Accuracy-Resource Tradeoff for Edge Devices in Internet of Things

Nima Mousavi
Electrical and
Computer Engineering
UC San Diego

Baris Aksanli
Electrical and
Computer Engineering
San Diego State University

Alper Sinan Akyurek
Electrical and
Computer Engineering
UC San Diego

Tajana Šimunič Rosing
Computer Science
and Engineering
UC San Diego

***Abstract*—Modern power grid has evolved from a passive network into an application of Internet of Things with numerous interconnected elements and users. In this environment, household users greatly benefit from a prediction algorithm that estimates their future power demand to help them control off-grid generation, battery storage, and power consumption. In particular, household power consumption prediction plays a pivotal role in optimal utilization of batteries used alongside photovoltaic generation, creating saving opportunities for users. Since edge devices in Internet of Things offer limited capabilities, the computational complexity and memory and energy consumption of the prediction algorithms are capped. In this paper we forecast 24-hour demand from power consumption, weather, and time data, using Support Vector Regression models, and compare it to state-of-the-art prediction methods such as Linear Regression and persistence. We use power consumption traces from real datasets and a Raspberry Pi 3 embedded computer as testbed to evaluate the resource-accuracy trade-off. Our study reveals that Support Vector Regression is able to achieve 21% less prediction error on average compared to Linear Regression, which translates into 16% more cost savings for users when using residential batteries with photovoltaic generation.**

I. Introduction

Internet of Things (IoT) is a collection of sensing and actuation that is supported by existing and growing Internet infrastructure [1]. As an application for IoT, Smart Grid aims to provide pervasive control for power grid [2]. This task is accomplished by creating intricate applications involving several elements in industrial and household environments. The edge devices in these systems consist of embedded computers that offer lower computational capabilities.

In addition to a shift in control potential, household users are now capable of generation and storage of electrical energy. For instance, the decreasing cost of solar photovoltaic devices has rendered them as one of the most economical off-grid generation methods [3], while other renewable energy sources such as wind are also used. The main challenge of integrating these sources is their intermittent and highly variable nature. The ability of batteries to act as generators as needed with volatile renewable energy resources can help smooth out the renewable output [4]. Researchers expect batteries to be more commonly used in houses to store energy generated by off-grid methods (i.e., solar PV or small wind turbines). This method can help reduce the dependency of users to the grid by increasing the consumption of locally generated energy from 30% up to 70% [5]. Evaluations show that battery storage is already economically viable for small PV systems under different future pricing scenarios, and it will become even more critical with the current trends in electricity pricing [6].

To optimally use batteries alongside renewable resources in a multi-tier pricing scheme, a fairly accurate prediction of household power consumption is required [4]. Smart Grid systems have millions of consumers and devices [7], it is technically challenging to implement these personalized algorithms centrally, i.e., with cloud based applications or by utilizing utility company's infrastructure.

Household power profile follows a sporadic pattern which is difficult to predict [8]. We made an information theoretical analysis and showed that household demand prediction is expected to have very low accuracy due to the highly random human behavior (Section V-B1). However, we also show that any improvement in accuracy results in significant cost saving for the users by optimal battery utilization.

A local power prediction algorithm implementation is implemented at the house level and thus needs to use inexpensive edge devices, e.g. small embedded computers, to collect and process data. These devices tend to have limited memory and processing capabilities. This makes the prediction task even more challenging, as most of the previous works on household power prediction ignore computational complexity and local embedded implementation feasibility of their algorithms [9] [10]. We fill this gap by analyzing accuracy, training time, and energy consumption of state-of-the-art prediction algorithms such as Linear Regression (LR), and comparing them to Support Vector Regression (SVR) [11] models for 24-hour daily power demand forecast. To assure feasibility of local implementation for these models, we only use recent power consumption and weather data as inputs. Our study reveals that SVR is able to achieve 21% less prediction error compared to LR, while LR shows better scalability characteristics for larger training sets. We show how this increased accuracy translates into 16% more cost savings for users through solving the optimal load flow problem for households that use batteries to smooth out solar generation variability. Also, we investigate the resource requirement of these prediction algorithms by implementing the models on a Raspberry Pi 3[1] device.

