City Pulse Traffic IoT Dataset Analysis

Anirudh Kuttiyil Valsalan, Febin Steve Jose, Malini Kottarappatt Bhaskaran, Nipun Agarwal, and Niraj Gadakari

***Abstract*—With the availability of IOT data for various cities, various applications are now been developed for tackling some serious problems. One of them is the growing impact of the vehicular traffic on the environment which results in high pollution. In this paper we present the analysis of real time and archived IOT data for the city of Aarhus, Denmark to study the impact of pollution caused by vehicular traffic on the environment. We also provide real time analysis of the traffic by suggesting traffic hotspots in the city of Aarhus.**

***Index Terms*—IOT, Spark, Traffic Hotspots.**

I. Introduction

TODAY the cities are becoming smarter with the advent of new technologies and valuable data available from various sensors across the cities which have led to more applications and services being developed [12]. The aggregation of these data can be put to use to solve various problems across the cities. In this project we have the IOT (Internet of Things)[12] data for the city of Aarhus, Denmark [1]. The data was collected from the sensors installed at various points across the city collecting the traffic information, weather information and the pollution information for those locations. Fig. 1 is the citypulse framework [8] which shows how the IOT data can be used for building various application on top of it. Our project focuses on aggregating these datasets to come up with some real time information from streaming data for analyzing and solving the problems across the city of Aarhus. We also did a static analysis on the archived data taken from [1]. Initially we worked on the archived data to combine these data sets after performing the necessary data cleansing on the raw data. We performed predictive analysis [4] on the dataset to predict the traffic status by building various models [2] using MLlib, an Apache Spark's machine learning library [3][5]. We also performed classification using an open source software library for numerical computation using data flow graphs, called TensorFlow [7]. We also worked on the streaming data and using various visualization techniques suggested the traffic hotspots. The report is divided into various sections. Section II describes the dataset, section III. Various data cleansing techniques applied on the data. Section IV describes the work flow for streaming and archived data and the machine learning techniques used. Section V discusses the various visualization and results obtained. Section VI finally concludes the report.

II. Dataset

We used the smart city pulsedata [1] for the city of Aarhus, Denmark. The dataset contained various files for the IOT data collected by various sensors for traffic, pollution and weather. We also had a metadata file with the details of the each dataset. The weather data had readings for dew point, humidity, pressure, temperature, wind direction and wind speed. The Vehicle Traffic data contains the information of vehicular count over a period of 2 months between 449 points of observation. The pollution data has measurements for various pollutants which are in conjunction with the vehicle traffic dataset.

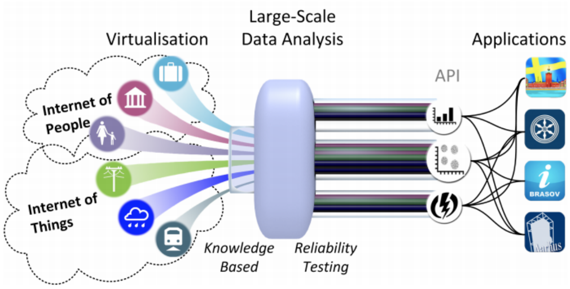


Fig. 1: CityPulse Framework.

III. Data Pre-processing

The original raw data was more than 3 GB in size. However, this raw data was not readily usable and we had to apply various data cleansing techniques to clean it and then finally join all these datasets into a single combined dataset for being able to perform analysis on it. The weather data was divided into individual data files for each attribute in JSON format. This has to be converted into one aggregated comma separated file which can be used for our purpose. In the pollution dataset we didn't had report ID complementing the other datasets. Also, there were many null values in the dataset which could have made results inconsistent. We had to remove all these rows. Also, the timestamp values in different dataset had different format and were in irregular time interval. We had to extrapolate the weather data to 5 minute interval. We had to create dataframes out of each dataset.

