A Supervised Machine Learning Approach to Control Energy Storage Devices

Gonzague Henri, *Student Member,* Ning Lu, *Senior Member,*

***Abstract*—This paper introduces a supervised machine learning approach to predict and schedule the real-time operation mode of the next operation interval for residential PV/Battery systems controlled by mode-based controllers. The performance of the mode-based economic model-predictive control (EMPC) approach is used as the benchmark. The residential load and PV data used in the paper are 1-minute data downloaded from the the Pecan Street Project website. The optimal operation mode for each control interval is first derived from the historical data used as the training set. Then, four machine learning algorithms (i.e. neural network, support vector machine, logistic regression, and random forest algorithms) are applied. We compared the performance of the four algorithms when using different number of features and length of the training sets extracted from different months of the year. Simulation results show that using the machine learning approach can effectively improve the performance of the mode-based control system and reduce the computation effort of local controllers because the training can be completed on a cloud-based Machine Learning engine. The work presented in this paper paves the way for using a shared-learning platform to design controllers of residential PV/storage systems. This may significantly reduce the cost for implementing such systems.**

***Keywords*—Energy storage; Machine learning; economic model predictive control; mode-based scheduling; discrete control**

NOMENCLATURE

$accuracy$	Percentage of modes predicted correctly;
B_{export}	Price to export electricity to the grid (\$/kWh);
B_{import}	Price to purchase electricity from the grid (\$/kWh);
B_{cycle}	Price of a battery cycle (\$/cycle);
C	Cost for the next time step (\$);
C_{base}	Yearly cost for the base case, without ESD(\$);
C_{optimal}	Optimal yearly cost when the ESD is used, EMPC with perfect forecast (\$);
C_{EMPC}	Yearly cost when using EMPC with perfect load forecast (\$);
C_{simu}	Yearly cost of a simulation case (\$);
Δt	Scheduling time interval for the mode-based operation (minute);
ΔT	Scheduling time interval of the EMPC algorithm (hour);
E	Battery energy level (kWh);
$E_{\max}$	Maximum battery energy level (kWh);
$E_{\min}$	Minimum battery energy level (kWh);
E_{rated}	Rated battery energy storage capacity (kWh);
f	Feasible mode number;
i	the i^{th} hour;
j	the j^{th} house in the dataset;
K	Number of features in the training set;
$length$	Total number of time steps;
m	Mode number;
$M_{predicted}$	Mode predicted by an algorithm;
$M_{optimal}$	Optimal mode for the time step;
η	One way efficiency of the battery (%);
N_{cycles}	Number of cycles;
$N_{feature}$	Total number of feature;
$\underline{P_B}$	Minute-by-minute battery power (kW);
$\overline{P_B}$	Hourly battery power (kW);
$P_B\|_m$	Battery power output at mode m (kW);
$P_B\|_f$	Battery power output at feasible mode f (kW);
P_{CCap}	Charging power cap (kW);
P_{charge}	Battery charging power (kW);
P_{charge}^{max}	Maximum battery charging power (kW);
P_{DCap}	Discharging power cap (kW);
$P_{discharge}$	Battery discharging power (kW);
$P_{discharge}^{max}$	Maximum battery discharging power (kW);
P_{export}	Power backfed to the grid (kW);
P_{export}^{max}	Maximum power backfed to the grid (kW);
P_{import}	Power supplied by the grid (kW);
P_{import}^{max}	Maximum power supplied by the grid (kW);
P_{load}	Household load (kW);
P_{net}	Net load (kW);
P_{rated}	Rated battery power (kW);
P_{sol}	Solar power output (kW);
P_{sol}^{f}	Solar power forecasted for next time step (kW);
$PMSA$	Percentage of the maximum achievable savings (%)
SOC	State of charge (SOC) of the battery;
$temperature$	Temperature data;
X_{test}	Input data for the testing part;
X_{train}	Input training data;
y_{train}	Optimal output corresponding to the

This work is supported by Total Inc.

