Predicting real life electric vehicle fast charging session duration using neural networks

Anthony Deschênes[1], Jonathan Gaudreault[1], Claude-Guy Quimper[1]

***Abstract*— Predicting the time needed to charge an electric vehicle from X% to Y% is a difficult task due to the non-linearity of the charging process and other external factors such as temperature and battery degradation. Using 28,000 real-life level 3 fast charging sessions from 15 different types of electric vehicles, we train models for this task. We compare learning models such as random forest, linear and second-degree regressions, support vector regressions, and neural networks. The models take into consideration the external temperature, battery capacity, nominal capacity of the electric vehicle, number of charges made during the same day, maximum charging time allowed by the electric vehicle, target voltage, maximum voltage and maximum current asked by the electric vehicle. The models also take into consideration the vehicle type and the charging station type. We use a data augmentation technique (SMOTE) and hyperparameters optimization to enhance our model performances. The structure of the neural networks is optimized using Bayesian Optimization. All models are trained and statistically compared in order to find the overall best model for all vehicle types. The overall best model is a neural network with a sub neural network pre-trained to predict the electric vehicle type.**

I. Introduction

With the rise of electric vehicles comes the rise of anxieties around them. The principal sources of anxiety are the purchase price, the autonomy and the battery charging time [1]. In this paper, we focus on charging time. We use machine learning and real data in order to train models to predict the time needed to charge an electric vehicle from X% to Y%. A precise model will help to significantly reduce the charging time anxiety (do I have time for a coffee or a complete meal?) and facilitates the adoption of electric vehicles. Moreover, it is crucial for algorithm that uses charging time to compute solutions (such as vehicle routing problems [2]) to have a precise algorithm to predict charging time. Errors in charging time prediction can indeed lead to non-optimal solutions and unnecessary costs.

We compare learning models (including neural networks) to predict the charging time using real-life fast charging sessions that takes into consideration external temperature, number of charges made during the same day, battery capacity, etc. The remaining of this paper is organized as follows. Section II presents a literature review. In Section III we present the dataset used for our experiments. Section IV presents the learning model variants tested. Section V presents the methodology used to train and test the different variants. Finally, Section VI presents the results of our experiments.

II. Literature review

Initial work related to electric vehicle supposed linear charging time [3] [4]. Montoya *et al.* [5] and Froger *et al.* [6] show that it is important to consider the non-linearity of the charging time. They use typical charging functions in order to predict the charging time. More recently we used a predetermined charging function to find the optimal charging decisions for a fixed route in electric vehicle (FRVCP-NLEM) [7]. It was found using a very limited number of historical data. Moreover, the charging function varies with respect to multiple external unknown factors such as the internal temperature of the battery. Nait *et al.* [8] used a predictive function to predict the charging rate of an electric vehicle. They then used this charging rate to predict the time needed to charge. Their approach needs to know the time needed to charge the electric vehicle to the maximum authorized value and the minimum authorized value.

Frendo *et al.* [9] used simulation (with sampled historical data) to train regression models predicting the departure time of an electric vehicle given an arrival and departure state of charge. They limited their analysis to an arrival state of charge of 20 % with a departure charge of at least 50 %. They used a single type of electric vehicle with always the same battery capacity. All charging stations (CS) are identical. They showed that irregularities in duration even for the same driver makes it difficult to predict. They showed that accurate charging duration predictions allow for a fair distribution when multiple electric vehicles need to simultaneously charge. The mean absolute error (MAE) of their best model is 82 minutes. Chung *et al.* [10] used an ensemble technique to predict charging duration. Their best model has a symmetric mean absolute percentage error (SMAPE) of 10.4 % using an ensemble predictive algorithm that uses support vector regression, random forests, and diffusion-based kernel density estimator.

