Machine Learning-Based Approach for Estimating the Quality of Mobility Policies

Miljana Sulajkovska
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
miljana.sulajkovska@ijs.si

Maj Smerkol
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
maj.smerkol@ijs.si

Erik Dovgan
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenija
erik.dovgan@ijs.si

Matjaž Gams
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenija
matjaz.gams@ijs.si

ABSTRACT

Cities are increasingly turning towards specialized technologies to address issues related to their significantly increased transport demand. Municipalities and transport authorities try to face these problems in order to achieve their objectives by taking various actions in the domain of public transport, air and noise pollution, road accidents, etc. The primary objective of this research is to explore the role of machine learning (ML) in mobility policy quality estimation using microscopic traffic simulations. The main idea is to use one simulation run as one training example. The features are represented by several group of parameters that are related to the input and output of the simulation, while the target variables are represented using key performance indicators (KPIs). The city of Bilbao is chosen as a use case. We have analyzed how closing the Moyua square in the city center and changing the number of cyclists there can affect the air pollution by estimating the CO2 emissions. Several machine learning algorithms are tested and the results show that by closing the main square in the city center and increasing the number of cyclists the CO2 emissions reduce.

KEYWORDS

machine learning, smart cities, mobility policy

1 INTRODUCTION

According to United Nations population estimation, the total population is exponentially increasing and by 2050 will reach 9 billion, i.e. it will increase for 2 billion from now [11]. This demographic growth will greatly impact on the transportation system in metropolitan areas since most population will be located there. As a result far more attention must go towards serving the needs and aspirations of the people with the aim to maintain the environmental, social, and economic costs at the same time [12].

In this context different mobility policies are tested and evaluated in order to achieve the desired city goals. Since implementing different scenarios in real life is an expensive process microscopic traffic simulations are widely used as a valuable support tool for evaluating transportation facilities or systems. Using the simulations we can see how some actions may impact the dimensions that we are interested in without making those changes in real life.

Currently, most of the mobility policy evaluation techniques rely on experts in urban/spatial planning using on simulation results [10]. Since the simulations create large amount of data, including data from optimization steps, various data analysis can be applied. In this context machine learning techniques can be applied to automate the evaluation of mobility policies and address the objectives of the cities.

As part of the URBANITE project we are developing a machine learning module using data from microscopic transport simulator that will help decision makers in the what-if analysis. More precisely, we propose a system to estimate the quality of previously simulated mobility policies using machine learning methods.

The rest of the paper is structures as follows. Section 2 explains the URBANITE approach and the relevant modules for this research. In Section 3 the data collection process is explained. Then, Section 4 presents the results of the machine learning module. Finally, Section 5 concludes the paper with ideas for future work.

2 OVERVIEW OF THE URBANITE APPROACH

The main objective of URBANITE approach is to build an intelligent platform that can use data from heterogeneous sources in order to help the city managers in the decision-making process. To achieve this aim, several modules are developed. In this section we will give an overview of only the relevant ones shown in Figure 1

The traffic simulator is used to simulate various mobility policies during the the policy evaluation process. The input files to this module are related to the network map, travel demand, public transit data etc. Based on the simulation output, target variables e.g. air pollution levels for the machine learning algorithms are calculated, based on which the models are build. This approach is able to process large amount of data in order to find the best mobility policy.

Besides automatic selection of mobility policy URBANITE also supports policy selection by the experts with the use of the decision support system. In addition to processing the simulation data this system also relies on expert knowledge in order to build the hierarchical decision models and satisfy the user preferences.

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

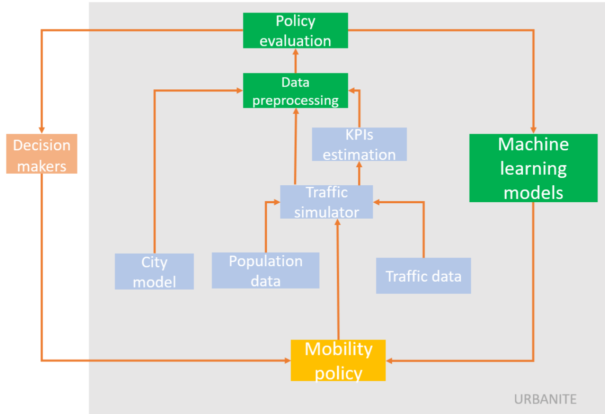

Figure 1: Modules of URBANITE approach.

Both approaches are used in the policy evaluation process with the difference that machine learning module relies on algorithms that automatically select the best mobility policy. In the next sections the simulation and machine learning process are described in detail.

