Microscopic modelling of bus travel time by using graph properties and Machine learning

Sara Jaber, Benoit Oillo, Mustapha Tendjaoui, Neila Bhouri

Microscopic modelling of bus travel time by using graph properties and Machine learning.

Sara Jaber

BEEMOTION
6 bis, rue Trousseau - 37000 TOURS. www.beemotion.eu
And
Univ. Gustave Eiffel ex- IFSTTAR/COSYS/GRETTIA,
F-77447 Marne la Vallée, Cedex 2- France
Email: sjaber10@gmail.com

Benoit Oillo

BEEMOTION
6 bis, rue Trousseau - 37000 TOURS. www.beemotion.eu
Email: b.oillo@beemotion.eu

Mustapha Tendjaoui
Univ. Gustave Eiffel ex- IFSTTAR/COSYS/GRETTIA,
14-20 Boulevard Newton, Cité Descartes, Champs sur Marne
F-77447 Marne la Vallée, Cedex 2- France
Email: mustapha.tendjaoui@ifsttar.fr

Neila Bhouri
Univ. Gustave Eiffel ex- IFSTTAR/COSYS/GRETTIA,
14-20 Boulevard Newton Cité Descartes, Champs sur Marne
F-77447 Marne la Vallée, Cedex 2- France
Phone: +33 1 81 66 86 89
Email: neila.bhouri@ifsttar.fr

Word count: 4833
Nr of Figures: 6
Nr of Tables: 4

Submission date: July, 30, 2021
Revised submission date:

ABSTRACT

The paper presents a microscopic modeling of surface public transport travel time. Results are performed on data collected with the DIALEXIS tool, which enables very precise measurement of the vehicle travel time at each step. We proposed hierarchical modeling; firstly, machine-learning techniques are used to find the most influencing set of components among the waiting time while doors are closed. Then to the global travel time, including the waiting time while doors are open and the running time.

We compared the results of the LASSO and the Random Forest Regression methods. After retrieving the results and evaluating the models, we applied the graph algorithm of PageRank, then we trained the generated importance coefficients. Finally, we evaluated and compared all the models on two datasets, a Rapid Bus Transit and a normal bus.

Further to the travel time modeling the paper shows that graphs can be used to feed machine learning models and find new features to use for training, subsequently speeding up artificial intelligence decisions. We also concluded that the Random Forest model is most performant and robust than the LASSO.

1. INTRODUCTION

Speed and regularity are important elements for users and operators of public transport networks. Therefore, the analysis of travel time and the sources of time loss of public transport vehicles requires the greatest importance, it could give indications on the approach to be followed to improve the speed and the regularity of the overall travel time (Bhouri & al., [2016]).

In this research, we propose microscopic modeling of the public transport vehicle's travel time as a function of several components. This data is measured by the Dialexis tool, marketed in France by Beemotion company. DIALEXIS enables very precise measurement of the vehicle travel time all around the route, by measuring precise waiting and running times and splitting them into categories, making it possible to identify and quantify the lost time of public transport systems in order to improve their operational performance (Oillo, [2021]).

Our approach is to apply machine learning and centrality algorithms to find the most important component(s) of the travel time by modeling it hierarchically. First, by splitting it into a running time, a waiting time in a station while doors are opened, which is the time of exchanging passengers, and a waiting time while doors are closed. This latter being the addition of the waiting times at the station before and after the exchange of passengers, the waiting time at traffic lights, at the roundabouts, at the pedestrian crossings, and on the links between two stop points.

To perform this approach, we conducted a methodology consisting of finding first the most influencing set of components on the waiting time while doors are closed. Once extracted, this set can be added to the waiting time while doors are opened, to retrieve the most influencing components on the overall travel time of the public transport system.

The proposed methodology consists of applying a feature selection machine learning method to reduce the number of features and retrieve the importance coefficients of these features to identify the most important ones. Then improving the used models by applying centrality algorithms and integrating them into our machine learning methods.

