Received: 30 October 2019 | Revised: 24 March 2020 | Accepted: 1 April 2020

DOI: 10.1002/eng2.12175

RESEARCH ARTICLE

WILEY

Application of deep learning techniques in predicting motorcycle crash severity

Mahdi Rezapour | **Sahima Nazneen** | **Khaled Ksaibati**

Wyoming Technology Transfer Center, University of Wyoming, Laramie, Wyoming

Correspondence
Mahdi Rezapour, Wyoming Technology Transfer Center, University of Wyoming, 1000 E. University Avenue, Department 3295, Laramie, WY 82071.
Email: rezapour2088@yahoo.com

Funding information
Wyoming Technology Transfer Center

Abstract

Machine learning (ML) techniques play a crucial role in today's modern world. Over the last years, road traffic safety is one of the applications where ML-methods have been successfully employed to prevent road users from being killed or seriously injured. A reliable data-driven predictive model is essential for this purpose. This could be achieved by successfully applying an intelligent transportation system to identify a driver at a higher risk of crashes. This study investigates the capabilities of different deep learning techniques to predict motorcycle crash severity. This study is based on 2,430 motorcycle crashes in a mountainous area in the United States over a 10-year period. Different deep networks (DNNs), including deep belief network, standard recurrent neural network (RNN), multilayer neural network, and single-layer neural network, were considered and compared in terms of prediction accuracy of motorcycle crash severity. Before conducting any analysis, feature reduction was performed to identify the optimal number of variables to include in the models by minimizing the error rate. Different metrics including the area under the curve and confusion matrix were used to compare the different models. Although the analyses were conducted on a relatively small dataset, the results indicate that almost all the DNN models better perform in predicting the severity of motorcycle crashes, compared with the single layer neural network. Finally, the RNN outperforms the other three neural network models. A comprehensive discussion has been made about the methodological approach implemented in this study.

KEYWORDS

deep belief network, machine learning, motorcycle crashes, multilayer neural networks, recurrent neural network

1 | INTRODUCTION

Motorcyclists are vulnerable road users because of their high death rate relative to other vehicle users in traffic crashes.[1] An absence of protective structural or the advance safety restraints, unlike other passenger vehicles, makes it more susceptible to severe crashes.[2] Identifying a predictive model to evaluate the relationship between motor cycle injury severity outcomes and the factors associated with this type of crashes are essential in addressing the motorcycle safety issues.

According to National Highway Traffic Safety Administration (NHTSA), motorcyclist fatalities are 27 times more frequent than passenger car occupant fatalities in traffic crashes.[3] Governors Highway Safety Association (GHSA) reported that in 2014, motorcyclist fatalities account for 14% of total traffic fatalities while in 1995, motorcyclist fatalities were 5% of all traffic fatalities.[4] Statistics also revealed that the number of motorcyclist fatalities increased by 5.1%, from 5,029 in 2015 to 5,286 in 2016.[5] The current elevated motorcyclist fatalities echo the necessity to find solutions to decrease this rate. In this regard, predicting the crash injury outcomes based on contributing factors has become one of the main areas of focus among traffic safety experts.

The number of motorcycle related fatalities in Wyoming reflects the elevated motorcyclist fatality rate among all vehicle road users. In 2017, motorcyclist fatalities accounted for 16.19% of all fatal crashes in the state, while involvement of motorcyclist was only 1.1% of total crashes.[6] Along with other factors, Mountainous topography and adverse winter weather condition may increase the risk of motorcycle fatal crashes in Wyoming. Identifying a reliable model that can predict the severity of crashes is crucial to reduce the risk of crashes.

In addition to the conventional statistical methods, machine learning algorithms have shown promising performance in crash severity prediction (eg, Reference 7). This study contributes to the existing motorcycle crash severity literature by evaluating various deep learning (DL) algorithms to predict motor cycle involved crash severities, and identification a best performed model based on DL algorithms.

In this study, crash severity injury prediction was performed for the Wyoming motorcycle crashes for 10 consecutive years from 2006 to 2016. Crash severity was classified as property damage only (PDO), or injury and severe injury/fatal crashes. This study developed motorcyclist crash injury severity prediction model based on recurrent neural network (RNN), single layer neural networks, multilayer neural networks (MLNNs), and deep belief network (DBN).

1.1 | Study contribution

With the development of the economy, the number of motorists on highways increase dramatically. With an increase in the number of motorists, for instance, more enforcement and infrastructure are needed to keep a safe network for road users. However along with the development of road network, it is not plausible to expand the road network and enforcement resources proportional to the development of the road network. Automated systems like artificial intelligence can act as a tool to lead rider to make a better decision and help enforcement in a more efficient way. This approach does not need much expenditure, and it is more practical to be used. For instance, consider an autonomous drivers which analyze the driving condition, speed of travel, and based on a trained model decide if any action would be hazardous for drivers, and consequently take an appropriate action. Or a scenario where automated enforcement observing the drivers and based on observed circumstances, decide if a driver needs to be flagged or alarmed.

Due to high severity of motorcycle crashes in the US and especially in Wyoming, this state is moving toward automated network so in addition of vehicles being connected (connected vehicles), policy makers would be connected to different vehicles/riders. Thus in an automated network crash severity would be minimized though identifying a motorists being more likely to be involved in a crash.

Although few studies have been conducted on the application of different DL techniques on traffic safety studies, the main contributions of this study are few folds. First, the first contribution lies in the uniqueness of the domain of this study which used dataset of crashes belonging to Wyoming with highest traffic fatality in the states.[8] It should be noted that also motorcycle crashes account for the highest rate of fatalities in this state.

Second although this study did not deal with big dataset due to low traffic in Wyoming, the results indicated that DL techniques performed better that a single layer neural networks. Moreover, although standard RNN, which could only deal with sequential dataset, the data in this study was convert into the format of nonsequential dataset to be used by this machine learning technique.

1.2 | Objectives

The main objective of this paper was to employ DL algorithms in predicting severity outcomes of motorcycle-related crashes. This study developed crash severity model based on RNN, and three neural-based models: single layer perceptron, multilayer perceptron, and deep architecture. The results were then compared to identify the models with the best prediction accuracy.

