). Among these, ELM is a novel learning algorithm and has never been used for accident frequency prediction. In addition, to evaluate the importance of each contributing factor, the RReliefF algorithm was used as a separate analysis.

In the following, a brief overview of past studies that were done in this area will be discussed. Subsequently, the data used in this research and the methodology of study will be explained comprehensively. Finally, the results of these methods were compared with each other to propose the best technique and relative weight of each variable was computed.

2 | LITERATURE REVIEW

ANNs have been widely applied to various transportation problems such as incident detection, travel behavior analysis, traffic parameters estimation, vehicle emissions, and traffic signal control. This inclination may come from their ability to deal with massive amounts of multidimensional data, their modeling flexibility, their learning and generalization ability, adaptability and their—generally—good predictive ability.

In the field of traffic safety, several studies have applied ANN to forecast traffic accidents based on a set of contributing factors (Chiou, 2006; Sameen & Pradhan, 2017). Furthermore, based on some previous studies results, ANNs have shown better performance in comparison with traditional approaches, especially when nonlinear safety effects of risk factors have been identified (Chang, 2005; Xie & Klionsky, 2007).

Chang (2005) compared an ANN with a negative binomial regression to predict the accident frequency using a specific accident data set obtained from a national freeway in Taiwan. According to the results, both approaches provided relatively similar results in terms of prediction performance in the process of data training and testing; however, the author mentioned that ANN is better able to analyze freeway accident frequency. Because the ANN not only do not require strong assumptions about the underlying relationship between dependent and independent variables but also can effectively tackle interrelation problems between independent variables.

Mussone et al., in two separate studies, applied a three-layer ANN approach to estimate the accident likelihood using contributing factors including accident sites, road characteristics, weather and surface conditions, and human error. Their modeling results showed that ANN could be a good alternative method for evaluating accident forecasting in Milan, Italy (Mussone, Ferrari, & Oneta, 1999; Mussone, Rinelli, & Reitani, 1996). In another study, Abdel-Aty and Pande (2005) employed a PNN model to forecast the likelihood of accident occurrence on the Interstate 4 corridor in Orlando. Based on the results, the PNN model could identify at least 70% of the crashes correctly. Abdelwahab and Abdel-Aty (2001) used an MLP in forecasting traffic accidents in signalized intersections. They compared its results with the fuzzy approach, in which the MLP classification showed relatively better performance. Akgüngör and Doğan (2009) employed an ANN to predict the number of accidents, injuries, and fatalities in Ankara and subsequently compared its results with a GA Approach. According to the results, the ANN showed better performance compared with the GA.

In addition to the previously mentioned ANN techniques, the present paper utilized a new approach called ELM, which is known as a novel learning algorithm with a single hidden layer feedforward networks. In spite of high ability, ELM has rarely been used in the field of transportation. Zhu, Miao, Hu, and Qing (2014) used an ELM to detect the virtual roads and vehicles in a driving simulation. To evaluate the capability of ELM, they compared its results with support vector machine (SVM) and backpropagation

(BP). They concluded that ELM had the fastest performance on road segmentation and vehicle detection with similar accuracy compared with the other techniques. It should be noted that the capabilities of this technique have not yet been used in traffic safety evaluation.

As mentioned, in spite of the steady progress of methodological innovation in the field of traffic safety, many fundamental issues have not been completely addressed. Therefore, in the present paper, the aim is to reveal more aspects of these advanced models as well as their probable capabilities.

3 | DATA DESCRIPTION

The collected data set in this research is traffic and accident data from roadway segments of the city of Mashhad, Iran, in the year 2014. There are total of 2,338 accidents on 194 roadway segments, and the three sets of variables in a disaggregated approach are: (a) the characteristics of the crash; (b) the characteristics of the traffic flow at the time of the crash, derived from macroscopic traffic simulation software; and (c) environmental conditions, such as highway geometry.

Most previous research has focused on determining the relationship between accidents and highway traffic volume, either at aggregated levels using annual average daily traffic (AADT) or disaggregated levels by hourly volumes (HV) (Imaninasab, Sekhavati, & Hajihoseinloo, 2016; Martin, 2002; Mensah & Hauer, 1998). However, aggregated traffic parameters cannot indicate prevailing traffic conditions at the time of an accident occurrence, whereas a disaggregated approach can reflect the true condition more accurately than previous parameters. Therefore, the present study uses hourly traffic parameter including V/C and speed as an independent variable. The simultaneous use of these two parameters makes it possible to determine the actual traffic conditions at the time of accident occurrence. Since the actual data extracted from the loop detectors were not available, VISUM[1] outputs were used. Similarly, Hadayeghi, Shalaby, and Persaud (2003) and Lord (2000) used EMME/2 outputs in developing accident prediction models. In addition to congestion parameters, according to several previous studies, speed has been considered as one of the main traffic-related crash contributory factors (Abdel-Aty, Uddin, & Pande, 2005; Elvik, Christensen, & Amundsen, 2004). The combination of V/C and speed as traffic flow characteristics could effectively reflect realities on the road segment before accident occurrence. VKT shows the amount of travel spent in a given segment. Indeed, the existence of this parameter in the model indicates how much the risk of traveling in a specific section is.

Whether parking is permitted in the right lane of the roadway, the presence or lack of median as a divider, and the width of the roadway are other independent variables regarding roadway characteristics. Table 1 shows the independent variables used in the models.

4 | METHODOLOGY

As mentioned, AI approaches include diverse types of analysis; however, in this paper, the focus is on ANN techniques, including RBF, MLP, PNN, and ELM, which were utilized to predict accident frequency in Mashhad city, Iran, using the data set described in the previous section. The reason why ANNs were chosen in the presented paper, among other AI techniques, is that they are able to solve road accident forecasting problems involving complex interrelationships between variables efficiently and provide realistic and fast ways for developing models (Abdelwahab & Abdel-Aty, 2001). ANN uses past experience to learn how to deal with new and unexpected situations; therefore, the statistical distribution of the data does not need to be known. Furthermore, there is no need for prior knowledge about the relationships among the variables used in the modeling process (Karayiannis & Venetsanopoulos, 1993).

Afterwards, the results of performing these methods were compared to each other to find out which technique better suits the data set. In the end, in order to investigate the importance of the input variables, each of them was weighted using the RReliefF algorithm. MATLAB 2010 software was used to perform the ANN techniques and RReliefF algorithm. To compare the predictive efficiency of the proposed models, four criteria of root mean square error (RMSE), Nash–Sutcliffe (NS), and mean absolute error (MAE) were employed to evaluate the modeling performance with respect to the data training and testing processes. MAE is a measure used to investigate how close the predictions are to the eventual outcomes. RMSE reveals the sample standard deviation of the differences between predicted and observed data. The NS coefficient indicates the efficiency of the model through relating the goodness-of-fit of the model to the variance of the measured data. This measure can range from $-\infty$ to 1, in which 1 represents a case with a perfect match between model outcomes and the observed values. The following formulas show how these measures can be calculated:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} \left|\widehat{y}_i - y_i\right|, \tag{1}$$

[1] VISUM is a comprehensive, flexible software system for transportation planning, travel demand modeling, and network data management.

