



HAL
open science

Online forecasting of electrical load for distributed management of plug-in electric vehicles

Kaustav Basu, Andres Ovalle, Baoling Guo, Ahmad Hably, Seddik Bacha, Khaled Hajar

► **To cite this version:**

Kaustav Basu, Andres Ovalle, Baoling Guo, Ahmad Hably, Seddik Bacha, et al.. Online forecasting of electrical load for distributed management of plug-in electric vehicles. REDEC 2016 - International Conference on Renewable Energies for Developing countries (REDEC 2016), Jul 2016, Beyrouth, Lebanon. 10.1109/REDEC.2016.7577557 . hal-01323253

HAL Id: hal-01323253

<https://hal.science/hal-01323253>

Submitted on 30 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Online forecasting of electrical load for distributed management of plug-in electric vehicles

Kaustav Basu, Andres Ovalle, Baoling Guo, Ahmad Hably *, Seddik Bacha , Khaled Hajar

Univ. Grenoble Alpes, GIPSA-Lab
F-38000 Grenoble, France.
ahmad.hably@gipsa-lab.grenoble-inp.fr

Abstract—The paper aims at making online forecast of electrical load at the MV-LV transformer level. Optimal management of the Plug-in Electric Vehicles (PEV) charging requires the forecast of the electrical load for future hours. The forecasting module needs to be online (i.e update and make forecast for the future hours, every hour). The inputs to the predictor are historical electrical and weather data. Various data driven machine learning algorithms are compared to derive the most suitable model. The results indicate that an online forecasting method has an error between 2-5% for the future 24-hour. The decentralized management system works well with the forecasting data.

Index Terms—Short term load forecasting, smart grid, plug-in electric vehicles, support vector machines, machine learning.

I. INTRODUCTION

Environmental issues have been a challenging concern all over the world, particularly the greenhouse gas emission and air pollution. Plug-in Electric Vehicles (PEV) as a new environmental friendly transportation alternative, have been emerging solutions to meet these concerns in recent years [1], [2]. Firstly, PEV can improve the air quality by reducing road transport emissions. Furthermore, in an intelligent energy management system, PEV can be assumed as a distributed energy storage , which is useful to balance the load curve by charging the battery when it is in low demand and discharging it towards the grid at peak time. Additionally, its energy storage potential can compensate the intermittency of renewable energy such as the wind and solar power. It is also useful to enhance grid power quality [3], [4].

Nevertheless, all benefits are not viable without

an advanced flexible energy management system. Otherwise, non-managed PEV load may cause abnormal voltage levels, highly unbalanced loading for the three phase distribution transformers, and other power quality issues. With the PEV technologies getting more mature, optimal load management becomes a key element to take full advantage of PEV benefits. In the past few years, many approaches have been proposed concerning this research issue. Authors of [5] propose a centralized approach which controls the PEV charging directly. The PEV aggregator acts as the charging service provider (CSP), individual charging schedules for each PEVs are considered in the charging control. Additionally, some methods of combining PEV load scheduling with wind power integration have been proposed. In [6], the objective is to control the PEV charging scheduling to meet the wind power production. These methods depend mostly on the wind power and PEV load forecast. In [7], a PEV charging algorithm is proposed following users price preferences. This algorithm takes both server side and user side into consideration. It puts up a pricing policy with simple bidding strategy considering aggregators predicted charging and desired load profile. Following a game theory approach in [8], a decentralized optimization frame is defined. A local Dynamic Programming (DP) algorithm for PEV load scheduling is proposed, taking into account the desired states of charge, power consumption limitation, charging power limitation. Defining PEVs as players in an N-person non-cooperative potential game, a PEV uses the DP algorithm in order to define the best response that corresponds to optimizing its payoff, given that

schedules of other PEVs (other players) are fixed. The payoffs for the potential game are formulated by an objective function that reduces the variance of the total load profile.

The paper is organized as follows. Section II presents the PEV decentralized management algorithm proposed in [4]. In Sect. III, different classic forecasting algorithms are introduced and compared. Section IV describes online learning model applied in this paper. In Sect. V, a comparison of the performance of different learning algorithms, and testing results with a real data are presented and analyzed. Section VI provides some conclusions and perspectives.