For this purpose, we applied two methods, one is linear which is the LASSO method, and the other is non-linear which is the Random Forest selection method. These methods allow us to select the features having the most impact on the travel time and eliminate those that are less important. After retrieving the results and evaluating the models, we applied the graph algorithm of PageRank, then we trained the generated importance coefficients. Finally, we evaluated and compared all the models to find the most accurate for our data.

To evaluate this approach and compare the models' accuracy we used the Root Mean Squared Error (RMSE), which is the square root of the variance, and R-Squared score that calculates the percentage of the variance in the dependent variable, as main criteria. The methods are programmed with Python Scikit-learn library.

To create the database and apply graph centrality algorithms we used the Neo4j tool, having a Graph Data Science (GDS) library, with a set of connected graph algorithms like degree centrality, eigenvector, and PageRank.

The following section of the paper describes the collected data, the specifications of the routes and the decomposition of the travel time; section 3 explains how datasets are modeled in a graph-oriented database. In section 4 we give a brief presentation about the feature selection methods we used in the research, namely LASSO and Random Forest. Section 5 is about the PageRank centrality algorithm and its use to retrieve the importance coefficient of each feature in the database. In section 6, we show the results of applying the LASSO and Random Forest methods on the two datasets of Line1 and Line2 before and after integrating the PageRank scores, we also evaluate the models and compare the results. Finally, section 7 gives the conclusion of the comparison.

2. DATA DESCRIPTION

The datasets used for the implementation are the DIALEXIS feeds for two bus lines in two different cities. The first, called Line1, is a chronobus (a Bus Rapid Transit) that takes advantage of the TSP (Transit Signal Priority), it ensures the connection between 15 stop points, each direction, over an area of 6 km. It is one of the busiest buses in the city, with a frequency of 4 minutes. The second bus, called Line2, is a normal bus without TSP, which serves 22 stop points, each direction, over an area of 3.5 km. It is a bus passing on the borders of the city, with a frequency of a bus every 10 minutes. The data is highly related to the topology of each line, that’s why a topology sheet is provided with each dataset.
We extracted on table sheet a total of 1851 completed bus trips for Line1, for one month (from 16 September 2019 – 18 October 2019), and 1761 completed bus trips for the Line2, scattered over 5 months (March, April, November, December 2019 and January, February 2020).

According to the DIALEXIS data, the travel time is the addition of the vehicle's running time to all the waiting durations:

$$TP = AS1 + EP + AS2 + AL + AF + AP + APP + ASF1 + ASF2 + TR. \quad (1)$$

- TP: The overall travel time of the vehicle.
- AS1: The waiting time of the vehicle before exchanging passengers in the station (loss of time while doors are closed before EP).
- EP: The waiting time of the vehicle while exchanging passengers in the station (loss of time while doors are open).
- AS2: The waiting time of the vehicle after exchanging passengers in the station (loss of time while doors are closed after EP) .
- AL: The waiting time of the vehicle while running on a link between two stop points.
- AF: The waiting time of the vehicle in a traffic light.
- AP: The waiting time of the vehicle in a roundabout.
- APP: The waiting time of the vehicle in a pedestrian cross getaway.
- ASF1: The waiting time of the vehicle before exchanging passengers in a station having a traffic light (loss of time while doors are closed before EP).
- ASF2: The waiting time of the vehicle after exchanging passengers in a station having a traffic light (loss of time while doors are closed after EP).
- TR: The link running time of the vehicle.

3. GRAPH-ORIENTED DATABASE

Graph technology is the fastest-growing category of databases in recent years. In a world where connected data represents a new source of business, graph technology is the obvious choice.

Since the objective of our study is to measure the impact of endogenous variables on the travel time of vehicles intended for public transport, using the microscopic composition of the travel time provided by DIALEXIS, we need to know the relationship between all these components, so that we will be able to study the correlation and apply feature selection methods. This means that we need to store complex relationships, query relationships based on highly connected data, and manipulate interconnected data. The graph database responds perfectly to these requirements, for the reasons of being non-relational, distributed, and horizontally scalable.