TABLE 1 Summary descriptions of relevant input variables

Variable	Description	Unit	Min.	Max.	Mean	SD
V/C	Volume-to-capacity ratio	Ratio	0.01	2.84	0.50	0.44
Speed	Average speed of traffic flow	km/h	7.33	65.09	37.14	10.93
ln (VKT)	ln (vehicle-kilometer traveled)	veh.km/h	6.80	16.96	13.24	1.35
Parking	Parking is permitted on the right lane of the roadway/otherwise	Binary	0	1	0.34	0.50
Two way	Road segment is two-way/otherwise	Binary	0	1	.072	0.47
Width	Width of roadway	Meter	4	30.61	14.60	4.73
Median	Roadway has median/otherwise	Binary	0	1	0.64	0.48

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\widehat{y}_i-y_i\right)^2}, \tag{2}$$

$$\mathrm{NS}=1-\frac{\sum_{i=1}^{n}\left(\widehat{y}_i-y_i\right)^2}{\sum_{i=1}^{n}\left(\widehat{y}_i-\overline{y}\right)^2}, \tag{3}$$

where n is the amount of data for the testing set, y_i and $\widehat{y}_i$ are predicted and observed number of crashes respectively, and $\overline{y}$ is the mean value. Following is a brief overview of ANN techniques used in this study.

4.1 | Artificial neural network

ANNs are made up of many artificial neurons, each having one input (x_i) with its own weight (w_i). The weights are floating point numbers that are adjusted in the final trained network. ANN can be simply expressed by

$$a=\sum_{i=1}^{n}w_i x_i, \tag{4}$$

where a is the output and n is the number of inputs.

ANNs used in this study are feedforward networks: Neurons in each layer feed their output forward to the next layer, known as hidden layers, until the final output is obtained from the neural network. Feedforward neural networks have the capability to (1) generate models for a large class of artificial and natural phenomena that are not easy to manage using classical parametric techniques, and (2) approximate complex nonlinear mappings directly from the input samples. Therefore, they can be successfully applied to many fields (Sajjadi et al., 2016). Figure 1 illustrates how it works.

Becoming familiar with ANN, each modeling technique is explained in details as follows.

4.1.1 | Multilayer perceptron

MLPs are able to describe any general feedforward network with backpropagation algorithm for training data. Perceptron learning theorem defines a perceptron as a threshold unit within a layer with the ability to learn anything that can represent OR, which is anything separable with a hyperplane. Since MLP cannot be separated linearly, it is not able to show an exclusive OR. In order to tackle such a restriction involved with an XOR problem, additional perceptron units can be introduced into the second layer of units. Equation (5) shows how the value of perceptron (y_i) can be calculated:

$$y_i=\sum\nolimits_{j=1}^{m}f\left(w_{ij}x_j+b_i\right), \tag{5}$$

where m is the number of inputs in the previous layer and y_i is the perceptron.

The main architectural characteristics of the MLP approach can be described as follows:

- Input or output units do not dictate the number of hidden units per layer; that is, it may be more or less than input or output units.
- There are no direct connections between output and input layers.
- More than three layers are often applied.
- There are no connections within a layer but there are full connections between layers.
- The number of output and input units do not require to be equal.

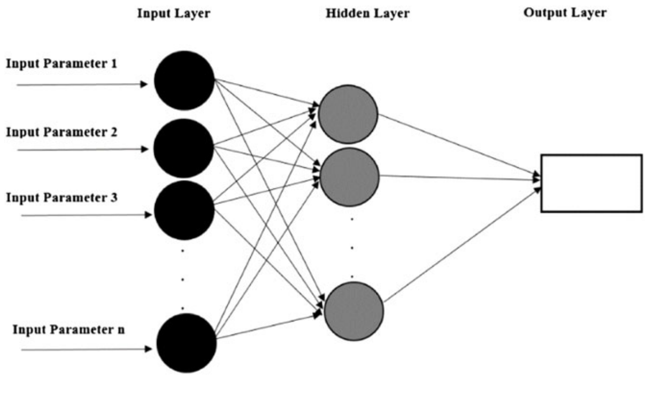

FIGURE 1 Feedforward network

The steps of MLP for forwarding activity and backpropagation of error are as follows:

- Choose a learning rate η and initialize weights at random.
- Until the network is trained, for each training example (input pattern and target output (s)):
  - Do a forward pass through the net (with fixed weights) to produce output (s), assuming j hidden layer nodes and n inputs for an MLP:

$$y_k = f\left(\sum_{j=0}^{j} w_{jk} o_j\right), \tag{6}$$

where o_j is output from each node:

$$o_j = f\left(\sum_{i=0}^{n} w_{ij} x_i\right)). \tag{7}$$

  - After the forward activity, there is backpropagation of error with the following steps:
    - For each output unit k, compute delta as follows:

$$\delta_j = o_j\left(1 - o_j\right)\sum_k w_{jk}\delta_k. \tag{8}$$

  - For all weights, change by gradient descent:

$$\Delta w_{ij} = \eta \delta_j y_i. \tag{9}$$

  - From input layer i to hidden layer unit j, the weight changes by

$$\Delta w_{ij} = \eta \delta_j x_i. \tag{10}$$

  - The number of changes in weight from hidden layer unit j to output layer unit k:

$$\Delta w_{ij} = \eta \delta_k o_j. \tag{11}$$

4.1.2 | Radial basis function

The hidden to output layer part is similar to the standard feedforward MLP network; that is, the output unit activations are given by the sum of the weighted hidden unit activations. The basic functions of φ_j (x, $\mu_{\rm j}$, σ_j) give the hidden unit activations and the functions are dependent on σ_j, the weights $\mu_{\rm ij}$, and input activations x_i in a nonstandard manner (Figure 2).

The Gaussian basis functions and y_k functions are computed using Equations (12) and (13):

$$\varnothing_j(x) = \exp\left(-\frac{\left|x-\mu_j\right|^2}{2\sigma_j^2}\right), \tag{12}$$

$$y_k(x) = \sum_{j=0}^{m} w_{kj} \varnothing_j(x), \tag{13}$$

where μ_j is the, mean σ_j is standard deviation, w_{kj} is weight, and m is the number of hidden layer units. These parameters are required to be wisely determined.

