Experiments on traffic data

Amartya Das Sharma

June 2019

1 Introduction

The data used is from the public Kaggle dataset 1.6 million UK traffic accidents retrieved on 27th April. This report briefly explains the data components, the goal of the experiments done, the deep learning methods used and the reasons for their inclusion, and results of the experiments.

2 Data Contents

According to the Kaggle homepage, the dataset contains traffic data collected by the UK government over the period of 2005 - 2014 (excluding 2008). The datasets of interest are compiled summaries of police reports of traffic accidents over this time period. The information that the dataset contains for each report (i.e. the columns of the dataset) are the following:

Accident Index A unique ID given to the accident.

Easting and Northing Local British coordinates of the accident location.

Longitude and Latitude Global latitude and longitude of the accident location.

Police Force Denotes which police force attended the scene of the accident by unique ID. For example, 1 denotes metropolitan police, while 5 denotes the force at Merseyside. There are 51 unique codes for this field.

Accident Severity How severe the accident was. 1 stands for fatal accident, 2 for severe, 3 for slight.

Number of Vehicles Number of vehicles involved in the accident.

Number of Casualties Number of casualties in the accident.

Date Date the accident occurred in dd/mm/yyyy format.

Day of Week The day of the week. 1 stands for Sunday, 2 for Monday, until 7 for Saturday.

Time Time of the accident in 24h format.

Local Authority (District) Unique ID for the district. For example, 204 denotes Leeds, 350 denotes Boston. There are 416 unique codes for this field.

Local Authority (Highway) Unique ID for the highway authority. For example, E08000025 denotes Birmingham. There are 208 unique values in this filed.

1st Road Class, 1st Road Number, 2nd Road Class, 2nd Road Number Values used only for junctions, denoting class and number of connecting roads.

Road Type 7 possible road types shown by ID. For example, 1 denotes roundabouts, 3 denotes dual carriageway, 2 denotes a one way street. -1 denotes data out of range or missing.

Speed limit Speed limit of the accident zone.

Junction Detail Type of junction. 9 total unique IDs denote 9 types of junctions. For example, 0 denotes that there is no junction within 20 meters, 1 denotes a roundabout and 2 denotes mini roundabouts.

Junction Control How the junction is controlled. 5 total unique IDs denote 5 different control mechanisms. For example, 2 denotes auto traffic signal, 3 denotes stop sign.

Pedestrian Crossing-Human Control Whether there is no pedestrian crossing within 50 meters (0), whether the crossing is controlled by school crossing patrol(1) or by an authorized person (2).

Pedestrian Crossing-Physical Facilities 6 unique IDs denoting the type of crossing. 0 denotes absence of physical crossing facilities within 50 meters. 1 denotes zebra crossing, 7 denotes footbridge or subway.

Light Conditions The condition of lighting. Values include daylight, darkness with street lights on, darkness with street lights available but unlit, darkness with no street lights etc.

Weather Conditions Values include fair with/without high winds, rainy with/without high winds, snowing with/without high winds, fog or mist, or other.

Road Surface Conditions Values include dry, wet or damp, snow, frost or ice etc.

Special Conditions at Site Whether there were any abnormal conditions at site, such as road works underway, road surface defective etc.

Carriageway Hazards Hazards relating to the road, such as whether there are other objects, whether there was a previous accident wtc.

Urban or Rural Area In this field, 1 denotes an urban area, and 2 denotes rural.

Did Police Officer Attend Scene of Accident Field contains yes, no and no (self reported accident via form).

LSOA of Accident Location LSOA areas are geographic areas uniquely used by England and Wales for reporting smaller area statistics. This field gives the area of the accident.

Year The year of the accident.

3 Experimental Setup

For the following experiments, the data field of interest was chosen to be the *Accident Severity* field. The goal was to see how other environmental factors affect the severity of an accident. However the dataset is very disbalanced, with 1,280,205 accidents recorded being slight accidents, 204,504 being severe accidents and only 19441 being fatal accidents. This means that traditional machine learning techniques have a lower chance of success due to overfitting.

Hence, machine learning techniques specific to anomaly detection were used in the following experiments, because they expect the anomaly class to have very few examples. The classes *slight accidents* and *severe accidents* were combined, and *fatal accidents* were treated as outliers. The techniques used were *deep autoencoder networks*, *one class SVMs* and *generative adversarial networks (GANs)*. These methods are explained in their respective sections.

