2015 IEEE Wireless Communications and Networking Conference (WCNC) - Workshop - Energy Efficiency in the Internet of Things,
and Internet of Things for Energy Efficiency

A predictive data-driven model for traffic-jams forecasting in Smart Santader city-scale testbed

Jérôme Treboux, Antonio J. Jara, Luc Dufour, Dominique Genoud
Institute of Business Information Systems
University of Applied Sciences Western Switzerland (HES-SO)
Sierre (Switzerland)
{jerome.treboux, antonio.jara, luc.dufour, dominique.genoud}@hevs.ch

***Abstract*—In this paper, a model for traffic jam prediction using data about traffic, weather and noise is presented. It is based on data coming from a Smart City in Spain called Santander. The project in this city is called ”Smart Santander” and provides a platform for large-scale experiment based on real-time data. This paper demonstrates the possibility of predicting traffic jams and is a basis to integrate in projects to improve the quality of services. In this work, a cross validation method to ratify our training set is proposed. Data intelligence analysis techniques are used for the prediction with an implementation of Neural Network and Decision Tree algorithms. These algorithms are using different parameters coming from Smart Santander and other external sources. Furthermore, a cross validation process is also integrated to improve the final result. The traffic jam prediction for the next 15 minutes reached an accuracy of 99.95%.**

***Index Terms*—Data Intelligence; Neural Network; Traffic flow prediction; Smart Cities; KNIME.**

I. Introduction

In Los Angeles, it takes 20 minutes to drive 5 miles in the city and around 4500 traffic lights to manage in this city [1]. Furthermore, the traffic intensity in the US has increased compared to the year 2012. The average time a driver spends in traffic jams during a year is 38 hours. [2]

Nowadays, the technology is evolving rapidly and data is coming from a lot of different sources, such as sensors, mobile phones or satellites. With this growing number of sources, the amount of data is growing as well. In order to manage this big data, specialized software is used. In this paper, it will be the open source data-mining platform called ”Knime”[1].

To execute this project, the data we use comes from the city of Santander in Spain, Europe. They are currently running a project called ”SmartSantander”[9]. It proposes an experimental research facility in support of applications and services for smart cities at a city-scale. Santander is the most data-intensive city in Europe. It has some 18000 stationary and mobile sensors of various types thought the municipality of around 180000 residents. These sensors monitor air pollution, noise, traffic, temperature, and other environmental conditions.

During this project, a main question will be answered: ”Are the data from the Smart Santander consistent enough to predict a traffic jam and build a project to improve the traffic management in the city?”

There are some tools giving a prediction of the traffic status during next minutes:

- IBM with their ”Smarter Traveller web application” [3]
- INRIX with their tools coming from the Microsoft project ”JamBayes” aggregating multiple sources such as sensors or police incident report. They provide a service that will give a better way for driver to avoid a traffic jam but only based on real-time information [2].
- Google & Waze with their most used mobile application in the world (54% of smartphone users use this application) [4]. They provide a real-time traffic flow analysis based on users localization (via smartphone) and community sharing (Waze).

The real-time traffic flow representation is good but some problems arise, like the human reaction. ”*An algorithm that reroutes precisely the right amount of traffic is still likely years away*” [5].

Unlike these tools, at the level of algorithms, stage of the prediction is advanced. Indeed, a lot of different projects to test the effect of weather or to predict the stock market were made. For example:

- The project ”Traffic Prediction System based on Probe Vehicles” emerged at the INRIA in 2006 [6] before the generalization of smartphones. The idea was to send multiple probe vehicles driving in a city and gather real-time traffic information. Because of smartphones they were preceded by Google.
- A project to evaluate the effect of the rain on the traffic parameters was carried out in 2009 [7]. The result is that the rain influences the traffic flow. For a thousand of vehicles miles travelled, the number of incidents changed from about 0.6 without rain to about 0.9 with rain at the incident moment.
- The last interesting project was talking about the prediction of the stock market, with a fusion model of Hidden Markov Model[2], Artificial Neural Network[3] and Genetic

[1] KNIME is the leading open platform for data-driven innovation - http://www.knime.com

[2] A Hidden Markov Model (HMM) is a statistical Markov model presented as the simplest dynamic Bayesian network.

