Application and machine learning methods for dynamic load point controls of electric vehicles (xEVs)

Danting Cao[1*], *Jonathan* Lerch[1], *Daniel* Stetter[2], *Martin* Neuburger[1], *Ralf* Wörner[1]

[1]Institute of Sustainable Energy Engineering and Mobility, University of Applied Sciences Esslingen, Kanalstreet 33, 73728 Esslingen am Neckar, Germany
[2]Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Nobelstreet 12, 70569 Stuttgart, Germany

Abstract. From the customer's perspective, the appeal of electric vehicles depends on the simplicity and ease of their use, such as flexible access to electric power from the grid to recharge the batteries of their vehicles. Therefore, the expansion of charging infrastructure will be an important part of electric mobility. The related charging infrastructure is a big challenge for the load capacity of the grid connection without additional intelligent charge management: if the control of the charging process is not implemented, it is necessary to ensure the total of the maximum output of all xEVs at the grid connection point, which requires huge costs. This paper proposes to build a prediction module for forecasting dynamic charging load using machine learning (ML) techniques. The module will be integrated into a real charge management concept with optimization procedures for controlling the dynamic load point. The value of load forecasting through practical load data of a car park were taken to illustrate the proposed methods. The prediction performance of different ML methods under the same data condition (e.g., holiday data) are compared and evaluated.

1 Introduction

The uncontrolled charging of xEVs might increase the system's peak demand and overload transformers. In order to ensure a more sustainable power system, a charge management for smoothing peak loads should be taken into consideration [1-3]. Precise dynamic charging load forecasting can effectively help to optimize a charge management system. In an optimal scenario, the reservation of parking spaces is made possible as an additional service. Furthermore, it would be conceivable to provide a user interface at the charging pile, through which the desired departure time, energy quantity and other information can be exchanged. Based on this and the underlying load predictions, a charging prioritisation of the individual xEVs in the event of an increased load volume can be controlled according to the situation. Costs for network expansion in conjunction with the expansion of upstream and downstream charging infrastructure through connection services, charges and transformers could be kept to a minimum. This approach directly benefits distribution network operators, property operators and even end consumers by harmonising their mobility needs. The use of AI methods in the concrete application case of prediction strategies for charging xEVs already proved to be efficient and more accurate than conventional probabilistic algorithms [4]. A comparison of different approaches was also possible. For super-short-term forecasting with deep learning, the long-short-term memory (LSTM) already showed very realistic results as described in [5].

Unlike before, the goal of this work is to use ML techniques to predict the charging load and modularly integrate it into an intelligent charging management system. As a further step, the dynamic load management methods can be applied to flatten the load curve with the help of predefined schedules. This is generally done by reducing the simultaneous factor and thus leads to a reduced load at the grid connection point. In addition, it is relevant to refer to the data that is actually available. Obtaining information from the vehicles like the state of charge (SOC) or energy demand seems realistic, but there would have to be digital communication with the charging station, which would require both the vehicle and the charging station to be ISO-15118 compliant (in Europe) [6]. As current instruments on the market do not yet fully support this technology, this criterion is not applicable. Therefore, in addition to the measurement data of the charging stations, which are provided by the charging station operators, it is advantageous to obtain further information on the environment. Thus, the features described in this paper come very close to those as they were set up in [7], whereby weather and time-related information such as holidays, weekdays etc. prove to be expedient.

1.1 The EVx charge management with integrated prediction module

KI-LAN is a project that began in Germany end of 2019, that researches smart charging in various usage scenarios. Within this framework funded by the Ministry of the

* Corresponding author: Danting.Cao@hs-esslingen.de

Environment, Climate and Energy, entitled "AI-based network load management for parking under different usage scenarios", the work is carried out in cooperation with a network of local energy suppliers, grid operators, charging station operators and renowned scientific institutes. The solutions will then be piloted at two real car park operating companies in Germany. First of all, the use case of an employer and organiser in urban areas as well as the car park of an employer in rural areas will be reproduced. Later, the sustainable concepts should be transferable to various other scenarios.