[1] We choose this platform because it is available at low cost, widely used within similar works, and offers great software and community support [12].

II. Related Work

A recent survey on electrical power demand forecasting by Hernandez et al. [8] classifies several methods for power consumption prediction. Their analysis scope mostly contains works focusing on aggregated power consumption, such as at city or country level. They also emphasize the challenge of prediction for individual users due to sporadic nature of time series. There are several other works which contain solutions and models for aggregated short-term load forecasting[2] and do not cover individual user level prediction [13] [14] [15].

Among the works on individual user level power prediction in smart grid, the goal is to maximize the benefit of the utility company [9] [10] [18]. Dudek [14] measures the running time of several power prediction algorithms on a desktop system. However, the energy draw of each algorithm and their suitability for a small, resource-limited edge device is not studied. Hence, these algorithms are either based on offline optimization methods, or they overlook the analysis of computational requirements of their algorithm, which itself casts doubt on their suitability for embedded edge devices.

Mateo et al. [16] and Edwards et al. [17] compare several machine learning algorithms by their capability to predict short term power consumption of a building. However, both these works lack analysis for computational demand of their algorithms, as well as potential cost savings. Furthermore, [17] is based on data from an experiment which simulates household occupancy with predefined behavioral patterns, which may not represent real household usage.

Logenthiran et al. [19] develop a day ahead demand side management algorithm which facilitates peak shifting in daily power profile. Their method does not involve savings from optimal battery usage, and study of limitations in embedded implementation is not covered. Wang et al. [20] present a power consumption prediction algorithm that is used alongside a control method for photovoltaic generation and batteries, but does not mention the computational requirement of their approach. Hossa et al. [21] describes the impact of power prediction on user's actions and electricity cost, but they do not study the computational demand of their algorithm or whether the algorithm suits embedded environments. Their prediction requires one year weather data, which may not be readily accessible at the house level, and requires large data storage.

Support Vector Regression (SVR) [11] has been used in time series prediction algorithms spanning several application domains [22] [23]. However, the models are usually trained offline. Therefore, the processing and memory demand of algorithms are not thoroughly analyzed.

Bajaj et al. [24], Hsieh et al. [25], and Haigh et al. [26] propose methods for implementing support vector classification in resource limited environments, however error for a power prediction application (or any other regression scenario) as well as algorithm energy consumption is not discussed.

Even though power consumption prediction has been subject to several studies, the main focus has been on creating elaborate methods that achieve the highest level of accuracy possible within a dataset. However, we argue that simpler models that achieve lower accuracy levels can provide cost savings for residential users by optimizing energy consumption and battery storage. Lower computational requirement of these models means that these methods can be implemented locally on edge devices.

III. Prediction Model

Many residential control applications, such as the optimal battery economic dispatch problem, rely on prediction of future power consumption of the household [4]. In Section II we discuss different methods that can be used to accomplish this task. We choose linear regression (LR) as a basic prediction model and baseline. We select LR because of its simple and low complexity solution, and compare it with support vector regression prediction. The latter is capable of predicting nonlinear usage patterns that are specific to each house. In this section we explain the implementation these methods.

The goal is to predict power demand profile $P(d)$ during the course of day d. This profile contains power predictions for every hour of a day. We use average power consumption of every hour in previous day and latest available weather data as inputs to our prediction model. To further tune our model we use time of day and day of week attributes to accommodate different usage patterns at different times. Equation (1) shows the general model that we seek to find,

$$\hat{P}(d) = f\left(P(d-1), T, H, CC, DP, AT, ToD, DoW\right), \quad (1)$$

where T, H, CC, DP, and AT correspond to weather data (temperature, humidity, cloud cover, dew point, and apparent temperature), ToD and DoW show time of day and day of week. $P(d)$ consists of a 24^{th} Markov order of 1-hour separated power consumption data, collected from day d,

$$P(d) = \{p_h(d,0), p_h(d,2), \cdots, p_h(d,23)\}, \quad (2)$$

where $p_h(d,t)$ corresponds to power consumption at day d during hour t. We use 24 separate models for each hour of the day, and train each model individually. By this mean, power consumption for every hour in day d is predicted by one of the 24 models with horizon between 1 hour to 24 hours. We need at least 24 models to be able to forecast power consumption throughout a full day.

