Accident Analysis and Prevention 151 (2021) 105950

Contents lists available at ScienceDirect

Accident Analysis and Prevention

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

Crash data augmentation using variational autoencoder☆

Zubayer Islam *, Mohamed Abdel-Aty, Qing Cai, Jinghui Yuan

Department of Civil, Environmental and Construction Engineering, University of Central Florida, Orlando, FL, 32816, USA

ARTICLE INFO

Keywords:
Variational autoencoder
Data augmentation
Crash prediction

ABSTRACT

In this paper, we present a data augmentation technique to reproduce crash data. The dataset comprising crash and non-crash events are extremely imbalanced. For instance, the dataset used in this paper consists of only 625 crash events for over 6.5 million non-crash events. Thus, learning algorithms tend to perform poorly on these datasets. We have used variational autoencoder to encode all the events into a latent space. After training, the model could successfully separate crash and non-crash events. To generate data, we sampled from the latent space containing crash data. The generated data was compared with the real data from different statistical aspects. *t*-Test, Levene-test and Kolmogrove Smirnov test showed that the generated data was statistically similar to the real data. It was also compared to some of the minority oversampling techniques like SMOTE and ADASYN as well as the GAN framework for generating data. Crash prediction models based on Logistic Regression (LR), Support Vector Machine (SVM) and Artificial Neural Network (ANN) were used to compare the generated data from the different oversampling techniques. Overall, variational autoencoder (VAE) showed excellent results compared to the other data augmentation methods. Specificity is improved by 8% and 4% for VAE-LR and VAE-SVM respectively when compared to SMOTE while the sensitivity is improved by 6% and 5% when compared to ADASYN. Moreover, VAE generated data also helps to overcome the overfitting problem in SMOTE and ADASYN since there is flexibility in choosing the decision boundary.

1. Introduction

Real-time crash likelihood prediction has been an important area of study for the past two decades. Expressways, being a vital part of any roadway network, must be evaluated not only from a capacity standpoint but also from a safety stance. With modern sensors deployed along the expressways, it is convenient to get different roadway information in real-time. A major drawback with such data is that the data is highly imbalanced. The ratio of crash to non-crash data can be as imbalanced as 1:11,000 (Cai et al., 2020). Machine learning algorithms struggle to correctly predict outcomes with such data. As we will show later in this study, a model trained with such data always predicts non-crash events.

Crash prediction is an important tool to ensure safety in freeways. Fig. 1 shows the number of deaths from motor vehicle accidents from 1992 to 2018 in a report (NSC and NHTSA Report, 2020). It is evident that the number of fatal deaths has remained more or less around the 40,000 mark in these 26 years. In 2018 alone, 4.5 million of the crashes needed paramedic support. These has led to several research attempts in the field of safety and the recent values show that more and more work is necessary. Since crash like situations develop within short-term turbulence of traffic flow (Lee et al., 2003), it is necessary to have real-time crash risk monitoring systems. To reduce crashes there have been research from a planning perspective in which studies try to quantify how the demographic characteristics aids in a crash. From an engineering perspective, it can be the lighting of a roadway or its geometric design and from the viewpoint of control solutions, it can be countermeasures like ramp metering and variable speed limit. To identify which method to use, data is a basic requirement. While statistical methods can be tweaked to counter being biased, machine learning method can easily become biased. As more and more machine learning crash risk predictive schemes are being proposed (Li et al., 2020a, 2020b; Yu and Abdel-Aty, 2013), it is important to balance these datasets. Synthetic data augmentation method like SMOTE, matched case control, random oversampling has been traditionally used with crash risk prediction (Ahmed and Abdel-Aty, 2011; Basso et al., 2018; Sun et al., 2020; Yuan et al., 2019). These methods come with several limitations including

☆ This paper has been handled by associate editor Tony Sze.

* Corresponding author.
E-mail addresses: zubayer_islam@knights.ucf.edu (Z. Islam), M.Aty@ucf.edu (M. Abdel-Aty), qingcai@knights.ucf.edu (Q. Cai), jinghuiyuan@knights.ucf.edu (J. Yuan).

https://doi.org/10.1016/j.aap.2020.105950
Received 28 July 2020; Received in revised form 6 December 2020; Accepted 10 December 2020
Available online 25 December 2020
0001-4575/

deformation of the decision boundary, overfitting and reduced dataset. While most data augmentation techniques using deep learning are applied to other areas of research (Frid-Adar et al., 2018; Perez and Wang, 2017), the number of studies relating to data augmentation in crash prediction is limited from a deep learning perspective (Cai et al., 2020). In this study, we propose to augment crash data using Variational Autoencoder (VAE) and therefore aim to overcome the limitations with traditional synthetic oversampling methods. In data generation with VAE, the decision boundary can be precisely selected which can help in reducing overfitting inherent to synthetic oversampling methods.

The paper is organized as follows: we first discuss the relevant work and then discuss the VAE structure in the next section. We describe how we initialized the data for quick convergence and also how the normal distribution was used to encode the data. The generated data was then compared with the real data using t-test, Levene-test and Kolmogorov–Smirnov test to compare mean, variance and distribution respectively. To better understand the data, it was finally tried on three crash risk prediction algorithms: Logistic Regression (LR), Support Vector Machine (SVM) and Artificial Neural Network (ANN). The metrics used for evaluation was specificity, sensitivity and area under receiver operating characteristics curve (AUC). The generated data was also compared with SMOTE and ADASYN to assert the improvement in performance.

2. Related work

2.1. Variational autoencoder

Autoencoders are neural network architectures that have three parts cascaded to each other: encoder, latent space and decoder. Encoder compresses the input data into a lower dimensional latent space. The decoder then tries to recreate the input data from the latent space. Variational Autoencoder (VAE) is a special type of autoencoder that were first introduced by Kingma and Welling (2014). The main challenges of why an autoencoder could not be used to generate data were addressed in this paper. A VAE consists of an encoder and decoder just like an autoencoder, but the loss term and the encoded layers of the autoencoder are modified so that VAEs could be used as a generative model. Since this study, VAEs have found applications in different fields. It has been used to learn images, labels and captions (Pu et al., 2016), to detect anomaly (An and Cho, 2015), for text classification (Xu et al., 2017) etc. Different improved version of the VAEs were also available (Kusner et al., 2017; Sønderby et al., 2016). In the transportation research VAEs have been used to identify missing data from sensor network (Boquet et al., 2019), traffic identification (D. Li et al., 2017), human mobility (D. Huang et al., 2019), anomaly detection (Boquet et al., 2020), etc. In this study, we aim to use VAE for generating crash data. To the best of the knowledge of the authors, this is the first time VAEs were studied from a crash data augmentation standpoint.

2.2. Crash data prediction

Various methods have been used over the past two decades to predict real-time crash likelihood. In most of the previous studies, the traffic data is aggregated at 5-minute intervals. At the start of the century there were several statistical methods that began to gain popularity. Initially case control logistic regression was studied (Abdel-Aty et al., 2004) which was improved upon with log-linear and Bayesian logistic models (L. Wang et al., 2015, 2019a, 2019b). Lately, there has been increasing work using learning algorithms like Support Vector Machine (SVM) (Basso et al., 2018; Yu and Abdel-Aty, 2013), Long Short Term Memory (LSTM) (Li et al., 2020a, 2020b), Random Forest (Lin et al., 2015) to predict crashes. Huang et al. (2020) used CNN to predict crashes in Interstates and found better results than shallow models. Bao et al. (2019) also used CNN to model citywide short-term crash risk prediction and reported that CNN was able to capture local spatial correlation. XGBoost (Sun et al., 2020), AdaBoost (Ke et al., 2019), Multilayer Perceptron (MLP) (Abou Elassad et al., 2020b, 2020a; Peng et al., 2020), Back Propagation Neural Net (BPNN) (Wang et al., 2019a) have also been studied with competitive accuracy scores.

2.3. Crash data augmentation

Traffic data related to crashes can be highly imbalanced since generally only none or a handful of crashes occur on a segment each year. With the advancement of more safety features and warning systems in recent vehicles, the crashes are expected to be reduced even further. If we down-sample the non-crash data to get a sample size similar to a crash data sample, extensive non-crash events are neglected. Therefore, it is necessary to up-sample the crash data to properly train it with any model, otherwise the model will be skewed to non-crash events.

A comprehensive evaluation of crash risk prediction using oversampling technique is presented in Table 1. Overall, three different methodologies have been used to handle imbalanced datasets. In several studies, matched case control sampling was carried out. In this method, the non-crash datapoints are sampled down so that the ratio between crash and non-crash events are suitable for training models. Abdel-Aty (2004) showed that there is no significant difference when the data ratio is changed from one to five. Several previous studies have used 4:1 (Ahmed and Abdel-Aty, 2011; Shi and Abdel-Aty, 2015; Xu et al., 2012; Yu et al., 2016, 2018; Zheng et al., 2010) however ratio of 10:1 (Yuan et al., 2019) and 20:1 (Xu et al., 2013) were also used. The main limitation of such method is that it does not fully capture the non-crash

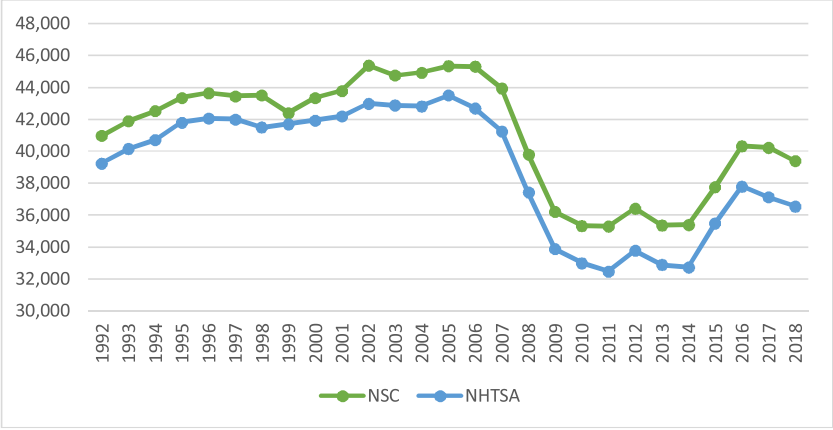

Fig. 1. Motor vehicle deaths, National Safety Council (NSC) and National Highway Traffic Safety Administration (NHTSA), 1992-2018.