In the context of a fleet of electric vehicles, Dong *et al.* [11] developed an algorithm to control the waiting time of an electric taxi at charging stations. They limited the charging time of an electric vehicle to a given maximum that allows them to dispatch vehicles

[1] CRISI Research Consortium for Industry 4.0 System Engineering, *Université Laval*, Québec, Canada `anthony.deschenes.1@ulaval.ca`, `jonathan.gaudreault@ift.ulaval.ca`, `claude-guy.quimper@ift.ulaval.ca`

in a way that avoids long waiting time and unbalanced utilization of charging stations. The maximum charging time is obtained from historical data. Tian *et al.* [12] employed a similar strategy to predict waiting time. Shahriar *et al.* [13] propose a complete literature review about predicting charging duration.

III. Dataset

Our raw dataset consists of 30 263 real fast charging sessions. We have access to data from 3 different types of level 3 charging stations for a period of one year. In Table I, we present the features used by our learning models. It also shows the source for each feature. In order to take into consideration the fact that temperature affects the charging process (as an approximation of the temperature of the battery), we use Environment Canada historical database [14] to get the external temperature for each charging session. The external temperature has a range of values from -40°C to 36°C. It roughly follows a normal distribution centered at 0°C. This allows models to have a good indication of the impact of temperature on the time needed to charge an electric vehicle. We also included a feature that represents the number of charges made the same day by the same user. It allows to have an indicator of the temperature of the battery.

A. Filters applied

Since the data is obtained from real charging sessions, it contains anomalies that need to be filtered. We first applied some basic filtering on the features from which we know the expected range. The *Start SOC* and *End SOC* must be between 0% and 100% and the *Start SOC* must be smaller than the *End SOC*. Charging sessions with a duration of less than 200 seconds are also excluded since they do not represent a normal fast charging behavior. We also exclude vehicle types from which we do not have more than an arbitrary value of 100 charging sessions because it is not enough to show the charging behavior of the vehicle.

We also applied some more advanced filters. First, we observe an abnormal behavior for the *Mitsubishi Outlander* when charging to 80% or 81%. We thus removed all charging sessions from the *Mitsubishi Outlander* that charged to these values. Second, we developed two indicators that allow to filter two types of abnormal charging session: (1) charging a long time to gain very little state of charge and (2) charging a short time to gain a very high state of charge. Equation (1) has a high value when the battery gained during the charging session has been obtained quickly. Equation (2) has a high value when the duration is high for a small gain in battery. We remove all data that is above the 97.5 percentile of each of these equations from our dataset. After applying the proposed filters, the dataset now has 28 034 charging sessions. In total, the dataset has 15 different vehicle types.

$$\frac{\text{End SoC} - \text{Start SoC}}{\text{Duration (min)}^2} \tag{1}$$

TABLE I

Available features for models

Feature	Description	Source
Start SoC (%)	Start state of charge of the electric vehicle	Operator Logs
End SoC (%)	End state of charge of the electric vehicle	Operator Logs
Temperature (°C)	External temperature during the charging session	Environment Canada Database
EVSE max power (kW)	Maximum power that the charging station can provide	Protocol Information
Max charging time (min)	Maximum charging duration authorized by the electric vehicle	Protocol Information
Battery capacity (kWh)	Battery capacity of the electric vehicle	Protocol Information
Nominal battery capacity (kWh)	Nominal battery capacity of the electric vehicle	Operator Logs
Target voltage (V)	The voltage that the electric vehicle would like to receive during the charging session	Protocol Information
Max voltage (V)	The maximum voltage that the electric vehicle can receive at any moment during the charging session	Protocol Information
Max current (A)	The maximum current that the electric vehicle can receive at any moment during the charging session	Protocol Information
CS type	The type of the charging station	Operator Logs
Vehicle type	The type of the electric vehicle	Operator Logs
Number of charges the same day	Number of charges made on the same day by the same client with the same vehicle.	Operator Logs

$$\frac{\text{Duration (min)}^2}{(\text{End SoC} - \text{Start SoC})^2} \tag{2}$$

IV. Models

We tested multiple learning models: linear and second-degree regressions (LR and PR), Random Forests (RF) of degree one and two, Support Vector Regression (SVR) and Neural Networks (NN).