Gonzague Henri and Ning Lu are with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, USA, 27695 USA. G. Henri is also with the Energy Management System department, Total, Paris, France. E-mails: gahenri@ncsu.edu; nlu2@ncsu.edu.

training data;

I. Introduction

ENERGY storage is a suitable technical solution for reducing the intermittency and uncertainty in power grid operations when a large amount of variable generation resources are integrated. In high PV penetration distribution systems, installing energy storage can smooth the output of PV power, store excess generation for future use, and provide real and reactive support to the grid operation [1]. In areas with high retail electricity prices and low feed-in-tariff, energy storage is becoming an economical option for residential PV installations [2]. Recently, Germany has lowered the feed-in-tariff and Hawaii no longer allows backfeeding for newly installed PV systems [3]. As a result, using energy storage devices (ESDs) to store excess solar power is an increasingly attractive option in those regions for residential and commercial PV customers.

The main ESDs used for residential PV applications are Lithium-ion and Lead-acid batteries [4]. Algorithms used to control battery storage in residential PV applications fall into two categories: optimization-based and rule-based. Optimization based algorithms include mixed-integer-programming (MIP) [5], model-predictive-control (MPC) [6], and dynamic programming methods [7]. Using the optimization-based approach usually yields the best performance when load and PV forecast are accurate. For realistic residential load and PV data sets, day-ahead load forecast error is approximately 20% [8], [9]. The forecast accuracy of PV power outputs depends on the day type. In a cloudy day, the forecast error can be over 50% [10], [11]. In this error range, the optimality of the schedule obtained by the MIP methods will no longer hold. To cope with the forecasting errors, multi-stage algorithms (e.g. [6], [12]–[14]) are proposed. First, an optimal schedule is developed based on 24-hour load and PV forecasts. Then, the schedule is adjusted every hour or every few minutes based on the updated forecast values.

However, those algorithms are also computationally intensive. Solvers for complicated optimization-based algorithms are usually not available on microprocessors, on which controllers are implemented. The rule-based control algorithms (e.g. [15]) are simple to implement, but the optimality of the solution is not guaranteed. In addition, because the rules of operation are customized to fit for different operation conditions, periodical updates may be required to performance avoid deterioration.

Table I: Simplified modes of the ESD controller

Idle	0	Idle
Charge	1	Charge by the net load, $\|P_{\mathrm{net}}\|$, if $P_{\mathrm{net}} < 0$
	2	Charge by the rated power, P_{rated}
Discharge	3	Discharge following the net load, $\|P_{\mathrm{net}}\|$, if $P_{\mathrm{net}} > 0$
	4	Discharge by the rated power, P_{rated}

Reinforcement learning (RL) has been previously used for demand response applications [16], Ruelens et al. use reinforcement learning for a model-free demand response application. In [17], a RL algorithm is used to learn the HVAC energy consumption model. RL has also been used to increase the performance of energy storage control [18], [19]. Wang et al. [18] use RL to compute the value of the residual energy in the battery at the end of each day. In [19], a Q-learning method solves the optimal battery management and control problem for residential applications.

RL can be seen as a problem where an agent tunes its policy to maximize its reward. However, the main drawback of RL is the need for a large number of samples to tune the policy. The policy tuning can occur in a simulator, but doing so in real-time with actual hardware might not be always possible. In some cases, RL is used to tune a parameter in the objective function of an optimization problem. In this setting, using RL does not help reduce the computational need to solve optimization problems because its target is to increase its performance.

This paper follows our previous work on the mode-based control approach for residential energy storage systems. In [**?**], [20], we introduced the mode-based control approach based on an Economic Model Predictive Control [21]. The main advantage of using the mode-based control mechanism is to select the control action from a finite set of modes in order to minimize the cost of operation. It is a two-stage algorithm, in the first stage the controller will select the best mode for the next hour based on day ahead forecasts and data. In the second stage, the battery operates in real-time in the mode selected. The results from our previous work demonstrated that the optimal actions of a battery controller can be limited to a small set (shown in Table I). This makes it possible to apply the machine learning approach for optimal mode selection.