Table 1
Literature Review of Data Balancing Methods used for Crash Risk Prediction.

Reference	Classification Model	Balancing Method
Ahmed and Abdel-Aty, 2011	LR	Matched-case control
Xu et al., 2012	Conditional LR	Matched-case control
Zheng et al., 2010	Conditional LR	Matched-case control
Shi and Abdel-Aty, 2015	Bayesian LR	Matched-case control
Yu et al., 2016	LR, negative binomial	Matched-case control
Yu et al., 2018	Bayesian LR	Matched-case control
Xu et al., 2013	Binary Logit	Matched-case control
Basso et al., 2018	SVM, LR	SMOTE
Parsa et al., 2019	SVM, PNN	SMOTE
Yahaya et al., 2019	RF	SMOTE
Li et al., 2020a	LSTM-CNN	SMOTE
Li et al., 2020a	LSTM-RNN	SMOTE
Zhou et al., 2019	Regression	SMOTE-ENN
Cai et al., 2020	SVM, LR, ANN, CNN	DCGAN
Yin et al., 2019	LR	SMOTE, under-sampling, Matched-case control
Elamrani Abou Elassad et al., 2020	SVM, MLP	SMOTE
You et al., 2017	SVM,	SMOTE, ADASYN
Sun et al., 2020	XGBoost	Random Over Sampler
He et al., 2018	MLP	Supervised data synthesizing
Yuan et al., 2019	LSTM-RNN	SMOTE, matched case control
Peng et al., 2020	RF, MLP	Youden Index
Wang et al., 2019a, 2019b	BPNN	SMOTE
Abou Elassad et al., 2020b	BL, kNN, SVM, MLP	SMOTE
Elamrani Abou Elassad et al., 2020	SVM, MLP	SMOTE
Ke et al., 2019	SVM, AdaBoost	SMOTE
Abou Elassad et al., 2020a	SVM, MLP, RF	SMOTE
Parsa et al., 2019	SVM, PNN	SMOTE
Basso et al., 2020	SVM, LR	SMOTE, Random over sampling

events because non-crash events are scaled down by factors of 1000. Moreover, linear algorithms such as LR, SVM can still make good predictions using a handful of data, but complex non-linear methods like different ensemble techniques have to be trained on relatively more data than the linear models. This fact has been proven by a series of experiments by Catal and Diri (2009).

The second commonly used method is random oversampling in which the data from the minority class is repeated until it matches the majority class (Basso et al., 2020; Sun et al., 2020). This method overfits the crash data which can sometimes result in poorer performance.

Thirdly, the technique Synthetic Minority Oversampling Technique (SMOTE) has been used mostly to oversample crash data (Abou Elassad et al., 2020a; Basso et al., 2018, 2020; Ke et al., 2019; Li et al., 2020a, 2020b; Parsa et al., 2019; Sun et al., 2020; Wang et al., 2019a, 2019b; Yahaya et al., 2019; Yuan et al., 2019). This method introduces synthetic data along the line segment joining the k minority class samples. Thus, it only takes into account the closeness of the datapoints and the intrinsic characteristics is not taken into consideration (Sáez et al., 2015). The main limitation of this blind oversampling includes creation of unwanted samples around the decision boundary of the majority and minority class thereby disrupting the natural boundary between classes. It also creates too many samples of the majority class which does not improve the minority class performance. These limitations has led to several versions of the SMOTE such as SL-SMOTE (Bunkhumpornpat et al., 2009), LN-SMOTE (Maciejewski and Stefanowski, 2011), SMOTE-IPF (Sáez et al., 2015), etc. A fourth technique called Adaptive Synthetic (ADASYN) (He et al., 2008) oversampling was also used by You et al. (You et al., 2017) but it was used to undersample the majority class to match the minority class. Moreover, Qing et al. recently applied Deep Convolutional Generative Adversarial Network (DCGAN) (Cai et al., 2020) to augment crash data and the results show how deep learning data augmentation techniques can be competitive to the statistical models like SMOTE.

Variational Autoencoder based synthetic data generation overcomes these limitations to a great extent. Firstly, it creates synthetic oversampling of only the minority class to match the majority class and therefore, the data limitation posed by the matched case control study is avoided. It also overcomes the random oversampling method because synthetic data is not generated based on duplication. Finally, the decision boundary disruption caused by SMOTE also does not occur in a VAE since the process of deciding the decision boundary can be expanded or limited with the help of confidence ellipsoid as will be explained in the proposed method section. In this study, ADASYN has also been used as a data oversampling technique to compare the performance of VAE and SMOTE. To the best of the knowledge of the authors, this is the first use of ADASYN to oversample crash data to the best of the knowledge of the authors.

3. Dataset

This paper used the crash dataset obtained from processing the Microwave Vehicle Detection System (MVDS) data from expressway SR 408 in Orlando. It is 21.4 miles long with an average 417,000 vehicles using it each day (Central Florida Expressway Authority Webpage, 2020). It has a total of 110 MVDS detectors. The data from the year 2017 was used in this study. The roadway is shown in Fig. 2.

The MVDS detectors in SR 408 are able to report speed, volume, lane occupancy every time a vehicle passes over the detectors. This was further processed to obtain average and standard deviation of speed and volume. All of the 24 features that were derived from the MVDS data are summarized in Table 2. Speed difference is calculated between inner and outer lanes. Inner lane is closest to the direction of opposing traffic and outer lane is the one farthest away from opposing traffic.

Each of the parameters has two values (as indicated by *1* and *2*) because for each event, two detectors upstream and two detectors downstream were considered. The data was aggregated in 5-minute intervals to prepare the features. There were 625 crash events in the year 2017 and about 6,749,447 non-crash events. For each of the crashes, 5−10 min before the event was labelled as a crash data. In addition, six hours prior to the crash were discarded since the data points before a crash may contain some traffic variation leading to the crash and therefore does not represent steady-state conditions. Moreover, six hours after the crash was also removed since it will take some time to reach steady state condition after a crash event.

4. Proposed method

Variational Autoencoders (VAE) are a type of autoencoder that has gained tremendous applications in computer vision recently. Proposed by Kingma and Welling (2014), it was quickly adapted to much of the vision available datasets. The image datasets are a collection of different pixel values. Similarly, traffic data obtained from different detectors are also a series of values. Of course, the range of values of the two datasets is vastly different but due to the normalization technique that is used before training many machine learning models, the values of any dataset maps within 0–1. Therefore, the intuition is that traffic data with a fixed window is numerically the same as that of an image of same dimension and VAEs can be used to augment traffic data as well.

To the best of the knowledge of the authors this is the first time variational autoencoders is being investigated from crash data augmentation point of view. In this section, we will first present the basics behind a VAE and then describe the model that gave the best results for crash data generation.

An autoencoder is a neural net that maps its input to an output of exactly the same dimension. Towards the middle of the neural net there

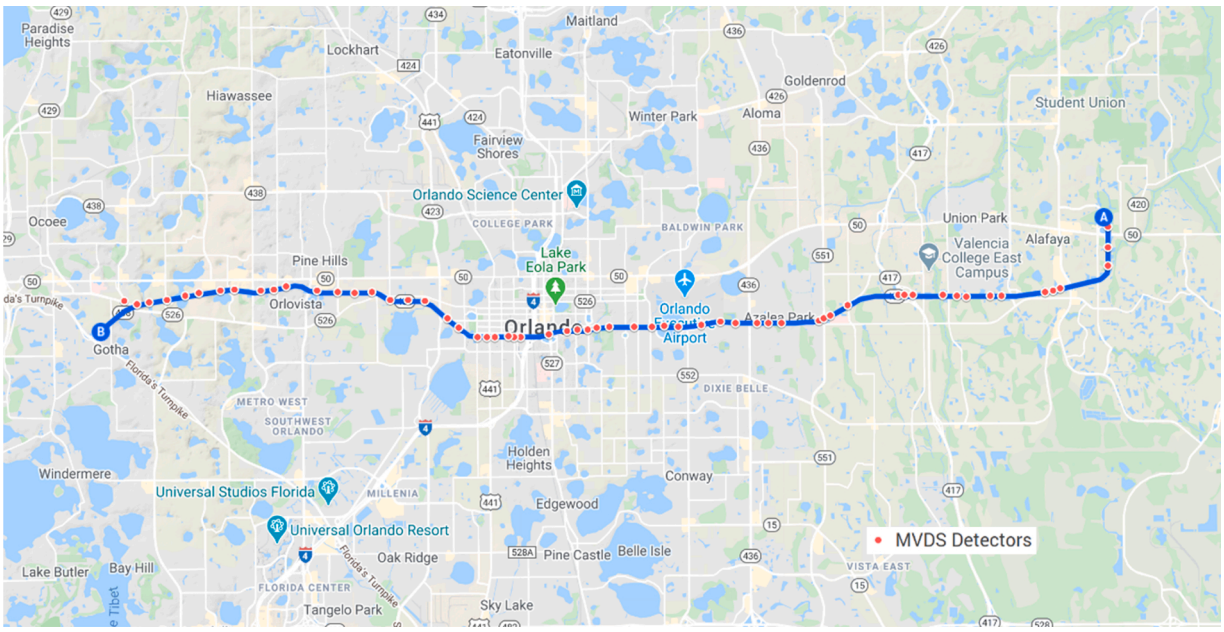

Fig. 2. Study area showing SR 408 along with the MVDS detectors.