The datasets are processed to create a graph database using Neo4j, which is a graph-based database management system that uses the Cypher language for requests. Our graph is containing two connected subgraphs, one for the topology of the lines containing the stop points, and the other is for the events happening all over the trip and storing the waiting time of each event. These events are the attributes provided by DIALEXIS and developed in the data description. Each event is connected to its related stop point with a weighted relationship.

In the topological subgraph, a sequence of geographic locations represents a bus trip through the transportation network, and the events subgraph represents a discrete sequence of events that occur on each excursion. Figure 1 shows the graph structure of this database.

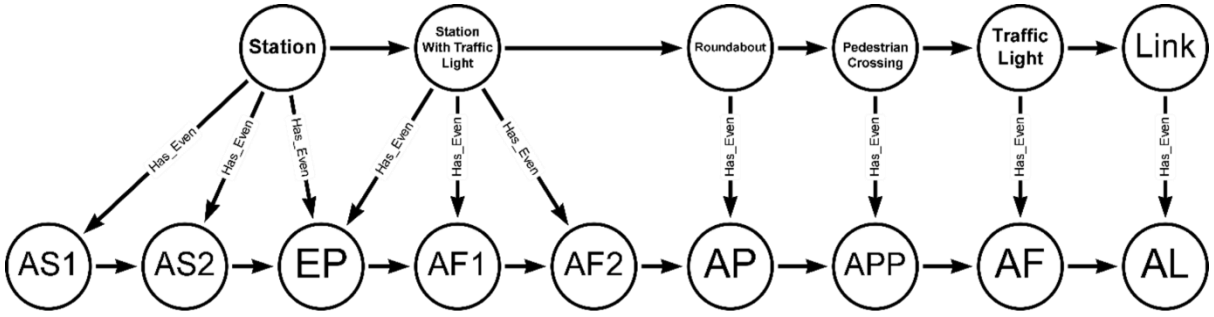

Figure 1: An overview of the graph

This database is scalable and can easily evolve in function of time and bus lines that will introduce DIALEXIS in their systems.

4. FEATURE SELECTION METHODS

The performance of a learning algorithm depends strongly on the features used in the learning task. The presence of redundant or irrelevant features can reduce this performance. Feature selection is generally denoted as a search process to find a relevant subset of features among those in the starting dataset. Feature selection is the process of determining the subset of extracted features that are most important or influential to a target goal. It is used to surface predictive importance as well as for efficiency (Needham & al. [2019]).

Determining feature importance and making feature selection in a machine learning model is an unmissable step. As a result of this step, we obtain a subset of features having each an importance measure. Thus, the most important features can be selected appropriately. This step can reduce the errors in the machine learning model caused by the noise of less important features.

In the literature, the existing feature selection methods are divided into three categories according to the type of selection criteria and how it is considered in the classification procedure. The first category, called “filter”, evaluates the importance of the variables according to measures that are based on the properties of the training data, the evaluation is usually done independently of a classifier (John & al. [1994]). The main disadvantage of the “filter” approach is that it doesn’t consider the correlation between features. The second category, called “wrapper”, evaluates features using a classifier that estimates the relevance of a given subset of features. The complexity of this algorithm makes “wrapper” methods very expensive, and the time required for the selection of features is longer than that of the “filter” approach. It also performs the evaluation by a single classifier, which is the second limitation of this approach. Finally, the third category, called "embedded", combines the selection of variables and the estimation of the model in a single task. This method is faster than the "wrapper" method because it avoids that the classifier restart from zero for each subset of features.

Based on this, and to avoid possible correlation problems between the components used to model the travel time (see equation 1), we chose to use the “embedded” approach for our

research to avoid the inconveniences mentioned for the “filter” and the “wrapper” methods. Among, embedded methods the LASSO (Least Absolute Selection and Shrinkage Operator) and the Random Forest seem the more suitable for our research.

4.1 Feature selection with LASSO:

The LASSO (Least Absolute Selection and Shrinkage Operator) is a penalized model approach that solves the problem of multicollinearity between variables in situations where all variables are kept. In LASSO, we look for estimators with a smaller variance by removing the effect of certain explanatory variables, which means assigning them a zero as an importance coefficient. This can result in a model with fewer explanatory variables. This penalization is very advantageous when the number of variables is high because the variance leads in some cases to strong prediction errors.