To fully capture the dynamic nature of the usage pattern, we need to update the predictor function frequently. We define parameter L_{test} as the number of days that a model remains valid before getting updated with the newest available data. In each update, we use the past L_{train} days of data for training. We compare two methods for model training, linear regression (LR) and support vector regression (SVR), that are described in the following subsections.

A. Linear Regression (LR)

Linear regression (LR) model predicts $\hat{P}(d)$ as a linear combination of all input variables. Equation (3) shows the general LR model. In this equation y is the predicted variable, which is defined as a linear combination of x_i predictors. The

[2] Since household consumption mainly comprises of daily or weekly usage patterns, the demand prediction in this application falls beneath short-term load forecasting.

goal in the training phase is to determine β parameters that achieve lowest error.

$$\hat{y} = \beta_0 + \beta_1.x_1 + \beta_2.x_2 + ... + \beta_n.x_n \tag{3}$$

In this problem, the predicted variable is $\hat{P}(d)$ which is predicted using $p_h(d-1,t)$ data, weather data, and time. This simple model has been used in several applications to predict linearly codependent variables described in (1) and (2). Although household power consumption data may not be ideally described by a linear model, we use this prediction method as a baseline to compare improvements that can be achieved using SVR (section III-B).

B. Support Vector Regression (SVR)

Support vector regression is a prediction model that can extract linear or non-linear relations between input and output variables. The advantage of SVR over other models for prediction is that it minimizes the structural risk, as opposed to minimizing empirical risk or training error [11]. This results in better generalization to unforeseen data in the test phase.

1) Linear SVR: The relation between input and output of a linear model can be formulated as

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \tag{4}$$

where $\mathbf{x}$ is input, $\langle ., . \rangle$ corresponds to dot product, and $\mathbf{w}$ and b are the parameters that define the characteristics of the model [11]. In order to find these parameters we have to solve an optimization problem:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \le \varepsilon + \xi_i \\ & \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \le \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \ge 0. \end{aligned} \tag{5}$$

In (5), ε denotes how much each predicted $f(x_i)$ is allowed to deviate from y_i, and $C > 0$ determines the penalty for these deviations. We can tune these parameters to trade between overfitting and generalization ability of the model. The details of the solutions will not be discussed here due to limited space, readers can refer to [11] for a more detailed explanation of solution to the optimization problem.

2) Non-Linear Extension: Non-linearity in support vector regression is achieved by mapping the input space into a higher dimensional feature space, and then applying the optimization problem that was described in the previous section [17]. In this work we use Radial Basis Function (RBF) kernels, as they achieved lowest prediction error for our power prediction scenario. The modified problem constraints and the RBF kernel are defined in (6).

$$\begin{aligned} y_i - \langle \mathbf{w}, \varphi(\mathbf{x}_i) \rangle - b \le \varepsilon + \xi_i \\ \langle \mathbf{w}, \varphi(\mathbf{x}_i) \rangle + b - y_i \le \varepsilon + \xi_i^* \\ K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \end{aligned} \tag{6}$$

3) Parameter Selection: The SVR model that was described in the previous section is mainly identified by ε, maximum allowed deviation from predicted value, C, the penalty assigned to these deviations, and γ, the kernel scaling parameter. We use cross-validated grid search [17] for each house in the data set, and found that $\varepsilon = 0.1$, $C = 10$, and $\gamma = 0.001$ achieve the best average result over all houses. Since we have implemented this model in an embedded environment, the runtime of prediction training is important and we cannot perform grid search for each training.