[3] Artificial Neural Network (ANN) is used to estimate functions that depend on a large amount of inputs generally unknown.

978-1-4799-8760-3/15/$31.00 ©2015 IEEE

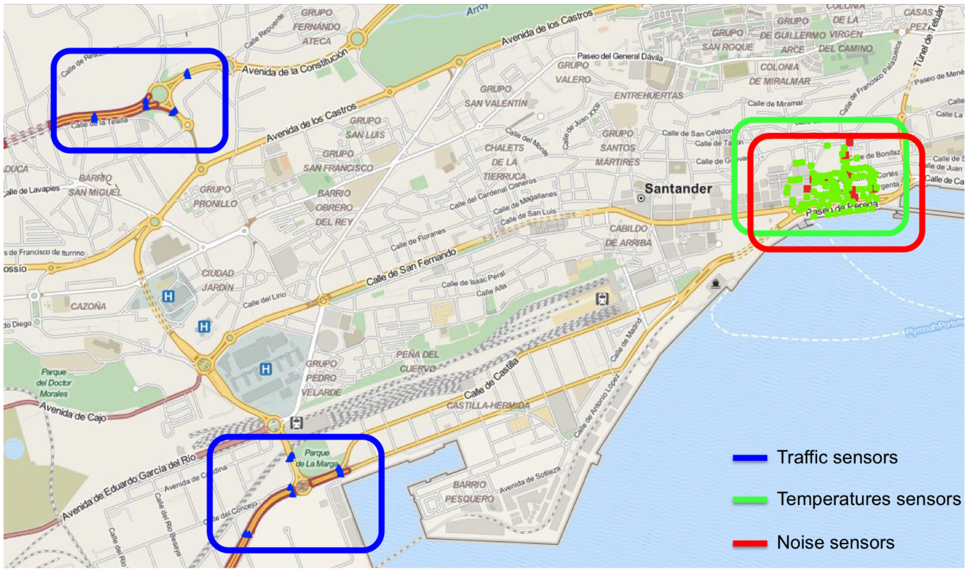


Fig. 1. Smart Santander sensors location

Algorithm [4].The GA will find the optimal parameters for the HMM and the ANN will introduce noise to the observation for a better fit [8].

It is full of other projects based on the traffic flow management such as the Intel project in the city of San José. With this project, a full experiment based on sensors data and meteorological data will be done. Unlike all projects presented, it is a full project beginning with data analysis and ending with an alert management going through algorithm testing and data cleaning. In this paper, the focus is on the data-mining part.

II. Methodology

Smart Santander provides 10 sensors for the noise, 99 sensors for the temperature and 38 sensors for the traffic intensity. As presented on Figure 1, sensors are dispatched all around the city. They are distributed in a very disparate manner across the city. Indeed, the noise would not be influenced by the traffic if there are some kilometres between them. The temperature is interesting for us, but the detailed weather is more important. For this reason, an external web service is used to have all information from the same source.

For the weather, the API from World Weather Online[5] is used. They provide one historical information every 3 hours. A general weather information is provided with mode detailed information, like the visibility, the temperature in Celsius and Fahrenheit, the cloud cover, the pressure, humidity and rain precipitation and wind speed.

Every minute, traffic intensity information is provided with the node ID and its localization (latitude and longitude), the current occupancy recorded and the number of vehicles on the lane. The specific date does not need to be taken because the prediction would not be on the same date. However, the day in the week is important (Monday, Tuesday, etc.) and also if it is a day off (like holidays or Sundays for example).

As presented in the introduction, the rain during and before a traffic jam is important. It is also important to know if the day is a normal day or not. Figure 2 and Figure 3 show the patterns present in the traffic depending if the day is a Sunday, a day off or a holiday day.