The hidden and output layers in RBF networks operate very differently from MLPs, with corresponding

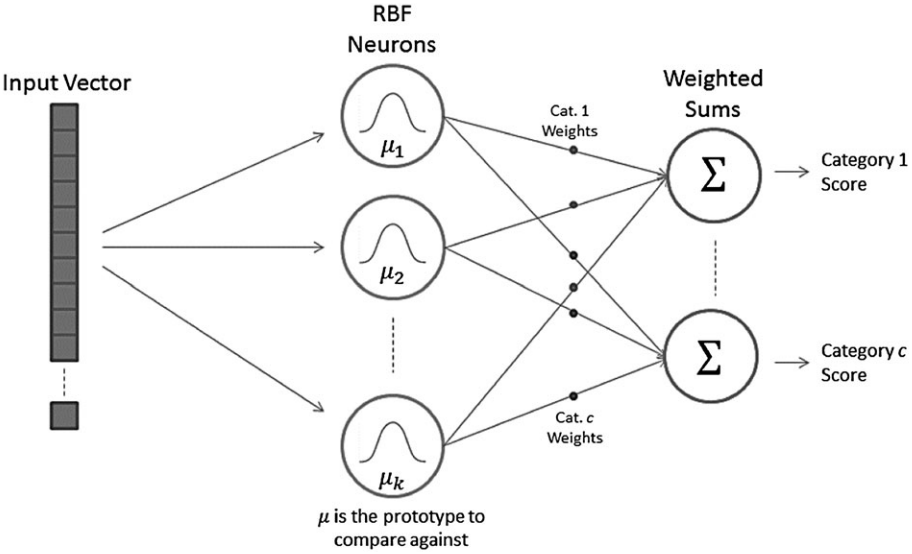

FIGURE 2 The RBF network architecture

weights having very different properties and meanings. Thus the use of different learning algorithms for them is recommended. One of the unsupervised learning techniques such as clustering-based approaches, fixed centers selected at random and orthogonal least squares can be used for training the input to hidden weights (i.e., basis function parameters μ_{ij} and σ_j). On the other hand, the simplest and quickest approach for setting the RBF parameters is to have their centres fixed at M points selected at random from the data points and it can be determined by comparing the results.

After finding the input to hidden weights, they are kept fixed while the hidden to output weights are learned. This second stage of training only involves linear output activation functions and a single layer of weights w_{jk}, and using simple matrix pseudoinversion weights can be found very easily and quickly. Since the hidden unit activations φ_j (x, μ_j, σ_j) are fixed while the output weights w_{jk} are determined, it is essential only to find the weights that optimize a single-layer linear network. Similar to MLPs, it can be defined using a sum-squared output error measure:

$$E = 0.5\sum_p\sum_k\left[t_k^p - y_k(x^p)\right]^2. \tag{14}$$

The gradients with respect to all the weights w_{ki} will be zero at the minimum of E, so

$$\frac{\partial E}{\partial w_{ki}} = \sum_p\left[w_{kj}\varnothing_j(x^p)\right]\varnothing_j(x^p) = 0. \tag{15}$$

It is more convenient to calculate weights using matrix form by defining matrices with components $(W)_{kj} = w_{kj}$, $(\Phi)_{pj} = \varphi_j\ (x^p)$, and $(T)_{pk} = \{t_k{}^p\}$. This gives

$$\varnothing^T\left(\mathbf{T} - \varnothing\mathbf{W}^T\right) = 0, \tag{16}$$

$$\mathbf{W}^T = \left(\varnothing^T\varnothing\right)^{-\mathbf{1}}\varnothing^T\mathbf{T} \tag{17}$$

There is a standard pseudo-inverse of $\varnothing$ which is $(\varnothing^T\varnothing)^{-\mathbf{1}}\varnothing^T$and is equal to unity matrix I.

4.1.3 | Probabilistic neural network

As shown in Figure 3, PNN consists of four layers—input, hidden, summation, and output layers—and can have many classes as well. The first layer includes N nodes, with one for each of the N input features of a feature vector. These are fan-out nodes that branch at each feature input node to all nodes in the hidden layer, so that each hidden node receives the complete input feature vector **x**.

Each hidden node in the group for Class k corresponds to a Gaussian function centered on its associated feature vector in the kth class (there is a Gaussian for each exemplar feature vector). All of the Gaussians in a class group feed their functional values to the same

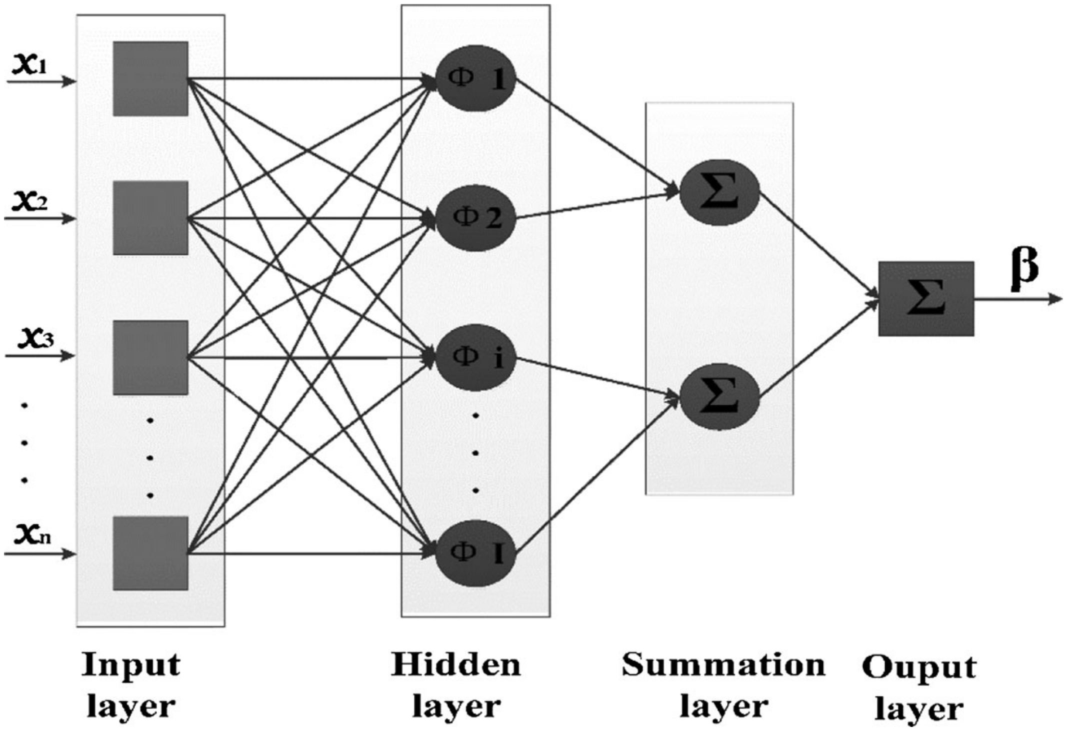

FIGURE 3 The PNN network architecture

output layer node for that class; therefore, there are K output nodes (Gu, Shi, & Wang, 2012).