IV. Application: Optimal Battery Flow Problem

PV installations, along with batteries, have become an economically viable option when exploiting the current trends in electricity pricing (e.g. time of use pricing) to provide monetary savings [6]. Smart control algorithms, such as optimum battery flow solutions, are required to maximize the benefits of these systems. Akyurek et al. [4] describes an optimal and low complexity control algorithms that smooths out the intermittent output from renewable electricity sources, while storing extra generated energy to minimize cost. The algorithm takes house power consumption profiles as an input. Although accurate time series prediction is a computationally heavy operation, using a powerful general purpose device in each house for this purpose is not cost-effective. Instead, we propose to perform time series prediction on resource-limited edge devices and use the optimal battery flow problem as a viable application that can tremendously benefit from such prediction. In this application, the controller requires the power demand forecast along with expected solar generation for the upcoming day (next 24 hours) in the beginning of every day. The optimization is described by:

$$\begin{aligned} \underset{bat}{\text{minimize}} \quad & Cost(bat, loadForecast, pvForecast) \\ \text{subject to} \quad & LPower < bat < UPower \\ & LCharge < SoC(bat) < UCharge. \end{aligned} \tag{7}$$

In (7), battery flow is denoted by bat and it is constrained by battery power bounds, $LPower$ and $UPower$. Battery state of charge is also marked by SoC which is bounded by battery energy bounds, $LCharge$ and $UCharge$. The true value for consumed power on each day along with optimal battery flow, bat, solar generation prediction (as explained in V-A), $pvForecast$, and power prediction for the day, $loadForecast$, is then used to calculate the savings from using these algorithms. The results are reported as cost reduction compared to the case were batteries are not used. Figure 1 shows the prediction, optimal battery flow solution, and cost saving computation.

A. Cost Definition

We use quadratic cost that approximates multi-tier pricing. This method has been used throughout literature (i.e., [27] [4]), as it encourages a flatter profile and it is employed by utility companies. We incorporate the same method from [4] and use prices from San Diego Gas & Electric [28]. We report cost reduction as a percentage of the actual cost.

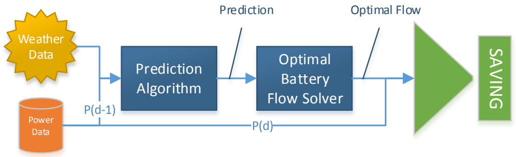


Fig. 1: Optimal load flow problem solution and cost saving calculation flowchart.

B. Algorithm Overhead

Since our goal is to run prediction algorithms on embedded edge devices, we need to limit the computational overhead of the prediction algorithms. The algorithm that is used to solve the battery flow problem matches this criteria, as it has $O(n^2)$ algorithmic complexity, where n corresponds to the maximum prediction horizon. Since n is bounded in our implementation ($n = 24$ hours), the overhead of this algorithm will be dominated by more complex SVR and LR predictors.

V. Results

In this section we present the detailed evaluation of the estimation methods presented in the previous section. We implemented these methods on an embedded computer (representing a typical edge computing device) and compared them in terms of accuracy, computational overhead and energy consumption. Computational overhead analysis is crucial since embedded devices offer limited CPU power and memory availability. Also, energy efficiency of embedded algorithms is important due to pervasiveness of embedded edge devices in IoT and scalability concerns.

While other works present Least Square Support Vector Machines (LS-SVM) [16] [17] and Neural Networks [18] [13] as accurate demand prediction algorithms, our initial analysis of runtime and memory requirements of these algorithms showed that they demand large datasets and abundance of processing power to achieve such accuracies. We therefore excluded these methods from our analysis.

