Real-Time Car Crash Prediction using Time Series

(Indy Car)

Akhil Nagulavancha
Intelligent Systems Engineering
Indiana University
Bloomington IN USA
aknagu@iu.edu

INTRODUCTION

IndyCar is one of the top racing car events in America. During the Period of race various events can occur like car stopping at pitstop, car crash, mechanical breakdown, driving ranking changes and many more.

The main goal of this project is to predict the event of car crash even before it happens so that the team can plan their further course of action of the car in the race. Over the progress of the project, the scope of the project was limited to detecting anomalies in the telemetry data from the Indy car time series data.

In the IndyCar Dataset there are two types of data sets, one is sequential data feed which is live data, and the other type of data is his historical or archived data.

As this task needs telemetry data of the car, we use the Log data which is received from CAN Bus Feed for the prediction of the crash event of the car. Telemetry Dataset extracted from the log files is currently being used for the implementation of the project.

Over the course of the race there can be many events which occur like car stopping, car crash, mechanical breakdown or pitstop. The goal of the project will be to detect all such events which occur during the race.

The current extracted datasets consist of telemetry, laps, cars and pitstops. The dataset which will be useful in our project will be telemetry dataset. The current telemetry dataset is limited in the set of features compared to the data available in the log file so have extracted all the required data from the Indy car race on 27th May 2018.

ARCHITECTURE AND IMPLEMENTATION

LSTM Autoencoders are used in this project for detection of anomalies by calculating the loss between the predicted and actual time series values.

We are essentially modelling the normalcy using time series data and calculating the deviation from the normalcy for predicting the time series anomalies.

Autoencoders are a combination of both encoder and a decoder. Encoders reduce the dimension of the data and decoders try to reconstruct the reduced dimension by ignoring the noise. Autoencoders are mainly used in learning efficient data encodings in an unsupervised manner. Autoencoders try to act as an identity function and try to copy the input as much as possible.

LSTM (Long Short-Term Memory) process the entire sequence of data unlike simple neural networks. Vanilla RNN's are not used due to the limitations of the vanishing gradient and exploding gradient problem.

Due to these advantages of Autoencoders and LSTM, LSTM Autoencoder is implemented in this project. It sequences data using Encoder-Decoder LSTM architecture. LSTM Autoencoder compresses into and retains the signal from latent space.

Due to the use of multivariate data, loss calculation for anomaly detection is calculated by considering the average loss of all the features of the model in the time series data.

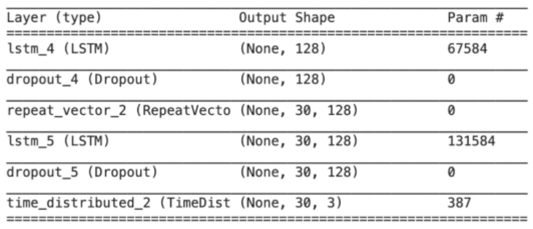
Experiments were made by extracting data from the log file of Indy car racing log file into csv which were converted into data frames used in the project.

Experiments were made on car 12 and car 9 by extracting data from log for both the cars. They were split into 80:20 ration of train and test data for both the cars.

This project is mainly implemented in TensorFlow, Keras and data manipulation is done by pandas' library.

The architecture of the LSTM Autoencoder Neural network consists of the following layers in the following order

1. LSTM
2. Dropout
3. Repeat Vector
4. LSTM
5. Dropout
6. Time Distributed



```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 128)               67584
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
repeat_vector_2 (RepeatVecto (None, 30, 128)           0
_________________________________________________________________
lstm_5 (LSTM)                (None, 30, 128)           131584
_________________________________________________________________
dropout_5 (Dropout)          (None, 30, 128)           0
_________________________________________________________________
time_distributed_2 (TimeDist (None, 30, 3)             387
=================================================================
```

Figure 1: Architecture of the Model used in the Project with respective output shapes and parameters

The characteristics of the data over the time are observed and feature engineering, data manipulation is implemented before feeding data into the model. Correlations between various features of the time series data obtained is checked for better understanding of the data, feature engineering and manipulation of the data.