2 | BACKGROUND

Considerable research studies in the literature have provided valuable resources and information to assist in solving motorcyclists' injuries and deaths caused by crashes. Rider variables including age, gender, impairment, helmet use, as well as temporal aspects such as season, day of the week, time of day, and geometric variables have consistently been identified in the literature as factors impacting the likelihood of crash severity. Various modeling approaches including parametric and nonparametric models have been employed throughout the literature to analyze crash severity by assessing injury-severity levels associated with the aforementioned factors.[2] Traffic, speed variation,[9] nationality, engine capacity,[10] motorcycle ownership,[11] motorist involved in approach turn collisions at signalized junction[12] are some of the factors being found to impact the severity of motorcycle crashes in the literature review.

In addition to traditional approaches, many researches are currently learning towards using machine learning and especially DL in predicting crash injury severity due to their higher prediction accuracy.

Machine learning algorithm has been used extensively in transportation safety studies. For instance, Random Forest Model is a promising data mining approach employed in many studies to prioritize the variables associated with crashes.[6,13,14] Researchers also found the better prediction performance of the support vector machine (SVM) in developing crash injury severity models compared to other parametric models.[15–17] Moreover, despite limited application in transportation sector, multivariate adaptive regression splines technique has outstanding predictive power in crash injury analysis.[18–19]

DL has recently become the new focus in transportation safety analysis. Being a branch of artificial intelligence, it models sophisticated data through a series of processing layers.[2] Several studies have applied DL on other transportation engineering aspects such as traffic data imputation, short-term traffic flow prediction, vehicle classification, and sustainable guideline development. The following paragraphs provide a brief review of studies conducted on using DL techniques in transportation engineering.

Das et al employed a DL technique to analyze 5 years (2010-2014) of Louisiana motorcycle at fault crashes. They developed a DL framework, named Deepscooter using R by H_2O platform to predict motorcycle involved crash fatalities. This model showed higher prediction accuracy than statistical method and machine learning algorithms with an accuracy of 100% for training data, and 94% for test data. This study also found that rider ejection, two-way road with no physical separation, single vehicle, curve aligned roadways, weekend, and young drivers are associated with higher likelihood of fatal crashes.[2]

Sameen et al analyzed various DL techniques in predicting the injury severities occurring on Malaysian highways.[20] This study proposed three network architectures based on a simple feedforward neural networks (NN), RNN, and convolutional neural network (CNN). These networks were optimized by a grid search optimization to calibrate the best hyperparameters of the models to predict the outputs with less computational costs. The results showed that among the tested algorithms, the RNN model with an average accuracy of 73.76% outperform the NN model (68.79%) and the CNN (70.30%) model based on a 10-fold cross-validation approach.

Yu et al developed a fine-grained vehicle classification approach which consisted of two parts: vehicle detection and classification model.[21] They employed Faster R-CNN method to extract single vehicle images from an image with clutter background. Then this single vehicle images were used to classify a vehicle by employing CNN with a joint Bayesian network.

Dong et al developed an improved machine learning model to investigate the complex interactions among roadways, traffic, environmental elements, and traffic crashes.[22] Different models were applied in this study including an unsupervised feature learning module to determine functional network between the explanatory variables, the feature representation, and a supervised fine-tuning module along with multivariate negative binomial model to perform traffic crash prediction.

3 | METHODOLOGY

This section first would outline the methodological approach taken in this study. It then outlines a theoretical background of various implemented methods. Then it would go over the data description, and finally it would detail various preprocessing methodological approaches taken in this study.

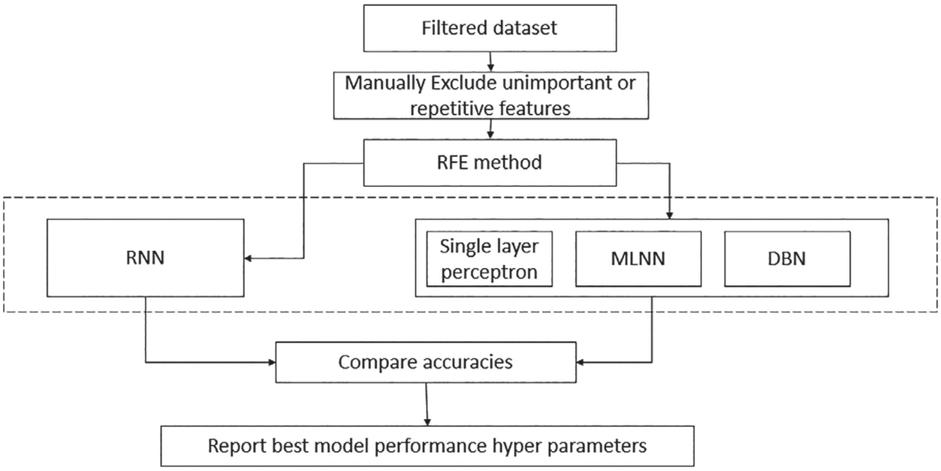

Filtered dataset
Manually Exclude unimportant or repetitive features
RFE method
RNN
Single layer perceptron
MLNN
DBN
Compare accuracies
Report best model performance hyper parameters

FIGURE 1 Methodological steps

3.1 | Methodological approach

The methodological step is presented in Figure 1. First the data were fileted to include a crash that involved at least a single motorcycle crashes. In the next step, unimportant variable such as crash ID were removed from the dataset. Variables that were found to be interrelated across predictors, or correlated with the response were manually excluded as well. These included features such as number of injury, the time took the ambulance to be present at the scene, estimated cost of crashes being correlated with the severity of crashes. For instance, the more severe the crash is, the less time is expected for the services such as an ambulance to be present at the scene of a crash. Also features were removed from the dataset, if there were more than 30% missing values for features, which make it imprecise for data imputation to fill the missing values those.

After manually removing noisy features from the dataset, the dataset reduced from more than 300 features to about 100 features. Still, not all the features could be incorporated in the model and the data need to be screened.

Thus, Recursive Feature Elimination (RFE) was implemented to discard unimportant/noisy features. A more description about this method and various machine learning algorithms would be presented in next few sections. Three methods that are based on neural network: single layer perceptron or MLNN, multilayer neural network, and DBN were considered in this study. On the other hand, recurrent (feedback) neural network (RNN) as a class of artificial neural network was considered as well. The model would be compared in terms of accuracy, and then a best fit model hyperparameters would be reported. The RNN can be viewed as a neural network in simple words. While doing prediction, the nodes, cells, need to be informed about the previous input before doing prediction. Thus RNN can be viewed as multiple feedforward neural networks, where the information would be passed from one to the others. In summary, all the included models could come under deep network.