A. Neural Networks

The structure of the neural networks is inspired by resnet [15]. Since our features do not contain spatial relationship, we use feedfoward neural networks. For all neural networks it consists of multiple repeated *Basic Blocks*. Each *Basic Block* consists of a configurable number of fully connected layers each with a configurable number of neurons. Each block has a batch normalization at its output and a skip connection going from the input to the output to help stabilize learning [16] [17]. The activation function used is always the rectified linear unit since it performs well in practice [18].

For all neural networks, we use a Stochastic Gradient Descent with a learning rate of $1 \cdot 10^{-5}$. We use Nesterov with a momentum of 0.9 and a weight decay of 0.0001. We used a validation set of 10% of the size of the training set. We reduce the learning rate by a factor of 0.5 if the validation loss has not reduced for 10 iterations. We use early stopping to stop the learning process if the validation loss has not reduced for 20 iterations and restore the model to the iteration that had the best validation loss when stopping the learning process.

B. *Model variants*

The vehicle type is provided by the EVSE network operator. Its value is uncertain since it is determined by the operator based on unverified criteria. Therefore, it likely contains errors such as vehicle types that are wrongly separated in two different vehicle types or too general segmentation. For example, the *Nissan LEAF* technology has evolved since the first generation and not all *Nissan LEAF* have the same charging behavior. For this reason, we propose to train variants for random forests and neural networks that do not include the vehicle type (since in theory the vehicle types can be identified using the features of our dataset).

Another option (only possible for neural networks) is to first train a sub neural network predicting the vehicle type (the value provided by the operator) using the Maximum Voltage, Maximum Current and the Battery Capacity. These three features allow to identify most of the vehicle types perfectly. This sub neural network can then be injected into the bigger network used to predict the time needed to charge the electric vehicle. Since none of the weights of the neural networks are frozen, the bigger network can correct errors made by the sub neural networks if needed.

Thus, the training of the sub neural network is a prior that we inject into the bigger neural network. As a result, this method could offer multiple advantages: it provides the neural network knowledge on the input data, it allows the neural network to correct errors made by the operator vehicle type and since the output of the sub neural network is a probability distribution it allows to encode more granular segmentation of electric vehicle types as a combination of existing vehicle type if needed.

Table II presents the different variants for each of the learning models. It also presents an alias that we use to refer to a specific variant. Each variant can use or not the vehicle type given by the operator. It can use second-degree interactions and for the neural networks they can use a sub neural network and for the sub neural network there is the possibility to copy the entries of the sub neural network to the bigger neural network or not.

V. Experiments

A. *Methodology*

TABLE II

Particularities of each variants and the alias used to identify a given variant.

Learning Model	**Variant**	**Use vehicle type?**	**Use second degree?**	**Sub Net-work?**	**Copy sub entries?**
Random Forest	RF	✓		-	-
	RF-NV			-	-
	RF2	✓	✓	-	-
	RF2-NV		✓	-	-
Regression	LR	✓		-	-
	PR2	✓	✓	-	-
SVR	SVR	✓		-	-
Neural Network	NNS			✓	
	NNS-A			✓	✓
	NN	✓			-
	NN-NV				-

Figure 1 presents an overview of the methodology used to train and test each of the variants from Table II. We do an 80-20 split of our dataset that ensures that there is the same percentage of each vehicle type and CS type. We then use the 20% to optimize the hyperparameters of the models and the remaining 80% is used to train and test our models. Each variant is then trained using four learning methods: using all data, per vehicle type, per CS type or per vehicle type per CS type. We then perform 10 repetitions of stratified K-Fold cross validation [19] with K=5. For each iteration of K-Fold, if data augmentation is used (see section V-C), we apply data augmentation to the training set and then train the variant on the augmented training set. We then compute the Mean Absolute Percentage Error (MAPE) of the variant on the test set. Once all variants are trained and tested, we perform a statistical comparison in order to determine which model has on average the best MAPE. All variants are tested on the same data for each repetition. This allows to compute a difference test to compare the different variants [20]. For most variants, we have no prior on the best way to train the variant and thus we need to train them in all the possible learning methods. However, for neural networks we do not train one variant for each vehicle type for each CS type since it generally performs better with more data [21] and for computational time limitations.