II. PEV DECENTRALIZED MANAGEMENT SYSTEM

Given that this paper is based on the algorithm of paper [4], a detailed introduction is given in this part. It proposes a PEV load management algorithm by applying a dynamics proper of the evolutionary game theory, known as the Mixed Strategist Dynamics (MSD). Besides, by introducing some analogies and applying the Maximum Entropy Principle (MEP), two objective functions are formulated: entropy measurements on the total load distribution and the local load distributions. The entropy maximization for local load distributions can preserve the batteries state of health; the entropy maximization for the total load distribution is intended to balance the transformer load curve to the greatest extent by shaving the peak and filling in the valley. A trade-off among them can be decided by the PEV owners convenience. Thus in the scheduling process, PEV owners have more right to determine how much they participate in, which can be economically incentivized by the utility grid manager tariff policy.

The decentralized management algorithm is described in Fig. 1. There exist two main actors: PEV and aggregator in the load management system.

A. The role of PEV

Each PEV is equipped with a local MSD algorithm, which manipulates its own load distribution. Once it has received new information from the aggregator, its charging power schedule is updated. Once the algorithm reaches the limiting amount of

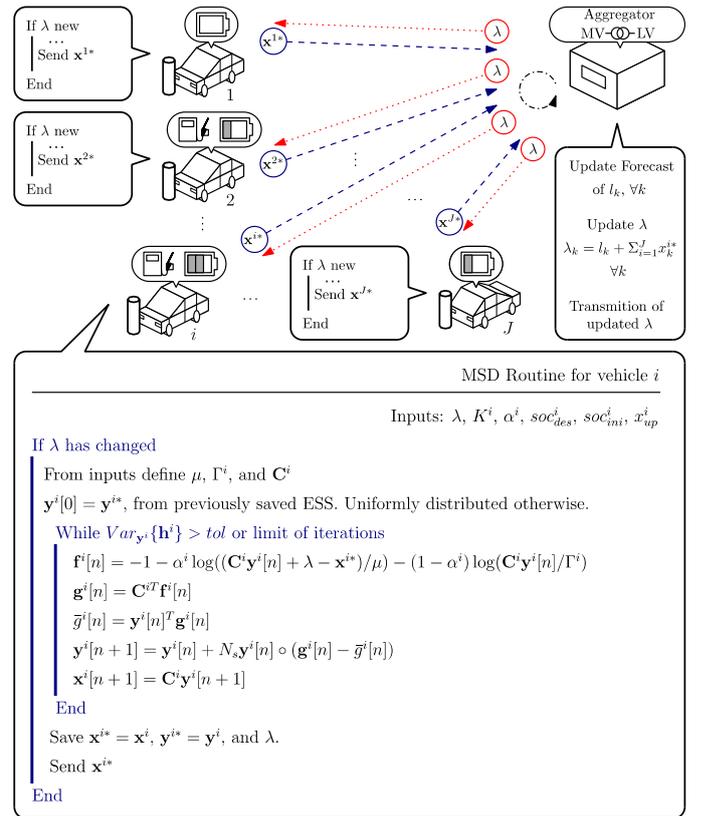


Fig. 1: Description of the decentralized MSD algorithm. With some important parameters as follow, λ is sum of total load and PEV distribution load; K^i is pure strategies, α^i is trade-off parameter; soc_{ini}^i and soc_{des}^i are initial and desired states of charge of PEV respectively. All the other parameters can be referred to [4].

iterations or meets a convergence criterion, the new profile is sent back to the aggregator.

B. The role of aggregator

It is the manager of the approach. It collects the load schedules from from PEVs and updates the total load distribution. Then, it redistributes the updated information of total load to PEVs under the transformer of the distribution system. To emphasize, the updating process is based on both the PEVs load schedules and the short time forecast of total transformer. The aggregator should continually receive and send updated information to each of the PEVs. The optimization algorithm depends strongly on the performance of the intrinsic load forecasting. In this paper, different forecasting

methods are studied and tested for the convenience of the distributed optimization algorithm.

III. SHORT TERM LOAD FORECASTING

The problem of forecasting has been a subject of research for a considerable number of years. The workability of a technique depends on its simplicity and comprehensibility of the model being used. Forecasting has been used in a number of domains, at first, one needs to look into the domain of *load forecasting* at the grid level and review the approaches used. In this section, a brief summary of load forecasting models is provided and subsequently the model used for transformer level load is discussed in details. Load forecasts can be divided broadly into three categories:

- 1) Short term forecasts: From one hour to one week.
- 2) Medium term forecasts: From a week to a year.
- 3) Long-term forecasts: Longer than a year.