A. Methodology

We implement our algorithms on a Raspberry Pi 3 device as an embedded testbed. We choose this platform as it is available at low cost, widely used within similar works, and offers great software and community support [12]. We use real residential power consumption and solar generation traces, obtained from Pecan Street dataset [29]. This dataset contains power consumption and solar generation of multiple houses with 15 minute granularity. To assure the generalizability of our conclusions, we compute the averages of statistics over 47 different house (4 of which include solar generation data) from Pecan Street power consumption time series to ensure that our selection covers sufficient variation across houses. Each house is represented by approximately 2 years of data. Since solar prediction is not the subject of our work, we add 14% noise to the real solar trace to simulate prediction with 14% error, which according to literature [30] is reasonable. We do not investigate the effect of solar prediction accuracy in this work, as it has been investigated by literature thoroughly [30], and instead focus on power consumption time series prediction and its impact on residential energy control

We measure the power prediction accuracy with normalized mean absolute error (NMAE). Equation (8) shows the error formula for predicting a vector $\mathbf{y}$ of length k (corresponding to power consumption time series) using the prediction vector $\hat{\mathbf{y}}$. This equation shows how to present the error as a percentage of the average of the predicted variable.

$$NMAE(\%) = \frac{\sqrt{\frac{\sum_{i=1}^{k} \left|\hat{y}(i) - y(i)\right|}{k}}}{average(\mathbf{y})} * 100 \tag{8}$$

In order to measure energy consumption of each method, we insert a 0.1Ω shunt resistor in the supply line and measure the voltage and current delivered to Raspberry Pi 3. We subtract the idle energy consumption from measurements to achieve the net energy draw of each method.

B. Model Accuracy

1) Information Theoretic Redundancy Analysis: As was mentioned before (Section III), prediction of power consumption over the course of day d is calculated using 24 models with horizons varying from 1 hour to 24 hours. In this subsection, we analyze the predictability of predicted sequence, $P(d)$ given $P(d-1)$ as predictor, using redundancy metric. We calculate redundancy by

$$Redundancy = \frac{I(P(d), P(d-1))}{H(P(d)) + H(P(d-1))}, \tag{9}$$

where I is the mutual information, and H is the entropy. Figure 2 shows the redundancy for different prediction horizons. In this figure, each vertical bar corresponds to a different house and the prediction horizon varies along the y-axis. Warmer colors show predictability, while blue corresponds to randomness. This figure allows us to gain an insight on how predictable power consumption of each house is given data from the day before. Ideally a perfect forecast can be obtained if the information theoretical redundancy is 1 and a value of 0 indicates no prediction is possible due to randomness. However, the redundancy values are very small for almost all of the houses that are available in this dataset, except for shortest horizon (which corresponds to persistence). Different houses show different predictability behavior, which will translate in a wide range of possible prediction errors. This also means that the expected prediction error is high. Our various experiments with different predictors confirm this analysis. However, we also demonstrate that every improvement to the prediction accuracy translates to considerable cost savings (Section V-C).

2) Accuracy vs. Training Set Length: First, we analyze the effect of training set length in the accuracy of the model. The training set consists of D days worth of power consumption data for 24 hours of the day, as well as environmental data for each day. Each trained model is tested for 30 days forecast, and the error is averaged over this period. Figure 3-(a) shows the effect of increasing training set length D on the accuracy of each model, which is characterized by NMAE (Equation

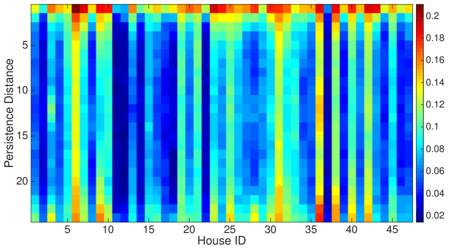


Fig. 2: Redundancy for different persistence horizons in different houses.

(8)). We observe that increased training length reduces error and improves accuracy of both models. However, using the same training length, SVR is able to achieve 21% less error on average compared to LR. In the highest accuracy case (30 day training length), SVR predicts with 14% less error. In the next subsection we show that increased accuracy translates into cost savings for the user.

C. Cost Savings

In this subsection we show the effect of model accuracy in providing cost savings for users by utilizing the optimal battery load flow algorithm with PV generation, discussed in Section IV. The predictions are input for ECO-DAC [4] which efficiently solves the load flow problem, and estimates the savings by comparing the projected usage pattern to the actual power consumption time series during that day. Solar prediction accuracy is fixed at 14% (see section V-A) as our goal here is to explore prediction accuracy of power consumption time series and its impact on residential energy management.