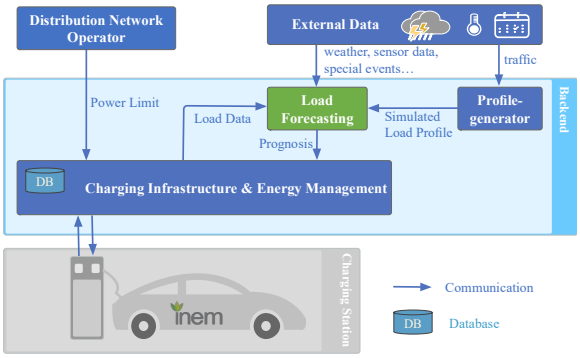

Fig. 1. Predictive charge management concept

As shown in Figure 1, the charging management indicates the embedding of a prediction model based on recorded load data and possible additional data. In principle, this is a modularly designed charging infrastructure which can be adapted by the smallest of changes. In the underlying architecture, the process unit of "Load Forecasting" should stand out as an innovative new element of a charging infrastructure. Its task is to forecast the load curves of the xEVs and to correlate the own ML prediction model with the simulation model generated by the "Profile-generator". It can be assumed that the “Profile-generator” is the more reliable source at the beginning until enough measurement data is accumulated. The “Charging Infrastructure and Energy Management” then apply methods for intelligent load and charge management control to flatten potential load curves with the help of the transmitted forecast. The maximum power that can be made available depends on the respective charging station as well as the power limit provided by the "distribution network operator" at the grid connection point. The recorded load data of the individual charging points is sent to the "Load Forecasting" backend at regular intervals. The mutual exchange of these two instances takes place via a configured application program interface (API). By means of one or more web crawlers, further external data from the web is analysed and processed. This can be information in various forms, which can be used as additional data for load forecasting training and increase the efficiency of the model. The profile generator also accesses external data. This is basically information on local traffic volumes.

2 Machine Learning methods

In this chapter, the most suitable machine learning methods for this application are presented, whose tests and results are subsequently compared. These have in common that they are regressors, which should serve the purpose of delivering a continuous value. Besides, they are basically supervised techniques that allow to collect data and to generate an output from previous experiences.

2.1 K-Nearest Neighbors

K-nearest neighbor is a non-parametric method which can be used both as a classifier and as a regressor. The basic idea is to assign the target value to one or more closest known values based on a density function related to the distance in a multidimensional space. The simplest way to calculate the target value is to calculate the average of all adjacent points. It should be noted that low values for k (number of nearest neighbors) make the regressor susceptible to variance, i.e. individual misclassified points in the reference data. However, if k is chosen too large, there is a risk of including points with a large distance to the regressor. In principle, the method can be robust against high noise of the training data and effective for large data sets due to the lower computational effort. However, one needs to determine the number of nearest neighbours. Caution is required when choosing the features. In general, the algorithm performs better, with a small number of input variables, because more data is needed, which leads to the problem of overfitting. [8-9]

2.2 Random Forest

A possibility that delivers good results with a high number of features and is resistant to overfitting is Random Forest as a regressor model. This is a meta-estimator that primarily forms several uncorrelated decision trees from numerous sub-samples. In addition, a weighted average is calculated to enhance the prediction accuracy. For a correct operation it is necessary to carry out a parameterization. In this work, the number of trees or estimators, the maximum depth of the trees and the number of features to be considered were defined. Unfortunately, it is partially difficult to interpret which features in the model play a decisive role in influencing the prediction. It is advantageous, however, that the decision trees can be set up and trained very quickly and paralyzed. Therefore, one of the strengths of the method is definitely the ability to work with large amounts of data and features. In addition, the variance of the individual decisions of different trees influences the overall result better. [10-12]

2.3 Decision Tree

Individual trees, where random samples are selected from a data set, the decision tree procedure uses the entire data set with all available input variables for its prediction model. On the one hand, the result can be better interpreted and only a few parameters need to be parameterized. But on the other hand, there is the danger

of over-fitting the tree, so actually cross-validation is even more essential in this case. Random Forest will rather reduce the error part of the variance than the part of the distortion. Thus, the procedure with Decision Trees on a test data set could be better at first sight. However, higher accuracy can be expected for Random Forests in reality or in case of an unexpected validation. [13]

2.4 Light GBM

This method also uses tree-based learning algorithms. Special attention is paid to the procedure because a large amount of data can be processed and less memory is needed to execute it, thus justifying the origin of the name. As in other boosting methods, the model builds up gradually. But in this regard, there is a decisive difference. Light GBM allows the tree to grow leaf-wise, i.e. vertically, while other algorithms grow level by level. Thus, when an identical leaf grows, the algorithm can reduce the loss more than level-wise algorithms. Moreover, the next leaf can be selected in such a way that the additional loss increases minimally. This supports the statement why the algorithm focuses on the accuracy of results. The parameter tuning is complicated here, since there is a large selection for this in order to achieve a great performance. Here again the overfitting is to be paid more attention to. The algorithm reacts particularly sensitively with small data sets. But in summary, the algorithm is one of the best in the field of machine learning and benefits from its high efficiency. [14]

3 The forecasting framework

The general process of xEVs charging load forecasting is shown in Figure 2. The framework consists of four following steps:

1. Data-Preprocessing
2. Model Building, Training and Test
3. Performance Evaluation
4. Model Management and Deployment

Charging load depends on many characteristics and influencing factors, such as historical load data, week characteristics, holiday or festival attributes, weather condition and temperature. These data can be collected in different ways. The two common ways are from exclusive data providers and the web.

