Traffic volume in Minneapolis-Saint Paul westbound I-94

1st Ali El-Sherif
University of Ottawa
Giza, Egypt
aelsh054@uottawa.ca

2nd Mohamed AbdelAal
University of Ottawa
Sohag, Egypt
mabde157@uOttawa.ca

3rd Omar Abdulatif
University of Ottawa
Cairo, Egypt
oabdu015@uottawa.ca

4th Abdelrahman Elshazly
University of Ottawa
Aswan, Egypt
aelsh052@uottawa.ca

5th Abdelrhman Rezkallah
University of Ottawa
Alexandria, Egypt
arezk095@uOttawa.ca

***Abstract*—The wrong decisions wasted billions, efforts, and time each year, but , lately reliance on the data in decisions become obligatory, which reduce the wrong decisions possibility, and instead of rely on the human experience, the date became our fascinating guide, so we tried to provide an informative and valuable information for decision makers and decision makers in the transport and road sector, this project help to predict the traffic volume in the future based on historical time-series data, detect the abnormal behaviour, two approaches were used here for prediction Regression using Xgboost and Time-Series prediction using ARIMA & SARIMAX & Extra Trees regressor, and three techniques for anomaly detection using Deviation & Seasonal & STL methods, and the project success to predict the traffic at next 8 hours with MAPE less than 4%.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

The project focuses on providing valuable information in the transport and road sector, it's helpful for reducing the jams in I-94 highroad in Minnesota, road expansions decisions which are costly, what is the rush hours, and how can be prepared to address, at which hours the Traffic officer existence is essential, all of that are just few of lot beneficial, Metro Interstate Traffic Volume dataSet have chosen, this is a time-series data and Includes weather information and holiday features from 2012-2018, using that dataset many algorithms were used for predicting the traffic volume in future, and two approaches used here, Regression model used to predict the traffic as interpolation problem and it archived high accuracy with high R^2, and because of the time-series nature of our dataset, time-series models used also to predict the extrapolation, many preprocessing steps and anomaly detection models which improved the accuracy significantly, which did not exist in previous work especially applying anomaly detection as a preprocessing step.

II. System Architecture

In this section, the model structure is discussed through this flow chart:

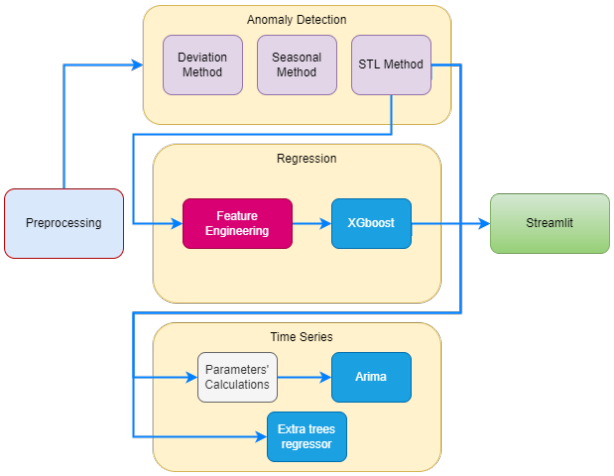

Fig. 1. Unexpected behaviour of traffic volume.

III. Modeling

A. Preprocessing

In this section, duplicated data and null values were removed. Nevertheless, Sorting the data frame in ascending order to deal with time series data later on, Set date time column as index of data frame. Finally, the time interval was set to be from July 1st, 2018 to September 29 th, 2018, yet, in time series, the last year of data was used.

B. Anomaly Detection

Anomaly detection is applied here as advanced preprocessing step through three different methods, because it affects the prediction significantly and prevents the signal

from being stationary, so addressing anomalies is crucial for accurate prediction, and that is represented in the abnormal peaks, and the unexpected behavior as illustrated in the following figure.

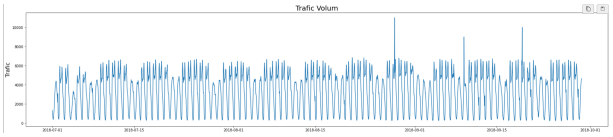

Fig. 2. Unexpected behaviour of traffic volume.

1) Deviation method It's a statistical method that depends on standard deviation and how it deviate about the normal one, the drawback of that method that it produces high error at first and that is because there is no data before the starting point, so we cannot predict the earlier data, so we have used more rubout one.