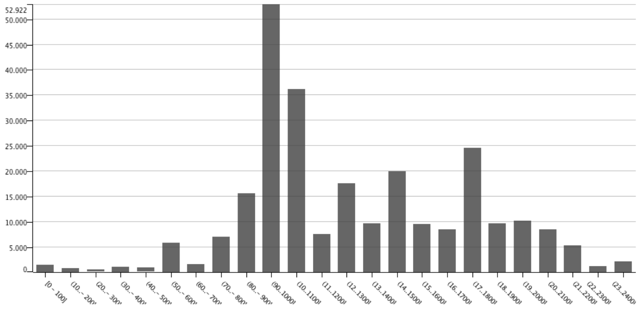


Fig. 2. Occupancy average during a weekday (including Saturday) by hour

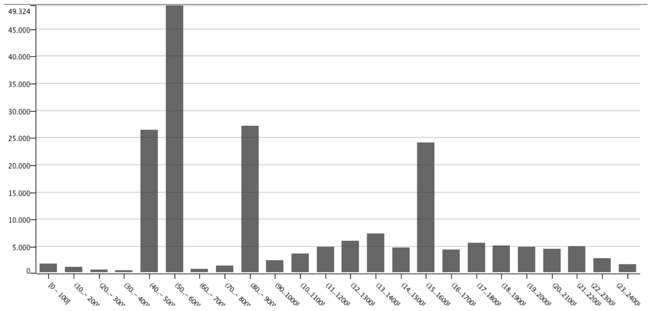
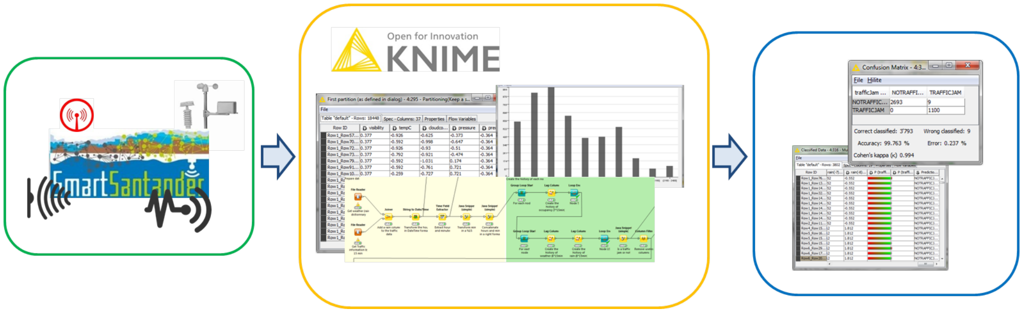


Fig. 3. Occupancy average during a Sunday or a day off by hour

To achieve goals over this project, multiple steps are manda-

[4] Genetic Algorithm (GA) is a search heuristic that mimics the process of neural selection.

[5] It is a website who provides information about the weather around the world - http://www.worldweatheronline.com/



Confusion Matrix - 4:3...

File Hilite

trafficJam ...	NOTRAFFI...	TRAFFICJAM
NOTRAFFIC...	2693	9
TRAFFICJAM	0	1100

Correct classified: 3'793 Wrong classified: 9

Accuracy: 99.763 % Error: 0.237 %

Cohen's kappa (κ) 0.994

Fig. 4. Project work flow

tory. First, a data cleaning process is necessary to analyse and confirm the data quality. This step is one of the most important to stabilise the project. This process will in particular handle differences between the number of cars and the occupancy value, and clean duplicate data at the same hour and place. The correction is done after a depth analysis through different visualization tools. A concatenating process between Smart Santander data and World Weather Online data is also carried out during the cleaning step.

Once data are ready to be handled, a pre-processing is done. That means some features are added like the days or the traffic categories (Fluid, Heavy, Busy and Traffic Jam). A grouping step to have a granularity of 15 minutes is also done during the pre-processing. After this step, because its size, the dataset has to be reduced. Indeed, the best way to train an algorithm would be to use all of the data from one year and test it on a second year, taking in account seasonal information. In this case, only a few months are taken into account for the training and testing process. The testing process is done on the same months as the training process, but one year after. This methodology has kept a potential seasonal information in the dataset.

For a prediction, a history needs to be created. Data are available but cannot be associated together. This history is artificially created through a lag process. This will create the new line with value in the past. For example, for a specific node at a precise time, the occupancy history (2 hours before) will be added as additional columns.

Because of the difference between the number of ”No Traffic Jam” and ”Traffic Jam” classes, a Bootstrap sampling is carried out. This will equalize both samples and the accuracy of the model is improved.