3.2 | Neural network models

Three neural networks: single layer neural networks, MLNN, and DBN would be discussed. For simplicity, a naïve general explanation of neural networks follow the following steps:

- Take the input.
- Apply activation functions and initialize variables' weights and biases.
- Predict the output.
- Compare the predicted output with the real outputs.
- Calculate the error terms, and back propagate the error back through the same path and adjust the weights.
- Repeat the above steps.

Continue until you get close to the real output.

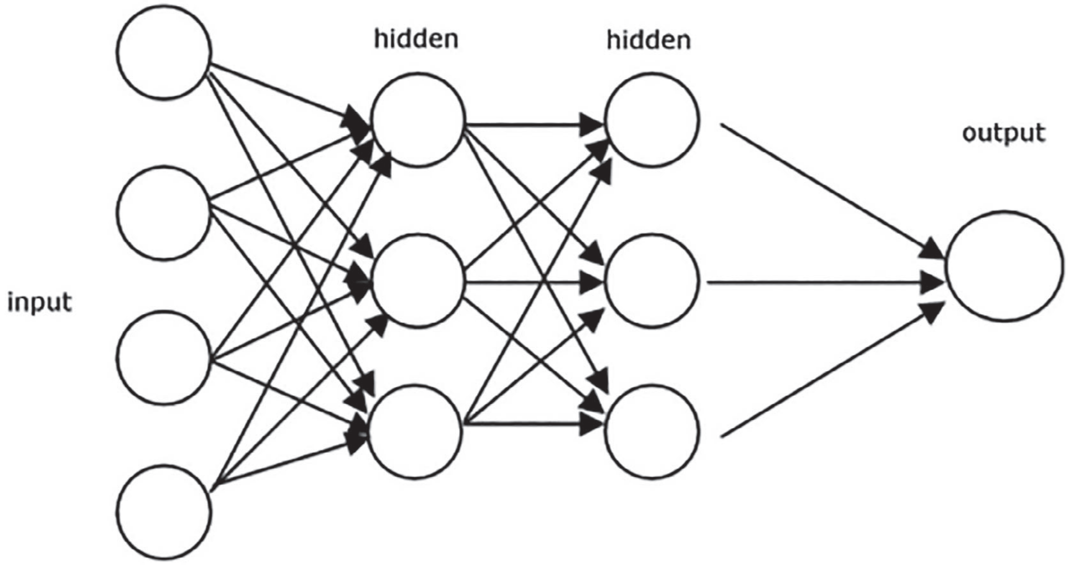


FIGURE 2 Feed forward neural network architecture

3.3 | Single layer perceptron

In this study, the model represents a nonlinear mapping between the input values (crash predictors) and the output parameters (injury severity levels). Neurons are systematic connection of weight vectors, which are usually structured in layers with full connections between successive layers.

In machine learning a simple NN model is an interconnection of neurons and nodes, consisting of three layers namely: inputs, hidden, and output layers. Feed forward neural network is one of the first artificial neural networks being invented. In this network, information only travels forward though the input, hidden, and output layers. This network is named as feedforward because there are no feedback connections in which output of the model are fed back into itself.[20–23]

Neural network was developed from single neuron named perceptron. It categorizes a set of input into one of two classes. The perceptron returns 1 if the weighted sum of the inputs exceeds a threshold, and 0 otherwise. A single layer neural networks is a feed forward network based on a threshold transfer function. This model is a simplest form of the artificial neural network and can only classify linearly separable cases with a binary target (1, 0). In this model, first it assigns weight to the inputs randomly. Then the weights would be optimized through backpropagation, and with the objective of minimizing the error in the model.

3.4 | Multilayer neural network

Combination of many hidden layers is named as MLNN or feedforward neural networks (see Figure 2).

This model has the same structure of a single layer neural networks with more hidden layers. This network uses backpropagation algorithm which consists of two phases: The forward phase where the activations are propagated from the input to the output layer, and the backward phase where the error between the actual and the predicted values in the output layer are propagated backwards in order to modify the weights and bias values. In other words, backpropagation was geared to decrease the error function by using an iterative approach as shown in Equation ((1)).[20]

$$E = \frac{1}{2} \sum_{i=1}^{L} (d_j - o_j^M)^2 \tag{1}$$

where d_j and $o_j{}^M$ denote output and current response, respectively, of the node "j" in the output layer, and "L" represents the number of nodes in the output layer. This algorithm uses the partial derivative of the error term E with respect to the weights of the network. This derivative is called the gradient descent which determines the variation in the error for a weight change. Weight keeps changing until largest error reductions is achieved. In a nutshell, corrections are made to weight parameters and added to the previous values as shown in Equation ((2)):

$$\Delta\ w_{i,j} = -\mu \frac{\partial E}{\partial w_{i,j}}$$
$$\Delta\ w_{i,j}(t+1) = \Delta\ w_{i,j} + \alpha\ \Delta\ w_{i,j}(t) \tag{2}$$

where Δ is the learning rate which controls the amount of adjustment, α is a momentum factor between 0 and 1 and "t" indicates the number of iterations. The parameter α is called smoothing factor as it adjusts the rapid changes between the weights.[21]

3.5 | Deep belief network

DBN consists of several layers of the restricted Boltzmann machine (RBM). The RBM is a stochastic artificial neural network that could learn a probability distribution over the sets of input. Supervised and unsupervised learning could be implemented for feature training, feature reduction, or topic modeling. The model is restricted as the neurons must form a bipartite graph. The RBM models are shallow constituting the structure of the DBN. In DBN, after the nodes construct the output, the model tried to reconstruct the input. After the model learnt the structure of the input data for the first hidden layer, the data would be passed to one layer down in the network. The steps for this method could be depicted as follows:

1. Train the first layer of RBM, modeling the raw input as its visible layer.
2. Now, the first layer would be used to obtain a representation of the input for the second layer.
3. Now, train the second layer like 1.
4. Iterate 2 and 3 for assigned number of layers.
5. Fine tune all the hyperparameters

In summary, DBN, as a probabilistic model, can be viewed as a network formed by training RBMs once at a time and stacking them. The Darch framework is based on the code written by Hinton and Salakhutdinov,[24] and is available in the MATLAB environment. This method provides a preformation with the contrasting divergence method, and fine-tuning with common training algorithms known as backpropagation or conjugated gradients. In addition, fine-tuning supervision can be improved with maxout and dropout, two recently developed techniques to improve fine-tuning for DL.