IV. Workflow

A. Real-Time Data Analysis

We have used the data API [6] to fetch the IOT streaming data provided by Open Data Aarhus which is updated every 5 minutes. We have used PySpark for analysis of real time data. The streaming data is coming in the JSON format. Here is the work flow for the streaming data:

- The JSON string converted to JSON Object and resultant JSON object then converted to pandas dataframe.
- Using PySpark the panda dataframe is transformed to spark dataframe.

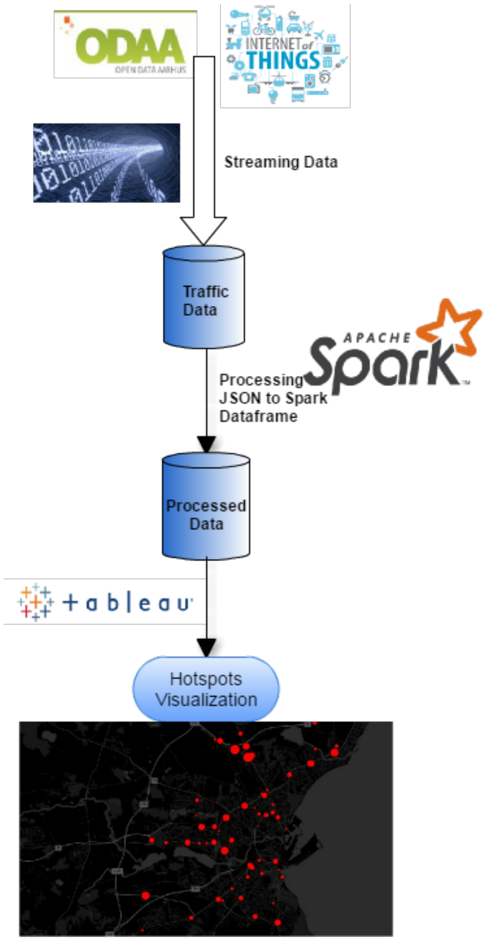
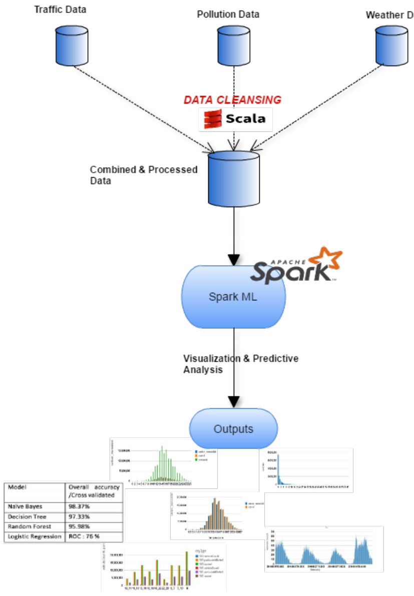


Fig. 2: Flowchart for streaming data analysis.

- Dataframe is created out of metadata with pyspark and then combined it with streaming vehicle data to get vehicle count and corresponding location.
- We have compared the vehicles count with allowed threshold value and plot hotspots for displaying critical regions

B. Archived data Analysis

We did statistical analysis on the archived data taken from [1]. Initially we worked on the archived data to combine these data sets after performing the necessary data cleansing on the raw data. We performed predictive analysis [4] on the dataset to predict the traffic status by building various models [2] using MLlib, an Apache Spark's machine learning library [3][5]. Here are the details of machine learning algorithms used on Traffic Dataset. We have combined the traffic, pollution and weather dataset based on report ID and timestamp. As part of the initial analysis, we have generated the analysis plots to identify patterns in the data. From the daily vehicle count plot we have identified that the day type and day time has an impact in the vehicle count. We have used the pollution data and weather data, day time and day type to predict the traffic status of the place. With the processed data, we have used machine learning algorithms like Nave Bayes, Decision tree, Logistic regression and Random Forest to predict the count. We have also performed cross validation to make sure that the models are not over fitted. The data cleansing and pre-processing was performed using Spark programs written in Scala. For machine learning part we have used PySpark kernel of Spark. Spark ML libraries are used for building the models. Models are built using the data frames rather than using RDD. Random Forest accuracy is better compared based on the confusion matrix. Our data is imbalanced as the class with traffic status high is less compared to the traffic status normal. So the classifier decision tree and logistic regression doesn't perform well even though the overall accuracy values are misleading. The confusion matrix of Nave Bayes and Random Forest, and accuracy values of various classifiers are as shown in Fig. 4 to Fig. 6.