3 DATA COLLECTION

3.1 Simulation

In order to collect the data, a microscopic traffic simulation tool was used. Several state-of-the-art solutions were tested and MATSim was chosen as the most suitable one. MATSim [6] is an open-source tool implemented in Java. It is used for microscopic modeling that enables us to simulate and analyze components on the network such as traffic flow, congestion, public transport, behavior of cyclists, etc. One of the core concepts is the co-evolutionary optimization where the individuals' plans are evolving in the presence of all other persons doing the same.

To run the simulator several input files need to be provided that are related to the city model, traffic and population data. For the creation of the transport demand e.g. persons with their daily plans and mode of transport real data from census and other travel surveys is required. Since there is no complete dataset containing the socio-demographic characteristics of individuals at a small geographic scale because of privacy concerns a transport demand was generated based on known random variables.

After providing all the required input we can run the simulation which is optimized by configurable number of iterations (see Figure 2). Each individual agent learns by maintaining multiple plans which are scored by executing them in the mobsim, selected according to the score and when needed, modified. The iterative process consists of the following steps:

- Mobsim simulation
- Scoring
- Replanning

Every iteration starts with an initial demand simulated by the mobility simulation and then evaluated by the scoring module as a central element of the simulator [9].

The MATSim scoring module evaluates the performance of a plan in a synthetic reality and determines the choice of person's plan in the next iteration. Next, only plans with higher scores are selected by the agent - others are deleted in the replanning step. The scores are computed using scoring function taking into account the performance of activities and travel time. A typical score is calculated as follows:

$$S_{\text{plan}} = \sum_{q=0}^{N-1} S_{\text{act},q} + \sum_{q=0}^{N-1} S_{\text{trav,mode}(q)} \tag{1}$$

Here utility functions are used to represent a basic scoring function or in other words the utility of a plan S_{plan} is computed as sum of all activity utilities $S_{\text{act},q}$, plus the sum of all travel utilities $S_{\text{trav,mode}(q)}$. N represents the number of activities. For scoring, the last activity is merged with the first activity to produce an equal number of trips and activities. Positive scores are obtained for desired events and negative for unwanted ones.

Finally, the optimization step takes part where four dimensions are considered: departure time, route, mode, and destination. Each of the agents has a memory of M plans that have been observed in the past and which have obtained a score. In the first step the replanning process checks whether the agent's memory exceeds the limit. If so one of the existing plans is removed according to previously computed scores. If the plan is removed that was currently selected for execution, a random one among the remaining ones is selected. After a certain number of iterations an equilibrium state on the network is reached improving the initial scores.

Several files are produced as output of the simulation that are related to specific iteration or they summarize a complete run, e.g. events file that contains every action taken on the network, and travel distance statistics showing the distance traveled per mode. These results are used to compute the features for the machine learning module and to define the target variable. More precisely the input features consists of simulation input and output data which can be directly influenced by the user. On the other hand the target variables depend on the simulation results and cannot be directly set by the user which makes them relevant for the decision-making process of particular mobility policy. These variables are summarized in Table 1.

Additional MATSim package was used to calculate the CO_2 emissions which are used as a target class in the prediction process. The tool calculates warm and cold-start exhaust emissions by linking MATSim simulation output to detailed emission factors for road transport.

3.2 Scenarios

In order to gather the dataset, 14 simulations were executed by applying different policies in the city of Bilbao, Spain. The main objective is to see the impact of closing the Moyua square in the city center for private traffic.

Two scenarios are implemented: the baseline scenario of the current network situation and the modified scenario representing the closure of Moyua square as one possible policy. All other

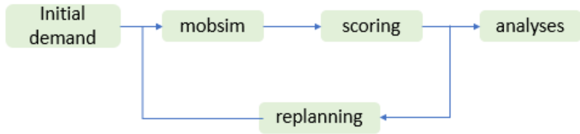

Figure 2: MATSim cycle.