Different hyperparameter could be adjusted in this model including batch size: which is the number of training samples that are presented to the network before weight update is performed, number of epoch: which is number of epochs of fine tuning, for example, length, or number of line search, and whether the data needs to be centered and scaled. This method is a class of neural network consisting of latent variables with no connections between the layers units but between layers. The model is similar to Restricted Bolzman Machine (RBM), where the nodes in each layers are connected to all previous layer's nodes. Learning occurs on a layer-by-layer basis. In other words, each layer receives a version of dataset: layers use the output from the previous layers.

3.6 | Recurrent neural networks

RNNs are networks with feedback connections. They are more powerful and rational than feedforward networks. RNN can identify model sequence by determining association between hidden layers that transmit through time. The feedback connections provide memory of past activations to the RNN, which enables it to learn the temporal dynamics of sequential data.[23]

Recurrent networks consider both current input and the output which was calculated previously to make a decision. The decision made at an instant also influences the decision that will be made immediately afterwards. In other words, recurrent networks combine two input sources to determine the response to new data.[23] Figure 3 reflects the RNN architecture.

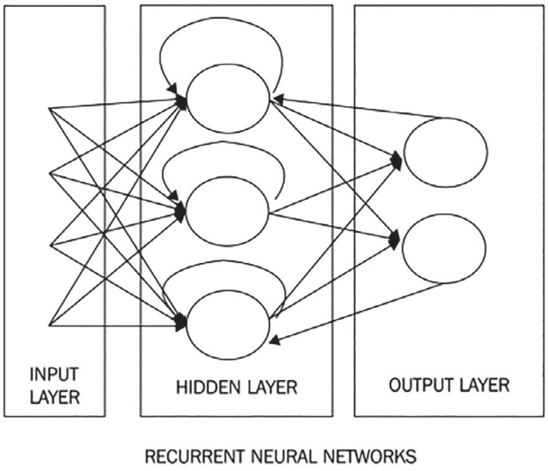


FIGURE 3 Recurrent neural networks architecture

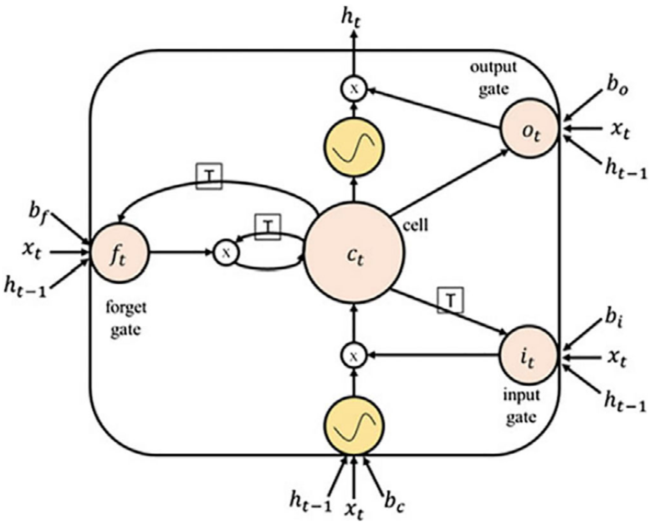


FIGURE 4 The structure of a memory cell in LSTM-RNN

Recurrent networks are geared to recognize patterns as a sequence of data and are useful in prediction. However, the standard or Vanilla RNNs could result in a problem named vanishing gradient or exploding gradient. Due to the short-term memory, all the previous information cannot be restored. Long Short-Term Memory (LSTM), introduced by Hochreiter and Schmidhuber in 1997, which could solve the vanishing gradient problem by incorporating memory units that allow networks to learn when to forget earlier inputs and when to update input given new information.[23]

LSTM consists of memory blocks that contain self-connected memory cells and three gates (input, output, and forget gates) instead of hidden layers. Each gate within a block might use different activation functions, for instance, to control whether they are triggered or not. The gates facilitate reading, writing, and resetting operations in the memory blocks. They act as switches and control the behavior of the memory blocks. Forget gate decides what information need to discard from the unit. The Input gate selects the value from the input to update the memory state whereas the output gate determines the output based on input and memory unit. Figure 4 depicts a diagram representing a single LSTM unit. Let c_t be the sum of inputs at time step t and its previous time step activations, the LSTM updates for time step i given inputs x_t, h_{t-1}, and c_{t-1} are.[25]

$$i_t = \sigma(W_{xi}.x_t + W_{hi}.h_{t-1} + W_{ci}.\mathrm{c}_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{xf}.x_t + W_{hf}.h_{t-1} + W_{cf}.c_{t-1} + b_f) \tag{4}$$

$$c_t = i_t.\tanh(W_{xc}.x_t + W_{hc}.h_{t-1} + b_c) + f_t.c_{t-1}\ ) \tag{5}$$

The main input of the network is a set of motorcycle crash-related factors, and the output is the corresponding crash severity class (ie, PDO or injury/fatality crashes). The output of each node in the LSTM layer, for instance, might result from different activation function such as the rectified linear unit (RelU) applied to a weighted sum of both inputs from the previous layer and the previous outputs of the layer.

The gates recurrent unit (GRU) is very similar to LSTM in terms of tracking long term dependencies while mitigating the exploding gradient problems. However, LSTM unit has separate input and target gate, while the GRU performs both together through reset gate. In this study standard (Vanilla) RNN only tried.

While all the included models accept the format of the datasets as they are, for RNN data need to be prepared as this technique only accepts sequential datasets. In order to transform the data into a format accepted by this technique, a function int2bin from the package (rnn) was used. This function works similarly as "to_categorical" function in Tensorflow package with this difference that the dataset would be trimmed not to exceed eight columns per predictors.

For instance, to_categorical function converts average annual daily traffic (AADT) column, into the dataset to more than 100 columns as this variable is continuous with 100 various values. The differences in the column numbers after conversion make it difficult to have the same time sequence for different included predictors. Thus, this function helps to have a dataset with sequence of eight for all the predictors. After converting all the predictors to binary values, all the predictors are merged to be used for the model.