In [22], we first introduced the possibility of using a supervised machine learning approach for operation mode selection. This paper focuses on using the machine learning approach to replace the EMPC component. By correlating each operation condition with its best operation mode, a machine learning algorithm can select the best operation mode without running an optimization-based algorithm, reducing the computational burden of the local controller.

We consider the main contribution of this paper to be *applying machine learning (ML) algorithms in real-time control mode selection,* which is the first step for developing intelligent controllers that can learn from its past operation experience and capture the relationship between operational conditions and optimal actions. In this paper, we compared four machine learning algorithms for mode selection considering different number of features and training lengths. Validation tests were conducted using 50 actual residential load and PV profiles collected in the Pecan Street Project [23]. The training sets were extracted from different months of the year. Simulation results show that the accuracy of the optimal mode selection outperforms the EMPC based approach and the dependency for an accurate load forecaster is significantly reduced.

The rest of the paper is organized as follows. The mode-based and EMPC approached are presented in Section II. Section III introduces the machine learning based mode selection process and the proposed algorithms. The simulation setups and results are discussed in Section IV. The conclusions and future work are summarized in Section V.

II. Problem Formulations of the Mode-based Control

In this paper, the application we select for illustrating how the mode-based approach works is: *minimizing the customers utility bill under the time-of-use rate*. This application is selected because it is the most commonly used application in practice for residential systems. In addition, because actual load data and utility rates can be used in case studies, meaningful performance comparisons can be made with existing methods (e.g. MIP-based algorithms) to benchmark the performance of the mode-based algorithm.

A. Battery Real-time Operation Modes

In this paper, we use five real-time operations modes, as shown in Table I. The modes will operate the battery in real-time while a mode selection process will schedule a mode for each time step, or each hour in this paper. The battery power output in each mode is calculated as follows.

First, calculate the net load, P_{net}], where P_{load} represents the load and P_{sol} the solar energy generated.

$$P_{\text{net}}(i)=P_{\text{load}}(i) - P_{\text{sol}}(i) \tag{1}$$

Then, calculate the battery charging power cap, P_{CCap}, for time interval i based on the current battery energy level, $E_B(i)$, and the battery energy limit, $E_{\max}$,

$$P_{\text{CCap}}(i)=(E_{\max} - E_B(i))/\Delta(t) \tag{2}$$

For the battery discharging power cap, P_{DCap}, for time interval i based on the current battery energy level, $E_B(i)$, and the battery energy limit, $E_{\min}$,

$$P_{\text{DCap}}(i)=(E_B(i) - E_{\min})/\Delta(t) \tag{3}$$

The power charging cap shows that, considering how much more energy can be stored in the battery, what the maximum charging power of the battery is for the i^{th} interval. Then, the battery charging/discharging power at the i^{th} time interval, $P_B(i)$ at mode m can be formulated as

$$P_B(i)|_{m=0}=0 \tag{4}$$
$$P_B(i)|_{m=1}=\max\Big(0, \mathrm{min(P_{CCap}(i), -P_{net}(i), P_{rated})}\Big) \tag{5}$$
$$P_B(i)|_{m=2}=\max\Big(0, \mathrm{min(P_{rated}, P_{CCap}(i))}\Big) \tag{6}$$
$$P_B(i)|_{m=3}=-\max\Big(0, \mathrm{min(P_{DCap}(i), P_{net}(i), P_{rated})}\Big) \tag{7}$$
$$P_B(i)|_{m=4}=-\max\Big(0, \mathrm{min(P_{rated}, P_{DCap}(i))}\Big) \tag{8}$$
$$i \in [1, N] \tag{9}$$

B. Control Logic of the Idling and Charging Modes

Because the main goal of charging batteries in residential PV applications is storing excess PV power or charging at low price periods for future use, we designed two charging modes to meet those control objectives. First, let the battery power output, P_B, be positive when charging and let the net load, P_{net}, be positive if the load exceeds the solar generation. Second, let the mode selection happen at the beginning of each hour and let the minimum operation period for each mode be an hour. During the i^{th} hour, the mode controller will adjust battery power, $P_B(j)$, every minute. Thus, $\Delta t = 1/60$ hour, $j = 1...60$, and $i = 1...24$.