Irrelevant classes in the dataset were dropped and the remaining dataset was sorted, normalized and cleaned of null values. The resulting dataset had 828,817 rows, with 822,426 accidents being nonfatal and 6391 accidents being fatal.

4 Deep Autoencoders

4.1 Introduction

In basic terms, an autoencoder is a neural network with as many output neurons as input neurons, with fewer hidden layer neurons than either layer. The output of an autoencoder is expected to be identical to the input, and it is trained to achieve such an effect. It can be thought of to comprise of two distinct halves, an encoder and a decoder. The encoder's job is to reduce the data to some representation with lower dimensionality, and the decoder reconstructs the original data from this representation.

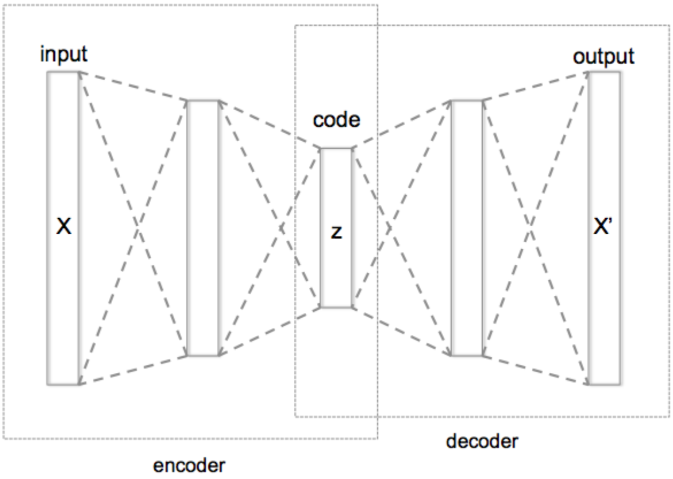


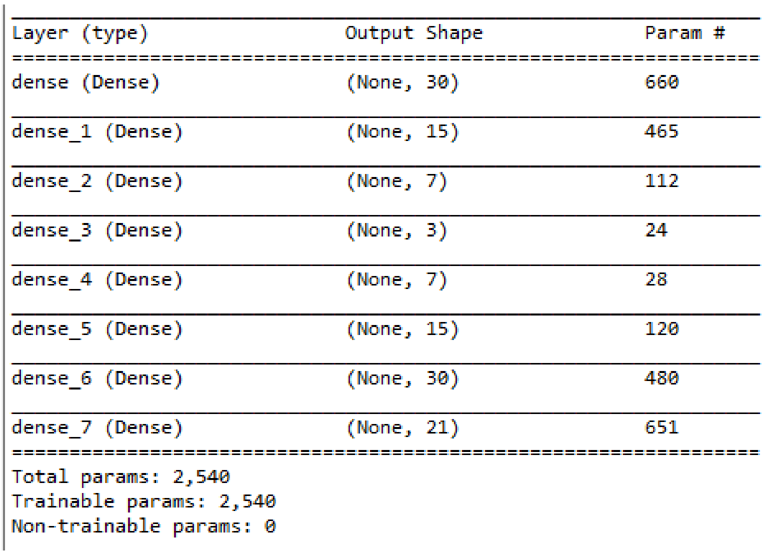
Figure 1: By Chervinskii - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45555552

If an autoencoder performs well, then it must mean that the code representing the data, while being compressed, captures the data accurately. Hence we can say that the network has learnt the data well. After training, the network becomes highly tuned to the data given, and while it can reconstruct data from the same class, it will not be able to reconstruct data from a different class, since it is not tuned to represent the relationships between data values of that class. This fact is used in anomaly detection, where anomalies have more reconstruction error (mean squared difference of prediction from actual data) than the members of the majority class.

4.2 Experiment details

The data was divided into training and testing sets, in an 80:20 split. The training data only contained information about nonfatal accidents, and the testing data contained instances of both fatal and nonfatal accidents. The understanding is that, if the network is trained exclusively on nonfatal accidents and is then made to reconstruct fatal accidents, it should make more errors doing so if fatal accidents are truly anomalous.

The autoencoder contains 9 layers, with architecture shown in Figure 2 (the input layer with 21 neurons is not shown but exists before the first dense layer): The autoencoder was trained for 200 epochs. The intention is to overfit the network on nonfatal accidents so that it cannot reconstruct fatal accidents (or reconstruct them with higher error than nonfatal accidents).