This method allows to compute the classifier and perform the variable selection at the same time. The idea is to find the estimator β that minimizes a penalized objective function:

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{argmin}_{\boldsymbol{\beta}} \{(\boldsymbol{\beta}, \boldsymbol{D}) + \boldsymbol{\lambda} * \boldsymbol{pen}(\boldsymbol{\beta})\} \quad (2)$$

The lambda parameter λ of the equation allows us to find a compromise between the complexity of the model and its approximation to the data. A λ that tends towards 0 will lead to a greater complexity and thus potentially bad predictions. Conversely, a λ that tends to infinity gives a more general model but can ignore information in the dataset.

The pen() function refers to the penalty function. A properly parameterized and computed penalty function will result in estimates of β strictly equal to 0. This is how variable selection is performed.

The LASSO regression is penalized by the L^1 norm of β coefficients. We define the penalization criterion by:

$$\boldsymbol{pen}(\boldsymbol{\beta}) = \sum_{\boldsymbol{j=1}}^{\boldsymbol{p}} |\boldsymbol{\beta_j}| \quad (3)$$

This method of selection is not adequate in the case where input variables are highly correlated with the target variable in the dataset.

4.2 Feature selection with Random Forest:

The Random Forest is often used for feature selection in a data science workflow. This is because of the tree strategies used by random forests naturally rank according to how they improve node purity. This algorithm is known to be one of the most efficient "out-of-the-box" classifiers (requiring little data preprocessing). It is among the most popular machine learning methods due to its relatively good accuracy, robustness and ease of use.

The search space for the construction of tree nodes is limited by P characteristics randomly chosen. The performance of the method depends directly on the parameter P. A small value of

P risks degrading the performance of the classifier. According to (Breiman [2001]), the optimal value of P is: P = $\sqrt{N}$, where N is the total number of features.

The importance of the feature is calculated as the decrease in node impurity weighted by the probability of reaching that node. The node probability can be calculated by the number of samples that reach the node divided by the total number of samples. The higher the value is, the more important the feature is considered.

In the context of ensembles of randomized trees, (Breiman, [2001, 2002]) proposed to evaluate the importance of a variable Xm for predicting Y by adding up the weighted impurity decreases $p(t)\,\Delta i(s_t, t)$ for all nodes t where Xm is used, averaged over all N_T trees in the forest:

$$\mathrm{Imp}(X_m) = \frac{1}{N_T} \sum_T \sum_{t \in T: v(s_t) = X_m} \mathrm{p(t)}\ \Delta \mathrm{i(st, t)} \qquad (4)$$

where p(t) is the proportion N_t / N of samples reaching t and $v(s_t)$ is the variable used in split s_t (Louppe & al. [2013]).

5. GRAPH CENTRALITY PAGERANK ALGORITHM

When searching for information using graphs, whether social graphs, transit graphs or communication graphs, the ranking of the results is based directly on the degree of importance (or authority) of a node in the graph.

Centrality algorithms are an excellent tool for identifying influencers in a network. There are many wide-ranging uses for centrality algorithms for a variety of analyzes, we cite among others betweenness centrality, closeness centrality, ArticleRank, degree centrality, eigenvector, and PageRank. This last is one of the most significant link analysis algorithms in the field of web search (Richardson & al. [2006]).

Since the goal of our study is to detect the influence of the endogenous factors on the bus travel time, and as these factors are modeled as nodes in a graph-oriented database, we can therefore apply the PageRank algorithm on a projected graph to give an importance score to each node based on the waiting time it stores. These scores can be integrated with the feature selection method to find the most influencing features and eliminate the less important ones.

Originally, the PageRank calculation mathematical formula is defined by having A1, A2, ..., An, n nodes pointing to a node B. Let us denote by PR (Ak) the PageRank of the node Ak, N (Ak) the number of outgoing links present on the node Ak, and by d the damping factor between 0 and 1, usually set at 0.85. The damping factor defines the probability that the next click will be through a link, so the PageRank score represents the likelihood that a node is visited through an incoming link and not randomly.