B. *Hyperparameters optimization*

Linear and second-degree regression do not require hyperparameters optimization. Therefore, we only optimized the hyperparameters of the random forests, SVR and the neural networks. We optimized their hyperparameters using Bayesian Optimization on 20% of the data of our dataset which usually allows to efficiently find good hyperparameters values for neural networks [22] [23]. Bayesian optimization of the random forests give the same results as the default one used by Scikit-learn [24]. It returns C=10, epsilon=2.97 with a linear kernel for the SVR.

1) Neural networks: We optimize the structure of the neural networks using Bayesian Optimization. For computational time limitations, we only optimize the

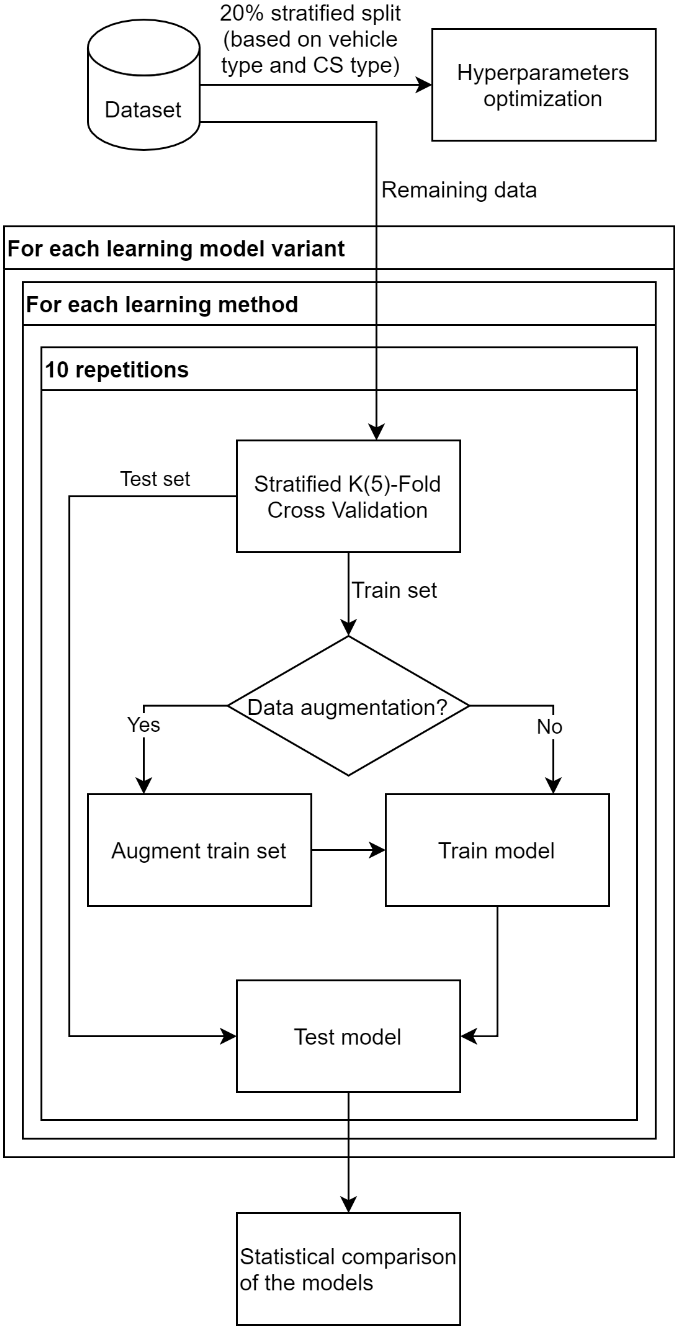
20% stratified split (based on vehicle type and CS type)
Dataset
Hyperparameters optimization
Remaining data
For each learning model variant
For each learning method
10 repetitions
Test set
Stratified K(5)-Fold Cross Validation
Train set
Data augmentation?
Yes
No
Augment train set
Train model
Test model
Statistical comparison of the models