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 30)                660
_________________________________________________________________
dense_1 (Dense)              (None, 15)                465
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 112
_________________________________________________________________
dense_3 (Dense)              (None, 3)                 24
_________________________________________________________________
dense_4 (Dense)              (None, 7)                 28
_________________________________________________________________
dense_5 (Dense)              (None, 15)                120
_________________________________________________________________
dense_6 (Dense)              (None, 30)                480
_________________________________________________________________
dense_7 (Dense)              (None, 21)                651
=================================================================
Total params: 2,540
Trainable params: 2,540
Non-trainable params: 0
```

Figure 2: Structure of the autoencoder

4.3 Results

The model converged with a loss of 0.0092 (mean squared error loss). However, when the network was tested, it could reconstruct fatal accidents as well as it could reconstruct nonfatal accidents.

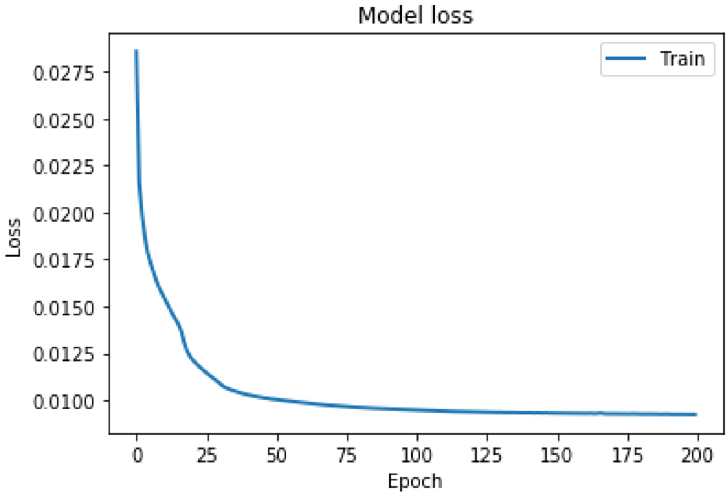


Figure 3: Training loss graph of the autoencoder

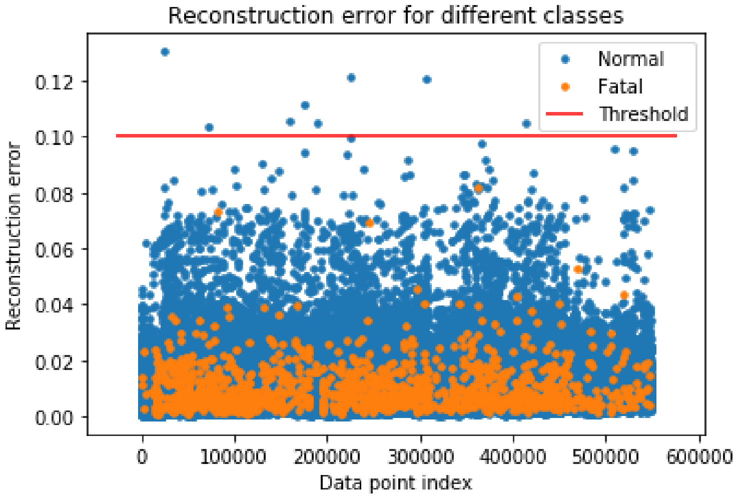


Figure 4: Reconstruction of testing data

The network had never seen fatal accidents before, but it could reconstruct the data despite that. Hence the representation that the network developed for nonfatal accidents works for fatal accidents too, despite overfitting. Therefore, fatal accidents, according to autoencoders, are not anomalies, based on only environmental data.

5 One-class SVM

5.1 Introduction

To use SVMs for anomaly detection, one-class SVMs are often used. Here, data from only one class is used, and an SVM boundary is fit around this data. The understanding is that members of other classes (anomalies) would fall outside this boundary.

5.2 Experiment details