At the output node for Class k, all of the Gaussian values for Class k are summed. Subsequently, the result is scaled to the probability volume under the sum function so that the sum forms a probability density function (PDF). The equation for each Gaussian centered on each class points x_i^p (feature vectors) can be calculated as follows:

$$g_i(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi\sigma^{2N}}}\right)\exp\left(\frac{-|\mathbf{x}-\mathbf{x}^p|^2}{2\sigma^2}\right). \tag{18}$$

The σ values can be taken to be one-half the average distance between the feature vectors in the same group, or at each exemplar it can be one half the distance from the exemplar to its nearest another exemplar vectors. The kth output node sums the values received from the hidden nodes in the kth group, called mixed Gaussians or Parzen windows. The sums are defined by

$$f_i(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi\sigma^{2N}}}\right)\frac{1}{p}\sum_{p=1,P}\exp\left(\frac{-|\mathbf{x}-\mathbf{x}^p|^2}{2\sigma^2}\right), \tag{19}$$

where $\mathbf{x}$ is any input feature vector, σ is the spread parameters (standard deviations) for Gaussians in each class, N is the dimension of the input vectors, P is the number of center vectors of the class, x^p is centered in the classes, and $|\mathbf{x} - \mathbf{x}^p|^2$ is the Euclidean distance (square root of the sum of squared differences) between x and x^p.

Any input vector $\mathbf{x}$ is put through both sum functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ and the maximum value (maximum a posteriori or MAP value) of f_i $(\mathbf{x})$ decides the class. For $K > 2$ classes the process is analogous.

There is no iteration or computation of weights. The error buildup can be significant for a large number of Gaussians in a sum. Therefore, thinning those that are too close to another one and making σ larger may reduce the feature vectors in each class.

4.1.4 | Extreme learning machine

The main problem with traditional learning algorithms is that they usually require a lot of time to train neural networks; accordingly, the need for faster and more accurate algorithms is always felt. To address this goal, ELM was developed as a novel learning approach that is able to solve problems far faster than traditional learning algorithms (Huang, Zhu, & Siew, 2006).

In Figure 4 the ELM network architecture is shown. For M arbitrary samples (x_i, t_i), in which $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in R^n$ and $t_i = [t_{i1}, t_{i2}, ..., t_{in}]^T \in R^m$, standard single hidden layer feedforward networks (SLFNs) with N hidden nodes and activation function g (x) are modeled as follows:

$$\sum_{i=1}^{M}\beta_i g_i(x_i) = \sum_{i=1}^{M}\beta_i g_i(w_i x_i + b_i) = O_j, j = 1, ..., N, \tag{20}$$

where $w_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ is the weight vector between input and hidden nodes, $\beta_i = [\beta_i{}^1, \beta_i{}^2, ..., \beta_i{}^n]^{\mathrm{T}}$ is the weight vector between output and hidden nodes, and $\beta_i s$ the threshold of the ith hidden node. The standard single hidden layer feedforward networks with M

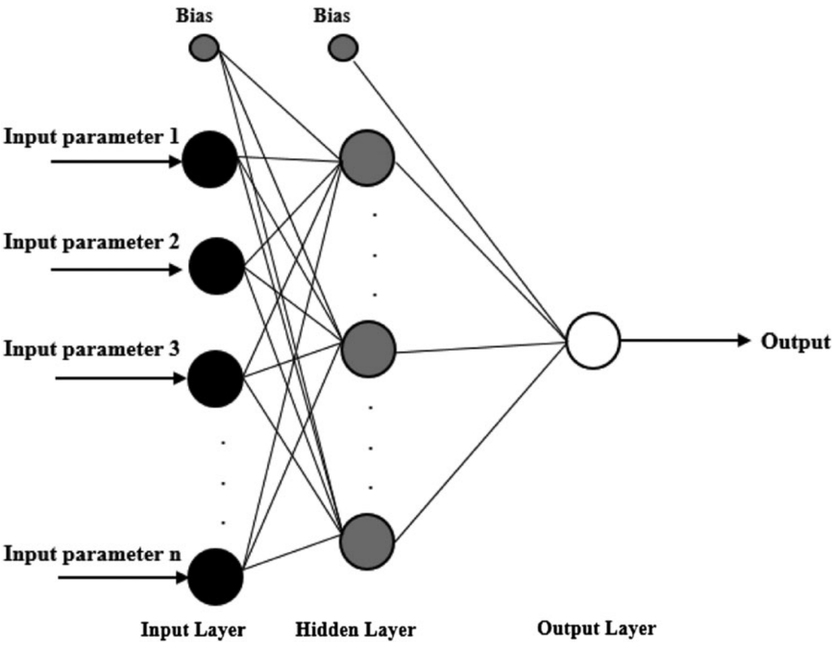

FIGURE 4 The ELM network architecture

hidden nodes with activation function $g\ (x)$ are shown as follows (Sajjadi et al., 2016):

$$\sum_{i=1}^{M} \beta_i g_i(w_i x_i + b_i) = t_j,\ j = 1, ..., N. \tag{21}$$

These equations can be written as follows:

$$\mathbf{H}\beta = \mathbf{T}, \tag{22}$$

where

$$H = \begin{pmatrix} g(w_1 x_1 + b_1) & \cdots & g(w_M x_1 + b_M) \\ \vdots & \ddots & \vdots \\ g(w_1 x_N + b_1) & \cdots & g(w_M x_N + b_M) \end{pmatrix}, \tag{23}$$

$$\beta = \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_M^T \end{pmatrix}, \tag{24}$$

$$T = \begin{pmatrix} t_1^T \\ \vdots \\ t_M^T \end{pmatrix}, \tag{25}$$

where H is the hidden-layer output matrix on neural network. The output weights can be made by finding least square solutions as follows:

$$\beta = H^{\dagger} T, \tag{26}$$

where $H^{\dagger}$ is the Moore–Penrose generalized inverse of the hidden layer output matrix H.