Fig. 1. Methodology used to optimize, train and test the different variants

structure of the NNS-A (see Table II) when trained on all the data without data augmentation. We choose the NNS-A because preliminary experimentations show that it performs the best between the different neural networks variants. All other neural networks will share the same structure. Training-specific hyperparameters are manually chosen because of computational time limitations.

The optimization process consists of deciding the number of *Basic Blocks* (from 1 to 10), the number of layers in each *Basic Block* (from 1 to 10) and the number of neurons in each fully connected layer (from 8 to 512). The number of configurations is 50 500 that ranges from a simple 1-layer neural network to a deep 100-layer neural network. After 128 iterations of Bayesian Optimization, the best configuration found is 3 *Basic Blocks* with 7 fully connected layers of 340 neurons each.

C. Data augmentation

Our dataset is highly unbalanced from one vehicle type to another. In reality, we want the variants to be equally accurate for each vehicle types.

In order to balance our dataset, we train every model with Synthetic Minority Over-sampling TEchnique (SMOTE) [25]. Since SMOTE can also worsen the model results, all models are also trained without SMOTE and a comparative analysis is performed. Without SMOTE the size of our training set is approximately 20 000 data. With SMOTE, the training set is now approximately 115 000 data, which is a significant increase.

VI. Results

Table III presents the average Mean Absolute Percentage Error (MAPE) over all vehicle types for each variant for each of the learning models. Since our dataset, is unbalanced the confidence interval is computed using post-stratification [26]. It also presents the number of vehicle types for which it dominates the other variants (a variant dominates when its MAPE is on average statistically lower to other variants when performing the difference test [20] with a confidence of 95 %). If for two variants for the same vehicle type we cannot conclude on average a statistical difference, we also consider that both models dominate.

The best variant is the neural network trained on all data using SMOTE that uses a sub neural network and copies the entries of the sub neural network to the bigger neural network (NNS-A). It dominates for 14 of the 15 vehicles types. For the *Kia Soul (2018)*, the dominant variant is the NN trained per vehicle. The MAPE of the NNS-A is 7.13 ± 0.0575 while the NN has a MAPE of 6.47 ± 0.095.

The best sixth variants are neural networks. Figure 2 presents the difference per vehicle type between the best variant (NNS-A) and the second-best variant (NN-NV). On average the difference of the MAPE between the best variant and the second-best variant is 0.24 with a maximum of 1.17 and a minimum of -0.2. The NNS-A dominates for all electric vehicle types. The two best variants do not use the operator vehicle type. Injecting a prior using a sub neural network allows for a slight improvement. The prior allows for a smaller confidence interval. For the neural network variants, it is better to train the neural networks on all data using SMOTE.

Figure 3 presents the difference of MAPE between the best variant (NNS-A) and the second-best variant that is not a neural network (RF2 trained on all data

TABLE III
AVERAGE MEAN ABSOLUTE PERCENTAGE ERROR (MAPE) OVER ALL VEHICLE TYPES AND NUMBER OF VEHICLE TYPES FOR WHICH IT DOMINATES OF EACH VARIANT FOR EACH OF THE LEARNING METHODS. THE TOTAL NUMBER OF VEHICLE TYPES IS 15.