Then the PageRank of node B is calculated from the PageRank of all the Ak nodes as follows:

$$PR(B) = (1-d) + d \times (PR(A1) / N(A1) + ... + PR(An) / N(An)). \quad (4)$$

This is an iterative formula that updates the rank of a node until it converges or meets the set number of iterations. This formula will be multiplied by the weight of the relation pointing to node B if the graph is weighted (quality measurement principle).

6. EXPERIMENTAL RESULTS

In order to reduce the dimension of the learning set, we applied the LASSO and the Random Forest methods on the two datasets of Line1 and Line2. The methods are applied, first, on the variables constituting the waiting time of vehicles while the doors are closed. That is, the waiting time formed by AS1, AS2, ASF1, ASF2, AL, and AF. This allows us to select the features that are making the big loss of time in closed doors, and we can then create a final set of features with the Exchange Passengers waiting time (EP) and the Running Time (TR) to identify the most influencing features on the overall travel time.

For this purpose, we created a new variable that stores all the waiting times in closed doors called APF.

$$APF = AS1+AS2+ASF1+ASF2+AL+AF+AP+APP. \quad (5)$$

6.1 Database description:

The tables 1 and 2 describe the datasets of Line1 and Line2 with statistical measures for each set of features like the count, the mean, the standard deviation, the minimum and the maximum values. As we can see, the TR (Running Time) and the EP (Passengers Exchange at bus stops) have the biggest mean and max values, and unlike other variables, they have a min > 0. This makes sense and comforts our methodology for the hierarchical analysis, splitting the travel time into three parts (TR, EP, and APF), where APF is the sum of all the wasted while doors are closed.

The size of the samples for each line and the number of features is not so big, that's why we tend to use feature selection methods that are applicable and useful with this kind of datasets. A recent paper (Lee & al. [2018]) shows that LASSO method is performent on small and large datasets, as well as Random Forest (Breiman [2001]).

Table 1: Line1 database description

	AS1	AS2	AP	APP	AF	AL	ASF1	ASF2	APF	EP	TR	TP
count	1851	1851	1851	1851	1851	1851	1851	1851	1851	1851	1851	1851
mean	3	19	3	3	11	1	0	39	82	237	832	1139
std	10	14	15	14	28	7	3	25	53	89	63	148
min	0	0	0	0	0	0	0	0	10	34	640	731
25%	0	11	0	0	0	0	0	21	46	179	787	1041
50%	0	15	0	0	0	0	0	36	70	227	828	1126
75%	0	23	0	0	9	0	0	56	102	285	874	1222
max	118	180	280	164	289	121	46	186	519	635	1070	1794

Table 2: Line2 database description

	AS1	AS2	AP	AF	AL	APF	EP	TR	TP
count	1761	1761	1761	1761	1761	1761	1761	1761	1761
mean	2	35	12	127	4	207	201	1264	1692
std	7	22	28	89	21	100	68	105	199
min	0	1	0	0	0	26	26	877	997
25%	0	22	0	78	0	146	158	1193	1566
50%	0	30	2	115	0	191	203	1261	1687
75%	0	42	16	155	1	245	241	1329	1823
max	91	226	796	862	555	985	539	2097	3028

To apply the LASSO and the Random Forest methods, it's important to measure the correlation between the input variables AS1, AS2, ASF1, ASF2, AL, AF, and the dependent one APF. Table 3 presents the correlation measures of all the input variables with the APF using the Pearson Correlation Method. The measures show that all the coefficients are less than 0.5, which means that the variables are not highly correlated.

Table 3: Correlation measures between the waiting in closed doors variables and APF for Line1 and Line2

	AS1	AS2	AP	APP	AF	AL	ASF1	ASF2
APF Line1	0.429	0.057	0.059	0.454	0.076	0.294	0.015	0.491
APF Line2	0.227	-0.047	0.056	-	0.039	0.211	-	-

6.2 Results without PageRank:

After fitting the train data into the LASSO method, it eliminated some features by giving them zero coefficient. We can visualize the feature importance calculated scores. Figure 2 shows the results of the LASSO training, while figure 3 displays the results of the Random Forest method on datasets of Line1 and Line2 respectively. The LASSO method eliminated the ASF1 for Line1 and AS2 for Line2, while the Random Forest choose to keep the AS2 and ASF2 for Line1 and the AF and AS2 for Line2.