4.2 | RReliefF algorithm

RReliefF is a regressional ReliefF that is itself the upgraded version of the Relief algorithm. Relief approximates the quality of attributes based on how well their values distinguish between instances that are near to each other. It searches for the two nearest neighbors of a randomly selected instance (R_i): one from the same class (nearest hit H) and another from a different class (nearest miss M). Depending on their values for R_i, M, and H, the quality estimation ($W\ [A]$) for all attributes (A) is updated. Having different values of the attribute A for instances R_i and H lead to separation of the two instances with the same class, which is not desirable, by the attribute A and, therefore, there is a decrease in the quality estimation $W\ [A]$. On the other hand if instances R_i and M have different values of the attribute A then the attribute A separates two instances with different class values which is desirable so we increase the quality estimation $W\ [A]$. The whole process is repeated m times, where m is a user-defined parameter. Equation 27 indicates that $W\ [A]$ is an approximation of the following difference of probabilities (Robnik-Šikonja & Kononenko, 2003):

$$W[A] = P(\text{different value of } A|\text{nearest instances from different class}) - P(\text{different value of } A|\text{nearest instances from same class}) \tag{27}$$

Since accident frequency prediction is a regression problem with a continuously predicted value, nearest hits and misses cannot be used. Therefore, instead of identifying whether two instances belong to the same class or not, a kind of probability that the predicted values of two instances are different is introduced. This probability can be modeled with the relative distance between the predicted (class) values of two instances. Thus computation of $W\ [A]$ can be reformulated as follows (Robnik-Šikonja & Kononenko, 2003):

$$W[A] = \frac{P_{\mathrm{diff}C|\mathrm{diff}A} P_{\mathrm{diff}A}}{P_{\mathrm{diff}C}} - \frac{\left(1 - P_{\mathrm{diff}C|\mathrm{diff}A}\right) P_{\mathrm{diff}A}}{1 - P_{\mathrm{diff}C}}, \tag{28}$$

where

$$P_{\mathrm{diff}A} = P(\text{different value of } A|\text{nearest instances}), \tag{29}$$

$$P_{\mathrm{diff}C} = P(\text{different prediction}|\text{nearest instances}), \tag{30}$$

$$P_{\mathrm{diff}C|\mathrm{diff}A} = P(\text{different prediction}|\text{different value of } A \text{ and nearest instances}) \tag{31}$$

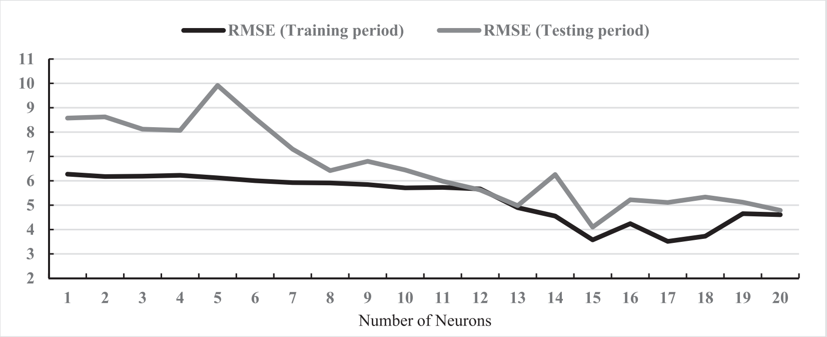

FIGURE 5 RMSEs for different numbers of neurons in the ELM model

TABLE 2 Comparison of different ANN techniques for forecasting accident frequency

	RMSE		NS		MAE	
Model	**Training**	**Testing**	**Training**	**Testing**	**Training**	**Testing**
ELM	3.576	4.1015	0.81	0.75	2.5062	1.8393
ANN	3.7077	4.1804	0.797	0.74	2.6709	2.9873
PNN	3.8384	4.2955	0.78	0.729	2.7705	3.2446
RBF	4.7619	5.6402	0.665	0.53	3.3313	4.2658

5 | MODELING RESULTS AND DISCUSSION

The data set is divided into two categories. Accordingly, 80% of the roadway segments' observations (i.e., 149 roadway segments) is used to calibrate the models and assess the models' fitting ability (i.e., first part or training set of samples). The remaining 20% of roadway segments' observations is utilized to compare the prediction capability of the models and compute error values (i.e., second part or testing set of samples). In this section, modeling development includes the evaluation of the models' fitting ability based on the training set of samples and

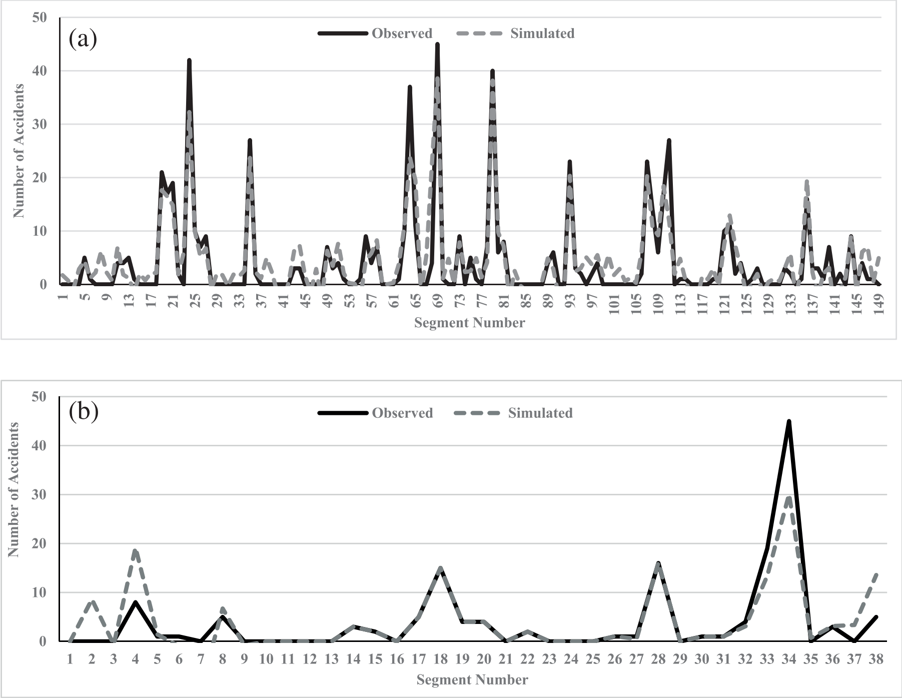

FIGURE 6 Observed and forecasted number of accidents using ELM model: (a) training period; (b) testing period

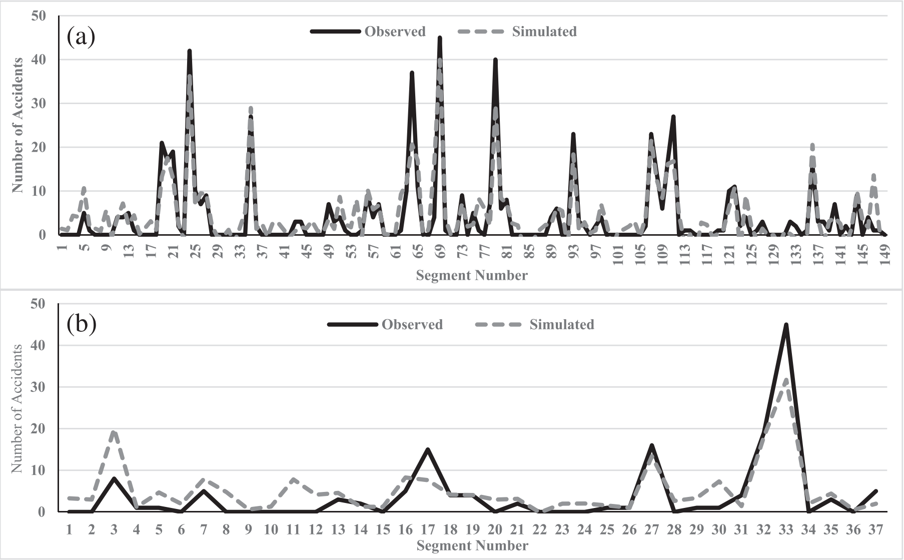

FIGURE 7 Observed and forecasted number of accidents using MLP model: (a) training period; (b) testing period

evaluation of the models' prediction ability based on the testing set of samples (Prediction). In addition, the parameters used in the modeling process were analyzed based on their importance as well as their impact on predicting the number of accidents.