1) Cost Reduction vs Battery Capacity: This section investigates the effect of battery capacity on energy savings for users, averaged over all participating houses with solar generation over 31 days. Figure 3-(b) shows the cost reduction for LR and SVR prediction algorithms, compared to the hypothetical case were we have can forecast with 100% accuracy (Perfect Prediction) and the case with prediction based on 24-hour persistence (predicting $p_h(d,t) = p_h(d-1,t)$ for all t). We cannot use shorter persistence because we require the data 24 hours in advance to solve the optimal load flow problem. We observe that although low predictability of power time resulted in high error, both LR and SVR models still provide a large cost saving compared to persistence.

2) Cost Reduction vs Accuracy: Since we are using a time series consisting of predicted power consumption values to schedule optimal battery load flow, we should analyze the relationship between prediction accuracy and cost savings. For this experiment, we fix the battery capacity to 20 kWh, and average the cost savings over one month period. Figure 3-(c) shows the cost reduction as a function of prediction accuracy for both SVR and LR models. We also mark the best fitted linear line over the data to emphasize the general trend. It is evident that as the model becomes more accurate (less error) the cost saving increases, and vice versa.

D. Algorithm Overhead Analysis

One of the crucial aspects of programs running on embedded edge devices is their computational overhead. This section analyzes the overhead of different prediction models by comparing their runtime on our embedded testbed (see V-A).

1) Training Runtime vs. Training Set length: It is expected that model training time should increase as the training set length increases. Figure 3-(d) proves this behavior on our testbed. The y-axis in this figure is plotted in logarithmic scale. It is clear that training time for SVR in almost all cases is larger than LR, and it increases with a higher rate as the training time rises. However, as figure 3-(a) shows, prediction error decrease rate is lower for models with training length more than 15 days. This demonstrates that using larger datasets may slightly reduce prediction error. But this reduction substantially increases the computational overhead.

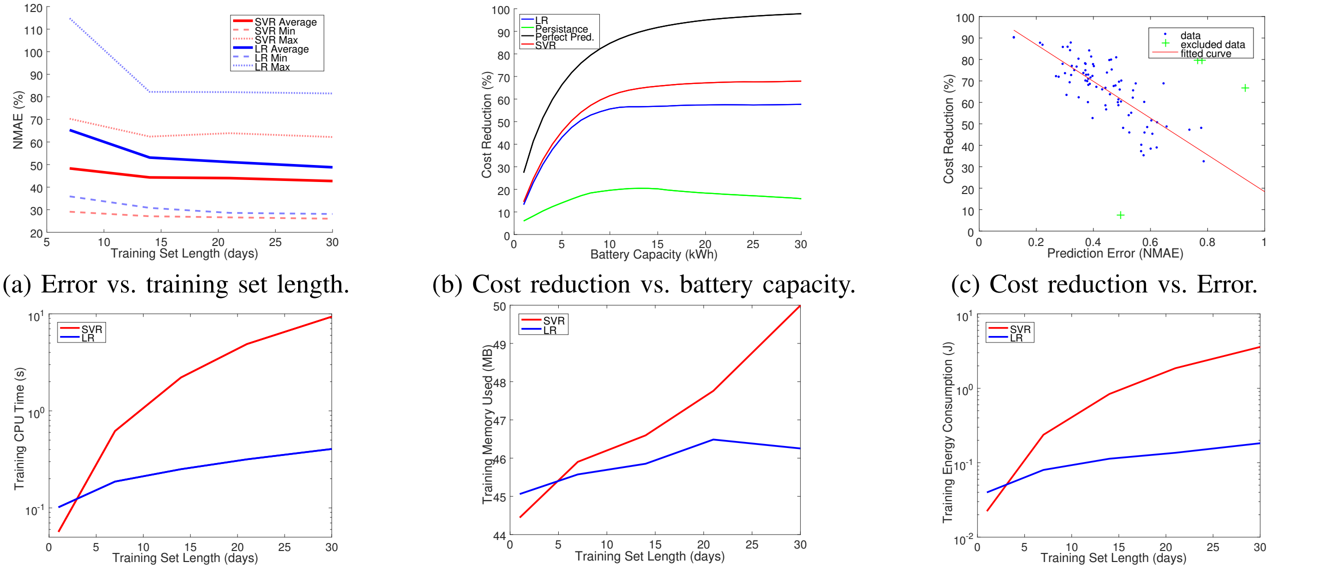
2) Memory Requirement vs. Training Set length: IoT applications rely on several embedded computers with small available memory. Since prediction performance depends highly on training dataset (consisting of historical data), memory size of a device can quickly become a bottleneck for an IoT application. We analyze memory requirement of the training stage of both SVR and LR models. We use a separate process to monitor the memory required by the training phase, and report the maximum observed value that is averaged over several experiments on different house data. Figure 3-(e) shows the maximum memory used during training as a function of training set length for both LR and SVR models. Memory requirement for small training sets is almost similar for both models. However, as the models get larger, SVR uses larger memory for the training phase, up to 50MB. This requirement can become a serious disadvantage for some edge devices, since they may have limited physical memory or multiple applications competing for the available memory space.