A subset of the data was used for training. 700 fatal accidents and 40000 nonfatal accidents were randomly sampled and divided in an 80:20 split for training and testing. A radial basis function was used as kernel.

5.3 Results

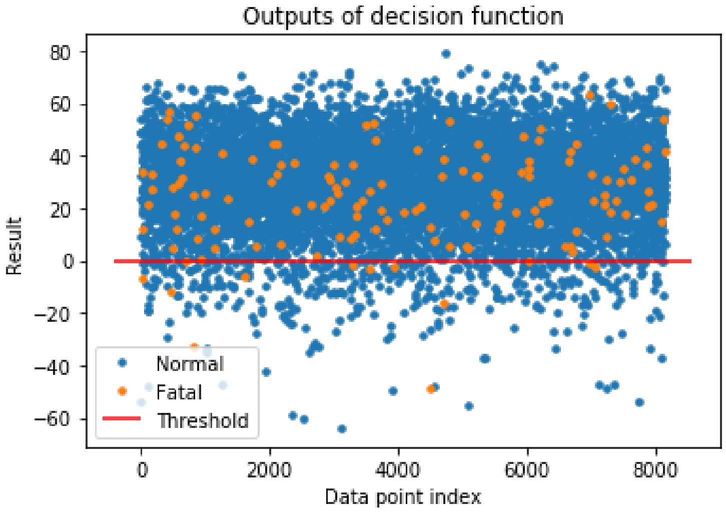


Figure 5: Result of the decision function on testing data

As shown in Figure 5, the one class SVM classifies most fatal accidents as nonfatal i.e. it does not treat them as anomalies.

6 Generative Adversarial Networks

6.1 Introduction

Generative adversarial networks (GANs), while usually used to generate realistic example of given data, can also be used for anomaly detection.

GANs consist of two parts: a *generator* and a *discriminator*. The discriminator judges whether data belongs to a certain class or not, while the generator takes random noise as input and tries to generate data that can fool the discriminator. Both neural networks try and work against each other, improving both their performance over time.The discriminator learns to generalize the data better, and the generator learns how to make its generated data resemble actual data more and more.

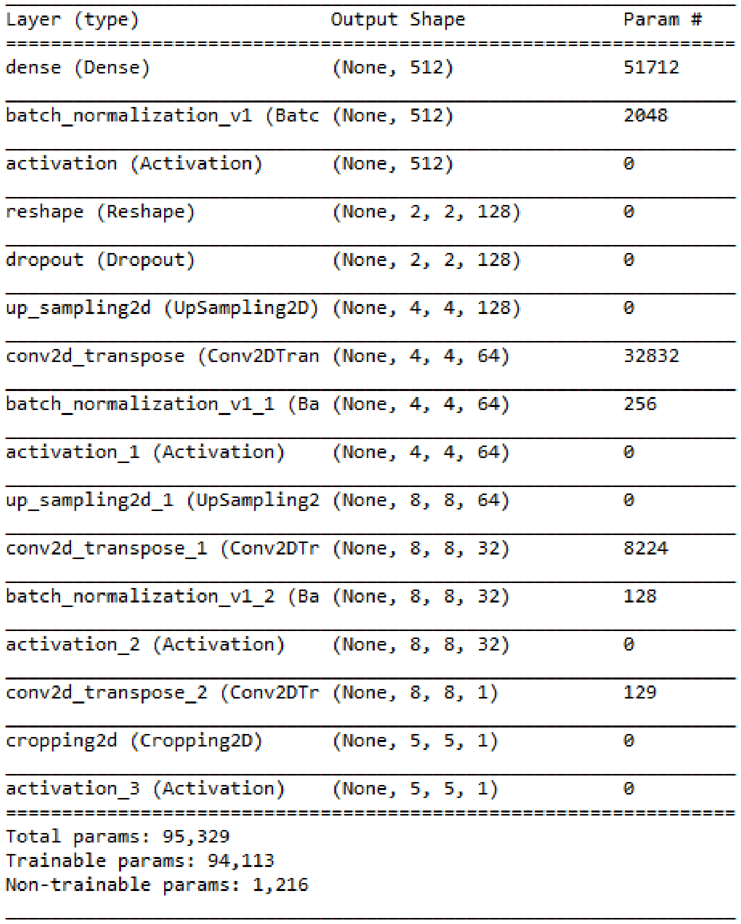
In this experiment, the discriminator is used for anomaly detection. After thorough training with data (from nonfatal accidents) and random noise, the discriminator should classify fatal accidents as noise, more often that it would classify it as legitimate data (nonfatal accident data).

6.2 Experiment details

Again, the data was split in an 80:20 ratio. Fatal accidents were removed from the training data and the residual data was used.

The architecture for the generative network can be seen in Figure 6 and the architecture for the discriminator in Figure 7. The two are then combined into one network, shown in 8.

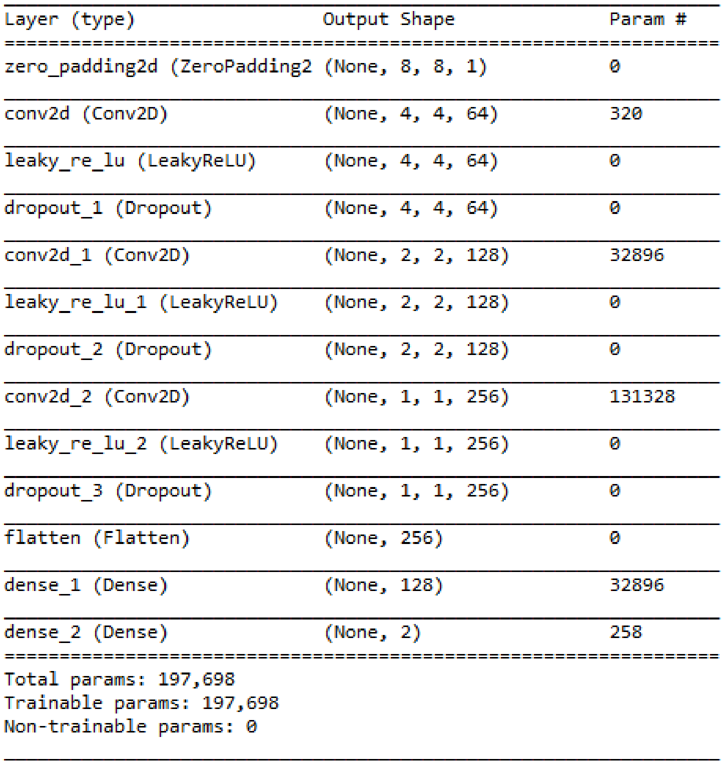
Each row of data contains 21 data points. This data was converted to a 5x5 array, adding extra zeroes so that a square shape was obtained. This 5x5 array was treated as an image, and was fed to the discriminator (which uses convolutional neural networks) for pretraining.