Table 1: Input data and target variable of the machine learning module

ML input		Target variable
Sim input	Sim output	
Surface of a road Capacity of road Number of lanes Type of district Number of bus stops	Number of cars Number of cyclists Number of public transport vehicles Average travel time	CO_2 emission

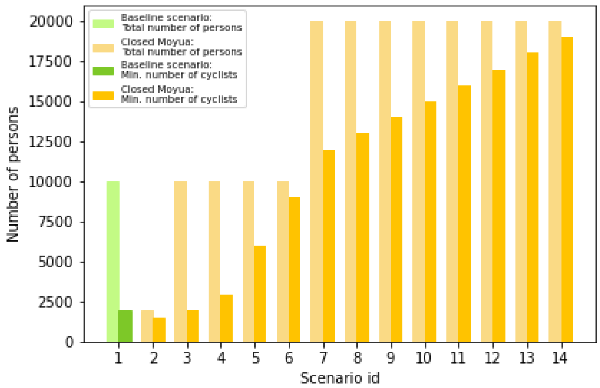

Figure 3: Number of persons and cyclists per scenario instance.

instances are variations of the second scenario where the number of cyclists varies from 1500 to 19000 while changing the number of inhabitants from 2.000 to 20.000, respectively. This change in number of cyclists does not represent a specific policy but shows what would happen if in conjunction with the applied policy also the number of cyclists changes. Figure 3 depicts these variations where with green is marked the baseline scenario and with orange the other scenario and all its variations. The first instance represents the baseline scenario with 10000 inhabitants and 2000 cyclists, while the remaining instances represent the second scenario with all variations. The second instance contains 2000 inhabitants with at least 1500 cyclists. The next four instances are representing 10000 population with up to 9000 cyclists while reducing the private transport. The rest of them represents 20000 inhabitants with up to 19000 cyclists. The number of public transport vehicles stays the same in all variations.

Figure 4 shows the results of the applied policy. More precisely it shows the relationship between number of cyclists and the level of CO_2 emissions. The x-axis represents the number of cyclists nearby the square and and y-axis represents number of cyclists in the center. The different colors denote the amount of CO_2 emissions as a target variable where with orange circle is marked the baseline scenario. This figure shows that by closing the main square for private traffic and reducing the number of private vehicles nearby it, the level of CO_2 emission is decreasing.

4 MACHINE LEARNING

4.1 Methods

Several machine learning models were applied using Orange [3]: k-Nearest Neighbors, Decision Tree, Support Vector Machines,

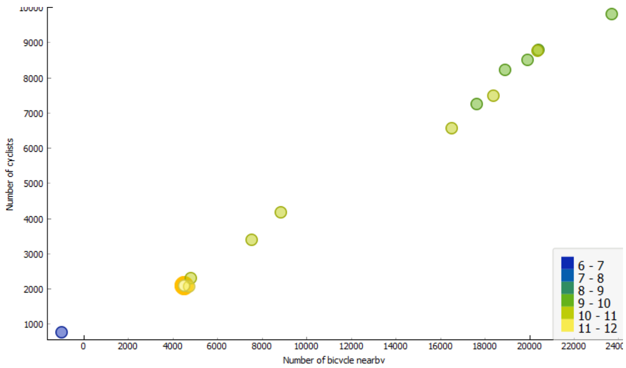

Figure 4: CO_2 **emissions in the Moyua square. The baseline scenario is marked with orange circle.**

Random Forest, Linear Regression, Gradient Boosting, and Neural Network.

The k-nearest neighbor (kNN) is a semi-supervised learning algorithm that requires training data and a predefined k value to find the k nearest data based on distance computation. If k data have different classes the algorithm predicts class of the unknown data to be the same as the majority class [1].

Tree splits the data into nodes by class purity. The top-most node is called root, the bottom ones leaves, and all other nodes are internal nodes connected to each other with edges. Each edge represents satisfaction of the node condition, and each leaf node determines the class assigned to the instances that met the conditions of the internal nodes on the path from the root node to the leaf node [5].

Support Vector Machines (SVM) is a two-grouped classifier where input vectors are non-linearly mapped to a high-dimension feature space. In this feature space a linear decision surface is constructed. Special properties of the decision surface ensures high generalization ability of the learning machine [2].

Random Forest consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes model's prediction. A large number of relatively uncorrelated models (trees) operating as a committee outperform any of the individual constituent models.

Linear Regression is commonly used in mathematical research methods, where it is possible to measure the predicted effects and model them against multiple input variables. It is a method of data evaluation and modeling that establishes linear relationships between variables that are dependent and independent [8].

Gradient Boosting tries to convert weak learners into strong ones by training many models in a gradual, additive and sequential manner where the gradient of the loss function is being minimized, with respect to the model values at each training data point evaluated at the current step [4].

Neural Network model simulates a large number of interconnected processing units that resemble abstract versions of neurons where the processing units are arranged into layers. The units are connected with varying connection strengths (or weights). Input data are presented to the first layer, and values are propagated from each neuron to every neuron in the next layer. Eventually, a result is delivered from the output layer. The network learns by examining individual records, generating a prediction for each record, and making adjustments to the weights whenever it makes an incorrect prediction. This process is repeated many times, and the network continues to improve its predictions until one or more of the stopping criteria have been met [7].