Based on those assumptions, at the j^{th} time interval, the netload, $P_{\text{net}}(j)$, is calculated as

$$P_{\text{net}}(j)=P_{\text{load}}(j) - P_{\text{sol}}(j) \tag{10}$$

Based on the battery energy level, $E_B(j)$, and the battery energy limit, $E_{\max}$, the battery charging power cap, $P_{\text{CCap}}(j)$, is calucated as

$$P_{\text{CCap}}(j)=\Big(E_{\max} - E_B(j)\Big)/\Delta t \tag{11}$$

Then, the battery charging power of mode m at the j^{th} time interval, $P_B(j)|_m$ is calculated as

$$P_B(j)|_1=\max\Bigg(0, \min\Big(P_{\text{CCap}}(j), -P_{\text{net}}(j), P_{\text{rated}}\Big)\Bigg) \tag{12}$$

$$P_B(j)|_2=\max\Bigg(0, \min\Big(P_{\text{rated}}, P_{\text{CCap}}(j)\Big)\Bigg) \tag{13}$$

$$\tag{14}$$

In the "idling" mode, the battery output is simple zero, so we have,

$$P_B(j)|_0=0 \tag{15}$$

C. Control Logic of the Two Discharging Modes

Because the main goal of discharging ESDs in residential PV applications is supplying load at high price periods or using self-generated power, we designed four discharging modes to meet those control objectives. Similar to the charging modes, at the j^{th} interval of the i^{th} hour, we fist calculate the battery discharging power cap $P_{\text{DCap}}(j)$, as

$$P_{\text{DCap}}(j)=\Big(E_B(j) - E_{\min}\Big)/\Delta t \tag{16}$$

Then, the battery discharging power of discharging mode m at the i^{th} time interval, $P_B(j)|_m$, is calculated as

$$P_B(j)|_3=-\max\Bigg(0, \min\Big(P_{\text{DCap}}(j), P_{\text{net}}(j), P_{\text{rated}}\Big)\Bigg) \tag{17}$$

$$\mathrm{P}_B(j)|_4=-\max\Bigg(0, \min\Big(P_{\text{rated}}, P_{\text{DCap}}(j)\Big)\Bigg) \tag{18}$$

$$\tag{19}$$

Based on (12)-(19), battery manufacturers can implement the build-in modes at the battery controller level [24]. An external controller, such as a home energy management system controller, can simply control the battery system by letting it operate at one of the five modes. This can greatly simplify the interface between the external controller and the battery systems, making the battery system plug-and-play with guaranteed performance.

D. EMPC-Based Mode Selection

In [21], the authors proposed the EMPC method to determine the setpoints for the ESD controller instead of minimizing the deviations from the setpoints sent to the ESD controller, the EMPC uses a cost function as its objective function. In our approach, we adapted the EMPC approach to determine the optimal hourly setpoint for the battery power over the next 24-hour period. The objective function of the EMPC problem is to minimize the 24-hour cost considering the cost of import and export energy as well as the cost for battery degradation, so we have

$$z(i)=\min\sum_{i=1}^{24}\Big(B_{\text{export}}(i)P_{\text{export}}(i)\Delta T+ B_{\text{import}}(i)P_{\text{import}}(i)\Delta T+N_{\text{cycles}}(i)B_{\text{cycle}}\Big) \tag{20}$$

s.t.