Finally, to evaluate the importance of each feature, a meta node available in KNIME is used. This meta node contains a Backward Feature Elimination Loop. This loop iterates on the dataset to find the error rate of each feature. The first iteration is executed with all input columns. In the next *n - 1* iterations, each of the input columns (target column exempted) is left once. The node will then discard the column that influenced the result the least. Then *n - 2* iterations follow where each of the remaining columns is left out once and so on. The final result is interesting because it shows that not all columns are important for the model. With over 36 features, only with 25 of them the error rate is interesting. For example, the pressure or the occupancy 1 hour before are not mandatory. On the other side, the rain and the occupancy 30 minutes before have a huge influence on the model.

This elimination feature helped a lot to define which input parameters are important. But it needs to be used carefully. Indeed, the final result was interesting with an error occurring each time. This wrong classification came from the time. Because of the re-sampling time was often the same and the classification was done over the time and not including the other features. Eliminating these parameters increased the overall accuracy and more precisely, eliminated this kind of error.

The accuracy of the prediction is evaluated with a scorer. It compares two columns by their attribute value and shows a confusion matrix, i.e. how many rows of which attribute and their classification match. Two outputs are provided. The first one is the confusion matrix with the number of matches in each cell. The second one reports statistics including True Positives, False Positives, True Negatives, False Negatives, Recall, Precision, F-measure and the overall accuracy. For more precision:

- True Positive: Traffic Jam correctly predicted as Traffic Jam (TP)
- False Positive: Fluid Traffic incorrectly predicted as Traffic Jam (FP)
- True Negative: Fluid Traffic correctly predicted as Fluid Traffic (TN)
- False Negative: Traffic Jam incorrectly predicted as Fluid Traffic (FN)
- Recall: It is defined as TP/(TP+FN)
- Precision: It is defined as TP/(TP+FP)
- F-measure: It is the weighted average of the precision and recall, defined as $\frac{precision*recall}{precision+recall}$

III. Settings

To do a prediction about the traffic flow, 25 features are necessary. It is split into 3 categories:

- Weather: visibility, temperature, cloud cover, humidity, wind speed, history of the rain and global weather (beautiful, cloudy,...)
- Traffic: node identification and history of the occupancy of the lane

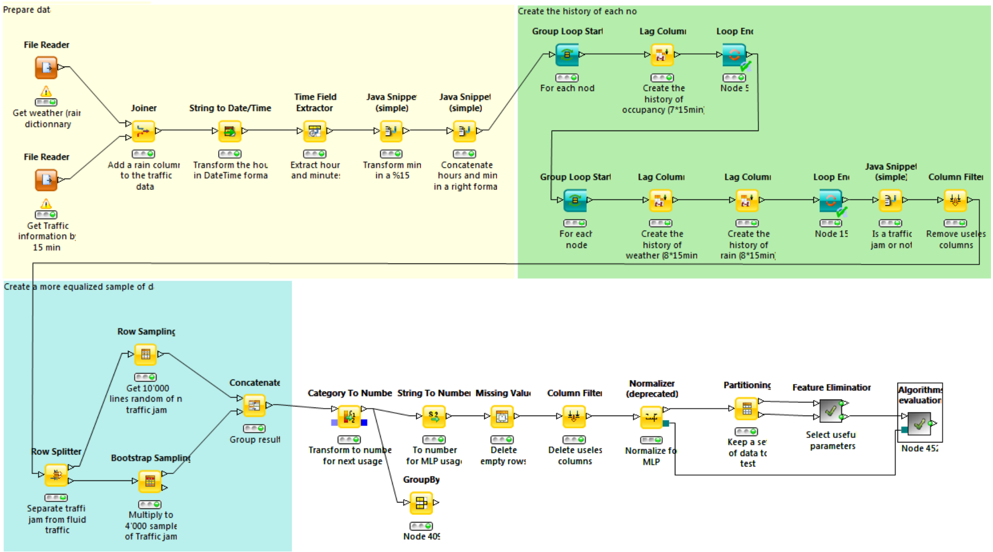


Fig. 5. Complete Workflow for the prediction model evaluation

- Other: day off or regular day

This history for the occupancy of the lane and the global weather are created for 90 minutes. For the rain, the history is for 2 hours.