EXPERIMENTS

Setting:

All the experiments of the project are conducted in jypyter lab of the feature grid virtual machines.

Language and Libraires: Python, Pandas, Keras, NumPy, TensorFlow, Matplotlib

Input: Indy Car log file of the race conducted on 27th May 2018.

STEPS:

Data sets for Car 9 and Car 12 are extracted from the Indy Car log file consisting of telemetry logs with letter starting $P.

The following features are extracted from the log file

- Time of the day
- Lap Distance
- Engine Speed
- Vehicle speed
- Gear
- Break
- Throttle

Exploratory Data Analysis is Performed on all the above features mentioned.

Break is further divided into 6 features through one hot encoding.

The visualization of various features is as follows for the car 12

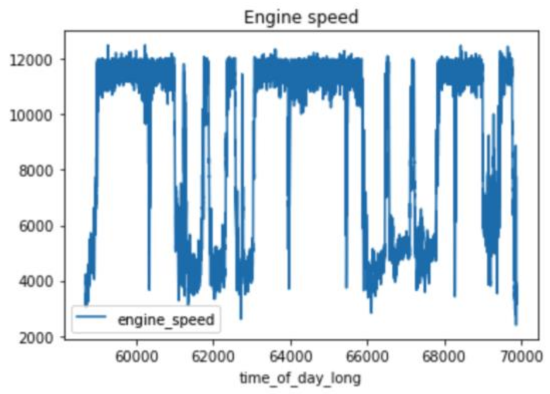


Figure 2: Visualization of Engine speed with time

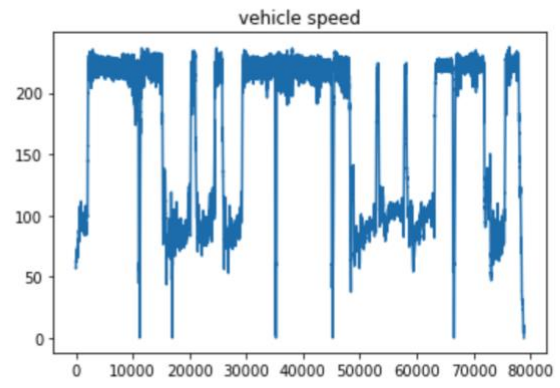


Figure 3: Visualization of Vehicle Speed with time

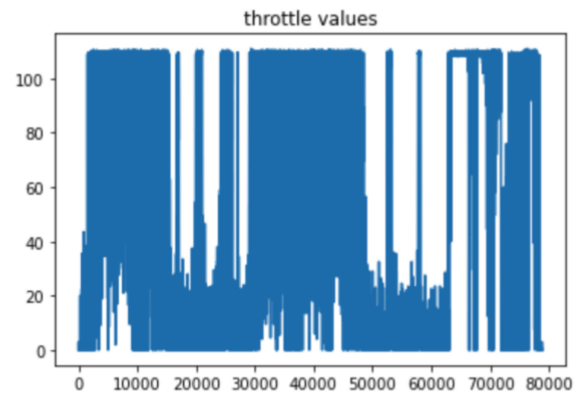


Figure 4: Visualization of Throttle values with time

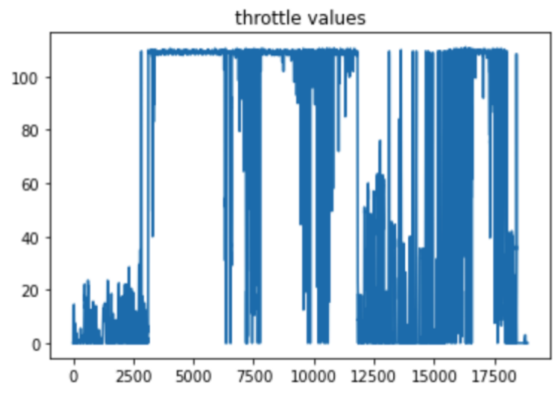


Figure 5: Visualization of Throttle values with time in the time interval of 60000 to 80000