Table 2
Features extracted from MVDS data.

Features	Upstream	Downstream	Description
Volume	*volume_up1, volume_up2*	*volume_down1, volume_down2*	*Directly obtained from MVDS*
Average Speed	*avg_speed_up1, avg_speed_up2*	*avg_speed_down1, avg_speed_down2*	*Directly obtained from MVDS*
Standard Deviation of Speed	*std_speed_up1, std_speed_up2*	*std_speed_down1, std_speed_down2*	*Calculated from average speed and actual speed*
Coefficient of Variation of Speed	*cv_speed_up1, cv_speed_up2*	*cv_speed_down1, cv_speed_down2*	*Calculated from average speed and actual speed*
Speed Difference	*speed_diff_up1, speed_diff_up2*	*speed_diff_down1, speed_diff_down2*	*Difference in speed between inner and outer lanes*
Congestion Index	*CI_up1, CI_up2*	*CI_down1, CI_down2*	$CI = \begin{cases} \frac{speedlimit - actualspeed}{speedlimit}, & CI > 0 \\ 0, & CI \leq 0 \end{cases}$

is a bottleneck that is the unique feature of an autoencoder. These can help in reducing noise or getting a lower dimensional representation of the input. A special variation of the autoencoder is the variational autoencoder. In general, all autoencoders have an encoder, followed by a latent space and a decoder. An encoder compresses the input x into a latent space z and a decoder decodes from this latent space z to get the reconstruction $\hat{x}$. The encoder and decoder could be interpreted as a recognition and generative model respectively. Therefore,

$$z \approx \text{Enc}(x) = q_{\varnothing}(z|x),\ \hat{x} \approx \text{Dec}(z) = p_{\theta}(x|z), \tag{1}$$

Here, $q_{\varnothing}(z|x)$ is the true posterior of $p_{\theta}(x|z)$ and $\varnothing$ and θ have been used to denote encoder and decoder models respectively.

For an autoencoder, the latent space is discrete and therefore sampling from this space would not result in anything meaningful. Variational autoencoder takes these discrete values and tries to find a known distribution of the latent space. Therefore, a variational autoencoder not only tries to reconstruct its input, but also tries to form a distribution in the latent space. The loss function L (Eq. (2)) consists of two terms; the first term penalizes the reconstruction error between the input x and the output $\hat{x}$ and the second term penalizes the error between the prior $p_{\theta}(z)$ and the learned distribution $q_{\varnothing}(z|x)$.

$$L = \log p_{\theta}(x|z) - D_{KL}(q_{\varnothing}(z|x)||p_{\theta}(z)) \tag{2}$$

In Eq. (2), $p_{\theta}(z)$ is the prior distribution and D_{KL} is the Kullback-Leibler divergence which can measure how much information is lost when q is used to represent p.

For crash data augmentation, we used the VAE model shown in Fig. 3. The leftmost layer labelled x is the input and the rightmost layer labelled $\hat{x}$ is the output. The bottleneck is the layer z which we have chosen to have a dimension 3 since better latent space was obtained with 3 rather than 2. All the weights and biases of the encoder and decoder were initialized with Xavier Glorot initialization (Glorot and Bengio, 2010). This initialization is able to bring faster convergence for deep learning networks. The latent space was encoded to a normal distribution with a mean of 0 and standard deviation of 0.1. Therefore, the encoder takes the input data and finds a latent space distribution which will be a normal distribution for our case. The adjustment in loss function is done so that there is a penalty if the encoder tries to encode the input to any value other than the specified distribution.

4.1. Encoder model of VAE

We considered a two layer densely connected encoder model for our VAE. The next two layers after the input x are the decoder. The number of neurons chosen was (24, 24). We also experimented with lower numbers such as (10,10), (15,15), (20,20) and (24,20). Best latent space visualization was obtained for (24,24).

4.2. Decoder model of VAE

The decoder takes the latent layer code and tries to reconstruct the original crash data. For the decoder we also had 24 neurons for the first

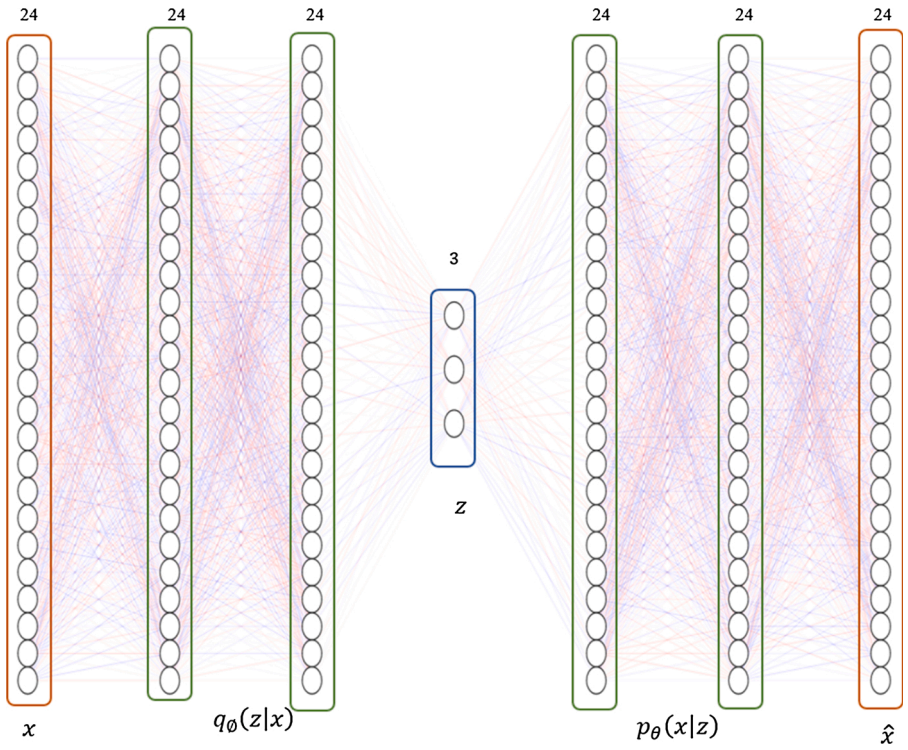

Fig. 3. VAE model presented in the paper (the numbers above each layer represents the number of neurons).

layer and the same for the second layer. At the end of the decoder a sigmoid activation function was used. The output layer would be the same as the input which is 24. All the neurons were densely connected to one another.

4.3. Data generation pseudocode

The pseudocode for training and data generation of VAE is shown Table 3. The input for training is the real data (70 % was used for training). The optimization techniques used was Adam with a learning rate of 0.0001. The batch size was selected as 874. Of these 874 samples, 437 was crash and 437 was non-crash. Therefore, we have used the same crash values over the entire epoch. This was necessary otherwise the VAE model would overfit with only the non-crash values. In each iteration, we also calculated loss on the train samples for visual evaluation. It was trained for a total of 12,000 epochs since this is required to consider all the non-crash data. For data generation, we sampled from the latent space generated from the training phase and passed it through the trained decoder to obtain synthetic data at least once.

It is also important to determine the latent space boundary from which we have to sample to get crash data. Generally, VAEs generate the latent space in which there is overlap between classes of data. If we want to sample any particular class, it is necessary to estimate the cluster boundary of the class. Visual interpretation may often lead to false selection of the boundary. For our case, since we are interested to sample crash data, it is important to find the cluster volume of the crash data. We have used confidence ellipsoid to draw this latent space boundary. A confidence ellipsoid draws a cluster boundary based on the confidence level. An 80 % confidence ellipsoid means that 80 % of the observed data falls within the boundary of the ellipsoid. We have used three confidence levels: 80 %, 85 % and 90 %. Next, we could easily sample from the space within the ellipsoid to get generated data.

Table 3
Algorithm for generating synthetic data.

1:	**Input:** Real Data
2:	**Initialize:** decoder, encoder, prior, latent space dimensions
3:	**Training:**
4:	**for** the number of iterations **do**
5:	**for** the number of batches **do**
6:	input the batch into the encoder
7:	calculate loss using equation 1 and optimize over the training dataset
9:	**Data Generation:**
10:	Determine crash boundary in latent space with confidence ellipsoids
11:	**for** number of samples **do**
12:	sample from the latent space
13:	use the trained decoder from training step to obtain synthetic data

5. Evaluation

5.1. Latent space visualization

The latent space for a VAE gives the impression as to how effective a model is in separating a dataset into its target. By properly interpreting the latent space boundary it is possible to create new samples of data. It is also an important indicator as to how effective is the KL divergence loss. In the Fig. 4, we display the results obtained from our VAE model.