The problem of online transformer level forecasting falls in the short-term load forecasting (STLF) category. Some of the state of the art approaches used in STLF are highlighted in [9]. The approaches are not independent of each other but rather complementary. An important aspect of forecasting algorithms is that the simpler models are more popular among utilities, of course without being trivial. The features of a forecasting algorithm depend on the business needs. The challenge is not only to be technical but analytical in the approach to build a proper model of forecasting tool.

A. Similar-day approach

This approach is based on searching historical data for days within one, two, or three years with similar characteristics to the day of forecast. Similar characteristics include weather, day of the week, and the date for example. In the case of a load forecasting, the load of the similar day is taken as a forecast [9].

This process is simple to comprehend but in relevant cases it can outperformed more complex mathematical approaches. It also must be mentioned that, though this method is simple, it is not trivial and requires a good understanding of the domain of interest.

B. Regressive methods

For electric load forecasting, regression methods are usually used to model the relationship of loads consumption and other factors such as weather, day type, and customer class. [10] presents several regression models for the next day peak forecasting. The linear regression is one of the basic models for forecasting and more complex algorithm (Decision tree learner, Support vector machines) use it as the base model. For a time series (X) The basic univariate regression can be expressed by the following equation:

$$X_{t+1} = a_1X_t + a_2X_{t-1} + a_3X_{t-2} + \dots + a_wX_{t-w} \quad (1)$$

where, w is the lag length and a is the coefficient which needs to be evaluated. The goal during training is to minimize the sum of the squared errors to fit a straight line to a set of data points. The multivariate linear regression is used in this application.

C. Time series

Time series methods are based on the assumption that the data have an internal structure, such as autocorrelation, trend, or seasonal variation. In particular ARMA, ARIMA and ARIMAX are the most used classical time series methods [11]. The idea of the time series approach is based on the understanding that a load pattern is nothing more than a time series signal with seasonal, weekly, and daily periodicities. Generally, techniques in time series approach work well unless there is an abrupt change in the environmental or sociological variables which are believed to affect load pattern. Our work takes into account the time series approaches using a data driven discriminative learning method.

D. Expert systems

Expert systems incorporate rules and procedures used by human experts in the field of interest into software. From that knowledge, these software are able to automatically make forecasts without human assistance. Knowledge-based expert system for the short-term load forecasting have already been successfully deployed in the world, for example for the Taiwan power system [12]. In this example, operators knowledge and the hourly observations of system loads along with weather parameters were

taken into consideration. Our model also proposes a general model which takes temporal behavior patterns into consideration.

E. The *K*-Nearest Neighbors classifier, KNN

The KNN classifier is an instance based learning method where the classification function is approximated by a majority vote of the neighbors using a distance metric. *K* is the number of neighbors which is calculated using cross-validation. The function is approximated locally and all computation is deferred until classification. Typically, the Euclidean distance is used as distance function. The mechanism of KNN is quite simple. For any new data instance, the attributes of the new case is compared with all the previously seen cases or instances in the training database. The comparison is typically based on a distance measurements. The nearest instances or cases in the training database are evaluated based on the distance metric. The new instance is assigned to the class of the majority of neighboring instances (process of classification). Mathematically, for any instance x_i in the database of size $n \times x$, the distance is expressed as:

$$d(x_i) = \min_j d(x_i, x) \text{ with } j \in \{1, \dots, n\} \quad (2)$$

The key point here is the use of the proper distance metric. The default metric is the Euclidean distance, which is calculated on the normalized value of the attributes. The Euclidean distance is a standard metric to calculate the distance between two vectors X, Y . It is given by the following equation:

$$d_E(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3)$$

where n is the total number of points and x, y are the corresponding points representing the vector (X, Y) .

F. Decision Tree Learner, DTL

The decision tree consists of nodes where a logical decision has to be made. Branches are connected according to the result of these decisions. For each node of the tree, one attribute of the data is selected that most effectively splits its set of samples into subsets enriched in one class or the other. Following a path of nodes and branches constitute

a sequence through a decision tree that reaches to a final decision. The DTL algorithms represent one of the preferred choices for load forecast as described in [13]. Indeed, decision trees are rule based and the built model is easy to visualize. A good quantitative measure of the significance of an attribute is a statistical property called *information gain*. It measures how well a given attribute separates the training examples according to their target classification. This measure is used to select among the candidates attributes at each step while the decision tree is growing. The attribute presenting the highest normalized information gain is chosen to make the decision.