In the Data-Preprocessing step the measurable property so-called features are extracted and selected from the original dataset. After the dataset has been cleaned, integrated and transformed, it is split into training and test set. The training set is used to train the model, in order to find hidden characteristics and temporal correlations between features and target values. In general, the majority of the data (about 80%) is divided into training sets for training models, and test sets are used to assess the predictive power of models. The model parameters are optimized ("tuned") by training process in order to maximize the model's predictive accuracy [15]. Once the best prediction accuracy of a trained forecasting model is built, it will be tested on the test set.

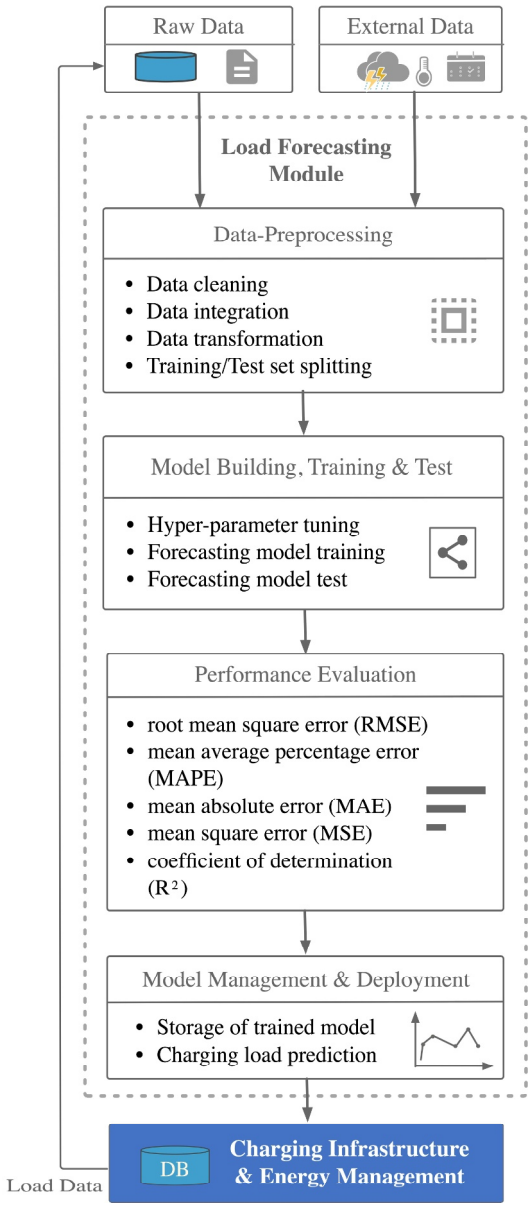

Fig. 2. The framework of ML based forecast

Then, the performance of predicted load with different algorithms will be compared to each other. Following metrics are commonly used for model evaluation: root mean square error (RMSE), mean average percentage error (MAPE), mean absolute error (MAE) and mean square error (MSE). The formulas of RMSE, MAPE, MAE and MSE are given in equations (1) – (4) [16].

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \qquad (1)$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100 \quad (2)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i - y_i\right| \quad (3)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \quad (4)$$

where
N is the number of samples,
$\hat{y}_i$ is the forecasted value of the i-th sample,
y_i is the actual value of the i-th sample.

In addition, a coefficient of determination named R square is also chosen as the important metric to estimate the goodness of fit. The calculation formula is defined as follows [17]:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \quad (5)$$

Where
N is the number of samples,
$\bar{y}$ is the average value of all samples
$\hat{y}_i$ is the forecasted value of the i-th sample

The R^2 ranges from 0 to 1, and the closer the value of R^2 is to 1, the more accurate the prediction would be.

Finally, the best trained forecasting models will be stored and can be used on new data. The new dataset includes features, while the target values are unknown. Based on the supplied features, the trained models will predict the unknown target values, which are the charging load values in this case.