Since there are three neurons in the bottleneck of the model, the latent space is best visualized in 3D as is shown in Fig. 4 (a). $z1$, $z2$, $z3$ are the three neurons which are being displayed. This figure gives the idea that the system is able to create different clusters for the crash and non-crash data. The crash data, which is encoded in yellow, is at the surface of the 3D plot as shown in Fig. 4 (a). We also show the 2D plots in Fig. 4 (b), (c) and (d). All of these figures show that even though the surface of the latent space mostly contains the crash data, the overlap between crash and non-crash events is also significant. This can be explained from the fact that certain crash events may not result in a crash due to number of other factors like driver reaction, automatic breaking or other safety features that are becoming more and more popular in cars. This is also clearly depicted in Fig. 5 where we plot one crash and one non-crash sample in each sub-figure (a) and (b). Both these datapoints are obtained from the real dataset. It shows how close the values of the two events might be. Nevertheless, the overlap is an

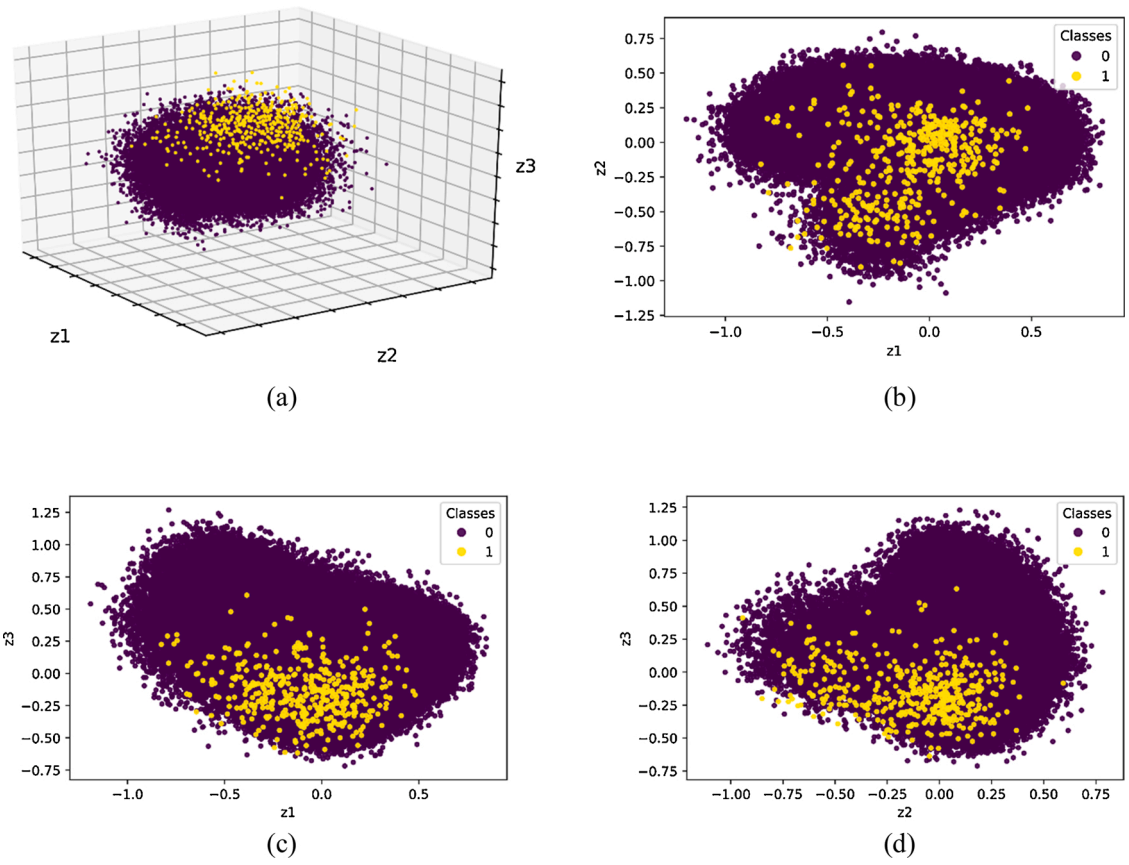

Fig. 4. Latent Space Visualization of the VAE Model.

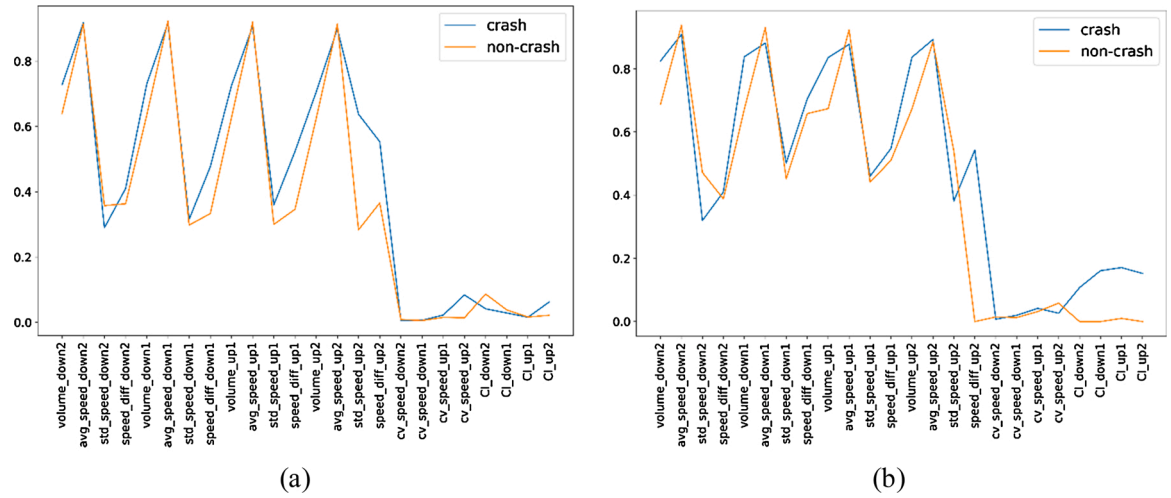

Fig. 5. Similarities between crash data and non-crash data (aggregated at 5-min interval).

indicator that the conditions are getting risky and that a crash is likely.

Fig. 6 shows the 80 % confidence ellipsoid that we have used to determine the latent space boundary of the crash data. This was a very important step because visual evaluation of the latent space may often lead to capturing false boundary and eventually false data. The yellow dots are the crash data and the purple ellipsoid is the 80 % confidence ellipsoid. This region was selected to sample generated crash data. The same was done for two other confidence levels: 85 % and 90 %.

5.2. Synthetic data evaluations

After training the VAE, we can sample from the latent space shown in Fig. 4 to generate more crash data. For our case, we have generated over 20 million crash events. In most data augmentation, the next step is to have a visual observation of the generated data (Jiwei Wang et al., 2018). We evaluate the data from a global viewpoint of mean and standard deviation. We also plot the data distribution and analyze if both real and generated data show similar pattern.

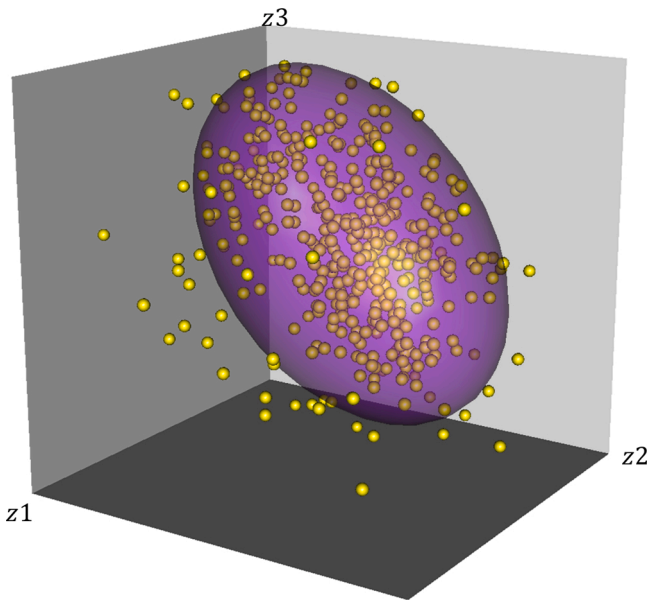

Fig. 6. Latent space showing the crash data samples and 80 % confidence ellipsoid.

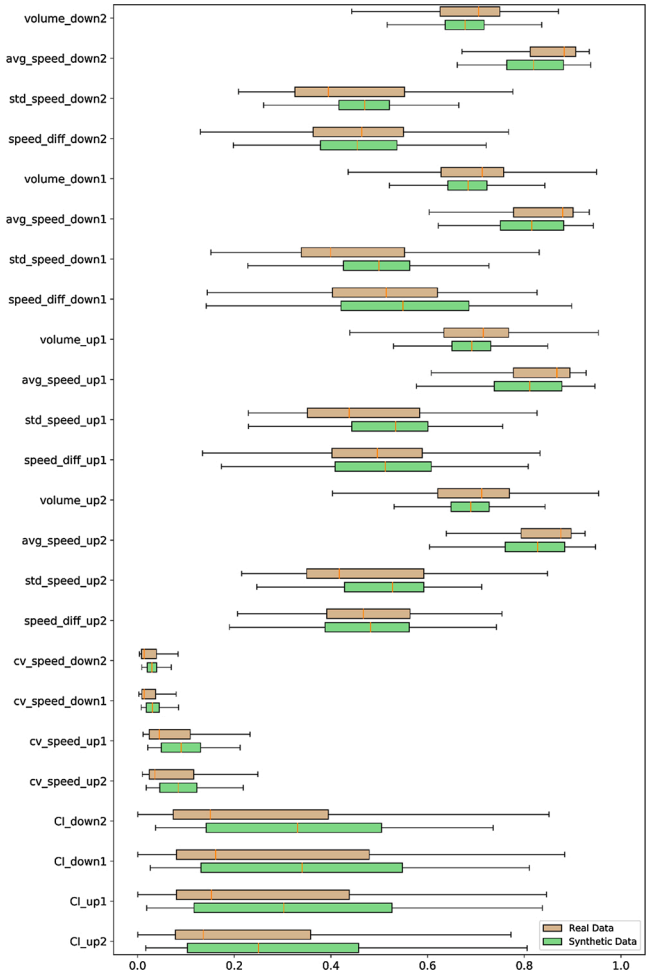

Fig. 7. Boxplot showing the mean and standard deviation of real data and synthetic data.

5.2.1. Global mean and standard deviation evaluation

We illustrate the mean and standard deviation of the generated and real data in the Fig. 7. We have compared 2 million synthetic crash samples with the 437 crash samples in the training data. Each VAE has two losses.