As the data need to change its format to have a three-dimensional scale, scale/centering is not practical for this method, and the process itself would account for that. Different hyperparameters can be set for this model such as learning rate, which would be applied for weight iteration, dimensions of hidden layers, batch size, which is the number of samples used at each weight iteration, and number of iteration that is the number of time that the whole dataset is presented to the network. In this study, only standard type of RNN model was considered.

3.7 | Data

Crash data were obtained from the Wyoming Department of Transportation (WYDOT) using Critical Analysis Reporting Environment (CARE) package. There were 2430 motorcycle crashes between 2006 and 2016 in Wyoming. The crash severity defined by KABCO include fatal (K), incapacitating injury (A), nonincapacitating injury (B), possible injury (C), and PDO.[26] In this study, based on the nature of motorcycle crashes and the performance of the model, two categories were used: K and A as one category, and B, C, and O in another category.[27] The data include highway and interstate motorcycle crashes.

As it was not practical to include all the predictors, and even the important predictors, the following lines would highlight few important characteristics of some predictors. The AADT of the included observation was about 6504. Out of 2430 motorcycle crashes, 828 (34%) accounts for severe/fatal crashes, while 1602 crashes were PDO. The majority of motorcycle crashes occurred on two lane highway system. On average, 19 motorcycle fatal crashes in a year were observed. Due to geometric characteristics of the state and severe winter conditions, motorist use the roadway in warm weather condition. As expected, a significant of motorists were Wyoming residence, 50%, and about 25% of crashes occurred in urban area. More than 50% of motorists hit a fixed objects after involvement in a crash.

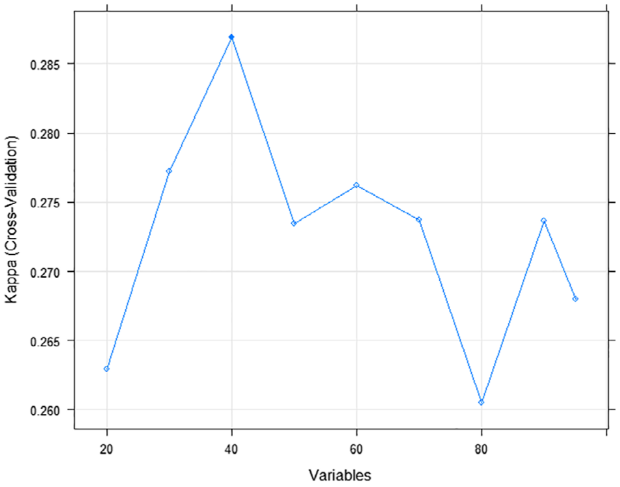
The study proposed four different architectures including a single layer neural networks, Different deep network (DNN), DBN, and standard RNN. Before conducting any analyses, a few steps were taken as described in the next few paragraphs.

3.8 | Data preprocessing

Before applying any machine learning techniques, the data need to be preprocessed. The first step was involved with deleting erroneous and repeated features/columns.[34] As there were limited number of motorcycle crashes, no incomplete observation was removed from the dataset unless the missing exceeds 30% of the whole onservations. Instead missing values were imputed using proximity from random forest. For continuous predictors, the imputed values were based on values of weighted average of nonmissing observations for continuous variables, and for categorical predictors, the imputed values was based on the largest average proximity.[35]

As the dimensions of each predictors were different especially for continuous variables, features had to be normalized. If the normalizing for machine learning disregarded, the algorithms would assign a higher value for higher values such as traffic. As for normalizing the data, this task was conducted inside each machine learning techniques. As discussed, the normalizing was not performed for RNN as the input for this machine was categorical. As can be expected, the test

FIGURE 5 RFE results of optimal number of predictors to be included in the models



data were isolated away from the training dataset for the different analyses. After imputing the data, the next conducted method was feature selection to identify the important feature which would be explained in the next section.

3.9 | Feature selection

Feature selection is crucial to have most important and relevant features in the model. Among various methods, RFE algorithm or a simple backward selection was used in this study. This algorithm aims to select the best performing feature subset. It fits a model and eliminates the weakest features repeatedly until the specified number of features is reached. It then ranks the features based on their importance. RFE attempts to remove dependencies and collinearity by recursively eliminating a small number of features per loop.

RFE requires to have a number of features to be kept in the model as an input. However it is often not known in advance how many features are valid. Kappa fold (10-fold) cross-validation is used with RFE to find the optimal number of features. This method scores different feature subsets and select the best scoring collection of features. Finally, the number of features in the model along with their cross-validated test score and variability are plotted and the optimum number of features are selected. As can be seen from Figure 5, optimum number of predictors for the model is 40. So, 40 predictors were selected as an optimal number of predictors for the models across all the machine learning techniques.

Initially about 100 predictors were incorporated in the dataset. Features such as roadway elevation, degree of curve, radius were excluded during RFE process as they found not be important to be included in the model.

The predictors include various binary and continuous variables: binary predictors such as driver under influence, whether a motorist hit a guardrail, time of a crash, road conditions at the time of crashes, and whether a crash occurred on roadway or off roadway. The continuous variables included 10 predictors such as AADT, and average annual truck traffic. Similar and multicollinear predictors along with irrelevant predictors were excluded from the model before running RFE technique.

Many predictors were also removed as they were found not to be helpful in prediction quality such as radios, road classification, rural vs urban, and lighting conditions of the roadways.

Included predictors were considered as features in various machine learning techniques, Binary categories of crash severity as response is considered and it is modeled based on different included features.

3.10 | Splitting data

The dataset was divided into two subsets: 80% of the data was randomly selected and used for training, while 20% of the data was randomly selected for testing. This criterion was chosen to ensure having adequate observations for test data and proper number of observations for training dataset, along with adequate observations for each response category. The DL algorithms consider training data to develop the learning framework. Then the performance of the model would be evaluated on test dataset.