In addition to the importance of input variables, the designing network should be able to identify the network structure that results in the most accurate prediction. This is due to the fact that changes in network structure, even without any change in input variables, can lead to quite different results. Therefore, the best network design is conducted using the trial-and-error approach.

5.1 | Modeling development

Network design is done using a trial-and-error approach, with the ultimate objective of finding the most effective structure that could result in more accurate predictions.

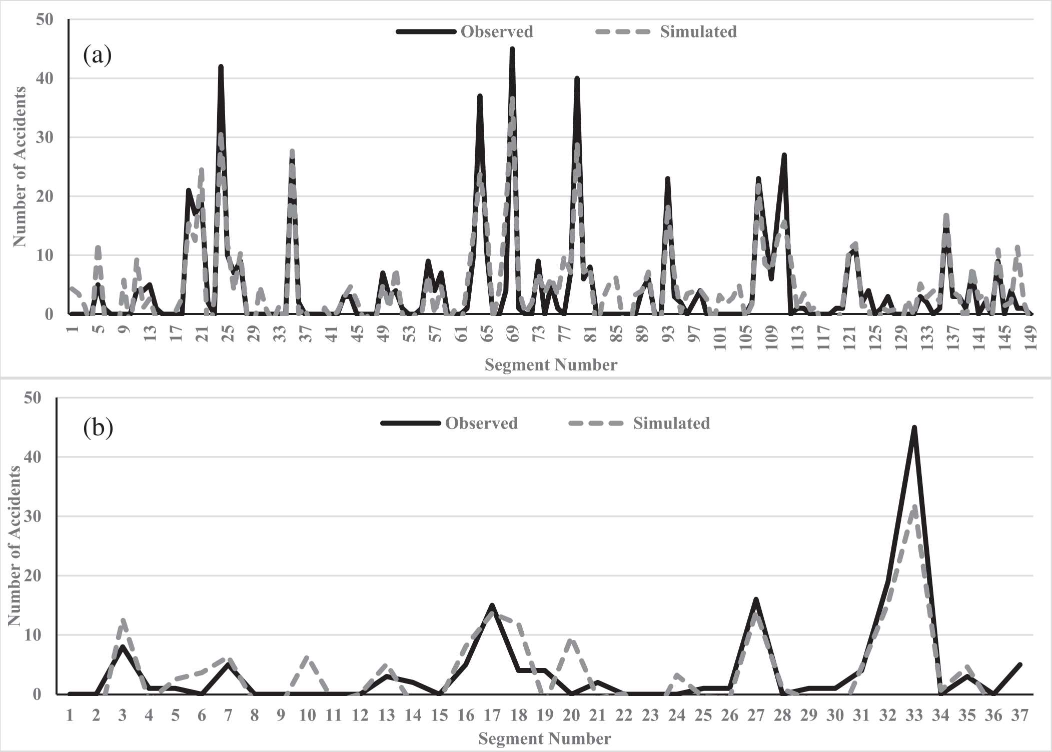

FIGURE 8 Observed and forecasted number of accidents using PNN model: (a) training period; (b) testing period

Figure 5 shows the performance of ELM within training and testing processes using RMSE versus the number of neurons. It can be seen that 15 is the optimal number of neurons leading to the most accurate results.

Unlike other ANN approaches that require setting considerable predefined parameters, which enhance the chance of getting into a local minimum problem, ELM only requires setting the number of hidden neurons and the activation function. It does not require adjusting the input weights and hidden layer biases during implementation of the algorithm, and it produces only one optimal solution. This enables ELM to approximate complex nonlinear mappings directly from the input samples. Therefore, compared with the other techniques, ELM is the fastest approach in the training process. In addition to being fast, based on Table 2, which presents RMSE, MAE, and NS model coefficient for comparing the performance of the proposed techniques, ELM is the most accurate technique in estimation accident frequency regarding both training and testing processes, followed by MLP, PNN, and RBF. The higher the amount of NS, the more accurate the model is, whereas higher amounts of RMSE and MAE indicate that the model has less predictive ability.

Figure 6 shows the prediction accuracy of ELM for both training and testing process. As shown, ELM represents strong compatibility, which resulted in the lowest values for MAE and RMSE, and the highest for NS. After ELM, MLP showed the best performance in dealing with the available dataset, in which RMSE, MAE, and NS within the training period are equal to 3.707, 2.679, and 0.797, respectively. Figure 7 compares the observed and forecasted accident frequency using MLP for both training and testing periods.

Generally, the output from PNN is probabilistic, which makes interpretation of output easy. However, in the present study, employing PNN led to the fairly accurate prediction of accident frequency. As shown in Table 2, the RMSE, MAE, and NS for the training process in this approach are estimated as 3.838, 2.77, and 0.78, respectively. In addition, PNN actually contains the entire set of training cases and is, therefore, space consuming and slow to execute. Figure 8 illustrates the accuracy of forecasting using PNN. As shown, compared with the two previous methods, there is less compatibility between observed and predicted values.