Because of the usage of the MLP, data are normalized. At the end of the process, to see which data are wrongly classified, data are denormalized.

The dataset created during the pre-processing is too big for a standalone machine (a server can be used) and need to be adapted. Two months in 2013 (April and May) for the training process and two months in 2014 (April and May) for the testing process. Using similar months but during two different years helps to find a potential pattern present over years.

The second step to reduce the size of the dataset is to eliminate some rows while maintaining the information complete. The granularity is changed to 15 minutes instead of 1 minute. After experiences, this granularity is precise enough to obtain a good result.

The final size of the dataset is about 9500 records. In this dataset, 60% of them are used for the training process and 40% of them for the test.

IV. Analyse Results

In this section, results are presented and discussed. To reach the best results, three experiments are developed. We explore the dataset and have a first result on the feasibility of this prediction, we improve the result and finally, we have a working proof of concept. During these experiments, multiple algorithms are used and described.

A. Experiment 1: Basic prediction

During this first experiment, all input parameters are used to predict the class (Traffic Jam or No Traffic Jam). Furthermore, the dataset is not balanced and the number of records per class changes a lot, as presented in the Table I the detection of traffic congestions during the next 15 minutes is very bad. Indeed, in over 17 records, only 4 are well detected.

Classes	TP	FP	TN	FN	Recall	Precision	F
Traffic Jam	4	2	111010	13	0.23	0.66	0.35

TABLE I
Result table of the Decision Tree without data sampling and all parameters. F is used to represent the value of F-measure

B. Experiment 2: MLP Prediction with tuning

In the next experiment, the MLP algorithm is used to predict the traffic status during the next 15 minutes. Multi-Layer Perceptrons (MLP)[10] is a multi-layer feedforward neural network that is capable of representing an arbitrary mapping. Furthermore, some tuning is done over the dataset. First, both classes are balanced through a bootstrap sampling. By this action, the number of Traffic Jam records is increased and will be included in the learning step. Indeed, the algorithm will take them in account because they have a better weight. In addition, the feature elimination is done over the input parameters. This elimination improves the quality of the prediction by excluding some noisy features. In total, 11 features are deleted, like the pressure or some historical parameters.

Classes	TP	FP	TN	FN	Recall	Precision	F
Traffic Jam	36	24	6761	4	0.9	0.6	0.72

TABLE II
Result table of the MLP algorithm with all parameters

Table II shows the result obtained after these manipulations. The improvement from the previous experiment is large. Indeed, in over 40 traffic jam records, 36 are correctly classified. This made the accuracy grow to 90%. This accuracy is 9.5% lower than the overall one. But in this case, to detect traffic congestions, it is more precise and relevant.

C. Experiment 3: Multiple algorithms prediction

The last experiment is the most accurate one. As presented in Table III, the accuracy of the Traffic Jam class is 100% during this specific test. To validate this experiment, a cross validation is carried out and explain below.

Classes	TP	FP	TN	FN	Recall	Precision	F
Traffic Jam	40	1	2708	0	1	0.98	0.99

TABLE III
FINAL RESULT TABLE WITH A MAJORITY VOTE DECISION INCLUDING TREE ENSEMBLE, FUZZY RULE AND PNN ALGORITHMS

To obtain these excellent results, real data, multiple algorithms and the tuning explained in the methodology are made. Indeed the overall accuracy reached 99.95%. To have this result, a unique algorithm is not enough. In effect, some results are misclassified. To improve this final a result, a combination between 3 prediction methods is done.

These 3 methods are:

- Tree Ensemble applying multiple tree algorithms predictors and using a majority vote to have the final accuracy
- Fuzzy Rule generates rules based on numeric data provides a fuzzy interval for each dimension plus the target classification column
- Probabilistic Neural Network (PNN) also generates rules based on numeric data defined as high dimensional Gaussian function

The use of these methodologies give the possibility to use a majority vote decision. The most frequent class found with each algorithm is retained, thus the overall accuracy is improved.