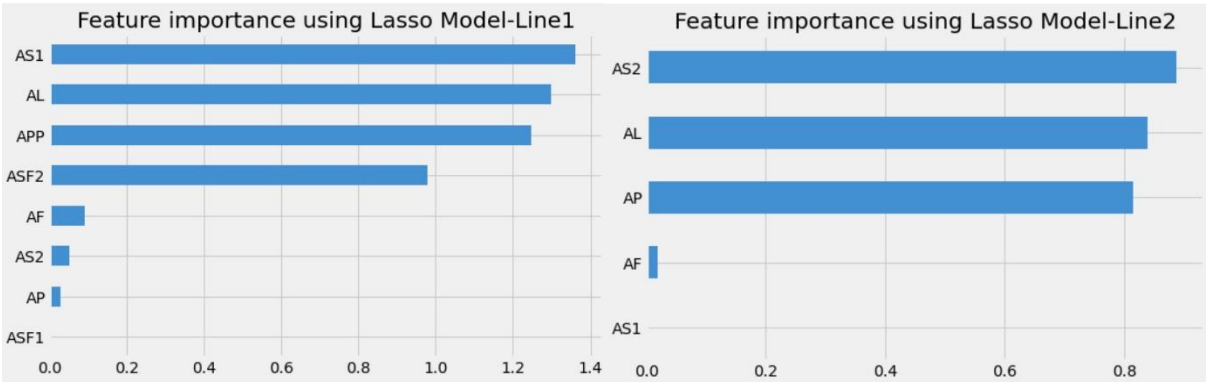

Figure 2: Feature importance using LASSO for Line1 and Line2

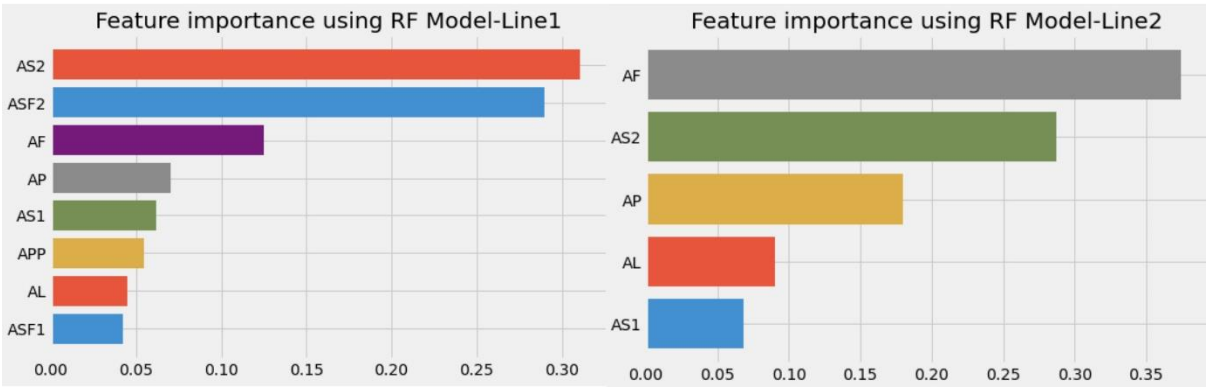

Figure 3: Feature importance using Random Forest for Line1 and Line2

6.3 Results with PageRank:

We created a Cypher projected graph in Neo4j containing all the nodes representing the waiting time in closed doors including the APF. It is a weighted graph having the waiting time as weight. Then we applied the PageRank algorithm with max iterations 20 and damping factor 0.85. Figure 4 shows the results of the integration of the PageRank scores with the LASSO model, it chooses the AS2 and ASF2 for Line1 and AF and AS2 for Line2, as the most impacting features on the APF. Figure 5 displays the results of the integration of the PageRank scores with the Random Forest model, it indicates that ASF2 and AS2 for Line1 and AF and AS2 for Line2 are the most important features that impact the APF.