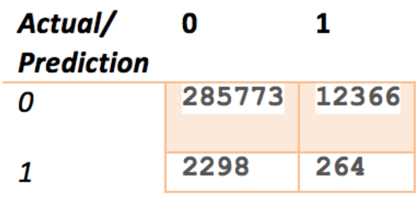
Model	Overall accuracy /Cross validated
Naive Bayes	98.37%
Decision Tree	97.33%
Random Forest	95.98%
Logistic Regression	ROC : 76 %

Fig. 3: Flowchart for static data analysis.

We also performed classification using an open source software library for numerical computation using data flow graphs, called TensorFlow. The problem was to build a convolution neural network to classify the Traffic status by using the features like location (i.e. latitude, and longitude), type of the day (i.e. weekday or weekend), and weather data at that particular location. The raw data was not normalized, so we had to normalize the data which was an important step for model building. We had to choose the model parameters on experimental basis. Following parameters were chosed: learning rate, training iterations and biases. We divided the data into a training and testing dataset to perform cross validation. 80% of the data was used for training the classifier and 20% data was used to test the classifier. This required a lot of time as the dataset was very large. Finally a TensorFlow graph was developed to fit a convolution neural network for the data. The accuracy achieved for predicting the traffic status to be "OK" or "NOT OK" was 98.5%.

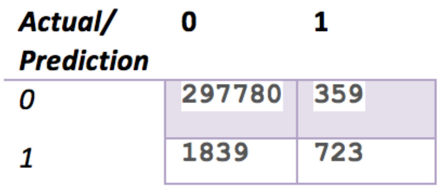
V. Visualization

We analyzed the data and have plotted various plot for the same. We also found the hotspots where we found the traffic



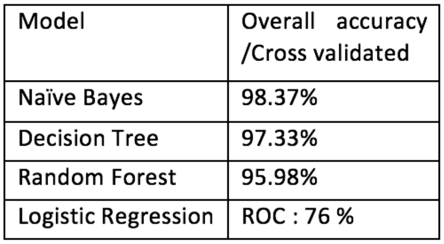
Actual/ Prediction	0	1
0	285773	12366
1	2298	264

Fig. 4: Naive Bayes.



Actual/ Prediction	0	1
0	297780	359
1	1839	723

Fig. 5: Random Forest.



Model	Overall accuracy /Cross validated
Naïve Bayes	98.37%
Decision Tree	97.33%
Random Forest	95.98%
Logistic Regression	ROC : 76 %

Fig. 6: Model Accuracy Comparison.

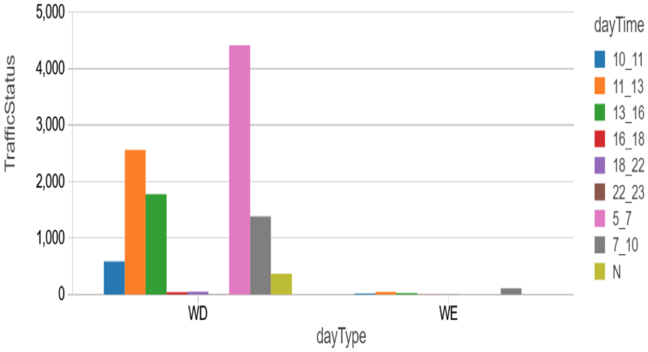


Fig. 7: Traffic Status VS Day type.

might be more than some threshold. The figures from Fig. 7 to Fig. 15 summarizes the results we got.