Even if some data are wrongly classified, the error could be worse. Indeed, the wrongly classified data is a ”No Traffic Jam” class predicted as a ”Traffic Jam” class. It is better to find one more than missing a traffic congestion.

The data provided by Smart Santander has a granularity of 1 minute. The project runs on a granularity of 15 minutes. For a prediction during the next 15 minutes, less storage is necessary. Also for the project, sensors don’t need to be as efficient. Furthermore, the most important historical data that is mandatory to predict the traffic status is the rain. The algorithm needs 2 hours of history. To improve the storage, older data should be deleted. Also with the elimination feature process, it shows that that not all features are necessary. They can then be deleted. The time for calculation, besides to the storage, will be lower.

Column	Min	Max	Mean	Std. Dev.
Accuracy	0.9978	1	0.9993	0.0006

TABLE IV
RESULT TABLE OF THE CLASSIFICATION CROSS VALIDATION

As presented in Table IV, during the cross validation, the accuracy is very stable. The minimum value does not go lower than 99.7% and the standard deviation is 0.0006. This table confirms a good algorithm by demonstrating constant result of the majority vote.

V. CONCLUSION & FUTURE WORKS

This work presented a methodology to create a process to predict traffic jams, based on real data coming from Santander in Spain. After the data are cleaned and well formatted, predictive algorithms were tested to have the best accuracy. This result for 15 minutes (accuracy of 99.7%) is very good and may be be seen as ”real time” traffic analysis. Indeed, it is interesting to have this information, but often too late to change plans. A good and useful prediction is about 1 hour. Drivers are able to adapt their route and avoid traffic congestion.

The second problem of having a prediction is the action taken after. When a traffic congestion happens and the traffic is redirected, this congestion will be moved to another place. The future applications have to think about the aftermath of a traffic modification: how many cars to each road? Which cars need to stay on the road with a traffic jam?

Now that the prediction for the next 15 minutes is very good, a longer time period prediction should be implemented. This will be the next step with the improvement of the time before detecting a traffic jam. A new analysis of the important features will be carried out. The prediction quality will decline over the period of time will increase.

The second work with this analysis and project will be to implement it into another project called ”Towards a Human Centric Intelligent Society”. This project, providing to the end user an automatic help in case of danger and showing him the status of the rescue, could implement the traffic jam prediction to reduce the time for ambulance to come on site.

ACKNOWLEDGMENT

The Authors would like to thank you the Institute of Information Systems, University of Applied Sciences and Arts Western Switzerland and KNIME.com for the support and the European project ”Smart Santander” for the real time data provided.

REFERENCES

[1] Wheatley, M., 2013, *Big Data Traffic Jam: Smarter Lights, Happy Drivers.* SiliconANGLE

[2] INRIX, 2014, *Who We Are.* INRIX Inc.

[3] Ashley, S., 2011, *IBM takes traffic-jam-prediction technology for test drive.* SAE International

[4] Fox, Z., 2013, *7 Stats Proving Google’s Global Internet Domination.* Mashable

[5] Matthews, S.E., 2013, *How Google Tracks Traffic.* Connectivist

[6] Furtlehner, C., de la Fortelle, A., Lasgouttes, J.-M., 2006, *Belief-Propagation Algorithm for a Traffic Prediction System based on Probe Vehicles*. INRIA

[7] Saberi, K. M., Bertini, R. L., 2009, *Empirical Analysis of the Effects of Rain on Measured Freeway Traffic Parameters*. Portland State University, Department of Civil and Environmental Engineering, Portland

[8] Hassan, R., Nath, B., Kirley, M., 2007, *A fusion model of HMM, ANN and GA for stock market forecasting*. The University of Melbourne, Computer Science and Software Engineering, Melbourne

[9] Sanchez, Luis and al., *SmartSantander: The meeting point between Future Internet research and experimentation and the smart cities*. Future Network and Mobile Summit (FutureNetw), IEEE, 2011.

[10] Lane, Stephen H and Flax, Marshall G and Handelman, David A and Gefland, Jack J, 1990, *Multi-layer perceptrons with B-spline receptive field functions, pp 684-692*. Proceedings of the 1990 conference on Advances in neural information processing systems 3