E. Energy Consumption

Energy is another important constraint of embedded edge devices, which have limited power budget, or have to rely on batteries during outages. To address this issue, we measured the energy consumption of the training phases of both SVR and LR models. The results are presented in Figure 3-(f), which shows that apart from the small and low accuracy cases, SVR always requires more energy on average per training forecast model. The gap between the energy requirement of SVR and LR gets bigger as the training set grows. This observation raises concerns about SVR's scalability for using bigger datasets on edge devices. However, within the analyzed model sizes, all models demand an amount of energy that is well within the battery capacity of a battery-powered device. By comparing the energy consumption values to the capacity of a normal cell phone battery (i.e., $2000mAh$), we observe that even the largest analyzed SVR model (training set length 30 days) is trained using only 0.01% of the battery capacity.

VI. Conclusion

We have analyzed SVR and LR prediction models in terms of their accuracy, runtime, memory requirement, energy con-



(a) Error vs. training set length.

(b) Cost reduction vs. battery capacity.

(c) Cost reduction vs. Error.

(d) Training Time vs. training set length.

(e) Memory vs. training set length.

(f) Energy Draw vs. training set length.

Fig. 3

sumption, and their capability to provide cost saving through optimal load flow problem in households with batteries and solar generation installed. We showed that although the household power time series has a sporadic nature (which translates into high prediction error), we are able to use these models to forecast 24-hour power consumption. Our study reveals that SVR is able to achieve 21% less prediction error compared to LR, while LR shows better scalability characteristics for larger training sets. We demonstrate how this increased accuracy translates into 16% more cost savings for users through solving the optimal battery load flow problem. Our work proves that both models are suitable for IoT applications implemented on embedded edge devices. However, for larger scale problems LR showed better scalability characteristics compared to SVR, while SVR's predictions were more accurate.

Acknowledgment

This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA and in part by ARPA-E NODES project DE-FOA-0001289.

References

[1] C. Perera et al., "Context aware computing for the internet of things: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[2] H. Farhangi, "The path of the smart grid," *IEEE Power Energy Mag*, vol. 8, no. 1, pp. 18–28, January 2010.

[3] *Renewables 2015 global status report*, 2015. [Online]. Available: "http://www.ren21.net/"

[4] A. S. Akyurek et al., "Eco-dac energy control over divide and control," in *IEEE SmartGridComm*, Oct 2013, pp. 666–671.

[5] *Battery Energy Storage for Smart Grid Applications*, 2013. [Online]. Available: "http://eurobat.org/"

[6] J. Hoppmann et al., "The economic viability of battery storage for residential solar photovoltaic systemsA review and a simulation model," *Renewable and Sustainable Energy Reviews*, vol. 39, pp. 1101-1118, 2014.