We showed in the previous section that the KL divergence loss limits the data within a fixed area while the reconstruction loss shows how close the generated data is to the real data. The mean and standard deviation gives an idea about this. As can be seen from Fig. 7, the quartile range of values for the generated data mimics the real data. The mean of some of the values have shifted considerably. This can be attributed to the huge difference of data samples between the two sets. An outlier in the real data can easily push the mean aside which cannot happen for the 2 million generated data.

5.2.2. Data distribution evaluation

The distribution of each of the 24 variables used in this study is shown in Fig. 8. We have used a violin plot to compare between the distributions of the generated data and the real data. This helps us to compare not only the distribution but also the gives an idea about the mean of the two datasets. Synthetic data is shown in green and real data in orange. For all of the variables, the generated data pattern closely resembles the distribution of real data. One important insight from this figure is that the spread (the difference between the highest and lowest values) of the generated dataset is always less than the spread of the real dataset. This is due to the reason that the real dataset only has 437 crash events whereas the synthetic dataset has 2,000,000 crash events. Also, while sampling, we intentionally left out some outliers that were too far from the crash cluster. Thus, the distribution standard deviation is lower for the larger dataset since it has less percentage of outliers.

5.2.3. Statistical evaluations

The minimum, maximum, mean and standard deviation of all the variables in the real and synthetic data is summarized in Table 4. In addition, three different tests were carried out to confirm that the two datasets do not differ from each other statistically. 30 random samples were taken from the two datasets. The p-values of each of the tests are presented in the table. For t-test, it can be noted that, the p value is greater than 0.05 for all of the variables. This confirms that the mean of the two datasets are not significantly different. Levene's test was carried out to compare the variances between the two datasets. The p-value is greater than 0.05 for all variables except five. This indicates that the variance of the datasets is not statistically. Similarly, the p values from Kolmogrove Smirnov test are also greater than 0.05 indicating that the distribution of the variables in the two datasets are not statistically significant. Therefore, we can conclude that the two datasets are statistically similar.

5.3. Classification using real and synthetic data

In the previous section, we examined the data from a statistical point of view. It is also important to see how the data performs under learning algorithms that have used in the past to classify crash and non-crash data. We used Logistic Regression (LR) to classify crash and non-crash data since this was a common methodology used in previous work (Cai et al., 2020; Li et al., 2020a, 2020b; Lin et al., 2015). We also used Support Vector Machine (SVM) and Artificial Neural Network that has also been previously studied (Basso et al., 2018; Cai et al., 2020; Yu and Abdel-Aty, 2013).

5.3.1. Data preparation

The real dataset was split into train and test datasets. 70 % data was used for training and the rest for evaluation. It can also be mentioned here that a set of 100 different random splits of the dataset was generated and then evaluated with the chosen dataset. Statistical t-test, Levene test and Kolmogorov–Smirnov test was carried out. It was noted that

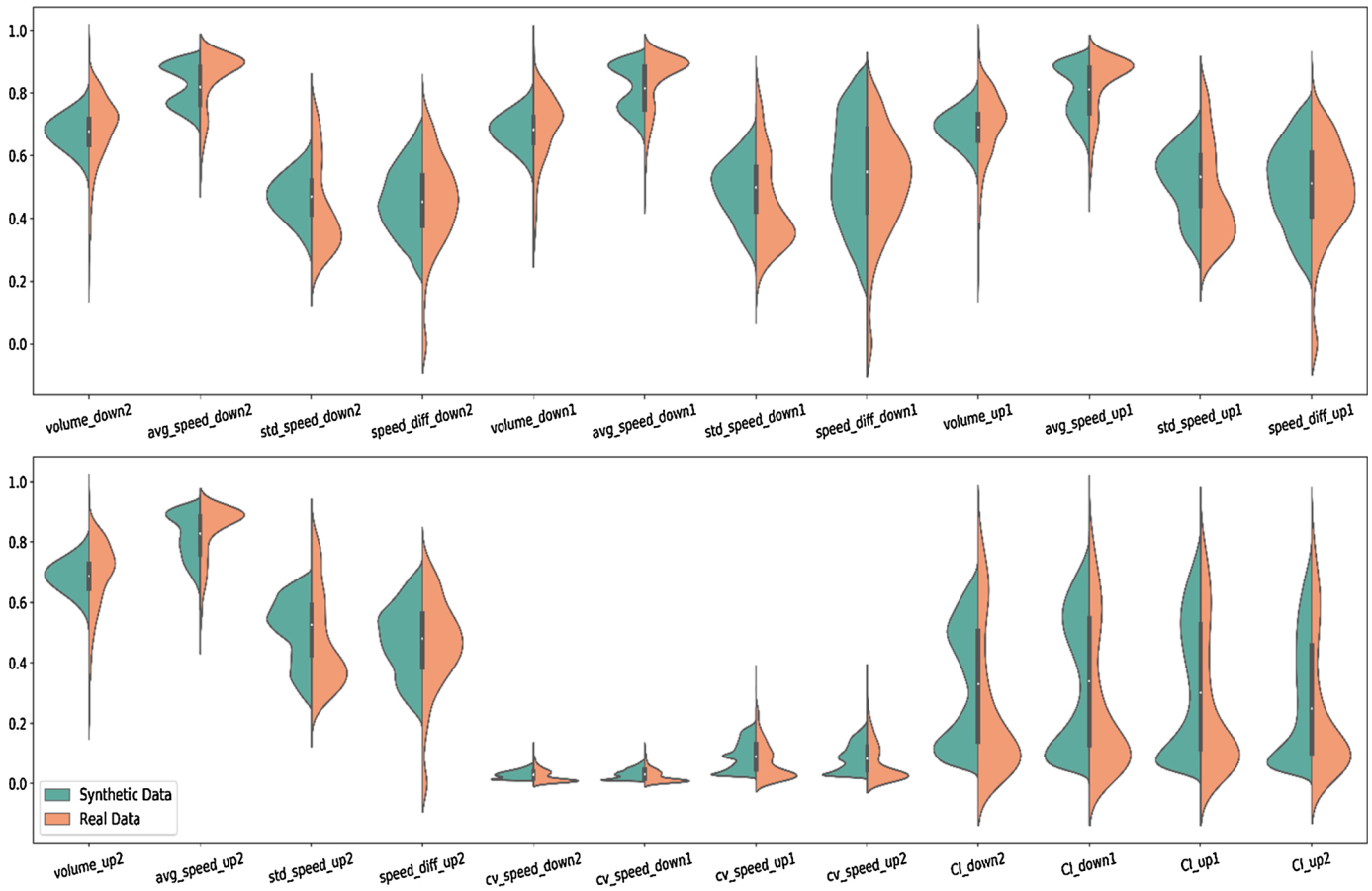

Fig. 8. Data distribution evaluation of generated data and noisy data.

Table 4
Descriptive statistical features of real and synthetic data as well as comparison between the two based on t-test, Levene-test and Kolmogrove Smirnov test (ks-test).

	Real Data				Synthetic Data				Statistical Tests between the real dataset and synthetic dataset		
Variable	Min	Max	Mean	STD	Min	Max	Mean	STD	t-test (p-value)	Levene-test (p-value)	ks-test (p-value)
volume_down2	0.20	0.95	0.68	0.10	0.47	0.83	0.67	0.05	0.92	0.93	0.39
avg_speed_down2	0.52	0.93	0.84	0.08	0.66	0.93	0.82	0.06	0.26	0.18	0.80
std_speed_down2	0.20	0.77	0.43	0.14	0.24	0.66	0.46	0.07	0.43	0.15	0.39
speed_diff_down2	0	0.76	0.44	0.15	0.19	0.72	0.45	0.10	0.49	0.74	0.59
volumne_down1	0.31	0.94	0.68	0.11	0.48	0.84	0.68	0.05	0.43	0.45	0.59
avg_speed_down1	0.46	0.93	0.83	0.08	0.62	0.94	0.81	0.07	0.32	0.26	0.39
std_speed_down1	0.15	0.83	0.44	0.14	0.22	0.72	0.49	0.09	0.42	0.52	0.13
speed_diff_down1	0	0.82	0.50	0.17	0.14	0.89	0.54	0.16	0.63	0.29	0.23
volumne_up1	0.20	0.95	0.69	0.10	0.48	0.84	0.68	0.05	0.58	0.56	0.03
avg_speed_up1	0.47	0.92	0.82	0.09	0.57	0.94	0.80	0.08	0.18	0.26	0.59
std_speed_up1	0.22	0.82	0.47	0.15	0.22	0.75	0.51	0.10	0.83	0.60	0.59
speed_diff_up1	0	0.83	0.48	0.16	0.17	0.80	0.50	0.13	0.94	0.74	0.39
volumne_up2	0.21	0.95	0.68	0.11	0.49	0.84	0.68	0.05	0.90	0.62	0.39
avg_speed_up2	0.48	0.92	0.83	0.08	0.60	0.94	0.81	0.07	0.15	0.33	0.39
std_speed_up2	0.21	0.84	0.47	0.15	0.24	0.71	0.50	0.10	0.66	0.86	0.80
speed_diff_up2	0	0.75	0.45	0.15	0.18	0.74	0.47	0.11	0.43	0.18	0.39
cv_speed_down2	0.003	0.12	0.02	0.02	0.008	0.07	0.03	0.01	0.57	0.03	0.23
cv_speed_down1	0.002	0.12	0.02	0.02	0.007	0.08	0.03	0.01	0.70	0.64	0.07
cv_speed_up1	0.011	0.35	0.07	0.06	0.02	0.21	0.09	0.04	0.70	0.23	0.13
cv_speed_up2	0.009	0.35	0.07	0.06	0.01	0.21	0.08	0.04	0.65	0.37	0.59
CI_down2	0	0.85	0.25	0.23	0.03	0.73	0.32	0.19	0.26	0.19	0.59
CI_down1	0	0.88	0.26	0.23	0.02	0.81	0.34	0.21	0.39	0.32	0.59
CI_up1	0	0.84	0.26	0.23	0.01	0.83	0.32	0.22	0.26	0.25	0.39
CI_up2	0	0.84	0.23	0.22	0.01	0.80	0.28	0.20	0.13	0.32	0.13

the p-values of all the tests were greater than 0.05 which concludes that the chosen dataset and the other 100 datasets were similar with respect to mean, variance and distribution.