Variant	**Learning method**	**MAPE (%)**	**Number of vehicle types for which it dominates**
NNS-A	All-SMOTE	12.19 ± 0.0106	14
NN-NV	All-SMOTE	12.43 ± 0.0151	9
NN	All-SMOTE	12.55 ± 0.013	6
NN	CS-SMOTE	13.02 ± 0.014	2
NN	CS	13.48 ± 0.025	1
NN	Vehicle	13.92 ± 0.1805	6
RF2	All-SMOTE	14.21 ± 0.0189	0
RF	All-SMOTE	14.41 ± 0.0196	0
NN-NV	All	14.46 ± 0.0323	1
RF2	CS-SMOTE	14.48 ± 0.0141	0
RF-NV	All-SMOTE	14.58 ± 0.0207	0
RF2-NV	All-SMOTE	14.59 ± 0.0196	0
RF	CS-SMOTE	14.6 ± 0.0192	0
NNS	All	14.66 ± 0.0235	0
NN	All	14.67 ± 0.0225	1
NNS-A	All	14.68 ± 0.0229	0
RF2	All	15.64 ± 0.0138	0
RF	All	15.69 ± 0.0086	0
RF2	CS	15.75 ± 0.0108	1
RF-NV	All	15.8 ± 0.011	0
RF	CS	15.86 ± 0.0107	0
RF	Vehicle	15.89 ± 0.0184	0
RF2	Vehicle	16.21 ± 0.0163	0
RF	Vehicle per CS	16.25 ± 0.0188	0
RF2-NV	All	16.54 ± 0.013	0
RF2	Vehicle per CS	16.58 ± 0.0165	0
PR2	Vehicle	20.29 ± 0.1977	0
PR2	Vehicle per CS	21.01 ± 0.1977	0
LR	Vehicle	23.65 ± 0.017	0
LR	Vehicle per CS	24.01 ± 0.0171	0
SVR	Vehicle	28.45 ± 0.0152	0
SVR	Vehicle per CS	29.25 ± 0.0153	0
SVR	CS-SMOTE	29.73 ± 0.0053	0
SVR	All-SMOTE	32.16 ± 0.0045	0
SVR	All	32.53 ± 0.0059	0
SVR	CS	32.55 ± 0.0064	0
LR	CS	38.61 ± 0.0038	0
LR	All	38.92 ± 0.0037	0
PR2	CS	39.04 ± 0.0043	0
PR2	All	39.17 ± 0.0039	0
LR	CS-SMOTE	83.21 ± 0.014	0
PR2	CS-SMOTE	83.93 ± 0.0142	0
LR	All-SMOTE	85.51 ± 0.0131	0
PR2	All-SMOTE	86.14 ± 0.0126	0

using SMOTE). On average the difference is 2.02 with a maximum of 3.72 and a minimum of 0.18. It dominates for all vehicle types and this difference is statistically significant for 14 of the 15 vehicle types. This difference increases for the other learning model such as SVR where they perform poorly on our dataset.

Figure 4 presents the MAPE of the best variant (NNS-A) for all vehicle types. Its MAPE over all vehicles is 12.19% with a maximum of 18.44 % and a minimum of 6.31 %. The mean absolute error (MAE) over all vehicles of the NNS-A is 3.10 minutes with a maximum of 5.60 minutes and a minimum of 1.51 minutes.

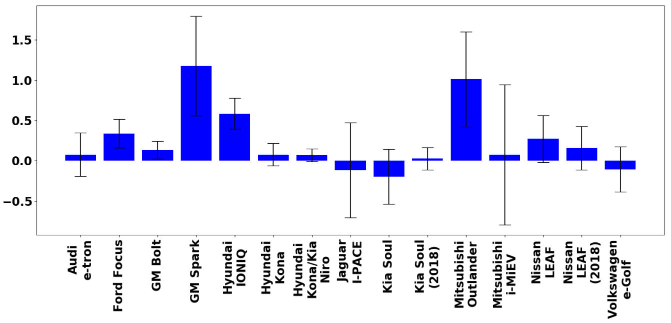

Fig. 2. Difference of the value of the Mean Absolute Percentage Error (MAPE) between the most dominant variant (NNS-A) and the second-best variant (NN-NV) for each vehicle type. Negative value means that the NN-NV has a smaller MAPE than the NNS-A for the vehicle type.