[7] Y. Yan et al., "A survey on smart grid communication infrastructures: Motivations, requirements and challenges," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 5–20, First 2013.

[8] L. Hernandez et al., "A survey on electric power demand forecasting: Future trends in smart grids, microgrids and smart buildings," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1460–1495, Third 2014.

[9] V. Almeida et al., "Hierarchical time series forecast in electrical grids," in *ICISA 2016*. Springer, 2016, pp. 995–1005.

[10] A. Tascikaraoglu and B. M. Sanandaji, "Short-term residential electric load forecasting: A compressive spatio-temporal approach," *Energy and Buildings*, vol. 111, pp. 380–392, 2016.

[11] A. Smola and V. Vapnik, "Support vector regression machines," *Adv Neural Inf Process Syst*, vol. 9, pp. 155–161, 1997.

[12] M. Richardson and S. Wallace, *Getting started with raspberry PI*. "O'Reilly Media, Inc.", 2012.

[13] L.-G. Chen et al., "Group-based chaos genetic algorithm and non-linear ensemble of neural networks for short-term load forecasting," *IET GENER TRANSM DIS*, vol. 10, no. 6, pp. 1440–1447, 2016.

[14] G. Dudek, "Pattern-based local linear regression models for short-term load forecasting," *ELECTR POW SYST RES*, vol. 130, pp. 139–147, 2016.

[15] A. Kavousi-Fard et al., "A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting," *EXPERT SYST APPL*, vol. 41, no. 13, pp. 6047 – 6056, 2014.

[16] F. Mateo et al., "Machine learning techniques for short-term electric power demand prediction." in *ESANN*, 2013.

[17] R. E. Edwards et al., "Predicting future hourly residential electrical consumption: A machine learning case study," *Energy and Buildings*, vol. 49, pp. 591–603, 2012.

[18] A. Songpu et al., "Domestic demand predictions considering influence of external environmental parameters," in *IEEE INDIN*, July 2015, pp. 640–644.

[19] T. Logenthiran et al., "Demand side management in smart grid using heuristic optimization," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1244–1252, Sept 2012.

[20] Y. Wang et al., "Adaptive control for energy storage systems in households with photovoltaic modules," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 992–1001, 2014.

[21] T. Hossa et al., "The comparison of medium-term energy demand forecasting methods for the need of microgrid management," in *IEEE SmartGridComm*. IEEE, 2014, pp. 590–595.

[22] T. Hansen and C.-J. Wang, "Support vector based battery state of charge estimator," *J. Power Sources*, vol. 141, no. 2, pp. 351 – 358, 2005.

[23] Y. Bao et al., "Multi-step-ahead time series prediction using multiple-output support vector regression," *Neurocomputing*, vol. 129, pp. 482 – 493, 2014.

[24] N. Bajaj et al., "Reduction of memory footprint and computation time for embedded support vector machine (svm) by kernel expansion and consolidation," in *IEEE MLSP*, Sept 2014, pp. 1–6.

[25] C.-J. Hsieh et al., "A divide-and-conquer solver for kernel support vector machines." in *ICML*, 2014, pp. 566–574.

[26] K. Z. Haigh et al., "Machine learning for embedded systems: A case study," Technical Report BBN REPORT 8571, BBN Technologies, MA, 2015.

[27] K. Worthmann et al., "Distributed and decentralized control of residential energy systems incorporating battery storage," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1914–1923, July 2015.

[28] *San Diego Gas and Electric*, San Diego Gas and Electric, 2016. [Online]. Available: http://www.sdge.com/

[29] *Pecan Street Dataport*, Pecan Street Inc., 2016. [Online]. Available: http://www.pecanstreet.org/

[30] A. Yona et al., "Application of neural network to one-day-ahead 24 hours generating power forecasting for photovoltaic system," in *ISAP, 2007*, Nov 2007, pp. 1–6.