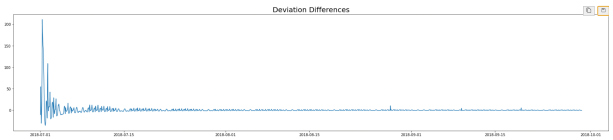

Fig. 3. Deviation method.

2) Seasonal method It's more rubout method for detecting the anomalies, it's using the stander deviation of each hour as points of interest, which is the vacation of traffic numbers for each hour in a day with the peak point between 5 and 10 Am which is an unexpected behavior. The drawback here is that each anomaly point can be addressed alone in each run.

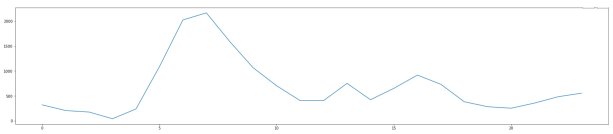

Fig. 4. Seasonal method.

3) Seasonal-Trend Decomposition using LOESS (STL) The most rubout one, it decomposes the Seasonality and Trends, and based on them it sorts a level of confidence as represented in the figure below, any peaks that going outside is considered as an anomaly as shown in figure 5.

   As it is shown in figure 6, all anomalies were detected and represented in the red points.

Fig. 5. STL method detection.

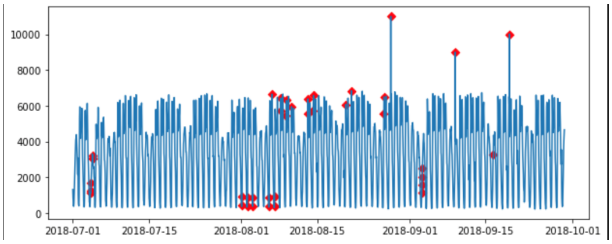

Fig. 6. STL method output.

finally, In figure 7, Those anomalies were tackled by taking the mean average of surrounded points, and after applying the SARIMAX model, there Mean Absolute Percent Error reduced from 0.6714 to 0.0471 which huge improvement.

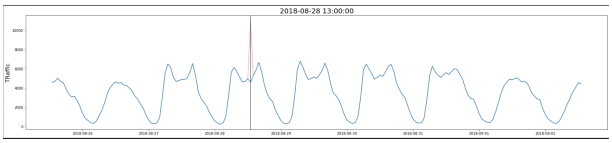

Fig. 7. STL with SARIMAX.

C. Regression

While using the data at first the R^2 had a very low value and it was enhanced after using regression model. In this part some feature engineering techniques were used which are: converting "date-time" column to date/time type of data, extracting months, days, day name and hours as features, one hot encoding for all of the mentioned features, including features with single column nature and concatenation of data. Various number of models were tried and the champion model is XGboost as its powerful approach for building supervised regression models and it showed high performance.

D. Time Series

This problem was handled as time series as there was some dependencies between errors, also, in real life, previous timed data should be taken into consideration. After removing all the anomalies, Autoregressive integrated moving average (ARIMA) was applied on the data. To apply ARIMA, three main parameters P,d,q are used to determine the d parameter the stationary data need to be checked using Dickey-Fuller

Test as in figure 8 which shows that the data is stationary so d equals 0.

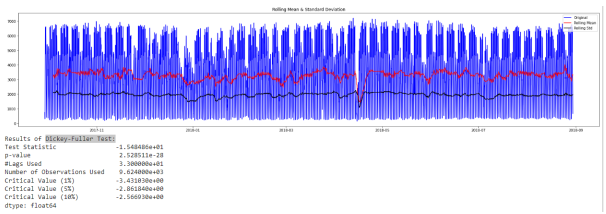

Fig. 8. Dickey-Fuller Test.

The parameter P describes how many previous readings can be taken into consideration to get the next value. To calculate it, partial auto-correlation function (PACF) is calculated in figure 9, which shows that the value of P equals 1.

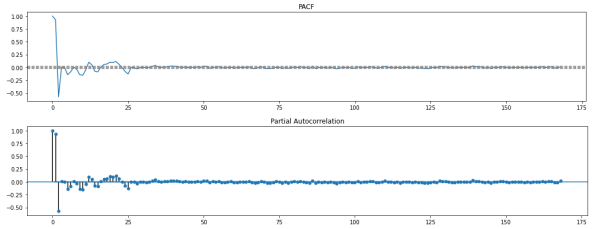

Fig. 9. Partial auto-correlation function

The parameter q describes how many previous errors can be taken into consideration to get the next value. To calculate it, auto-correlation function is calculated in figure 10, which shows that the value of q equals 7.