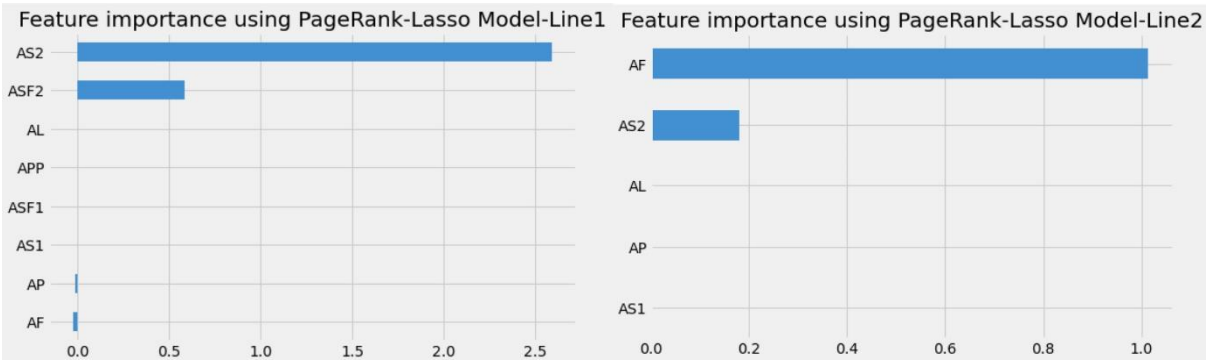

Figure 4: Feature importance using PageRank-LASSO for Line1 and Line2

Figure 5: Feature importance using PageRank-Random Forest for Line1 and Line2

6.4 Evaluation of the models

To measure the performance of the models, we evaluated them using the regression metrics. We used the mean absolute error MAE to calculate the prediction error, the root mean squared error RMSE to indicate the absolute fit of the model to the data, this is an occuration measure for regression models. The less is the value of RMSE is, the better the model is. We also used the r2 score to evaluate how well the regression model fits the observed data, it corresponds to the squared correlation measure between the actual value in the dataset and the predicted one. A low score of r2 is a sign of a bad model.

Table 4 resumes the evaluation scores for the two lines datasets trained with LASSO and Random Forest with and without PageRank integration.

Table 4: Comparison of evaluation results

Evaluation	Line1				Line2			
	LASSO		RF		LASSO		RF	
PageRank	**Without**	**With**	**Without**	**With**	**Without**	**With**	**Without**	**With**
RMSE	29.66	0.4	55.66	0.31	89.65	0.07	118.9	0.052
R2	0.77	0.97	0.44	0.98	0.086	0.996	0.45	0.99

Comparing the results, we can detect an impressive improvement in the machine learning models after integrating the graph algorithm of PageRank. The LASSO model without

PageRank had a root mean squared error of 29.66 for Line1 and 89.65 for Line2, meanwhile, the same error gave a value of 0.4 for Line1 and 0.07 for Line2 after using the PageRank scores in the LASSO training dataset. Also, the Random Forest model had a root mean squared error of 55.66 for Line1 and 118.9 for Line2 before integrating the PageRank scores, and it got 0.31 and 0.052 for Line1 and Line2 respectively after the integration.

After comparison, we can deduce that the models are improved after integrating the PageRank algorithm, and we can notice also that the Random Forest model gives the same subset of features as the model with PageRank even if the error was bigger than the error in the LASSO model, which means that the Random Forest model was more performant and more robust in respect to noise than the LASSO.

6.5 MODELING THE RESULTS

Now that we know for each bus line what are the features that influence the most on the waiting time while the doors are closed, the next step is to model this subset of features with the waiting time while doors are opened (EP) and the running time (TR) to find the most influencing features on the overall travel time (TP).