4 Practical Examples and the Analysis

In the Framework of KI-LAN, pre-tests are used to verify the predictive performance of different ML algorithms. In this paper, a real-world dataset of a car park in Germany is applied as practical example.

The original data contains the accumulated charging load of all the charging piles and the time interval is 10 minutes. After a preliminary analysis of the original data, it is apparent that the distribution of the total charging load on weekends and weekdays is completely different. The total load curves of the car park on weekdays present highly periodical characteristics and peak properties, while on the weekend (Saturday and Sunday), there is almost no charging load. The possible reason is that the car park is not operated on weekends.

So-called time-based features can be extracted from the historical data of the car park using the time stamp. Time-based features include the month of the year, the hour of the day and weekday property etc. Simultaneously, the holiday information has also been collected. In addition to the above, features such as weather-based features are also taken into account. They can be extracted from weather service provider in different intervals, like hourly or daily temperature, rainfall and sunshine duration.

The forecasting module is developed in the Python programming language, which is popular in the field of data analytics. Especially as it has many libraries, such as pandas, numpy, Scikit-learn, Tensorflow and Keras.

4.1 Results of forecast curve

Figure 3 shows the charging load of the car park forecasted by different ML algorithms, including K-Nearest Neighbors Regressor (KNNR), lightGBM (LGBM), Random Forest Regressor (RFR) and Decision Tree Regressor (DTR). In this case, the feature values contain weather-based data. In order to better compare their prediction performance, the results of the same week including the weekend are considered here, not only just one day. In the figure, the dark line shows the current charging load of the car park. As mentioned above, it could be easily seen that, the curve has a certain periodicity and nonlinear characteristics on weekdays, which also makes it more difficult to build a general model for accurate prediction. From the weekday, it can be observed that xEVs began to charge at about 7'o