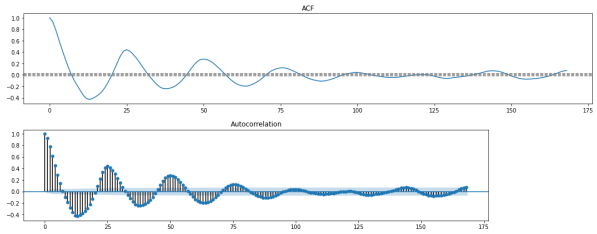

Fig. 10. Auto-correlation function

E. Streamlit

In this part, the regression model was used in streamlit which is an application framework for machine learning.

First of all, the environment should be set up by using a docker image by the use of a make file to simplify docker commands as in figure 11. Generally, to build this image, run make build. Secondly, make docker-up should be runned then the app can be accessed by typing "localhost:8501" in browser as figure 12.

```
.Phony: docker-up build down

docker-up:
    docker-compose up -d

build:
    docker-compose up -d --build

down:
    docker-compose down
```

Fig. 11. Streamlit application

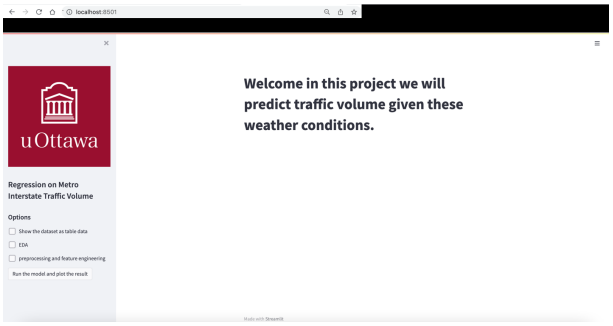

Fig. 12. Streamlit application

There are three options:

1) show the dataset as table data, which shows the main data that has been used.
2) EDA, which shows exploratory data analysis and it contains the following:
   - Distribution of temperature
   - Rain density
   - Holidays (with/without normal days)
   - Cloud density
   - General weather features
   - Relationships between traffic density and other features.
   - Correlation heat map between traffic volume, clouds' density, snow of one hour, rain of one hour and temperature.
3) Preprocessing and feature engineering, which shows graphs of:
   - conversion of "date-time" column to date/time type of data.
   - Extraction of month feature
   - Extraction of Day feature
   - Extraction of hour feature
   - Extraction of day name feature
   - Extraction of month feature
   - One hot encoding to all the extracted features
   - Utilization with apply method
   - include features with single column nature
   - Concatenation of one hot encoded features

Now "Train the model and plot the result" can be pressed for training, prediction and plotting the results.

PERFORMANCE EVALUATION

F. Regression

As previously mentioned, XGboost was trained in regression on test data and it resulted R^2 of 94.63%, Mean squared error of 207995.49 and mean absolute percentage error of 13.93%. The result of predictions versus actual data is shown in the following figure.

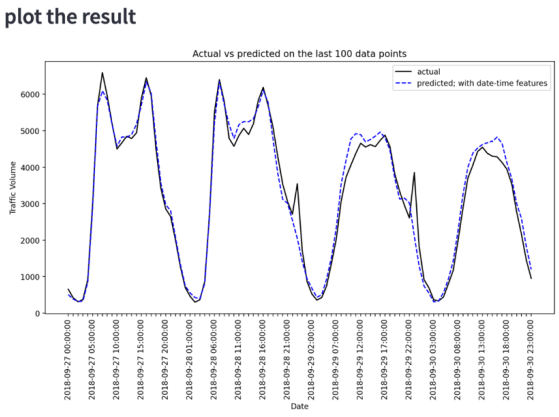

Fig. 13. XGboost model prediction output

G. Time Series

The ARIMA model was trained on test data and resulted Mean squared error of approximately 429.516, root mean squared error of 644.085 and mean absolute percentage error of 19.077%. The result of predictions versus actual data is shown in the following figure.

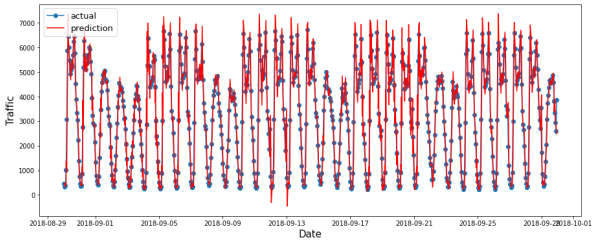

Fig. 14. ARIMA model prediction output

In figure 15, The mean absolute error of different forecasting times is as shown. It is clear that the error between 8 and 12 hours is acceptable compared to that of 24 hours.