The last ANN technique that was used in the present study is RBF. Generally, RBF can model any nonlinear function using a single hidden layer, which removes some design decisions concerning numbers of layers. Furthermore, the simple linear transformation in the output layer can be optimized fully using traditional linear modeling techniques, which are fast and do not suffer from problems such as local minima, which plague MLP training techniques. However, according to the results, RBF was not as successful as other ANN techniques. Another drawback associated with RBF is that, although it is quick to train, when training is finished it can be far slower to use than an MLP; therefore, where

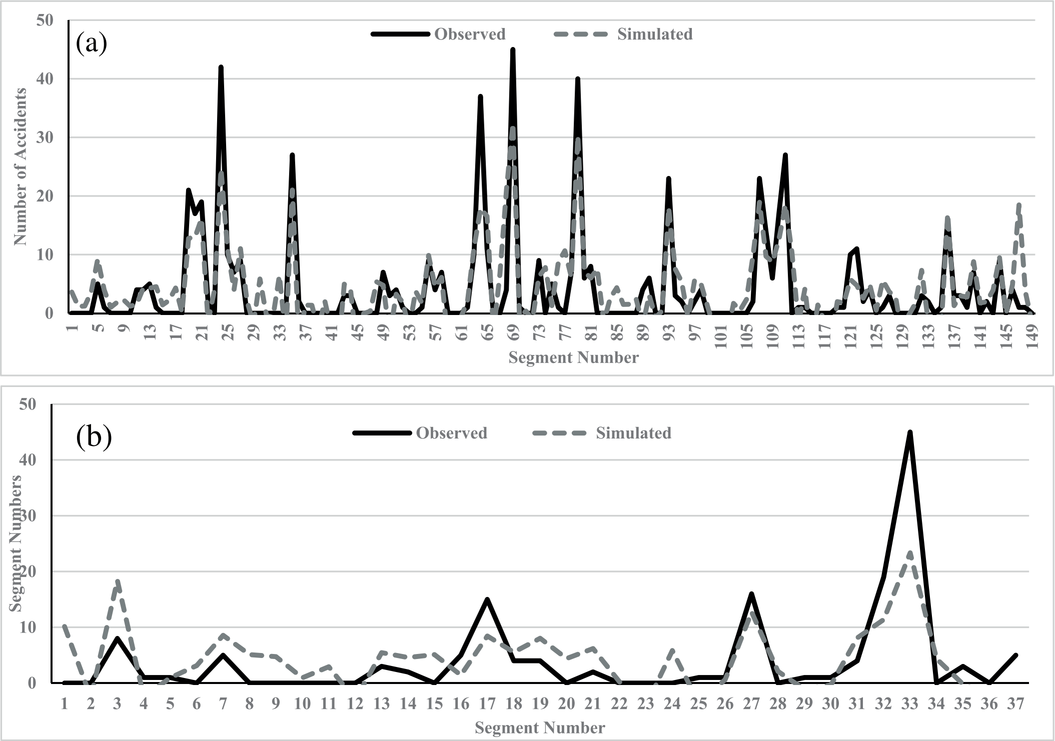

FIGURE 9 Observed and forecasted number of accidents using RBF model: (a) training period; (b) testing period

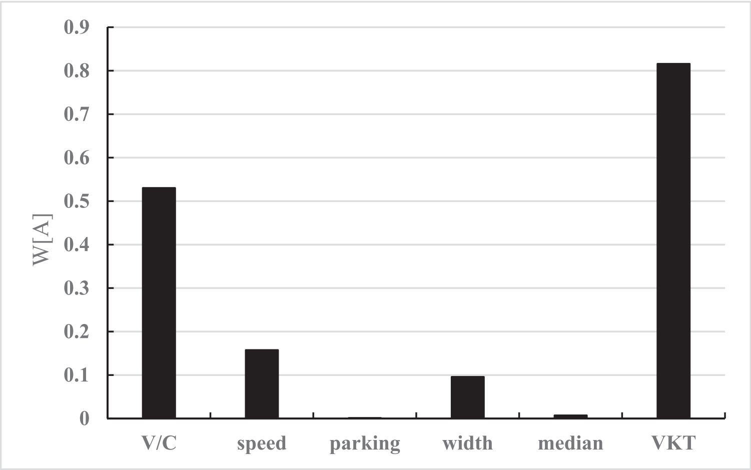

FIGURE 10 Importance of each explanatory variable obtained from RReliefF

speed is a factor, RBF does not seem to be an appropriate approach. As shown in Figure 9, RBF is not an accurate approach in forecasting accident frequency compared to the others.

5.2 | Analysis of parameters

Results of conducting RReliefF for seven variables that were expected to be influential in accident frequency are given in Figure 10. Parked vehicles in the right lane of the roadway cause a reduction of useful roadway width and an increase in traffic interactions when vehicles enter or exit from the parking space. However, it can be observed that this variable is not significant in the model. Moreover, the existence of a median can help to decrease the number of specific types of accidents, such as head-on crashes. Nonetheless, according to the weights shown in Figure 10, this parameter is not significant in the model.

Also, VKT was found to be the most influential explanatory variable reflecting the number and length of trips made. An increase in VKT (higher vehicle volume on the roadway) will increase the frequency of accidents. Indeed, the existence of VKT in the model shows the risk of accident occurrence based on the amount of travel made in a given segment.

To identify the possibilities for improvement and in order to better understand developments in traffic safety, it is necessary to gain a clear understanding of how traffic flow processes influence safety. Accordingly, in the presented paper, V/C and speed were used in the modeling process and found to be significant in accident frequency. This indicates how influential hourly traffic variables such as congestion parameters are on the validity of the results. Shefer and Rietveld (1997) suggested that congestion may cause a positive effect on traffic safety by decreasing the number of fatalities as speeds decrease. Although this statement seems logical when looking at the traffic conditions in more detail, the effects of congestion on safety are less apparent. However, under unstable traffic flow conditions, any small disturbance may result in accident occurrence, especially rear-end crashes, because of large differences in speed.

6 | CONCLUSIONS

One of the main objectives of this study was to predict accident frequency based on the contributing factors. Instead of the more popular traffic characteristics such as ADT or AADT, hourly volume was used. This makes it possible to predict accident severity and type based on prevailing traffic conditions at the time of accident occurrence, which results in a more reasonable modeling process. Furthermore, using V/C and speed together as traffic parameters could be beneficial in identifying the real causes of traffic collisions. This information can be employed in various traffic simulation tools to evaluate the safety performance of all segments of an urban network.

In the present paper, four types of ANN technique including RBF, MLP, PNN, and ELM were developed to predict accident frequency according to the study's limitations and objectives. According to the results, ELM showed the best performance compared with other approaches. ELM selects hidden nodes randomly and computes the output weights of single-hidden layer feedforward neural networks (SLFNs) analytically rather than tuning parameters iteratively. This feature makes ELM able not only to approximate complex nonlinear mappings directly from the input samples but also to provide models for a large class of natural and artificial phenomena that are difficult to handle using classical parametric techniques.

Following ELM, MLP performed more accurately when faced with the available data set. However, MLP neural network models are highly vulnerable to adversarial noise and can make very wrong predictions when fed with such examples as their inputs. Although outputs in PNN are probabilistic (which makes interpretation of output easy) and PNNs are much faster than multilayer perceptron networks, employing PNN led to the fairly accurate prediction of accident frequency. RBF showed the weakest performance in comparison with other ANN techniques, in spite of the ability to bring much more robustness to the prediction, because they are more limited in application compared to other kinds of neural networks such as MLP.

Overall, compared with most of the training techniques developed according to the gradient, which may result in the local minimum problem and slow convergence of parameters, ELM structure and characteristics give this approach the potential to be a more effective predictive analytics method in dealing with crash data. Briefly, using ELM as a feedforward neural network with random weights could be quite beneficial, especially when the amount of labeled data is relatively small.