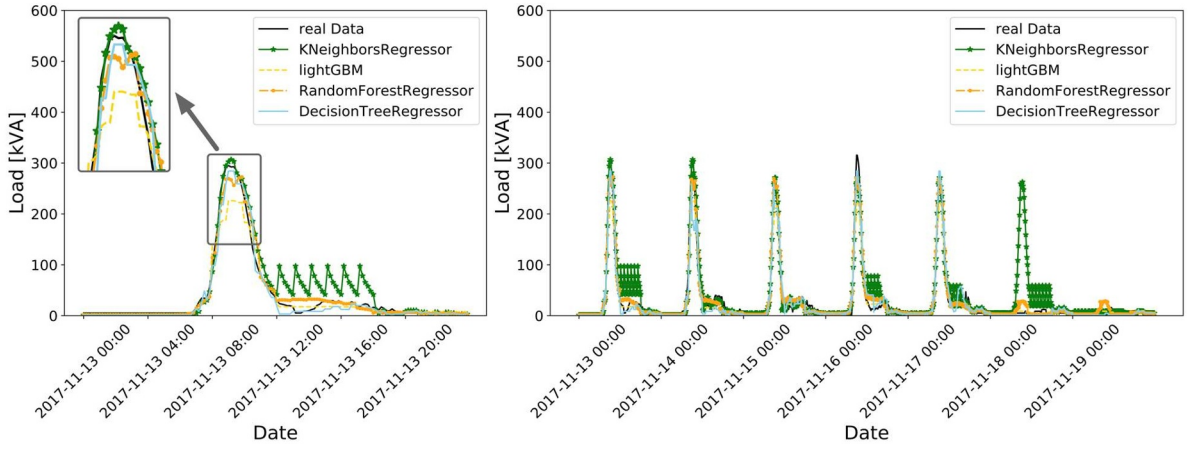

Fig. 3. Load forecast curve of one day and one week with weather data

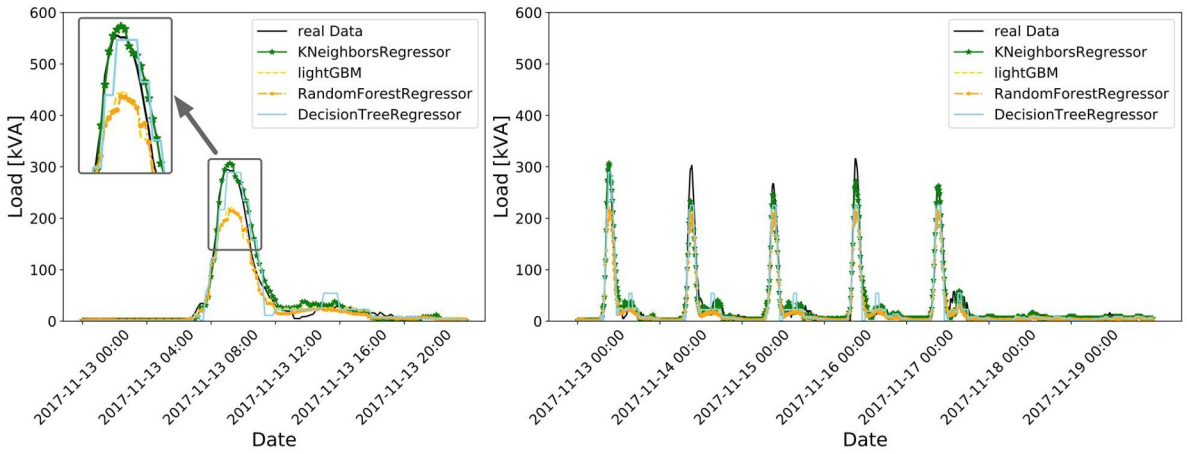

Fig. 4. Load forecast curve of one day and one week without weather data

clock in the morning and then it reached the peak load around 9 o'clock, which is also consistent with the usual activities and behavior of people, including xEV users. The load curve with green point line represents that the KNNR model can reasonably forecast the value of each point, and the predicted load is nearly close to the actual load. But at the same time there are relatively large errors in the prediction of lower charging load values (after 12 o'clock). In contrast, the LGBM (yellow dotted line) model can capture the general trend changes of the load, but for the steep changes of the load, especially for the data near the peak load, it cannot be accurately predicted. The DTR (light blue line) and RFR (orange dotted line) models can fit the peak load relatively well. Their prediction performance is close, but it seems that DTR is more sensitive to rapid rising changes, because the predicted load curves show sharper peaks on weekdays. However, the performance of KNNR on the weekend is very poor, from the peak of the charging load about 300 KVA on weekdays to almost 0 on the weekend, such changes in this case cannot be predicted. Although the predictions of RFR and DTR also have errors, they are far smaller than the KNNR.

For comparison purposes, the weather-based features are removed from the case of Figure 4. However, prediction accuracy of KNNR on Saturday is much better. It no longer forecasts a peak. With the comparison of these two figures, it is clear that the weather-based features can be disturbing influences on the prediction performance of KNNR. The KNNR algorithm uses "feature similarity", so-called "nearest neighbor" principle, to predict the value of new data point. This means that the weather-based features make this day's features closer to a weekday rather than a weekend day. Therefore, in real application cases, feature values should be appropriately selected based on the individual datasets. Generally speaking, the forecasting performance with weather data is better than without weather data during the weekdays, especially when observing the prediction results near the peak load on Tuesday (2017-11-14) in this practical example.

4.2 Performance Evaluation

From the analysis of the original data, the sum of the charging loads of the charging piles can be 0 for a certain period of time, especially at the weekend. Therefore, MAPE is not suitable as an evaluation metric here, because y_i in formula (2) as a denominator is mathematically not allowed to be equal to 0. Thus, RMSE, MAE and R^2 are selected as the criteria for evaluating the prediction accuracy of the models.

Table 1. Performance comparison of one day with (grey area) and without (white area) weather data for all methods.

Metrics	K-Nearest Neighbors Regressor	Decision Tree Regressor	Random Forest Regressor	Light GBM
RMSE	26.8783	12. 3508	12.2347	17.1227
MAE	16.8377	7.7829	7.4394	7.5534
R^2	0.8605	0.9705	0.9710	0.9179
RMSE	12.5734	17.7091	21.4342	20.6072
MAE	6.7385	10.3856	10.6058	9.2320
R^2	0.9558	0.9394	0.9113	0.9180

The results of performance comparison for one day in two cases are shown in Table 1. The grey area in the table represents the forecast results with weather data. It demonstrates that RFR and DTR models have the lowest error and best suitability of fit with weather data compared to other methods. The R^2 values of both models reached 0.97. But only if the load values of peak area are considered, the prediction performance of KNNR actually seems to be better than other methods. This has also been proven in the case without weather data in the lower part of the table. KNNR model has the best performance with the R^2 score, which is more than 0.95, the MAE and RMSE values are only 6.7385 and 12.5734. But as mentioned above, the performance of the KNNR model may be not robust enough with more features. In addition, the prediction performance of the DTR model is also relatively good, the R^2 score has reached 0.93, and the MAE values are only 10.3856.