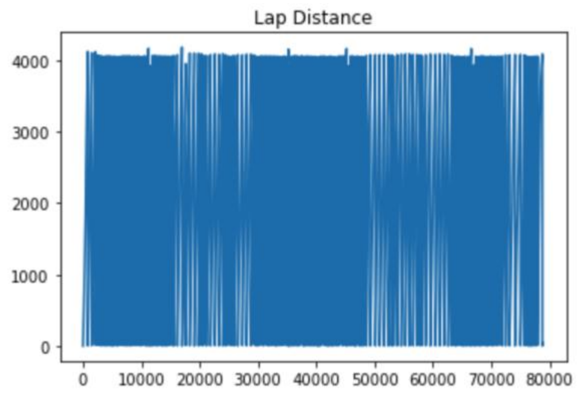


Figure 6: Visualization of Lap Distance with time

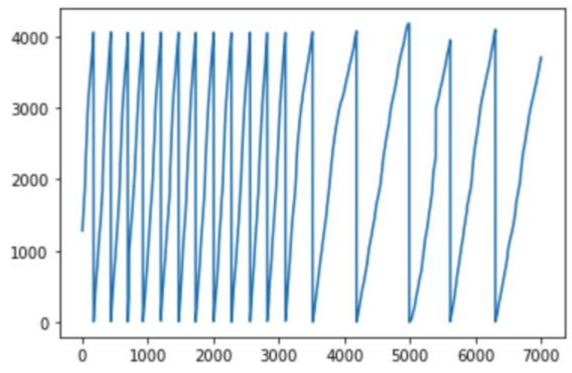


Figure 7: Visualization of Lap Distance between the time interval 12000 and 19000

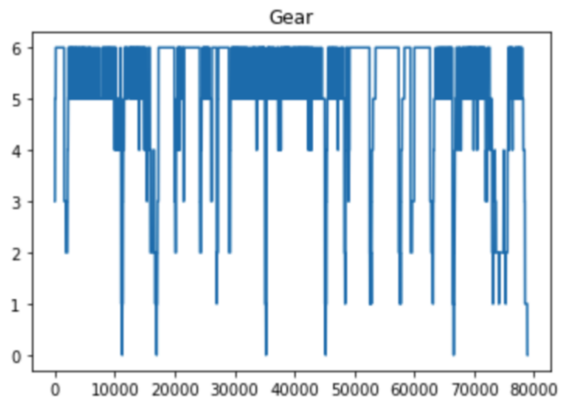


Figure 8: Visualization of Gears of the car 12 with respect to time

Correlations between various features of the data extracted

Feature 1	Feature 2	Correlation
Vehicle speed	Throttle	0.809
Lap Distance	Engine speed	-0.009
Engine Speed	Vehicle	0.956
Lap Distance	Throttle	0.0589
Lap Distance	Break	0.11

The Dataset is scaled for each feature from zero to one and fit into the model. The Autoencoder gives the following results for throttle

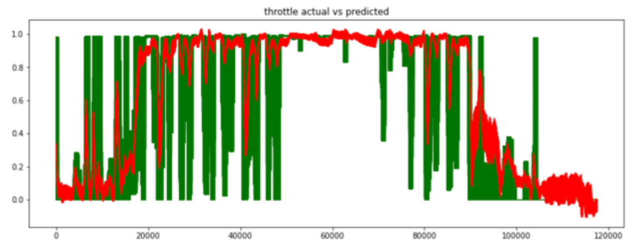


Figure 9: Autoencoder Prediction for Throttle

The correlations were highly negative for the input and autoencoder output, so Vehicle Speed, Lap Distance and Engine speed are the only features which are further considered in the experiment.