$$P_{\text{import}}(i)-P_{\text{export}}(i)-P_{\text{charge}}(t)+P_{\text{discharge}}(t) = P_{\text{load}}(i)-P_{\text{sol}}(i) \tag{21}$$

$$E_{\text{min}} \leq E_B(i) \leq E_{\text{max}} \leq E_{\text{rated}} \tag{22}$$

$$0 \leq P_{\text{charge}}(i) \leq P^{\text{max}}_{\text{charge}} \tag{23}$$

$$0 \leq P_{\text{discharge}}(i) \leq P^{\text{max}}_{\text{discharge}} \tag{24}$$

$$0 \leq P_{\text{import}}(i) \leq P^{\text{max}}_{\text{import}} \tag{25}$$

$$0 \leq P_{\text{export}}(i) \leq P^{\text{max}}_{\text{export}} \tag{26}$$

$$N_{\text{cycles(i)}} = \frac{|E_{\text{B}}(i)-E_{\text{B}}(i-1)|}{2E_{rated}} \tag{27}$$

$$E_B(i) = E_B(i-1)+\eta P_{\text{charge}}(i)\Delta T - \frac{P_{\text{discharge}}(i)\Delta T}{\eta} \tag{28}$$

$$P_{\text{import}}, P_{\text{export}}, N_{\text{cycles}}, P_{\text{charge}}, P_{\text{discharge}} \geq 0$$

$$i \in [1, ..., 24]$$

Equation (20) represents the objective function, the goal is to minimize the cost of electricity for the next 24 hours considering the cost of the battery degradation. The energy balance constraint is represented in (21). The constraint on the battery energy is represented in (13), while (14)-(15) represent the constraints on the battery power. Equations (16)-(17) are used to set a limit to the power that can be imported or exporter from/to the grid. Equation (19) show how the battery energy quantity is updated.

Note that because the EMPC calculates hourly schedules, $\Delta T = 1$ can be omitted from the problem formulation. We also used a simplified method to calculate the number of battery cycles in (27). More sophisticated methods for estimating effective battery cycles to account for degradation can be used, but because our focus is to formulate the mode-based control algorithm, we chose to use a simplified battery cycle calculation as an illustration for how to account for battery degradation.

From solving (20), we obtain the battery power for the next time step, if the the optimal operation is to charge, we will run the two charging mode for the next time-step, from this calculation we obtain the battery updated variables and the cost for the next time step. Then, (20) is ran over the 23 remaining hours. The 24 hours cost is obtained by adding the cost associated with the time step using the mode and the cost from the corresponding 23 hours optimization. The mode associated with the lowest 24 hours cost is selected. The same process is used for discharging modes.

In the next section, we are going to introduce the machine learning method for predicting the next operation mode.

III. Machine Learning Based Mode Selection

In this section we will first describe the process of selecting training data sets and features. Then, the selection of the best performing architecture for Neural Networks (NN) and Random Forests (RF) methods is discussed. Lastly, we will present the machine learning based mode selection algorithm.

A. Why Machine Learning?

The problem selected, minimizing the electricity bill under a TOU tariff in a household with PV and an energy storage system, is based on rational decisions. The controller select the mode of the ESD in order to minimize the bill. In Fig. 1, the count of each mode for each hour of the day is presented for every Monday of the same year. Mode 1 represents charging from solar, mode 3 charging at full power, mode 5 discharging following the load and mode 7 discharging at full power. A darker green indicates a larger number of count, while a blue color indicates that this specific mode has never been selected at this hour. From this figure, it can be observed that the mode selection follows a pattern, more precisely it tends to charge during the day, when there is excess of solar energy, and discharge in the evening or later.

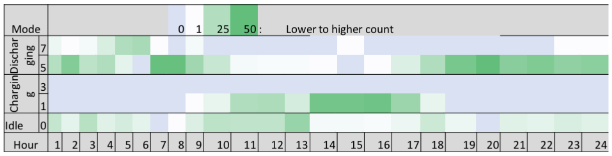

Fig. 1: Count of the mode selected at each hour of day for every Monday of a year

B. Selection of Training Data Sets and Features

Data from the Pecan Street Project [23] have been used. We selected 50 houses with PV installations located in Austin, Texas, which have 8760 hourly points that were collected in 2015. Using the actual load and PV data (i.e. assuming that the forecast is perfect), we run the EMPC-based mode selection algorithm (refer to [20] for details) for selecting the best operation modes, $m^*(i,j)$ (i.e. for hour i at house j). This process generates the optimal modes at each of the 8760 hours for all 50 houses.