Only the 70 % data was used to train our VAE model and it was tested on the rest of the data that the model did not see. This helps us to identify if the model can actually perform on real data that it has never been used before. We have also used this data to generate data with other popular data augmentation techniques like ADASYN (He et al., 2008) and

SMOTE (Chawla et al., 2002). SMOTE generates minority samples along the line joining the *k* (mostly takes as 5) minority samples. On the other hand, ADASYN is a sampling approach that is focused on generating samples of data that are harder to learn. It assigns different weights to different minority samples. Higher weights mean higher difficulty in learning and therefore more data points are generated in that vicinity based on kNN. We have also used an undersampled dataset to see how sampling a little fraction of the non-crash data and all of the crash data perform to train a model. Random undersampling method was used to create the undersampled dataset. Finally, the data generated from the VAE model is also evaluated. The summary of the train dataset and test dataset are provided in Table 5. We have also included the sample size of each dataset in the table. We generated 3 sets of VAE data for three different confidence ellipsoids. The data generated from the latent space of 80 % confidence ellipsoid is referred to as VAE Data A. The data from 85 % and 90 % confidence intervals are labelled as VAE Data B and VAE Data C respectively. The number of crash samples increased from VAE Data A to C is because of the increase in volume of the confidence ellipsoids.

5.3.2. Evaluation procedure

To evaluate whether the generated data performs as expected, we propose the methodology in Fig. 9. The real data from the MVDS detectors are firstly divided into train and test sets. Only the trained data is used thereafter, and test data is left for final evaluation.

The train data is processed into five different datasets. "Real Data" is exactly the train data and is used as a reference. "Undersampled Data" contains all the crash data in the train data and a sample of the non-crash data. It is perfectly balanced meaning that the number of crash and non-crash samples are exactly equal. "ADASYN Data" also contains a balanced dataset based on the work by Haibo et al. (2008). The minority samples of crash data are increased in this method. SMOTE is also a minority oversampling technique as proposed by Chawla et al. (2002). "SMOTE Data" contains the data from this algorithm. Finally, "VAE Data A" (also B and C) contains the samples generated by the VAE model described above and also the 437 real crash data. All these five datasets are fed into two learning algorithms namely Logistic Regression and Support Vector Machine. Finally, the model is evaluated based on three values: specificity, sensitivity and AUC (area under the ROC curve). These metrics have been used in different data augmentation techniques (Chawla et al., 2002; He et al., 2008).

Specificity is defined as the ability of a model to predict true negative values. For our case, this is the ability to predict a non-crash as a non-crash.

$$Specificity = \frac{true\ negative}{true\ negative + false\ positive}$$

Sensitivity is the ability to predict true positive values. For the case of crash data, this shows how good a model is to predict a crash as a crash.

$$Sensitivity = \frac{true\ positive}{true\ positive + false\ negative}$$

AUC or area under the receiver operating characteristics curve is used to tell how effective a model is in distinguishing between classes. In this scenario, it refers to the ability to segregate crash and non-crash events.

5.3.3. Classification results and discussion

It is also important to look at the confusion matrix to better understand the true positives and the false positives. Confusion matrix lists all the predicted values and true values of the test dataset. Table 6 presents different confusion matrix calculated. The actual number of the classes are shown in each cell along with its percentage in italics.

The confusion matrix for the models LR, SVM and ANN are shown along with the different confidence ellipsoids. True labels presented along the rows are the labels obtained from the test data while predicted labels along the columns show the model predictions. Let us refer to the values "True = YES" and "Predicted = YES" as True Positives (TP) and "True = NO", "Predicted = NO" as True Negatives (TN). "True = NO", "Predicted = YES" is referred to as False Positives (FP) and the inverse as False Negatives (FN). As we increased the confidence level, it was seen that the TP values steadily increased. Out of the 188 crashes in the test dataset, the LR model correctly classified 139, 150 and 154 crashes along the different confidence levels while the SVM model correctly classified 163, 165 and 169, respectively. The ANN model peaked at classifying 176 crashes.

Also, TN values decreased as the confidence level was increased. This means that even though the models have predicted crashes more accurately, it sometimes classified some non-crash event as a crash. This is the expected trend since increasing the confidence ellipsoid, increases the size of the ellipse and therefore it captures not only more of the crash data but also non-crash data that may be located between the newly captured crash events. We would also like to present our argument in favor of the fact that, these newly captured non-crash data that are predicted as crash lies in the grey boundary between the crash and non-crash data. These may or may not result in a crash owing to a number of other factors like driver skill or safer vehicles, but it is safe to consider that these events have a good chance of resulting in a crash. Therefore, the increase of FP values can also be an indicator that the models are performing towards our expectations. Based on this reasoning, we can argue that the 90 % confidence ellipsoid should be selected as the method going forward. The best result for LR is marked green, SVM in blue and ANN in yellow in Table 6.

Table 7 shows the comparison of the confusion matrix of VAE with SMOTE and ADASYN. From this table it can be seen that the maximum number of crashes correctly classified is 176 for the ANN model. While the performance of the LR model is better for SMOTE and ADASYN, for SVM and ANN, the performance of VAE data is better than the two minority oversampling methods.

We also compared VAE model to the results obtained from the work of Qing et al. (2020). The authors reported similar performance metrics as we have used in our study. The data used by them was also for SR 408 for the year 2017. The comparison on the test data is shown in Fig. 10. In this case as well, we observe that the specificity, sensitivity and AUC improvement is remarkable for both the VAE models. The improvement in specificity is 8% for the LR model and 4% for the SVM model. The

Table 5
Datasets used for evaluation.

Train Dataset			Test Dataset		
Description	# of crash samples	# of non-crash samples	Description	# of crash samples	# of non-crash samples
Real Data (70 % of the MVDS Data)	437	4,724,612			
Undersampled Data	437	437			
ADASYN Data	4,724,612	4,724,612			
SMOTE Data	4,724,612	4,724,612	Real Data (30 % of the MVDS Data)	188	2,024,835
VAE Data A	5,583,020	4,724,612			
VAE Data B	6,738,161	4,724,612			
VAE Data C	8,336,539	4,724,612			

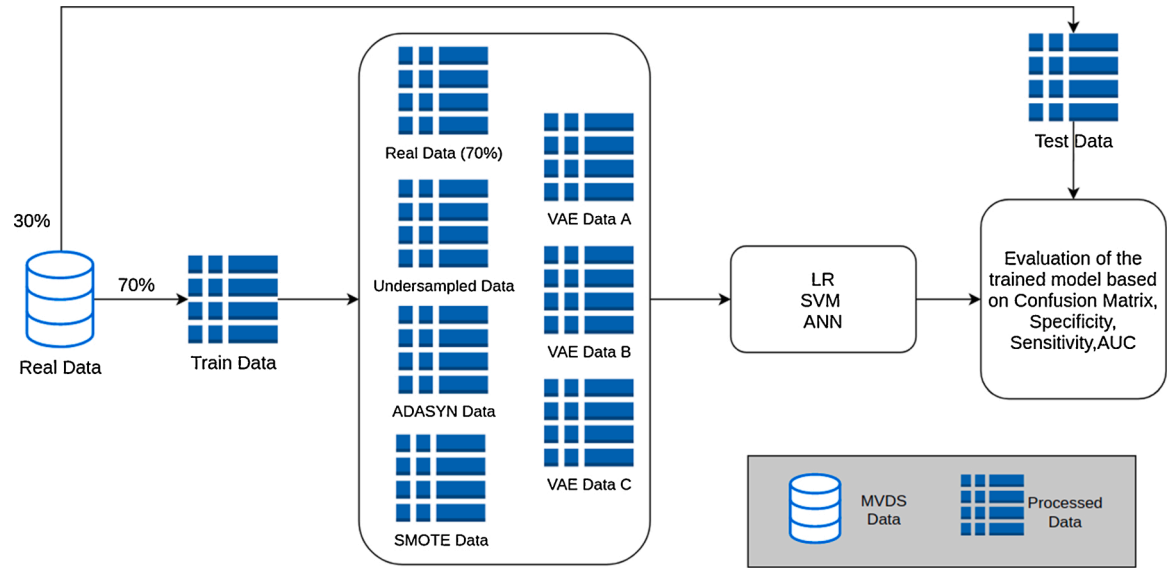

Fig. 9. Flowchart for the evaluation of generated data.

Table 6
Confusion Matrix of the models with different confidence ellipsoid.

	80% Confidence Ellipsoid (VAE Data A)		85% Confidence Ellipsoid (VAE Data B)		90% Confidence Ellipsoid (VAE Data C)		
	Predicted = NO	Predicted= YES	Predicted= NO	Predicted= YES	Predicted= NO	Predicted= YES	
True = NO	1877900 *0.93*	146935 *0.07*	1816510 *0.90*	208325 *0.10*	1776680 *0.88*	248155 *0.12*	LR
True = YES	49 *0.26*	139 *0.74*	38 *0.2*	150 *0.8*	34 0.18	154 *0.82*	
True = NO	1792742 *0.89*	232093 *0.11*	1762788 *0.87*	262047 *0.13*	1693220 *0.84*	331615 *0.16*	SVM
True = YES	25 *0.13*	163 *0.87*	23 *0.12*	165 *0.88*	19 *0.10*	169 *0.90*	
True = NO	1680575 *0.88*	344260 *0.12*	1654439 *0.81*	370396 *0.19*	1601701 *0.79*	423134 *0.21*	ANN
True = YES	16 *0.08*	172 *0.92*	13 *0.07*	175 *0.93*	12 *0.07*	176 *0.93*	

improvement of sensitivity is also increased by 6%, 5% and 4% for LR, SVM and ANN respectively. The AUC is improved by 7% and 1%, respectively for LR and SVM. Therefore, we can conclude that the VAE performs better for the LR and SVM models than the DCGAN method. The AUC of the DCGAN-ANN method is 1% better than VAE-ANN but the sensitivity for the ANN-VAE is better by 4%.