TABLE 1 Confusion matrices containing actual and predicted crash severity along with error rates for the model-based (training data) and for cross-validation (testing data), multilayer perceptron and single layer perceptron

Multilayer perceptron						**Single layer perceptron**				
	Actual	**Predicted**		**Total error rate**	**AUC**	**Actual**	**Predicted**		**Total error rate**	**AUC**
		0	1				0	1		
Model-based	0	1010	269	35%	0.57	0	1274	19	32%	0.52
	1	413	252			1	164	13		
Cross-validation	0	250	73	37%	0.58		301	8	35%	0.53
						0				
	1	109	54			1	164	13		

TABLE 2 Confusion matrices containing actual and predicted crash severity along with error rates for the model-based (training data) and for cross-validation (testing data), recurrent neural network and deep belief network

Recurrent neural network						**Deep belief network**				
	Actual	**Predicted**		**Total error rate**	**AUC**	**Actual**	**Predicted**		**Total error rate**	**AUC**
		0	1				0	1		
Model-based	0	1094	246	22%	0.65	0	1245	39	31%	0.63
	1	185	419			1	568	92		
Cross-validation	0	266	85	29%	0.74	0	257	61	29%	0.68
	1	57	78			1	77	91		

3.11 | Model performance evaluation

Two methods were used to evaluate the performance of different included models: confusion matrix, and area under the curve (AUC). Confusion matrix provides detailed information about how the data performed by different models. This method would provide information for the number of response categories.

AUC also was used to provide a greater understanding of a model performance. Although confusion matrix provides a detailed performance of the model based on prediction for each category, sometimes AUC is favored as it presents a comparison based on a single value. AUC calculates area under the receiver operating curve (ROC). ROC is created by plotting a true positive rate, against the false positive rate. AUC ranges in value from 0 (100% wrong), vs 1 (100% right).

4 | RESULTS

Tables 1 and 2 summarize the results of the methods used in this study. Table 1 summarizes the result of the MLNN and single layer perceptron in terms of error rate and AUC, while Table 2 depicts the results for standard RNN and DBN. The following sections outline the performance of various models.

4.1 | Single layer perceptron

This method was included as the baseline for comparison especially across neural-based methods. NNet package in R was used to fit a feed-forward neural networks with a single-hidden-layer neural network.[36] Different hyperparameter could be adjusted to have a best tuned model. These parameters include size, which is a number of units in the hidden layer, and the maximum allowable number of weights. As expected, these hyperparameters were found to impact the accuracy of this model significantly.

The findings indicated that 30 as a number of hidden layers and weight of 5000 would result in an optimal performance. The weights of a neural network would be optimized during each step. The step is called learning rate. This value defines how slowly or quickly the neural network model learns. The step would be conducted during backpropagation. A very low learning rate would impact the speed of the model learning significantly. While large learning rate would increase the speed of training dramatically, preventing a model from reaching an optimal solution by passing this value. In this study due to having a low number of observation/crashes this value was set as 2e-16. The results indicated that the model perform with error rate of 32% and 35% for training and test dataset, respectively.

4.2 | Multilayer neural networks

Different hyperparameters were adjusted for this model. Number of output nodes/units was set as two as the number of categories for the response was binary. Number of unites in the input layer was set as 30, where the number of included predictors was 40. Batch size controls the number of training samples before model parameter being updated. The value could range between one and number of observation/crashes. Based on the accuracy results, this value was set as 150 resulting in a best performance.

In order for this model to be able to make sense of the input data and perform well, the input data need to be converted to a nonlinear format that could be used by a model. Different activation functions could be used based on the ranges of input and type of response. Softmax activation function was used for this study as it could normalize the input vector into a probability distribution between 0 and 1. Based on the discussed adjustment this model resulted in error rate of 35% and 37% for training and test dataset, respectively.

4.3 | Deep belief networks

As discussed the reason why this method was chosen in this study was this model is similar to MLNN with the difference that the MLNN has been improved by converting a high dimensional data to low dimensional codes through training of MLNN with a small central layer.[24]

For the included model, number of epochs was set at 1000 with batch size being adjusted to be as default, 32, and the data were scaled and centered. Scaling was based on min-max normalization to rescale features between 0 and 1. Number of epochs define the number of times that the algorithm works through the whole training dataset. For this study number of epoch equal to 1000 resulted in most optimum outcomes meaning that the training dataset has the opportunity to update its hyperparameter. It was specified in the model that the model is binary.

Although DBN and RNN both perform similarly, in terms of prediction accuracy, RNN performs slightly better than DBN, with 31% and 29% of error rates for model base and cross-validation data.

4.4 | Recurrent neural network

From Table 2, it can be concluded that the RNN outperformed the other three models based on the AUC values. This model used stochastic gradient descent as a rule to update the weight. This is an iterative method used for optimizing/minimizing an objective function (loss function). RNN can predict crash severity with an error rate of 22% for training dataset and 29% for test data, which resulted in a best performance compared with the other models. As this model performs better the results of hyperparameter tuning would be presented in the following sections.

5 | RNN MODEL PERFORMANCE

Different hyperparameters would impact the performance of the different machine/DL techniques significantly, so it is important to tune the models' parameters before reporting results. While tuning was performed on all the included models, as RNN model performs better, the tuning results would be presented only for this model only.

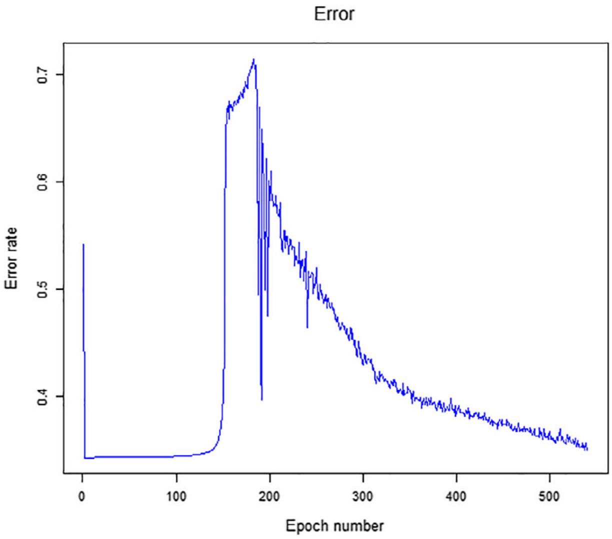


FIGURE 6 Impact of number of epoch on error rate

An initial model was developed by using RNN. The training was run on a batch size 100 with 500 epochs. The best learning rate was 0.01. In addition, the dimension of the hidden layers was 20. The following sections will discuss the impact of the aforementioned hyperparameters in the model performance.