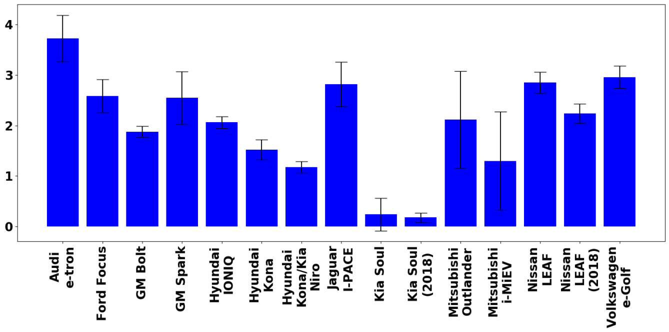

Fig. 3. Difference of the Mean Absolute Percentage Error (MAPE) between the best variant (NNS-A) and the second-best variant that is not a neural network (RF2) for each vehicle type

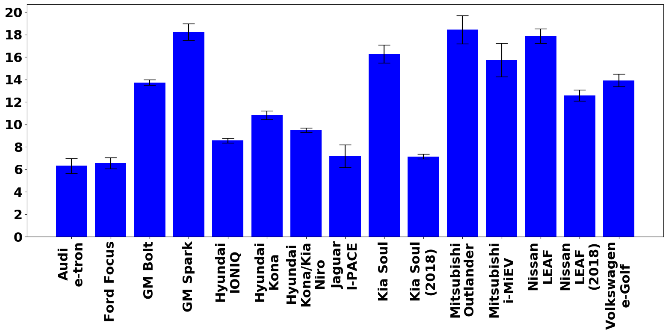

Fig. 4. Mean Absolute Percentage Error (MAPE (%)) of the NNS-A for each vehicle type

A. *SMOTE analysis*

The four best variants use SMOTE: NNS-A, NN-NV, NN trained on all data and NN trained per CS. For the NNS-A, the difference between using SMOTE or not is 2.34. All improvements are on average statistically significant except for the *Kia Soul*. The best MAPE improvement is 16.62. We also note that for all vehicle types, the use of SMOTE always improve the MAPE. For the NN-NV, the average difference is 2.03. For the

NN trained on all data the average difference is 2.12 and finally for the NN trained per CS it is 0.46. For all of these variants, using SMOTE allow to obtain a smaller confidence interval. We also observe that all the best variants for the random forests use SMOTE with an average difference of 1.43 for the RF2 trained on all data. Finally, the use of SMOTE for the regressions and support vector regressions generally worsen the results with the worst learning models being linear and second-degree regressions.

VII. Conclusion

We presented a comparison of multiple learning model variants predicting the time needed to charge an electric vehicle from X% to Y% on a fast-charging station. We used 30,000 real fast charging sessions over 15 vehicle types. The best variant is a neural network with a sub neural network (NNS-A). Its MAPE over all vehicle types is on average 12.19% ± 0.0106 with a MAE of 3.10 minutes. In comparison with the second-best variant that is not a neural network, the NNS-A has on average a smaller MAPE (2.02). We showed that using SMOTE to balance our dataset allows the NNS-A to obtain on average a MAPE that is 2.34 smaller with a maximum improvement of 16.62 for the *Jaguar I-PACE*. As for future work, this paper did not present integration methods to a real-world application. We want to personalize the neural networks to adjust to user vehicle specific behaviors to consider the vehicle state of health.

References

[1] A. TANȚĂU and I. Gavrilescu, "Key anxiety factors for buying an electric vehicle." *Management & Marketing*, vol. 14, no. 2, 2019.

[2] S. Karakatič, "Optimizing nonlinear charging times of electric vehicle routing with genetic algorithm," *Expert Systems with Applications*, vol. 164, p. 114039, 2021.

[3] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl, "The electric fleet size and mix vehicle routing problem with time windows and recharging stations," *European Journal of Operational Research*, vol. 252, no. 3, pp. 995–1018, 2016.