We also compared the results with other data sampling methods. The results are shown in Fig. 11. The real dataset is also shown for reference. For this dataset, we see that the model predicts everything as non-crash, since the specificity is 1 and sensitivity is zero. The undersampled dataset also performs well but the specificity is lower than the other techniques since it was only trained with a handful of non-crash data and therefore cannot recognize properly some of the non-crash events in the test data. ADASYN and SMOTE are also well recognized minority sampling techniques and therefore perform better than random undersampling. The performance of the VAE data is shown toward the right end of Fig. 11.

The VAE model has improved specificity for the LR model (about 2% increase) than that of both ADASYN and SMOTE while the sensitivity is lower by 4%. For the SVM model, VAE generated data has better

Table 7
Confusion Matrix of the models with VAE, SMOTE and ADASYN.

	SMOTE		ADASYN		90% Confidence Ellipsoid (VAE Data C)		
	Predicted = NO	Predicted= YES	Predicted= NO	Predicted= YES	Predicted= NO	Predicted= YES	
True = NO	1744736 *0.86*	280099 *0.14*	1748582 *0.86*	276253 *0.14*	1776680 *0.88*	248155 *0.12*	LR
True = YES	24 *0.13*	164 *0.87*	24 *0.13*	164 *0.87*	34 0.18	154 *0.82*	
True = NO	1773173 *0.88*	251662 *0.12*	1762788 *0.86*	262047 *0.14*	1693220 *0.84*	331615 *0.16*	SVM
True = YES	24 *0.13*	164 *0.87*	22 *0.12*	166 *0.88*	19 *0.10*	169 *0.90*	
True = NO	1739254 *0.86*	285581 *0.14*	1709736 *0.84*	315099 *0.16*	1601701 *0.79*	423134 *0.21*	ANN
True = YES	20 *0.10*	168 *0.9*	18 *0.10*	170 *0.9*	12 *0.07*	176 *0.93*	

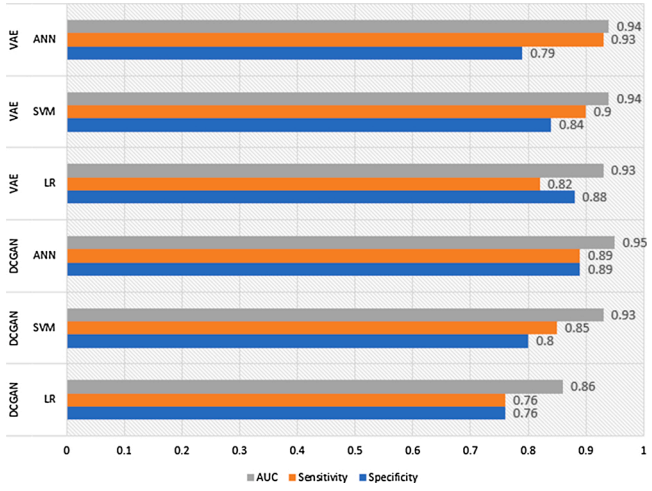

Fig. 10. Comparison between DCGAN and VAE as a data augmentation method.

sensitivity. The specificity is comparable to ADASYN and SMOTE while the sensitivity is improved by 3%. The AUC (for SVM) of the VAE generated data is also better than both ADASYN and SMOTE by 1%. The reason for this variation can be attributed to the selection of the confidence ellipsoids. A higher confidence level would mean better crash prediction, but also higher false alarms and lower confidence interval would lower the crash prediction and also lower false alarms. We are, therefore, convinced that the confidence level of the ellipsoid is an important hyperparameter to be tuned. From a crash prediction point of view, it is safe to have a higher confidence level since the false alarms would be a type of indicator that traffic conditions are likely to result in a crash. We also propose that, sensitivity is a better measure of the performance of generated crash data owing to this intuition. The SVM model outperforms the minority oversampling techniques with respect to sensitivity. For the ANN model, the sensitivity value peaks for all the models that have been trained. This model successfully classifies 93 % of the crashes.

6. Conclusions

To predict crashes in real-time is of utmost importance in safety. It is also important to mark certain areas as unsafe where there are frequent crashes so that authorities can take long term measures to better the location. Due to insufficient data, the analysis of such metrics does not always provide conclusive results. Variational autoencoder can help leverage this by producing synthetic data. Given a small bunch of sample data, it can, theoretically, produce infinite data samples.

To summarize, the work in this paper we have generated over 20 million crash data with only a handful of 437 crash samples. We also used the concept of confidence ellipsoid to accurately capture crash data from the latent space of a VAE. We then compared the generated data with the real data from statistical standpoints. The mean, standard deviation of the two datasets were similar. While it is known that autoencoders tend to reduce the noise in the data and thus the generated data could lose some important feature in the process, the data distribution of each of the 24 variables (in Fig. 8) shows that the synthetic dataset closely follows the distribution of the real data. Finally, we evaluated the performance of the dataset with three crash prediction models: LR, SVM and ANN. The metrics chosen for evaluation was specificity, sensitivity and AUC. The real crash data was also augmented with state-of-the-art minority sampling techniques like SMOTE and ADASYN. These datasets were also used on LR, SVM and ANN to obtain the evaluation parameters. The results of the VAE model showed

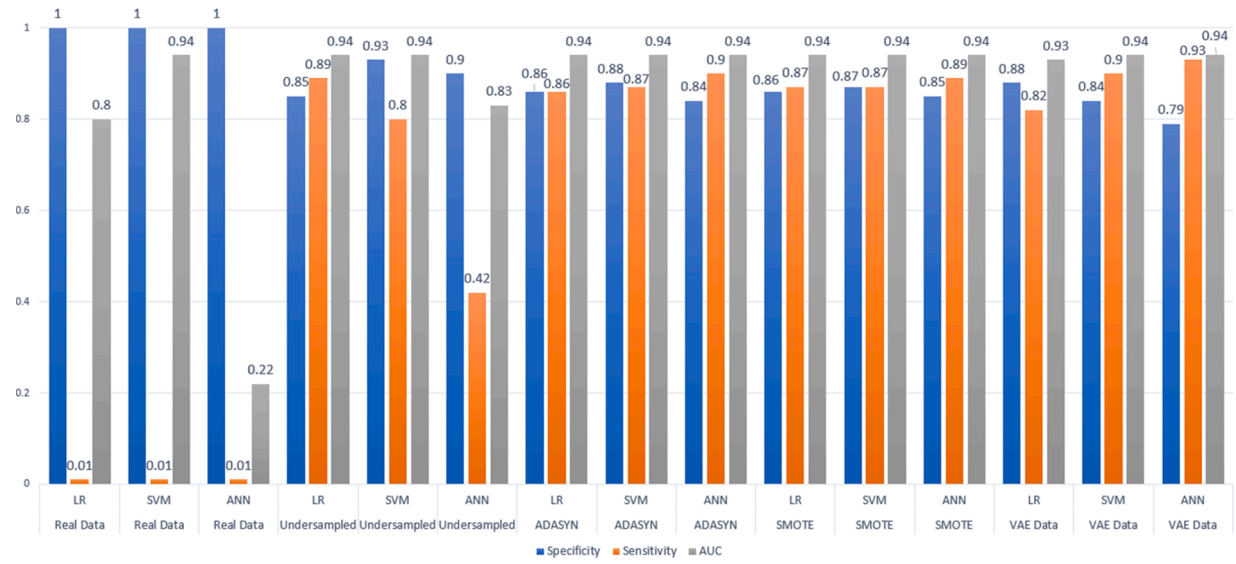

Fig. 11. Comparison of the evaluation metrics across various models and data augmentation techniques.

improvements from the view of specificity, sensitivity and AUC. The results were also compared with data augmentation technique involving DCGAN. Our VAE model outperformed most studies in the literature. The results from the confusion matrix, that was used to compare across different confidence ellipsoids, was also very insightful. It indicated that increasing the confidence ellipsoid increases the chances of getting false alarms but that is also an indicator that conditions are favorable for a crash. Overall, the results were encouraging, and can aid to balance an imbalanced crash dataset.

Future studies can train crash prediction models on more complex and non-linear algorithms that do not perform well on small datasets. Using VAE as a data generation tool in the pipeline would definitely aid in generating substantial data to train on non-linear models. Furthermore, there could be more work relating VAE that could have one more class in between crash and non-crash: crash-prone. The training data of this region could be derived from the false positives from our VAE model.

Author contributions

The authors confirm contribution to the paper as follows: study conception and design: Zubayer Islam, Mohamed Abdel-Aty, Qing Cai; data collection: Qing Cai, Jinghui Yuan, Zubayer Islam; analysis and interpretation of results: Zubayer Islam, Mohamed Abdel-Aty; draft manuscript preparation: Zubayer Islam, Mohamed Abdel-Aty. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Competing Interest

The authors report no declarations of interest.

Appendix A. Supplementary data

Supplementary material related to this article can be found, in the online version, at doi:https://doi.org/10.1016/j.aap.2020.105950.

References

Abdel-Aty, M., Uddin, N., Pande, A., Abdalla, M.F., Hsia, L., 2004. Predicting freeway crashes from loop detector data by matched case-control logistic regression. Transp. Res. Rec. 1897 (1), 88–95.

Abou Elassad, Z.E., Mousannif, H., Al Moatassime, H., 2020a. A proactive decision support system for predicting traffic crash events: a critical analysis of imbalanced class distribution. Knowl. Based Syst. 205, 106314.