```
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 512)               51712
_________________________________________________________________
batch_normalization_v1 (Batc (None, 512)               2048
_________________________________________________________________
activation (Activation)      (None, 512)               0
_________________________________________________________________
reshape (Reshape)            (None, 2, 2, 128)         0
_________________________________________________________________
dropout (Dropout)            (None, 2, 2, 128)         0
_________________________________________________________________
up_sampling2d (UpSampling2D) (None, 4, 4, 128)         0
_________________________________________________________________
conv2d_transpose (Conv2DTran (None, 4, 4, 64)          32832
_________________________________________________________________
batch_normalization_v1_1 (Ba (None, 4, 4, 64)          256
_________________________________________________________________
activation_1 (Activation)    (None, 4, 4, 64)          0
_________________________________________________________________
up_sampling2d_1 (UpSampling2 (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr (None, 8, 8, 32)          8224
_________________________________________________________________
batch_normalization_v1_2 (Ba (None, 8, 8, 32)          128
_________________________________________________________________
activation_2 (Activation)    (None, 8, 8, 32)          0
_________________________________________________________________
conv2d_transpose_2 (Conv2DTr (None, 8, 8, 1)           129
_________________________________________________________________
cropping2d (Cropping2D)      (None, 5, 5, 1)           0
_________________________________________________________________
activation_3 (Activation)    (None, 5, 5, 1)           0
=================================================================
Total params: 95,329
Trainable params: 94,113
Non-trainable params: 1,216
_________________________________________________________________
```

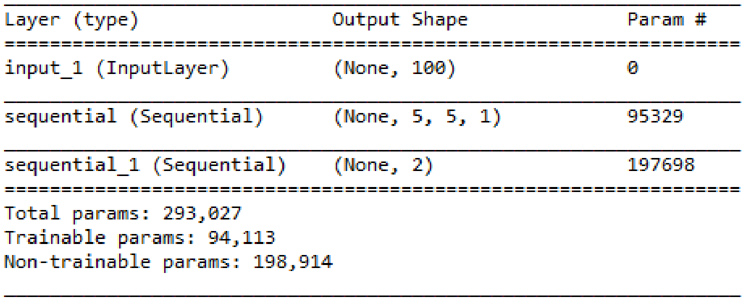
Figure 6: Generator network

After pretraining, the two networks are trained together. The generator takes 100 random bits, and converts them into a 5x5 image. The discriminator then classifies the 5x5 images as valid data or noise. This is done for 8000 epochs.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
zero_padding2d (ZeroPadding2 (None, 8, 8, 1)           0
_________________________________________________________________
conv2d (Conv2D)              (None, 4, 4, 64)          320
_________________________________________________________________
leaky_re_lu (LeakyReLU)      (None, 4, 4, 64)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 4, 4, 64)          0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 2, 2, 128)         32896
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 2, 2, 128)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 2, 2, 128)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 1, 1, 256)         131328
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 1, 1, 256)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 1, 1, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 258
=================================================================
Total params: 197,698
Trainable params: 197,698
Non-trainable params: 0
_________________________________________________________________
```