The metric used in practice is the gain ratio which corrects the information gain by taking the intrinsic information of a split into account. Then, the algorithm applies it recursively on the sub-lists. The *M5P* ([14]) algorithm is used from and the parameter is optimized using parameter section algorithm during training.

The DTL usually leads to a good understanding of the significant features for the load forecast. Based on the disparity measurement, the attribute with the highest normalized information gain is chosen as the root of the decision tree. Information gain is measured in bits and is given a probability distribution, the information required to predict an event is the distribution's entropy, given by:

$$S(p_1, p_2, \dots, p_n) = -p_1 \log(p_1) - \dots - p_n \log(p_n) \quad (4)$$

G. Support Vector Machines, SVM

Support vector machines perform a non-linear mapping (by kernel functions) of the data into a higher dimensional feature space. Then the algorithm uses simple linear functions to create linear decision boundaries in the new space. SVM model can be used to predict daily load demand, for example for the next month [15]. The SVM algorithm is a powerful tool for data classification described in [16]. The first major step of a SVM classification is to build a decision plane that separates a set of objects with different class memberships. It guarantees the best function to distinguish between members of classes by maximizing the margin between them. The maximal margin hyper-planes allow the best generalization abilities and thus the best

classification performances on the training dataset. This procedure requires finding the solution of the following optimization problem:

$$\min_{w,b,\xi} \left(\frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \right) \quad (5)$$

$$\text{subject to } \begin{cases} y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \quad (6)$$

with l the total number of sub-sequences, w the normal vector of the hyper-plane, b the offset of the hyper-plane, C the penalty parameter of the error term ξ and ϕ the kernel function.

The second major step is to choose the kernel function of the algorithm. The Radial Basis function kernel gave the better performance in the tests (not presented). For two groups i and j , the training vectors x_i and x_j are mapped to a higher dimensional space by the kernel function ϕ defined as:

$$\begin{cases} K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j) \\ K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2); \gamma > 0 \end{cases} \quad (7)$$

where γ is a parameter of the kernel. A grid-search has been conducted on the parameters C . The SVM algorithm is computationally more expensive than rule based algorithms such as DTL. The Sequential Minimal Optimization (SMO) implementation of [14] is used with a grid search for parameter optimization during training. There has been other works using the hierarchical structure of grid load to make innovative forecast [17].

IV. THE ONLINE LEARNING MODEL

A stepwise outline of the future usage forecast implementation is enumerated below. These steps can be cycled on a predefined time period, transforming a discriminative learning algorithm into an online learning model.

- 1) All the power values are obtained as input at a 1-hour sampling.
- 2) Sub-sequences are generated using temporal sliding window with a window size of 24 units.
- 3) The temporal data (hour, date) and meteorological data (temperature, humidity) are used as input for each next forecast.

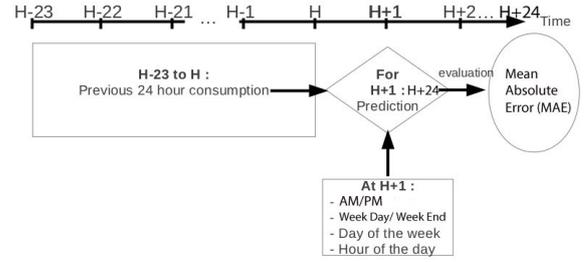


Fig. 2: Proposed method at a given time instance [18].

- 4) The discriminative learner (e.g. Svm) with the generated features as inputs and the future hour consumption as output are trained.
- 5) The model is learned iteratively and is tested in an online learning procedure (cycling this process to the first item once again).

Fig. 2 describes the principle of the proposed model. At every sampled time instance, the future 24-hour is foretasted in an iterative process. The inputs to the model can be categorized thereby :

- The consumption at the transformer level each hour for the previous 24 hours.
- The time of event and meteorological information.