The Autoencoder output is as below when only the left over three features are considered

The Results of Train and Test loss are as below after final feature engineering

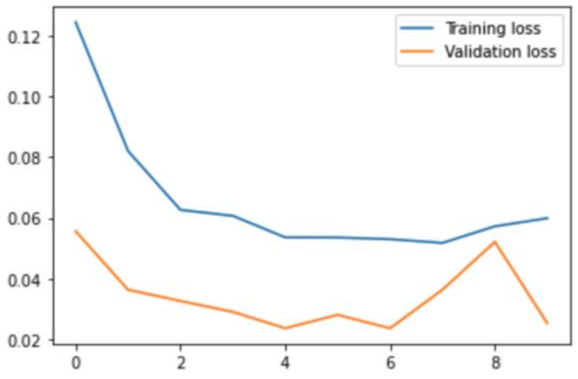


Figure 10: Training and Validation Loss of the Autoencoder

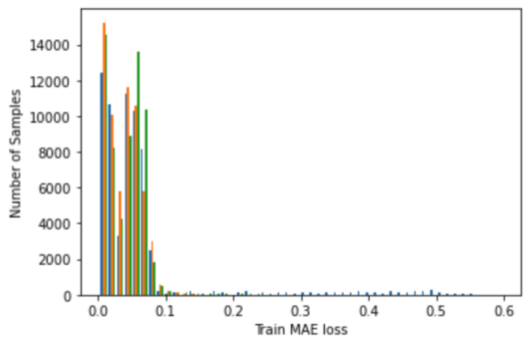


Figure 11: MAE Train losses sample wise for Train Data

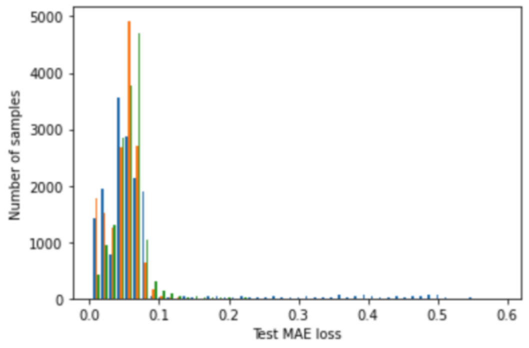


Figure 12: MAE Test losses sample wise for Test Data

The Predictions of the Autoencoder as follows for the three features finally Considered

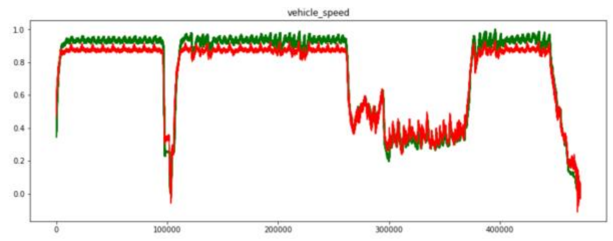


Figure 13: Autoencoder Prediction for Vehicle Speed with Test Data

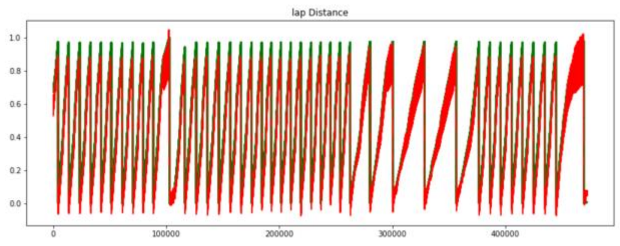


Figure 14: Autoencoder Prediction for Lap Distance with Test Data

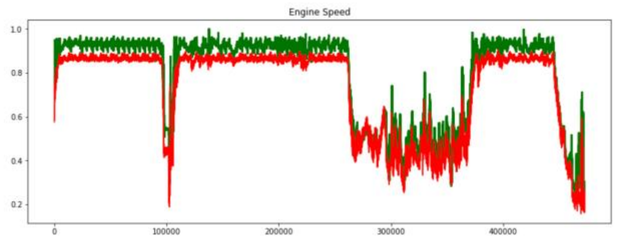


Figure 15: Autoencoder Prediction for Engine Speed with Test Data

The Anomalies Detected are as follows with respect to different thresholds for car 12 and car 9 respectively