4.2 Evaluation Results

We have evaluated the machine-learning algorithms described in Section 4.1. The data for evaluation of the algorithms is split randomly 10 times and the average results are computed. We have compared four evaluation metrics:

- Mean squared error (MSE) measures the average of the squares of the errors or deviations (the difference between the true and estimated values).
- Root mean squared error (RMSE) is the square root of the arithmetic mean of the squares of a set of numbers (a measure of imperfection of the fit of the estimator to the data).
- Mean absolute error (MAE) used to measure how close forecasts or predictions are to eventual outcomes.
- R2 is interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variables.

The results are shown in Table 2. According to MSE, RMSE, and R2 the best model is kNN, while according to MAE the best model is SVM.

Table 2: Evaluation results

Model	MSE	RMSE	MAE	R2
kNN	4.718	2.172	1.416	-0.372
Tree	5.387	2.321	1.431	-0.567
SVM	4.953	2.225	1.298	-0.441
Random Forest	5.093	2.257	1.404	-0.482
Neural Network	11.264	3.356	2.583	-2.277
Linear Regression	8.076	2.842	2.041	-1.349
Gradient Boosting	5.193	2.279	1.324	-0.511

5 CONCLUSION

In this paper we showed how machine learning can be used in mobility policy evaluation helping the urban development in cities. As large amount of data is produced from simulations, machine learning techniques can be applied to automatically choose the best policy.

We defined the mobility policy for Bilbao. Then, using microscopic traffic simulation the two scenarios were implemented: baseline scenario of the current network state and the modified scenario representing closure of Moyua square for private traffic. In order to gather more data, variations of the second scenario were produced by changing the proportion of cyclists and private car users. After gathering sufficient data, machine learning techniques were applied to evaluate the performance of the policy. Changing the number of cyclists in combination with the second scenario showed that the level of CO_2 emissions can be decreased or in other words, the proposed policy proved fairly good.

In future work, more policies will be tested and evaluated using the proposed approach. Then, advanced machine learning and deep learning techniques will be applied to improve the current results. Finally, data from simulation runs in the optimization step can be used to expand the current dataset.

ACKNOWLEDGMENTS

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 870338. The authors also acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0209).

REFERENCES

[1] Kittipong Chomboon, Pasapitch Chujai, Pongsakorn Teerarassamee, Kittisak Kerdprasop, and Nittaya Kerdprasop. 2015. An empirical study of distance metrics for k-nearest neighbor algorithm. In *Proceedings of the 3rd international conference on industrial application engineering*, 280–285.

[2] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20, 3, 273–297. doi: 10.1007/bf00994018. http://dx.doi.org/10.1007/BF00994018.

[3] Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, Miha Štajdohar, Lan Umek, Lan Žagar, Jure Žbontar, Marinka Žitnik, and Blaž Zupan. 2013. Orange: data mining toolbox in python. *Journal of Machine Learning Research*, 14, 2349–2353. http://jmlr.org/papers/v14/demsar13a.html.

[4] Jerome H. Friedman. 2002. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38, 4, 367–378. doi: 10.1016/s0167-9473(01)00065-2. http://dx.doi.org/10.1016/S0167-9473(01)00065-2.

[5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. *Springer Series in Statistics*. Springer New York Inc., New York, NY, USA.

[6] 2016. *Introducing matsim*. *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, 3–8.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521, 7553, 436–444.

[8] Dastan Maulud and Adnan M Abdulazeez. 2020. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, 1, 4, 140–147.

[9] 2016. *A closer look at scoring*. *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, 23–34. doi: 10.5334/baw.3. http://dx.doi.org/10.5334/baw.3.

[10] Soledad Nogués, Esther González-González, and Rubén Cordera. 2020. New urban planning challenges under emerging autonomous mobility: evaluating backcasting scenarios and policies through an expert survey. *Land Use Policy*, 95, 104652. doi: 10.1016/j.landusepol.2020.104652. http://dx.doi.org/10.1016/j.landusepol.2020.104652.

[11] Gilles Pison. 2019. How many humans tomorrow? the united nations revises its projections. *The Conversation*, 1–6.

[12] Eleni I Vlahogianni, John C Golias, and Matthew G Karlaftis. 2004. Short-term traffic forecasting: overview of objectives and methods. *Transport reviews*, 24, 5, 533–557.