VI. Conclusion

In this report we presented in-depth analysis of real time streaming IOT data and archived data for the city of Aarhus, Denmark. We found some temporal trends in the vehicular traffic. We also did a predictive analysis on the vehicular traffic to check the status as being OK or NOT OK using MLlib library in spark and tensor flow. We also plotted the traffic hotspots to suggest high traffic area. All of the above techniques were performed using the community edition of the databricks.

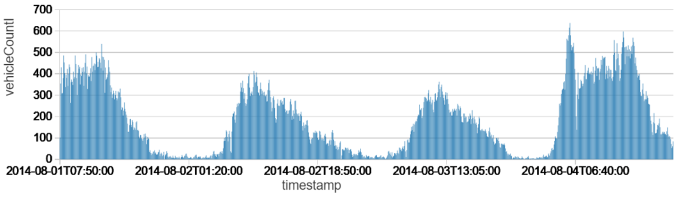


Fig. 8: Vehicle count VS Time stamp.

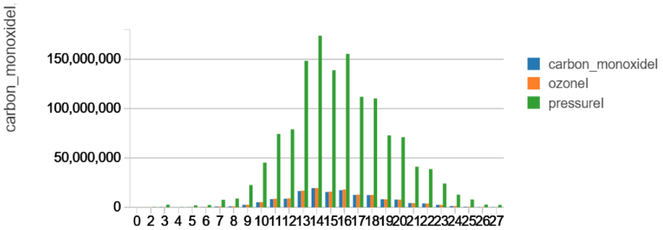


Fig. 9: Pollutants Vs Temperature.

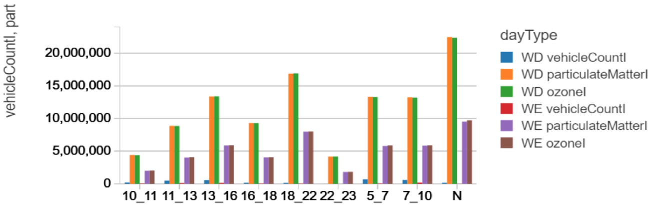


Fig. 10: Vehicle Count, Pollutants Vs daytime (Grouped by Day type).

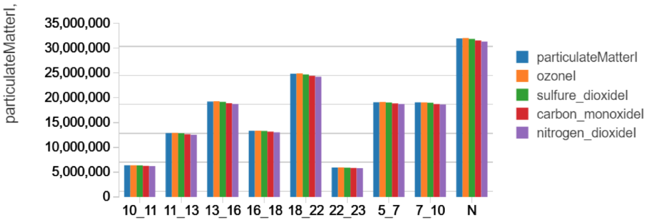


Fig. 11: Pollutants VS Day Time.

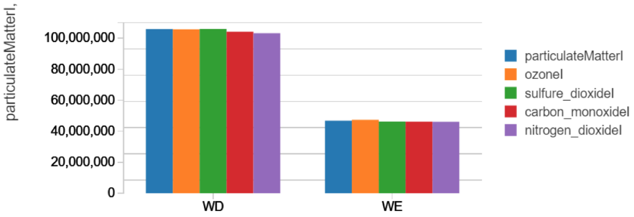


Fig. 12: Pollutants Vs DayType.

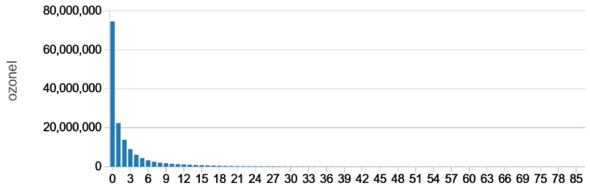


Fig. 13: Ozone Vs Vehicle Count.