The inputs to the system are shown in the Fig. 2 for a given time instance (hour H). At each time instance the system predicts the coming hour and then the window is shifted one hour for the next forecast. The input stream consists of the consumption of the past load sequence. The initial step is to populate the sliding window with sufficient historical data that aims at creating a single test instance to start the closed loop forecasting process for the future time steps. The subsets of the original time series are then shifted in time creating thereby the sub-sequences and preserving time dependency among sub-sequences. Instances containing these sub-sequences are finally presented as standard propositional instances to the classification algorithm. This process is illustrated in the Fig. 3. As the variables in each sub-sequence are considered as independent the time dependency is lost. The temporal sliding window principle injects back this dependency among the sub-sequences. Once the classifier produces a forecast for the next time step, this classified value moves into the sliding

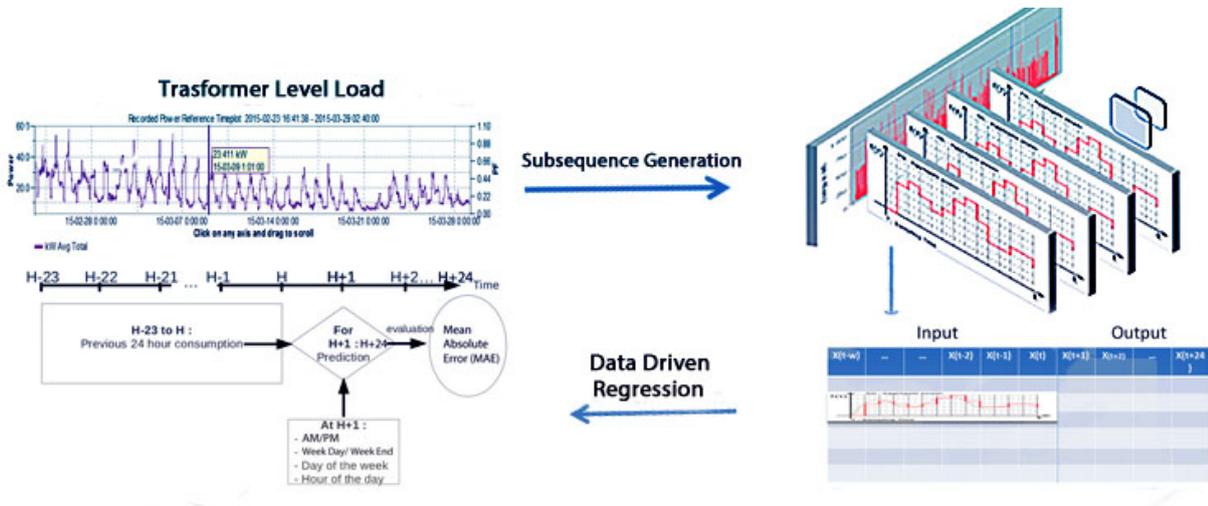


Fig. 3: Data representation in a standard propositional learning format.

time window as the most recent value of the target and the oldest value in the window falls out. Another test instance is then created from the history window and the next time step is classified. This is also known in the literature as “closed loop forecasting”. Once the classifier is trained, it only needs to be “primed” each time for classifying future instances. Two additional remarks have to be made at that point. First, this methodology results in a high amount of data which is processed subsequently at the forecasting stage. Second, a significant constraint is that the states of the loads of the previous hours are sequential. This information is known only after the availability of the previous events. Then, it is only possible to predict the Hour ($H + 1$) if we have all the needed information about the Hour H . On the other hand the time of event and meteorological information (temperature and humidity) are available for future time instances and are relatively independent of the current time. Furthermore, the time of event is expressed as two periodic variable: hour of the day and day of the week.

This way to consider the time allows to take into account the periodic nature of human behavior. Note that we could have also added the week of the year, or the season of the year, etc. Focusing on a daily forecast, taking too much time related inputs would not have changed the results.

V. RESULTS

The results are presented in two parts. Firstly, different forecasting methods are compared to ascertain the most suitable algorithm for such application and the performance. Finally, the forecast result at 24-hour future step is used for the underlying PEV scheduling application. The data is measured on a distribution transformer from SOREA utility grid company in the region of Savoie, France.

A. Performance of forecasting algorithms

A comparison of 1-step ahead errors of four classic learning algorithms are listed in TABLE I. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are two of most widely used parameters to evaluate the forecasting performance. From the comparison, it can be seen that the forecasting algorithm SVM has the smallest error. The experiments in section V-B are conducted with the above forecasting algorithm. 1-step ahead forecasting result of SVM algorithm is shown in Fig. 4.