5.1 | Epoch number

Figure 6 shows the accuracy performance for 500 epochs (iterations) using the datasets. The accuracy is fluctuating because of the use of the dropout technique in the model, which results in different error rate during every iteration. The dropout technique controls over-fitting and presents randomness to the network. In the first iteration, the accuracy was 55%. As the model trains during the first pass through the data, error rate declines indicating that the model is learning the structure of the traffic crash data, and possibly its temporal correlations. The higher the epoch number, up to optimum values, the lower the error rate.

5.2 | Sensitivity analysis of learning rate

Different values of learning rate could change the accuracy of a model drastically. For this hyperparameter different learning rates were tried (see Figure 7), and the learning rate that resulted in a lowest error rate of a model was chosen as an optimum value for learning rate. As can be seen from Figure 7, while the learning rate did not impact the error rate of test dataset significantly, even the slight changes in the value of learning rate would impact the results of training data error rate significantly. The highest validation accuracy was achieved when a learning rate of 0.01 was chosen as it resulted in a lowest error rate for test dataset.

5.3 | Number of hidden layers

Number of hidden layer plays a key role in designing the overall structure of the neural network architect. It gives path for different prediction. As can be seen from Figure 8, the constant change across the test data error rate might result from small changes (one unit) of this hyperparameter while the changes are clearer for the training dataset.

5.4 | Number of batch size

This hyperparameter defines the number of observation (crashes) that would be propagated in the network. In other words, this value is equal to the number of observations in one forward/backward pass. There is a positive correlation

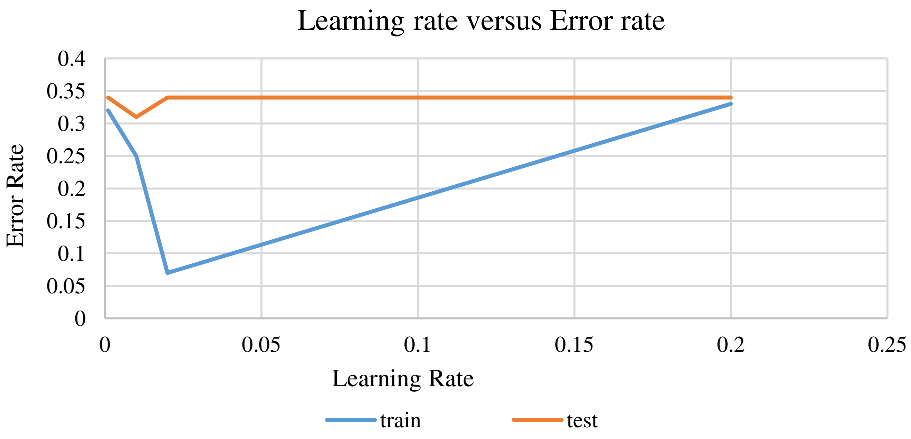


FIGURE 7 Impact of learning rate on the error rate of training and test datasets

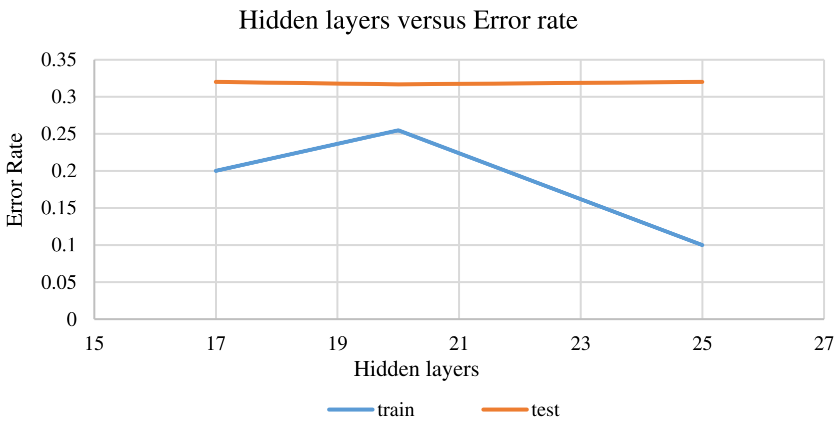


FIGURE 8 Impact of hidden layers on error rate of training and test datasets

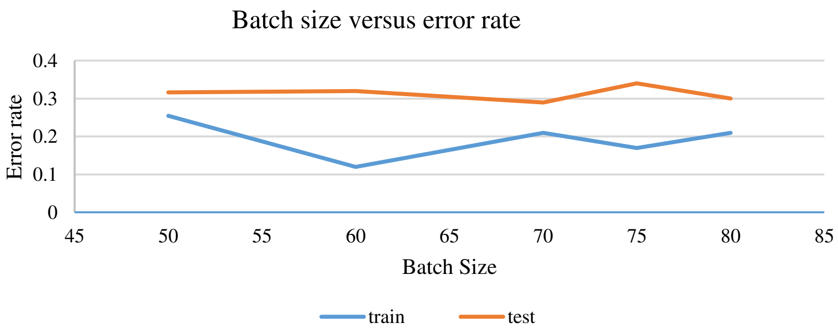


FIGURE 9 Impact of batch size on error rate of training and test datasets

between the number of batch size and the used memory. As can be seen from Figure 9, with the interval of 5 points, the error rate of both training and test dataset changes dramatically, highlighting the importance of setting a right value for this hyperparameter.

6 | CONCLUDING REMARKS

Traffic severity prediction is an important step in intelligent transportation system and traffic management to identify the drivers at higher risk of severe crashes, and thus preventing those crashes. The machine learning techniques have attracted much attention in various areas from voice to computer vision recognition. In this study, DNNs and one-layer neural network were considered for evaluation of the severity of motorcycle crashes. Feature reduction was performed before doing any analyses to come up with an optimum number of included predictors. Models were tuned up to have optimum values for hyperparameters. Single layer neural networks was included as a base model to see if application of various DL techniques could improve the performance of the model compared with this simple machine learning model. The DBN is a developed version of multilayer perceptron so both of this model included in the comparison as well.

On the other hand, the RNN was used due to the benefit that this model has in terms of incorporating loops allowing information to persist. The results of this study indicated that standard RNN and deep architect perform similarly in predicting the validation data (29%). On the other hand DBN, MLNN, and single layer neural network perform poorly on validation dataset, with error rate of 37% and 35% of test data, respectively.