According to the prediction performance of reference [7], the ML algorithms used in this case study have a higher prediction accuracy, especially when referring to the values of RMSE and MAE. And compared with [18], this paper also provides more options of multi-steps forecasting with comparable prediction accuracy. However, the prediction result must depend on the characteristics of the load curve of the xEVs in the specific application.

5 Conclusions

The load forecasting module can play an important part embedded in a real charging management system to help the control module, so-called "charging infrastructure and energy management" to better solve the dynamic load control problem. A comparison of different ML methods could be made by accessing real-world measurement data. Associated model predictions in the pre-test already proved a sufficient R^2 of the data and thus confirmed the good suitability of the introduced methods for the present use case.

In the framework of the Project KI-LAN, future work will focus on measuring the efficiency of various features and on identifying further useful input data. Some system-based features may also have a large impact on the prediction results, such as some information about charging stations and xEV users, including travel plans and activities. More methods will be tried and also revised by analysing the best possible influencing parameters in order to contribute to an increase in prediction accuracy and develop a more general model with various time horizons of forecasting.

References

1. K. Mahmud, M. J. Hossain and J. Ravishankar, Peak-Load Management in Commercial Systems With Electric Vehicles, *IEEE Systems Journal*, vol. **13**, no. 2, pp. 1872-1882 (2019)
2. Q. Dang, Electric Vehicle (EV) Charging Management and Relieve Impacts in Grids, *2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems* (PEDG), Charlotte, NC (2018)
3. D. Yu, M. P. Adhikari, A. Guiral, A. S. Fung, F. Mohammadi and K. Raahemifar, The Impact of Charging Battery Electric Vehicles on the Load Profile in the Presence of Renewable Energy, *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE*), Edmonton, AB, Canada (2019)
4. AS. Al-Ogaili, TJT. Hashim, NA. Rahmat, et al, Review on Scheduling, Clustering, and Forecasting Strategies for Controlling Electric Vehicle Charging: Challenges and Recommendations, IEEE Access, vol. **7** (2018)
5. J. Zhu, Z. Yang, M. Mourshed, Electric Vehicle Charging Load Forecasting: A Comparative Study of Deep Learning Approaches, Energies, MDPI, Open Access Journal, vol. **12**(14) (2019)
6. ISO 15118-2 (CD, 2017), Edition 2: Road vehicles – Vehicle to grid communication interface – Part 2: Network and application protocol requirement
7. Q. Sun, J. Liu, X. Rong, et al, Charging load forecasting of electric vehicle charging station based on support vector regression, *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference* (APPEEC, Xi'an, 2016)
8. Okfalisa, I. Gazalba, Mustakim and N. G. I. Reza, Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification, *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta (2017)
9. J. Unpingco, Python for Probability Statistics, and Machine Learning, **2**, pp. 214-218 (Springer, 2019)
10. W. Deng, Y. Guo, J. Liu, A missing power data filling method based on improved random forest algorithm, Chinese Journal of Electrical Engineering, vol. **5** (4) (2019)
11. VK. Jain, A. Phophalia, M-ary Random Forest, *Pattern Recognition and Machine Intelligence, 8th International Conference, PReMI 2019, Tezpur, India, December 17-20, 2019, Proceedings, Part II,* pp.161-169 (Springer, 2019)
12. D. Paper, Hands-on Scikit-Learn for Machine Learning Applications: Data Science Fundamentals with Python, **1**, pp. 110, 120 (Apress, 2020)
13. A.V. Joshi, Machine Learning and Artificial Intelligence, **1**, pp. 53-61 (Springer, 2020)
14. G. Ke, Q. Meng T. Finely, et al, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, *Advances in Neural Information Processing Systems 30* (NIP 2017)
15. F. Hutter, L. Kotthoff, J. Vanschoren, Automated Machine Learning: Methods, Systems, Challenges, **1**, pp. 3-8 (Springer, 2019)
16. A. Jindal, M. Singh, N. Kumar, Consumption-Aware Data Analytical Demand Response Scheme for Peak Load Reduction in Smart Grid. IEEE Trans. Ind. Electron. (2018)
17. H. Kaneko, Beware of r2 even for test datasets: Using the latest measured y ‐ values (r2LM) in time series data analysis*, Journal of Chemometrics* 33.2 (2018)
18. J. Zhu, Z. Yang, Y. Chang, A novel LSTM based deep learning approach for multi-time scale electric vehicles charging load prediction, *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia, Chengdu, 20.*