B. PEV optimization performance

MSD decentralized optimizing algorithm introduced in section II simulates a real-time system, thereby, 24-hour future forecasting results should be updated per hour to realize a real-time optimization. Each PEVs’ arrival and departure time are defined with statistically modeled random values. The number of PEVs in current time is shown in Fig. 5.

TABLE I: Comparison of 1-step ahead errors of four classic learning algorithms

Algorithms	MAE	RMSE
Linear regression	2.1204	2.787
KNN	3.4173	4.5554
DTL	1.9946	2.6354
SVM	1.7986	2.4433

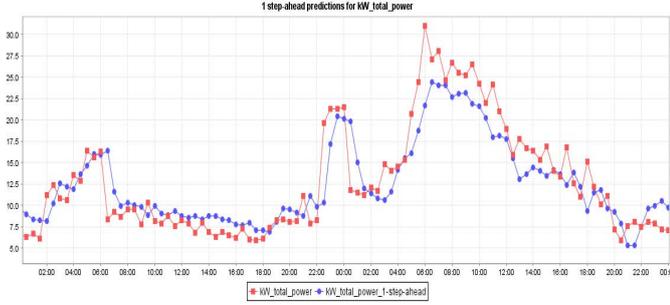


Fig. 4: 1-step ahead forecasting results with SVM forecasting algorithm. The red curve is the actual data measured on a distribution transformer, the blue curve is the 1-step ahead forecasting results on the base of actual history data.

The 24-hour optimization performance using the actual history and forecasting data respectively are presented in Fig. 6. The comparison results show the optimizing algorithm works well with forecasting data, which nearly has the same performance as the actual data.

VI. CONCLUSION

The paper addresses the online forecasting of active load at the transformer level for the electrical vehicle charging application. The future forecast of 24-hour load at 1-hour step is modeled. This method is tested for a real French database of 1-month duration. The results indicate that Support Vector Machine for regression performance is better among the algorithms compared. The error for the future 24-hour is between 2 to 5% and result indicate that the underlying EV optimization performance remains approximately similar. The forecasting methods may be enhanced by using other contextual factors and data from previous years which were not available for this work.

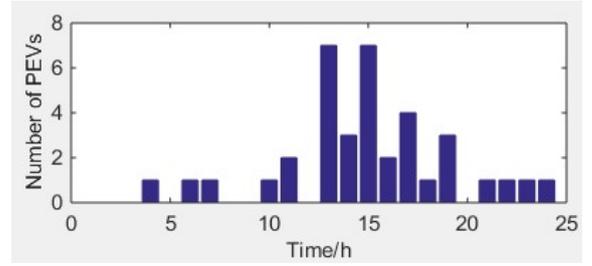
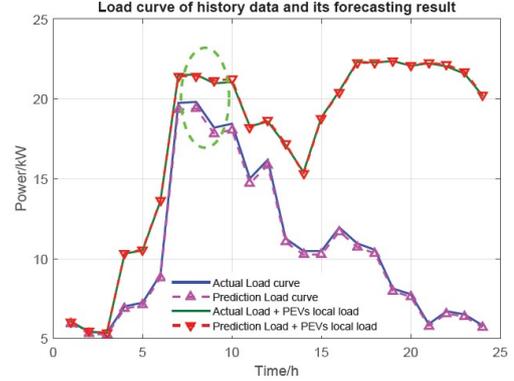
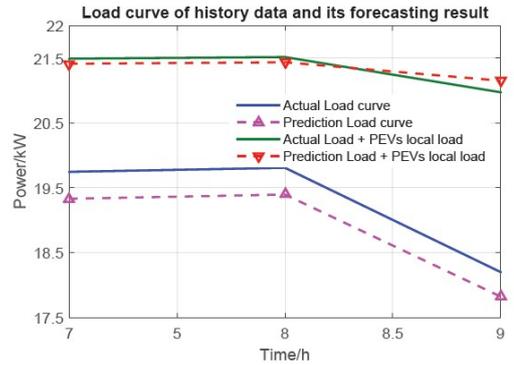


Fig. 5: Number of PEVs in current time.