While this study showed that the RNN and MLNN would outperform the other DL and single layer neural networks, more research is needed for the future studies to incorporate other DL techniques in predicting motorcycle crash severity. It should be noted that DL techniques have been commonly practiced on a very large dataset so the algorithms could train themselves for best performance on test dataset. However, although this study used less than 3000 observation for training the model, the results show a promising output in prediction of motorcycle crashes.

For the future studies more number of observations is recommended to predict crash severity. Due to imbalanced nature of crash dataset used in this study, the algorithms were not able to reach an acceptable accuracy for severe/fatal category. This is due to the fact that machine learning techniques work in favor of overrepresented data category, PDO. For the future dataset, it is recommended to balance dataset through under sampling, or synthetic minority over-sampling (SMOTE) technique to balance dataset before using any machine learning techniques.

The issue of standard RNN is the vanishing gradient. This resulted from the fact that each time-step during training, the model uses the same weight for calculating y. Thus during backpropagation errors would become smaller and smaller due to multiplications. This means that the model would be unable to remember the previous information, making the prediction on only the most recent information. The standard RNN only pass the input and the hidden state through a single tanh layer. However, the LSTM network has been improved by introducing additional gated and a cell state. This could address the issue of keeping or forgetting the context. For future studies, application of LSTM method on nonsequential dataset is recommended.

ACKNOWLEDGEMENTS

This work was supported by the Wyoming Technology Transfer Center (WYT2).

PEER REVIEW INFORMATION

Engineering Reports thanks Enrique Puertas, Apostolos Ziakopoulos, and other anonymous reviewers for their contribution to the peer review of this work.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

ORCID

Mahdi Rezapour https://orcid.org/0000-0003-0774-737X

REFERENCES

1. Vaca F. National Highway Traffic Safety Administration (NHTSA) notes. Drowsy driving. *Ann Emerg Med*. 2005;45(4):433.
2. Das S, Dutta A, Dixon K, Minjares-Kyle L, Gillette G. Using deep learning in severity analysis of at-fault motorcycle rider crashes. *Transp Res Rec*. 2018;2672(34):122-134.
3. Motorcycle safety is a two-way street. https://www.nsc.org/road-safety/safety-topics/motorcycle-safety.
4. NHTSA. Traffic safety facts motorcycles. Tech. Rep. DOT HS 812 492; 2016.
5. Hedlund J. *Motorcyclist Traffic Fatalities by State*. Governors Highway Safety Association; 2013.
6. Harb R, Yan X, Radwan E, Su X. Exploring precrash maneuvers using classification trees and random forests. *Accid Anal Prev*. 2009;41(1):98-107.
7. Rezapour M, Molan AM, Ksaibati K. Analyzing injury severity of motorcycle at-fault crashes using machine learning techniques, decision tree and logistic regression models. *Int J Transp Sci Technol*. 2019.
8. Weber A, Murray DC. *Evaluating the Impact of Commercial Motor Vehicle Enforcement Disparities on Carrier Safety Performance*. American Transportation Research Institute; 2014.

9. Theofilatos A, Ziakopoulos A. Examining injury severity of moped and motorcycle occupants with real-time traffic and weather data. *J Transp Eng A Syst*. 2018;144(11):04018066.
10. Quddus MA, Noland RB, Chin HC. An analysis of motorcycle injury and vehicle damage severity using ordered probit models. *J Safety Res*. 2002;33(4):445-462.
11. Lin M, Kraus JF. A review of risk factors and patterns of motorcycle injuries. *Accid Anal Prev*. 2009;41(4):710-722.
12. Pai C, Saleh W. An analysis of motorcyclist injury severity under various traffic control measures at three-legged junctions in the UK. *Saf Sci*. 2007;45(8):832-847.
13. Saha D, Alluri P, Gan A. A random forests approach to prioritize Highway Safety Manual (HSM) variables for data collection. *J Adv Transp*. 2016;50(4):522-540.
14. Hossain M, Muromachi Y. Understanding crash mechanism on urban expressways using high-resolution traffic data. *Accid Anal Prev*. 2013;57:17-29.
15. Li Z, Liu P, Wang W, Xu C. Using support vector machine models for crash injury severity analysis. *Accid Anal Prev*. 2012;45: 478-486.
16. Li X, Lord D, Zhang Y, Xie Y. Predicting motor vehicle crashes using support vector machine models. *Accid Anal Prev*. 2008;40(4):1611-1618.
17. Gu X, Li T, Wang Y, Zhang L, Wang Y, Yao J. Traffic fatalities prediction using support vector machine with hybrid particle swarm optimization. *J Algorithms Comput Technol*. 2018;12(1):20-29.
18. Friedman JH. Multivariate adaptive regression splines. *Ann Stat*. 1991;19:1-67.
19. Chang L, Chu H, Lin D, Lui P. Analysis of freeway accident frequency using multivariate adaptive regression splines. *Procedia Eng*. 2012;45:824-829.
20. Sameen MI, Pradhan B, Shafri H, Hamid HB. Applications of deep learning in severity prediction of traffic accidents. Paper presented at: Global Civil Engineering Conference; 2017:793-808.
21. Yu S, Wu Y, Li W, Song Z, Zeng W. A model for fine-grained vehicle classification based on deep learning. *Neurocomputing*. 2017;257:97-103.
22. Dong C, Shao C, Li J, Xiong Z. An improved deep learning model for traffic crash prediction. *J Adv Transp*. 2018;2018.
23. R. D. Guide. *American Association of State Highway and Transportation Officials*. Washington, DC; 1996.
24. Zheng M, Li T, Zhu R, et al. Traffic accident's severity prediction: a deep-learning approach-based CNN network. *IEEE Access*. 2019;7:39897-39910.
25. Savolainen P, Mannering F. Probabilistic models of motorcyclists' injury severities in single-and multi-vehicle crashes. *Accid Anal Prev*. 2007;39(5):955-963.
26. RColorBrewer S, Liaw MA. *Package 'randomForest'*. Berkeley, CA: University of California; 2018.
27. Ripley B, Venables W, Ripley MB. Package 'nnet'. *R Package Version*. 2016;7:3-12.

How to cite this article: Rezapour M, Nazneen S, Ksaibati K. Application of deep learning techniques in predicting motorcycle crash severity. *Engineering Reports*. 2020;2:e12175. https://doi.org/10.1002/eng2.12175