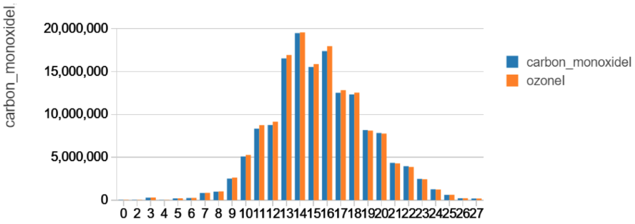


Fig. 14: CO, O3 Vs Temperature.

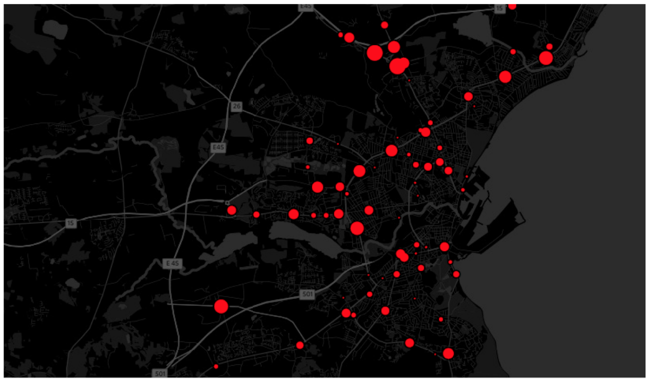


Fig. 15: Various traffic hotspots as shown in red for the city of Aarhus plotted using Tableau[9].

Acknowledgment

The authors would like to thank Dr. Latifur Khan for providing timely guidance to successfully complete this IoT Big-Data project.

References

[1] Muhammad Intizar Ali, Feng Gao and Alessandra Mileo *CityBench: A Configurable Benchmark to Evaluate RSP Engines Using Smart City Datasets*, In proceedings of ISWC 2015 - 14th International Semantic Web ConferenceBethlehem, PA, USA.

[2] Srini Penchikala *Big Data Processing with Apache Spark - Part 4: Spark Machine Learning*
https://www.infoq.com/articles/apache-spark-machine-learning

[3] Juliet Hougland and Sandy Ryza *How-to: Predict Telco Churn with Apache Spark MLlib*
https://blog.cloudera.com/blog/2016/02/how-to-predict-telco-churn-with-apache-spark-mllib/

[4] Bill Haffey *Predictive Analytics and Machine Learning: An Overview*
https://www-01.ibm.com/events/wwe/grp/grp004.nsf/vLookupPDFs

[5] *Spark MLlib*
http://spark.apache.org/mllib/

[6] *Real-Time Traffic Data*
https://www.odaa.dk/dataset/realtids-trafikdata/resource/b3eeb0ff-c8a8-4824-99d6-e0a3747c8b0d

[7] *TensorFlow*
https://www.tensorflow.org/

[8] Payam Barnaghi, Ralf Tonjes et. al. *CityPulse: Real-Time IoT Stream Processing and Large-scale Data Analytics for Smart City Applications*, poster session, European Conference on Networks and Communications. 2014.

[9] *Tableau*
https://www.tableau.com/stories/topic/maps

[10] R. Tnjes, P. Barnaghi, M. Ali et. al.*Real Time IoT Stream Processing and Large-scale Data Analytics for Smart City Applications*, poster session, European Conference on Networks and Communications 2014

[11] Kolozali, Sefki, et al.*A knowledge-based approach for real-time IoT data stream annotation and processing.*Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom), IEEE. IEEE, 2014.

[12] *IoT*
https://www.wired.com/insights/2014/11/the-internet-of-things-bigger/

[13] Barnaghi, Payam, Amit Sheth, and Cory Henson. *From Data to Actionable Knowledge: Big Data Challenges in the Web of Things [Guest Editors' Introduction].*IEEE Intelligent Systems 28.6 (2013): 6-11.