(a)



(b)

Fig. 6: Example of 24-hour optimization curves with actual history data and forecasting data, trade-off factor is set 0.95 and iteration times are equal to 50. Fig.6(a) is a zoom to part of Fig.6(b).

VII. ACKNOWLEDGEMENT

This work has been a joint collaboration by the project PARADISE which is sponsored by the National Research Agency (ANR). The work is implemented using weka [19].

REFERENCES

- [1] K. M. Tan, V. K. Ramachandaramurthy, and J. Y. Yong, "Integration of electric vehicles in smart grid: A review on vehicle

- to grid technologies and optimization techniques,” *Renewable and Sustainable Energy Reviews*, vol. 53, pp. 720–732, 2016.
- [2] A. Poullikkas, “Sustainable options for electric vehicle technologies,” *Renewable and Sustainable Energy Reviews*, vol. 41, pp. 1277–1287, 2015.
- [3] S. Habib, M. Kamran, and U. Rashid, “Impact analysis of vehicle-to-grid technology and charging strategies of electric vehicles on distribution networks—a review,” *Journal of Power Sources*, vol. 277, pp. 205–214, 2015.
- [4] A. Ovalle-Villamil, J. Fernandez, A. Hably, and S. Bacha, “An electric vehicle load management application of the mixed strategiest dynamics and the maximum entropy principle,” *Industrial Electronics, IEEE Transactions on*, 2016, to be published.
- [5] O. Sundström and C. Binding, “Flexible charging optimization for electric vehicles considering distribution grid constraints,” *Smart Grid, IEEE Transactions on*, vol. 3, no. 1, pp. 26–37, 2012.
- [6] Q. Huang, Q.-S. Jia, and X. Guan, “A review of EV load scheduling with wind power integration,” *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 223–228, 2015.
- [7] B. Wang, B. Hu, C. Qiu, P. Chu, and R. Gadh, “EV charging algorithm implementation with user price preference,” in *Innovative Smart Grid Technologies Conference (ISGT), 2015 IEEE Power & Energy Society*. IEEE, 2015, pp. 1–5.
- [8] A. Ovalle-Villamil, A. Hably, and S. Bacha, “Optimal management and integration of electricvehicles to the grid: Dynamic programming and game theory approach,” in *IEEE International Conference on Industrial Technology (ICIT 2015)*, 2015.
- [9] E. Feinberg and D. Genethliou, “Load forecasting,” in *Applied Mathematics for Restructured Electric Power Systems*, ser. Power Electronics and Power Systems, J. H. Chow, F. F. Wu, and J. Momoh, Eds. Springer US, 2005, pp. 269–285.
- [10] R. Engle, C. Mustafa, and J. Rice, “Modelling peak electricity demand,” *Journal of Forecasting*, p. 241251, 1992.
- [11] V. Tran, V. Debusschere, and S. Bacha, “Hourly server workload forecasting up to 168 hours ahead using seasonal arima model,” in *Industrial Technology (ICIT), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1127–1131.
- [12] K. Ho, Y. Hsu, C. Chen, T. Lee, C. Liang, T. Lai, and K. Chen, “Short term load forecasting of Taiwan power system using a knowledge-based expert system,” *Power Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 1214–1221, nov 1990.
- [13] J. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, pp. 81–106, 1986.
- [14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update,” *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
- [15] B. Chen, M. Chang, and C. Lin, “Load forecasting using support vector machines: A study on eunite competition 2001,” European Network on Intelligent TEchnologies for Smart Adaptive Systems, Tech. Rep., 2001.
- [16] T. Onoda, G. Rätsch, and K.-R. Müller, “Applying support vector machines and boosting to a non-intrusive monitoring system for household electric appliances with inverters,” 2000.
- [17] X. Sun, P. B. Luh, K. W. Cheung, W. Guan, L. D. Michel, S. Venkata, and M. T. Miller, “An efficient approach to short-term load forecasting at the distribution level,” *Smart Grid, IEEE Transactions on*, vol. pp, no. 99, pp. 1–12, 2015.
- [18] K. Basu, V. Debusschere, and S. Bacha, “Appliance usage prediction using a time series based classification approach,” in *Industrial Electronics (IECON), IEEE Conference on*, October 2012, pp. 1217–1222.
- [19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, “The weka data mining software: An update,” <http://www.cs.waikato.ac.nz/ml/weka>, 2009.