[4] Y. CHENG and W. Szeto, "Artificial bee colony approach to solving the electric vehicle routing problem," *Journal of the Eastern Asia Society for Transportation Studies*, vol. 12, pp. 975–990, 2017.

[5] A. Montoya, C. Guéret, J. E. Mendoza, and J. G. Villegas, "The electric vehicle routing problem with nonlinear charging function," *Transportation Research Part B: Methodological*, vol. 103, pp. 87–110, 2017.

[6] A. Froger, J. E. Mendoza, O. Jabali, and G. Laporte, "Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions," *Computers & Operations Research*, vol. 104, pp. 256–294, 2019.

[7] A. Deschênes, J. Gaudreault, L.-P. Vignault, F. Bernard, and C.-G. Quimper, "The fixed route electric vehicle charging problem with nonlinear energy management and variable vehicle speed," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 1451–1458.

[8] A. Nait-Sidi-Moh, A. Ruzmetov, M. Bakhouya, Y. Naitmalek, and J. Gaber, "A prediction model of electric vehicle charging requests," *Procedia Computer Science*, vol. 141, pp. 127–134, 2018.

[9] O. Frendo, N. Gaertner, and H. Stuckenschmidt, "Improving smart charging prioritization by predicting electric vehicle departure time," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6646–6653, 2020.

[10] Y.-W. Chung, B. Khaki, T. Li, C. Chu, and R. Gadh, "Ensemble machine learning-based algorithm for electric vehicle user behavior prediction," *Applied Energy*, vol. 254, p. 113732, 2019.

[11] Z. Dong, C. Liu, Y. Li, J. Bao, Y. Gu, and T. He, "Rec: Predictable charging scheduling for electric taxi fleets," in *2017 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2017, pp. 287–296.

[12] Z. Tian, T. Jung, Y. Wang, F. Zhang, L. Tu, C. Xu, C. Tian, and X.-Y. Li, "Real-time charging station recommendation system for electric-vehicle taxis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3098–3109, 2016.

[13] S. Shahriar, A. Al-Ali, A. H. Osman, S. Dhou, and M. Nijim, "Machine learning approaches for ev charging behavior: A review," *IEEE Access*, vol. 8, pp. 168 980–168 993, 2020.

[14] E. Canada. (2021) Historical data. [Online]. Available: https://climate.weather.gc.ca/historical_data/search_historic_data_e.html

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] A. E. Orhan and X. Pitkow, "Skip connections eliminate singularities," *arXiv preprint arXiv:1701.09175*, 2017.

[17] S. Santurkar, D. Tsipras, A. Ilyas, and A. Mądry, "How does batch normalization help optimization?" in *Proceedings of the 32nd international conference on neural information processing systems*, 2018, pp. 2488–2498.

[18] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *towards data science*, vol. 6, no. 12, pp. 310–316, 2017.

[19] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, and S. Ridella, "The 'k'in k-fold cross validation," in *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. i6doc. com publ, 2012, pp. 441–446.

[20] M. F. Triola, *Elementary statistics*. Addison Wesley Publishing Company, 1992.

[21] G. Foody, M. McCulloch, and W. Yates, "The effect of training set size and composition on artificial neural network classification," *International Journal of Remote Sensing*, vol. 16, no. 9, pp. 1707–1723, 1995.

[22] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[23] H. Cho, Y. Kim, E. Lee, D. Choi, Y. Lee, and W. Rhee, "Basic enhancement strategies when using bayesian optimization for hyperparameter tuning of deep neural networks," *IEEE Access*, vol. 8, pp. 52 588–52 608, 2020.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[26] J. T. Kulas, D. H. Robinson, J. A. Smith, and D. Z. Kellar, "Post-stratification weighting in organizational surveys: A cross-disciplinary tutorial," vol. 57, no. 2, pp. 419–436. [Online]. Available: http://doi.wiley.com/10.1002/hrm.21796