The following figures are the forecasting of traffic density for the next 8 hours in figure 16 and 12 hours in figure 17.

From these dates it is noticed that there is a seasonality and the ARIMA does not detect it so different algorithms can be applied to detect the seasonality and compare between the results which are in figure 18.

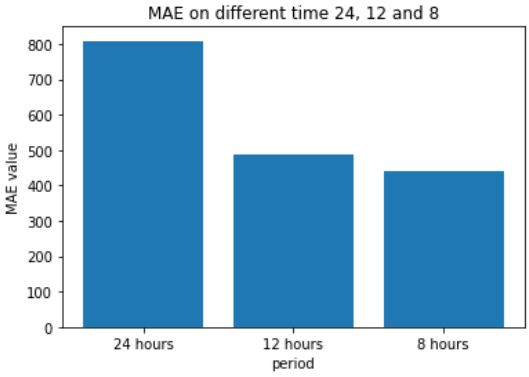

Fig. 15. MAE of different forecasting hours

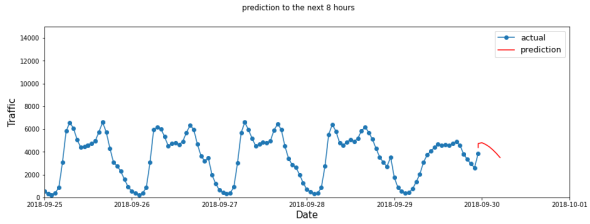

Fig. 16. Traffic forecasting for 8 hours

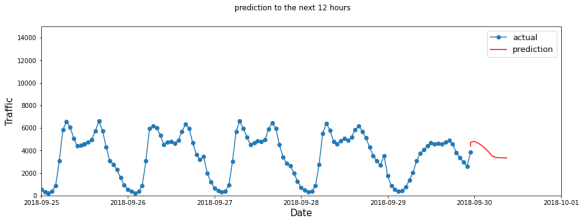

Fig. 17. Traffic forecasting for 12 hours

	Model	MAE	RMSE	MAPE	SMAPE	MASE	RMSSE	R2	TT (Sec)
et_cds_dt	Extra Trees w/ Cond. Deseasonalize & Detrending	159.3119	260.6752	0.0554	0.0566	0.0822	0.1084	0.9786	10.6800
rf_cds_dt	Random Forest w/ Cond. Deseasonalize & Detrending	166.6087	279.8376	0.0581	0.0597	0.086	0.1164	0.9765	33.0733
knn_cds_dt	K Neighbors w/ Cond. Deseasonalize & Detrending	181.0107	296.3685	0.0708	0.0707	0.0934	0.1232	0.9727	3.0600
lightgbm_cds_dt	Light Gradient Boosting w/ Cond. Deseasonalize...	191.9246	298.8277	0.0749	0.0754	0.0991	0.1243	0.9724	2.2867
dt_cds_dt	Decision Tree w/ Cond. Deseasonalize & Detrending	251.654	364.1347	0.0951	0.095	0.1299	0.1514	0.959	1.8833
gbr_cds_dt	Gradient Boosting w/ Cond. Deseasonalize & Det...	281.9427	389.6504	0.1015	0.1	0.1455	0.162	0.9552	16.8533
huber_cds_dt	Huber w/ Cond. Deseasonalize & Detrending	274.0972	409.1249	0.1257	0.1202	0.1415	0.1701	0.9538	1.5067
br_cds_dt	Bayesian Ridge w/ Cond. Deseasonalize & Detren...	426.4876	551.8206	0.2487	0.2005	0.2201	0.2295	0.9198	1.3667
en_cds_dt	Elastic Net w/ Cond. Deseasonalize & Detrending	426.7937	551.9826	0.2494	0.2008	0.2203	0.2295	0.9197	1.7733
ridge_cds_dt	Ridge w/ Cond. Deseasonalize & Detrending	426.7959	551.984	0.2494	0.2008	0.2203	0.2295	0.9197	1.4133
lasso_cds_dt	Lasso w/ Cond. Deseasonalize & Detrending	426.7925	551.982	0.2494	0.2008	0.2203	0.2295	0.9197	1.8100
lr_cds_dt	Linear w/ Cond. Deseasonalize & Detrending	426.7959	551.984	0.2494	0.2008	0.2203	0.2295	0.9197	1.4200
llar_cds_dt	Lasso Least Angular Regressor w/ Cond. Deseaso...	511.7508	672.1992	0.2973	0.2297	0.2641	0.2796	0.8915	1.3833
omp_cds_dt	Orthogonal Matching Pursuit w/ Cond. Deseasona...	620.0735	791.7623	0.416	0.2852	0.3201	0.3293	0.8466	1.3733
ada_cds_dt	AdaBoost w/ Cond. Deseasonalize & Detrending	1165.2028	1426.2362	0.7926	0.4496	0.6014	0.5931	0.4996	5.2967