ORCID

Amir Mohamadian Amiri http://orcid.org/0000-0002-1459-5856

REFERENCES

Abdel-Aty, M., & Pande, A. (2005). Identifying crash propensity using specific traffic speed conditions. *Journal of Safety Research*, *36*(1), 97–108.

Abdel-Aty, M., Uddin, N., & Pande, A. (2005). Split models for predicting multi-vehicle crashes during high-speed and low-speed operating conditions on freeways. *Transportation Research Record*, *1908*, 51–58.

Abdelwahab, H., & Abdel-Aty, M. (2001). Development of artificial neural network models to predict driver injury severity in traffic accidents at signalized intersections. *Transportation Research Record*, *1746*, 6–13.

Akgüngör, A. P., & Doğan, E. (2009). An artificial intelligence approach to traffic accident estimation: Model development and application. *Transport*, *24*(2), 135–142.

Bittermann, M. (2010). Artificial intelligence (AI) versus computational intelligence (CI) for treatment of complexity in design. In J. S. Gero (Ed.), *Design computing and cognition: DCC10*. Berlin, Germany: Springer.

Chang, L. Y. (2005). Analysis of freeway accident frequencies: Negative binomial regression versus artificial neural network. *Safety Science*, *43*(8), 541–557.

Chiou, Y. C. (2006). An artificial neural network-based expert system for the appraisal of two-car crash accidents. *Accident Analysis and Prevention*, *38*(4), 777–785.

Elvik, R., Christensen, P., & Amundsen, A. (2004). *Speed and road accidents: An evaluation of the Power Model* (TØI Report 740. Oslo, Norway: Institute of Transport Economics.

Gu, Y., Shi, Y., & Wang, J. (2012). Efficient intrusion detection based on multiple neural network classifiers with improved genetic algorithm. *Journal of Software*, *7*(7), 1641–1648. (2012)

Hadayeghi, A., Shalaby, A. S., & Persaud, B. N. (2003). Macro-level accident prediction models for evaluating the safety of urban transportation systems. Presented at the 82nd Annual Meeting of the Transportation Research Board, TRB, Washington, DC.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, *70*(1), 489–501. (2006)

Imaninasab, R., Sekhavati, E., & Hajihoseinloo, M. (2016). Modeling crash frequency of heavy vehicles in rural freeways. *Journal of Traffic and Logistics Engineering*, *4*(2), 98–102.

Ivancevic, V. G., & Ivancevic, T. T. (2007). *Computational mind: A complex dynamics perspective*. Berlin, Germany: Springer.

Karayiannis, N. B., & Venetsanopoulos, A. N. (1993). *ANNs: Learning algorithms. Performance Evaluation and Applications*. Boston, MA: Kluwer.

Lord, D. (2000). The prediction of accidents on digital networks: Characteristics and issues related to the application of accident prediction models. PhD dissertation, Department of Civil Engineering, University of Toronto (2000).

Martin, J. L. (2002). The relationship between crash rate and hourly traffic flow on inter-urban motorways. *Accident Analysis and Prevention*, *34*(5), 619–629.

Mensah, A., & Hauer, E. (1998). Two problems of averaging arising in the estimation of the relationship between accidents and traffic flow. *Transportation Research Record*, *1635*, 37–43.

Mittal, S. (2016). A survey of techniques for approximate computing. *ACM Computing Surveys*, *48*(4). Art. No. 62 (2016)

Mussone, L., Ferrari, A., & Oneta, M. (1999). An analysis of urban collisions using an artificial intelligence model. *Accident Analysis and Prevention*, *31*(6), 705–718. (1999)

Mussone, L., Rinelli, S., & Reitani, G. (1996). Estimating the accident probability of a vehicular flow by means of an artificial neural network. *Environment and Planning. B, Planning & Design*, *23*(6), 667–675.

Nassiri, H., Najaf, P., & Amiri, A. M. (2014). Prediction of roadway accident frequencies: Count regressions versus machine learning models. *Scientia Iranica Transaction A, Civil Engineering*, *21*(2), 263–275.

Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, *53*(1–2), 23–69. (2003)

Sajjadi, S., Shamshirband, S., Alizamir, M., Yee, L., Mansor, Z., Manaf, A. A., … Mostafaeipour, A. (2016). Extreme learning machine for prediction of heat load in district heating systems. *Energy and Buildings*, *122*, 222–227.

Sameen, M. I., & Pradhan, B. (2017). Severity prediction of traffic accidents with recurrent neural networks. *Applied Sciences*, *7*(6), 476.

Shefer, D., & Rietveld, P. (1997). Congestion and safety on highways: towards an analytical model. *Urban Studies*, *34*(4), 679–692.

Xie, Z., & Klionsky, D. J. (2007). Autophagosome formation: Core machinery and adaptations. *Nature Cell Biology*, *9*(10), 1102–1109.

Zhu, W., Miao, J., Hu, J., & Qing, L. (2014). Vehicle detection in driving simulation using extreme learning machine. *Neurocomputing*, *128*, 160–165.

Hamid Behbahani received the B.Sc. degree in civil engineering from Iran University of Science and Technology, and the M.Sc. and P.hD. degree in road and transportation engineering from University of Florida, USA. He is currently working as a professor in Iran University of Science and Technology, Tehran, Iran. His main research interests include traffic safety and pavement design.

Amir Mohamadian Amiri received the M.Sc. degree in highway and transportation engineering from Sharif University of Science and Technology, Tehran, Iran. Currently, he is PhD student in Iran University of Science and Technology. His main research interests are traffic safety, traffic flow, intelligent optimization, data mining, and discrete choice.

Reza Imaninasab was born in Bandarabbas, Iran, in 1988. In 2013, he received his MS degree in highway and transportation engineering from Iran University of Science and Technology, Tehran, Iran. Currently, he is PhD student in Purdue university. His main research interests are traffic safety, pavement materials characterization and evaluation, and pavement work planning optimization.

Meysam Alizamir received the B.Sc. degree from the Mohaghegh Ardebili University, Ardebil, Iran, in 2010, and the M.Sc. degree from the K. N. Toosi University of Technology, Tehran, Iran, in 2013. He is currently working as a research assistant at the department of civil engineering, Hamedan branch of Islamic Azad University, Hamedan, Iran. His main research interests include machine learning, intelligent optimization, data mining, multiobjective optimization, computational intelligence and estimation algorithms.

How to cite this article: Behbahani H, Amiri AM, Imaninasab R, Alizamir M. Forecasting accident frequency of an urban road network: A comparison of four artificial neural network techniques. *Journal of Forecasting*. 2018;37:767–780. https://doi.org/10.1002/for.2542