In the experimental results, we found that the Random Forest model is most performant and robust than the LASSO. We found also that the integration of the PageRank scores improve impressively the feature selection models. Based on this, we will adopt the subset of features chosen by the Random Forest with PageRank model: (ASF2, AS2) for Line1 and (AF, AS2) for Line2, to be the subset of feature that most influence the waiting time in closed doors. Thus, after reducing the dimension of the learning set of features, the travel time formulas will be as following:

Line1: $TP = TR + EP + AS2 + ASF2.$ (6)
Line2: $TP = TR + EP + AS2 + AF.$ (7)

Figure 6 displays the results of applying Random Forest with PageRank method on each subset for Line1 and Line2. The model chooses EP as the most influencing feature on the travel time (TP) for Line1, and TR for Line2. This result shows that the factors that impacts the most the travel time (TP) are strongly dependent on the nature of the bus route. As Line 1 is a BRT, it has a very important number of passengers, therefore an important waiting time to load (EP). As a BRT has a dedicated road lane it has a low running time (TR). This is the contrary of the normal bus route (Line2) where the running time is the most important, followed by the loading time and stops at traffic lights at third.

Evaluation of the model: for Line1, RMSE = 0.004 and R2 = 0.999, and for Line2, RMSE = 0.12 and R2 = 0.72, this indicates that the 2 models are performant.

Figure 6: Feature importance using PageRank-Random Forest with TP as target for Line1 and Line2

7. CONCLUSION

We propose in this paper the modelization of a transportation network dataset in graph-oriented database via Neo4j, and to use the LASSO and the Random Forest methods for feature selection to retrieve the most influencing features on the waiting time of two bus lines while the bus doors are closed. We proposed also to enhance these methods with the graph algorithm PageRank that gives an importance score for every feature.

In conclusion, graphs can be used to feed machine learning models and find new features to use for training, subsequently speeding up artificial intelligence decisions. Graph centrality algorithms such as PageRank identify influential features to feed more accurate machine learning models and measurable predictive lift. Once we have extracted connected features, we can improve our training by using graph algorithms like PageRank to prioritize the features with the most influence. This enables us to adequately represent our data while eliminating noisy variables that could degrade results or slow processing. With this type of information, we can also identify features with high co-occurrence for further model tuning via feature reduction.

Concerning the bus travel time, our study shows the importance of the topology and nature of the bus line on the microscopic model. It shows that for a BRT (bus Line1) the most important part of travel time corresponds to the loading time, followed by running time. The third most important part of the travel time is the waste time doors closed at bus stops after loading passengers, either this stop is linked to a traffic light or not. For a normal bus route (Line 2) the most important is the running time, followed by the loading time and comes after the waiting time at traffic lights.

REFERENCES

1. Breiman, L., “Random Forests”, *Machine Learning* 45, 2001, pp. 5–32.
2. Breiman, L., “Manual on setting up, using, and understanding random forests v3.1”, Statistics Department University of California Berkeley, CA, USA., 2002, pp. 1-29.
3. Richardson, M., Prakash, A., Brill, E., “Beyond PageRank: Machine Learning for Static Ranking”, *Association for Computing Machinery*, 2006, pp. 707-715.

4. Needham, M., Hodler, A., “Graph Algorithms: Practical Examples in Apache Spark & Neo4j”, *O’Reilly*, 2019, pp. 1-257.
5. Oillo, B., “La performance opérationnelle des systèmes de transport collectif : pour une analyse microscopique des conditions d’exploitation”, *Groupement pour l'Étude des Transports Urbains Modernes*, No. 138, 2021, pp. 34-40.
6. Bhouri, N., Aron, M., Scemama G.; “Gini Index for Evaluating Bus Reliability Performances for Operators and Riders”. Transportation Research Board, 95th TRB annual meeting. 10–14, 2016, Washington D.C.
7. John, G., Kohavi, R., Pfleger, K., “Irrelevent features and the subset selection problem”, *Morgan Kaufmann Publishers,* 1994, pp. 121-129.
8. Louppe, G., Wehenkel, L., Sutera, A., Geurts, P., “Understanding variable importances in forests of randomized trees”, *NIPS*, 2013, pp. 1-9.
9. Lee, C. Y., Cai, J. Y., “LASSO variable selection in data envelopment analysis with small datasets”, *Omega (United Kingdom)*, 2018, pp. 1-9.