Fig. 18. Models' comparison

As previously shown, the best model that has achieved the highest results is ”extra trees deseasonalize and detrending”. Figure 19 shows the result after forecasting the next 1000 hours.

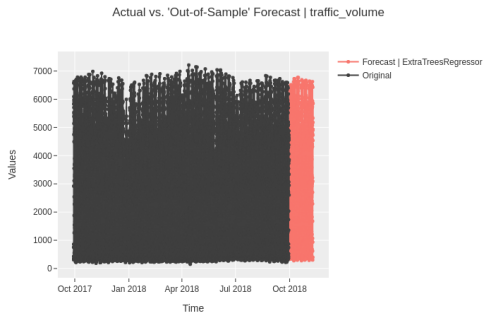

Fig. 19. Extra trees regressor

And here is a sample output:

	y_pred
2018-09-30 00:00	1878.3266
2018-09-30 01:00	1185.1143
2018-09-30 02:00	763.5546
2018-09-30 03:00	424.9395
2018-09-30 04:00	429.5728
...	...
2018-10-08 03:00	204.6250
2018-10-08 04:00	314.6074
2018-10-08 05:00	947.5564
2018-10-08 06:00	3084.3752
2018-10-08 07:00	5733.6868

Fig. 20. Extra trees regressor’s output sample

Summary and Conclusion

The project succeeded to predict the traffic volume at next 8 hours in accurately without any computational power complexity by using time series which is more realistic than regression, and it is applicable to use in all countries and different sectors, and with some improvement it can predict more than 8 hours with low error, also the improvement is always going up throw time, where more data will be collected, and improving the algorithms may achieve higher accuracy specially the late methods using deep learning but it needs a lot of data to train and huge computational power which is not preferable in real-time prediction.

Bibliography

References

[1] A. Chandel, “An Accurate Estimation of Interstate Traffic of Metro City Using Linear Regression Model of Machine Learning,” SSRN Electronic Journal, 2020, doi: 10.2139/ssrn.3598310.

[2] A. Ermagun, S. Chatterjee, and D. Levinson, “Using temporal detrending to observe the spatial correlation of traffic,” PLOS ONE, vol. 12, no. 5, p. e0176853, May 2017, doi: 10.1371/journal.pone.0176853.

[3] “UCI Machine Learning Repository: Metro Interstate Traffic Volume Data Set,” archive.ics.uci.edu. https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume

[4] “I-94/I-35E in St. Paul project - MnDOT,” www.dot.state.mn.us. https://www.dot.state.mn.us/metro/projects/i94-i35e-stpaul/

[5] Kanokwan Khiewwan, Phrommate Weeraphan, Khumphicha Tantisontisom, and Jindaporn Ongate, “Application of Data Mining Techniques for Classification of Traffic Affecting Environments,” Journal of Renewable Energy and Smart Grid Technology, vol. 15, no. 1, 2020, [Online]. Available: https://ph01.tci-thaijo.org/index.php/RAST/article/view/240698

[6] E. A. Mrs. B.Karthika, “Traffic Flow Prediction Using An Improved Fuzzy Convolutional LSTM Algorithm,” Turkish Journal of Computer and Mathematics Education (TURCOMAT), vol. 12, no. 10, pp. 5541–5549, Apr. 2021, doi: 10.17762/turcomat.v12i10.5361.

[7] Z. Yin and P. Barucca, “Stochastic Recurrent Neural Network for Multistep Time Series Forecasting,” Neural Information Processing, pp. 14–26, 2021, doi: 10.1007/978-3-030-92185-9_2.

[8] J. Zhao et al., “Do RNN and LSTM have Long Memory?,” proceedings.mlr.press, Nov. 21, 2020. https://proceedings.mlr.press/v119/zhao20c.html (accessed Aug. 02, 2022).