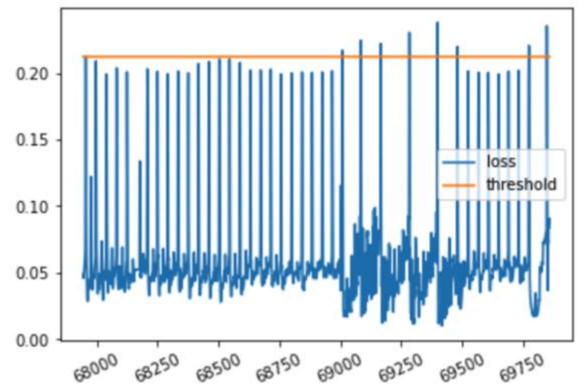


Figure 16: Anomalies Plot for Car 9

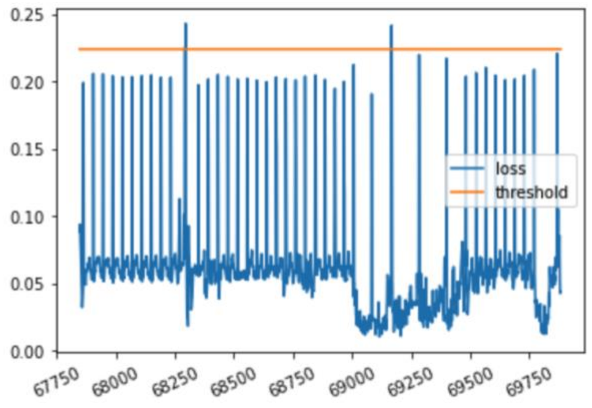


Figure 17: Anomalies Plot for Car 12

CONCLUSION

After Data Manipulation, Feature Engineering and finalizing the features the autoencoder is successfully able to predict the anomalies for the Project.

The Prediction of Anomalies for the test data set for the last 20 percent of the time period can be seen matching to the manually plotted anomalies through HTM Studio attached below.

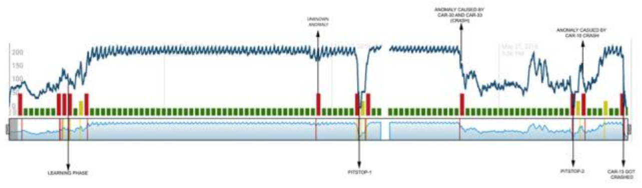


Figure 17: Manually Plotted Anomalies for Car 9

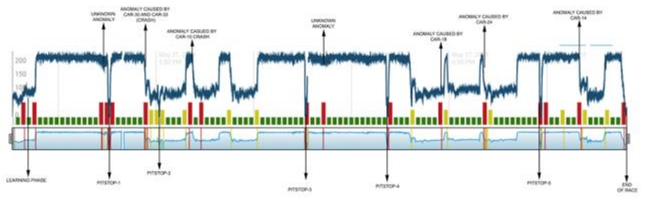


Figure 18: Manually Plotted Anomalies for Car 9

The Features which are impacting the Journey of the car (Vehicle speed, Lap Distance, Engine speed) are found.

In future implementations for anomaly detection for Indy Car and other multivariate time series datasets implementations like GRU networks can be explored as the method used for multivariate anomaly detection is unique to this project.

ACKNOWLEDGMENTS

I would like to acknowledge Professor Judy Fox, My Project supervisor Jiyau Li and Associate Instructor Selhattin akkas for all the input, mentorship and guidance during the course of the project

REFERENCES

[1] O.I. Provotar, Y.M Linder and M.M veres, “Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders”

[2] Mohammad Braei, Dr.-Ing. Sebastian,” Anomaly Detection In Univariate Time-Series: A Survey ON The State-Of-The-Art “

[3] Ihssan Tinawi,” Machine Learning for Time Series Anomaly Detection”, https://dspace.mit.edu/bitstream/handle/1721.1/123129/1128282917-MIT.pdf?sequence=1&isAllowed=y