Abou Elassad, Z.E., Mousannif, H., Al Moatassime, H., 2020b. A real-time crash prediction fusion framework: an imbalance-aware strategy for collision avoidance systems. Transp. Res. Part C Emerg. Technol. 118, 102708.

Ahmed, M.M., Abdel-Aty, M.A., 2011. The viability of using automatic vehicle identification data for real-time crash prediction. IEEE Trans. Intell. Transp. Syst. 13 (2), 459–468.

An, J., Cho, S., 2015. Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE 2 (1), 1–18.

Bao, J., Liu, P., Ukkusuri, S.V., 2019. A spatiotemporal deep learning approach for citywide short-term crash risk prediction with multi-source data. Accid. Anal. Prev. 122, 239–254.

Basso, F., Basso, L.J., Bravo, F., Pezoa, R., 2018. Real-time crash prediction in an urban expressway using disaggregated data. Transp. Res. Part C Emerg. Technol. 86, 202–219.

Basso, F., Basso, L.J., Pezoa, R., 2020. The importance of flow composition in real-time crash prediction. Accid. Anal. Prev. 137, 105436.

Boquet, G., Vicario, J.L., Morell, A., Serrano, J., 2019. Missing data in traffic estimation: a variational autoencoder imputation method. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2882–2886.

Boquet, G., Morell, A., Serrano, J., Vicario, J.L., 2020. A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection. Transp. Res. Part C Emerg. Technol. 115, 102622.

Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C., 2009. safe-level-smote: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. Pacific-Asia Conference on Knowledge Discovery and Data Mining 475–482.

Cai, Q., Abdel-Aty, M., Yuan, J., Lee, J., Wu, Y., 2020. Real-time crash prediction on expressways using deep generative models. Transp. Res. Part C Emerg. Technol. 117, 102697.

Catal, C., Diri, B., 2009. Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem. Inf. Sci. 179 (8), 1040–1058.

Central Florida Expressway Authority Webpage. (n.d.). Retrieved July 18, 2020, from https://www.cfxway.com/for-travelers/expressways/408/.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.

Elamrani Abou Elassad, Z., Mousannif, H., Al Moatassime, H., 2020. Class-imbalanced crash prediction based on real-time traffic and weather data: a driving simulator study. Traffic Inj. Prev. 21 (3), 201–208.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H., 2018. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing 321, 321–331.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics 249–256.

He, H., Bai, Y., Garcia, E.A., Li, S., 2008. ADASYN: adaptive synthetic sampling approach for imbalanced learning. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) 1322–1328.

He, Y., Wu, D., Beyazit, E., Sun, X., Wu, X., 2018. Supervised data synthesizing and evolving–a framework for real-world traffic crash severity classification. 2018 IEEE 30th International Conference on Tools With Artificial Intelligence (ICTAI) 163–170.

Huang, D., Song, X., Fan, Z., Jiang, R., Shibasaki, R., Zhang, Y., Wang, H., Kato, Y., 2019. A variational autoencoder based generative model of urban human mobility. 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) 425–430.

Huang, T., Wang, S., Sharma, A., 2020. Highway crash detection and risk estimation using deep learning. Accid. Anal. Prev. 135, 105392.

Ke, J., Zhang, S., Yang, H., Chen, X., 2019. PCA-based missing information imputation for real-time crash likelihood prediction under imbalanced data. Transp. A Transp. Sci. 15 (2), 872–895.

Kingma, D.P., Welling, M., 2014. Auto-encoding variational bayes. In: 2nd International Conference on Learning Representations. ICLR 2014 - Conference Track Proceedings, Ml, pp. 1–14.

Kusner, M.J., Paige, B., Hernández-Lobato, J.M., 2017. Grammar variational autoencoder. Proceedings of the 34th International Conference on Machine Learning Volume 70, 1945–1954.

Lee, C., Hellinga, B., Saccomanno, F., 2003. Real-time crash prediction model for application to crash prevention in freeway traffic. Transp. Res. Rec. 1840 (1), 67–77.

Li, D., Zhu, Y., Lin, W., 2017. Traffic identification of mobile apps based on variational autoencoder network. 2017 13th International Conference on Computational Intelligence and Security (CIS) 287–291.

Li, P., Abdel-Aty, M., Cai, Q., Yuan, C., 2020a. ⋆ This paper has been handled by associate editor Tony Sze. The application of novel connected vehicles emulated data on real-time crash potential prediction for arterials. Accid. Anal. Prev. 144, 105658.

Li, P., Abdel-Aty, M., Yuan, J., 2020b. Real-time crash risk prediction on arterials based on LSTM-CNN. Accid. Anal. Prev. 135, 105371.

Lin, L., Wang, Q., Sadek, A.W., 2015. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. Transp. Res. Part C Emerg. Technol. 55, 444–459.

Maciejewski, T., Stefanowski, J., 2011. Local neighbourhood extension of SMOTE for mining imbalanced data. 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM) 104–111.

NSC and NHTSA Report. (n.d.). Retrieved December 5, 2020, from https://injuryfacts.nsc.org/motor-vehicle/overview/comparison-of-nsc-and-nhtsa-estimates/.

Parsa, A.B., Taghipour, H., Derrible, S., Mohammadian, A.K., 2019. Real-time accident detection: coping with imbalanced data. Accid. Anal. Prev. 129, 202–210.

Peng, Y., Li, C., Wang, K., Gao, Z., Yu, R., 2020. Examining imbalanced classification algorithms in predicting real-time traffic crash risk. Accid. Anal. Prev. 144, 105610.

Perez, L., Wang, J., 2017. The effectiveness of data augmentation in image classification using deep learning. ArXiv Preprint. ArXiv:1712.04621.

Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., Carin, L., 2016. Variational autoencoder for deep learning of images, labels and captions. Adv. Neural Inf. Process. Syst. 2352–2360.

Sáez, J.A., Luengo, J., Stefanowski, J., Herrera, F., 2015. SMOTE–IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Inf. Sci. 291, 184–203.

Shi, Q., Abdel-Aty, M., 2015. Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways. Transp. Res. Part C Emerg. Technol. 58, 380–394.

Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O., 2016. Ladder variational autoencoders. Adv. Neural Inf. Process. Syst. 3738–3746.

Sun, S., Zhou, B., Zhang, S., 2020. Analysis of factors affecting injury severity in motorcycle involved crashes. CICTP 2020, pp. 4207–4219.

Wang, L., Abdel-Aty, M., Shi, Q., Park, J., 2015. Real-time crash prediction for expressway weaving segments. Transp. Res. Part C Emerg. Technol. 61, 1–10.

Wang, Jiwei, Chen, Y., Gu, Y., Xiao, Y., Pan, H., 2018. SensoryGANs: an effective generative adversarial framework for sensor-based human activity recognition. Proceedings of the International Joint Conference on Neural Networks, 2018-July 1–8. https://doi.org/10.1109/IJCNN.2018.8489106.

Wang, Junhua, Kong, Y., Fu, T., 2019a. Expressway crash risk prediction using back propagation neural network: a brief investigation on safety resilience. Accid. Anal. Prev. 124, 180–192.

Wang, L., Abdel-Aty, M., Lee, J., Shi, Q., 2019b. Analysis of real-time crash risk for expressway ramps using traffic, geometric, trip generation, and socio-demographic predictors. Accid. Anal. Prev. 122, 378–384.

Xu, C., Liu, P., Wang, W., Li, Z., 2012. Evaluation of the impacts of traffic states on crash risks on freeways. Accid. Anal. Prev. 47, 162–171.

Xu, C., Tarko, A.P., Wang, W., Liu, P., 2013. Predicting crash likelihood and severity on freeways with real-time loop detector data. Accid. Anal. Prev. 57, 30–39.

Xu, W., Sun, H., Deng, C., Tan, Y., 2017. Variational autoencoder for semi-supervised text classification. Thirty-First AAAI Conference on Artificial Intelligence.

Yahaya, M., Jiang, X., Fu, C., Bashir, K., Fan, W., 2019. Enhancing crash injury severity prediction on imbalanced crash data by sampling technique with variable selection. 2019 IEEE Intelligent Transportation Systems Conference (ITSC) 363–368.

Yin, Y., Huang, Y., Zhang, L., Gao, Z., 2019. Influence of different sampling techniques on the real-time crash risk prediction model. 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA) 1795–1799.

You, J., Wang, J., Fang, S., Guo, J., 2017. An optimized real-time crash prediction model on freeway with over-sampling techniques based on support vector machine. J. Intell. Fuzzy Syst. 33 (1), 555–562.

Yu, R., Abdel-Aty, M., 2013. Utilizing support vector machine in real-time crash risk evaluation. Accid. Anal. Prev. 51, 252–259.

Yu, R., Wang, X., Yang, K., Abdel-Aty, M., 2016. Crash risk analysis for Shanghai urban expressways: a Bayesian semi-parametric modeling approach. Accid. Anal. Prev. 95, 495–502.

Yu, R., Quddus, M., Wang, X., Yang, K., 2018. Impact of data aggregation approaches on the relationships between operating speed and traffic safety. Accid. Anal. Prev. 120, 304–310.

Yuan, J., Abdel-Aty, M., Gong, Y., Cai, Q., 2019. Real-time crash risk prediction using long short-term memory recurrent neural network. Transp. Res. Rec. 2673 (4), 314–326.

Zheng, Z., Ahn, S., Monsere, C.M., 2010. Impact of traffic oscillations on freeway crash occurrences. Accid. Anal. Prev. 42 (2), 626–636.

Zhou, B., Zhang, X., Zhang, S., Li, Z., Liu, X., 2019. Analysis of factors affecting real-time ridesharing vehicle crash severity. Sustainability 11 (12), 3334.