Figure 7: Discriminator network



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 100)               0
_________________________________________________________________
sequential (Sequential)      (None, 5, 5, 1)           95329
_________________________________________________________________
sequential_1 (Sequential)    (None, 2)                 197698
=================================================================
Total params: 293,027
Trainable params: 94,113
Non-trainable params: 198,914
_________________________________________________________________
```

Figure 8: Combined GAN network

6.3 Results

The loss graph of the GAN is shown in Figure 9.

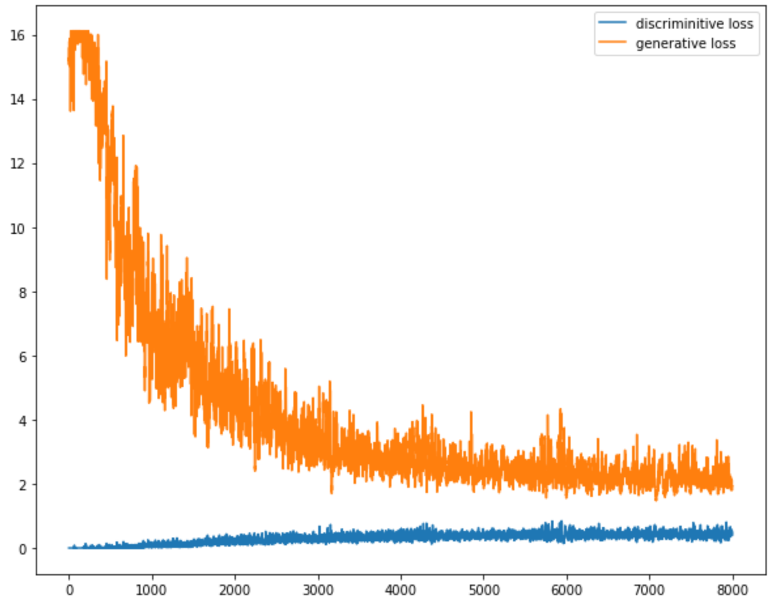


Figure 9: GAN network training loss

After training, the testing data is fed to the discriminator. The ROC graph is obtained and shown in Figure 10. For true anomaly detection, we expect to see a graph with TPR growing faster than FPR. However, we see an ROC curve characteristic of an untrained network making random guesses.

Again, we see that the model is unable to differentiate between fatal and nonfatal accidents, even though it was trained exclusively on nonfatal accidents. We can conclude that fatal accident conditions are closer to nonfatal accident conditions, than they are to random data.

7 Conclusion

From the results, the conclusion that is drawn is that fatal and nonfatal accidents occur in similar environmental conditions, and there are no conditions (or combination of conditions) that cause exclusively fatal or nonfatal accidents. Intuitively, this makes sense, because the driver of the car is a much bigger factor in determining the severity of an accident, and better drivers

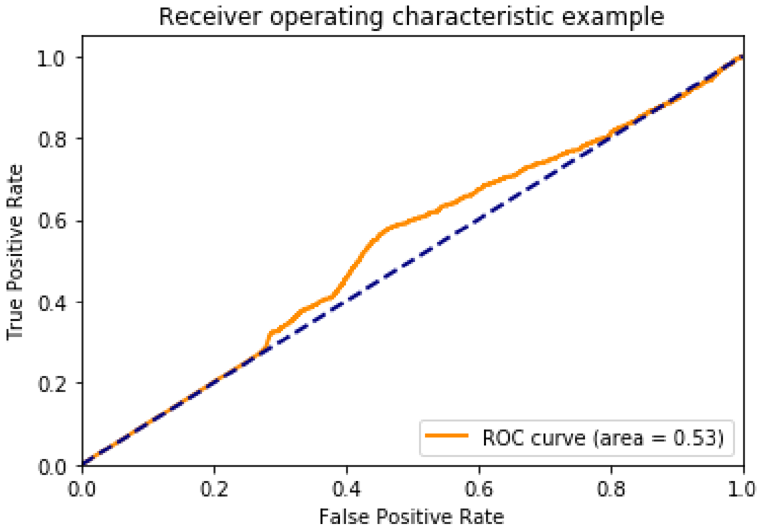


Figure 10: ROC curve of the discriminator

drive better in all conditions. However, this work only looks at 3 approaches to verify this hypothesis, and further work is desirable to derive stronger conclusions.