Those data will then be used to train the ML algorithms. For dispatch interval i, we also ran the EMPC-based mode selection algorithm using an average load forecaster. Note that this is the algorithm with the best performance as we demonstrated in [20]. Note that for the average load forecast algorithm, the yearly average load of the household is considered as the constant value for the forecaster. Thus, two sets of modes are obtained: modes selected using a perfect load forecaster and modes selected using an average load forecaster. The perfect forecast mode vector will be used for training and to calculate the mode selection accuracy, the mode vector based on the average load forecast will be used as a benchmark to demonstrate the improvement of the ML approach against the mode-based EMPC based approach.

At each time step, we record the following 14 features: 1) the state of charge (SOC); 2) the PV output at the last time step; $P_{\mathrm{sol}}(i-1)$; 3) the load at the last time step; $P_{\mathrm{load}}(i-1)$; 4) the PV forecast at the current time step; $P^{\mathrm{f}}_{\mathrm{sol}}(i)$; 5) the remaining energy storage capacity to charge; $E_{\max}-E(i)$; 6) the remaining energy storage capacity to discharge; $E(i)-E_{\min}$; 7) the sum of the forecasted PV for the next 24 hours; $\sum_{i=1}^{24} P_{\mathrm{sol}}(i)$; 8) temperature; 9) hour; 10) day of the week; 11) month; 12) day of the month; 13) weekday/weekend; and 14) the prices to import electricity, B_{import}.

As a first step, we want to identify which features among the 14 dominate the mode selection process. To achieve this goal, we conducted multiple analysis of variance (ANOVA) tests [25] so that the K-best features are selected. Note that $K \in [1, N_{\mathrm{feature}}]$, where $N_{\mathrm{feature}} = 14$. We divided the 12 months data sets into the training and testing sets such that the training set contains 11 months of data and the testing set contains one month of data. Four of such data-sets are constructed for each house by letting the testing months be January, April, July, and November and the 11 months serve as the training sets.

In the second step, we want to determine the optimal number of features for achieving the highest accuracy in the mode prediction. To do so, we trained the ML algorithms with different numbers of the K features following the order of importance. Then, each algorithm will be tested on the same test sets to find which set of features of the algorithm can achieve the best performance.

In the third step, we want to quantify the impact of the length of the training set on the accuracy of mode selection. To do so, we increase the length of the training set from one to 11 months with an increment of one month at a time. The ML algorithms achieving the best overall performance will be selected as the algorithms to serve for the control simulation. Then, the ML algorithm accuracy will be tested for each mode in order to verify that the increase in accuracy is not just for any specific mode. The algorithms to be tested are support vector machines (SVM), NN, RF and logistic regression. According to [26] NN and SVM should have strong performance with a large dataset of continuous features. The selection of the best learning architecture is obtained after a sensitivity analysis is conducted on the feature scenarios, the training set length and ML algorithm architectures. The architectures and algorithms considered in this paper are listed in Table II. Neural networks of different size have been selected in order to determine with which number of neurons could the NN start to approximate the non-linearity of our objective function.

Using the methodology introduced in Section III-B, the algorithm with the best performance will be selected. The performance of the algorithms will be measured against the EMPC-base mode selection algorithm with an AL forecaster.

Table II: List of the Different Algorithm Used

Case	Algorithm	Architecture	Name
1	NN	1 hidden layer/10 neurons	clf_nn_101
2	NN	2 hidden layer/10 neurons each	clf_nn_102
3	NN	1 hidden layer/20 neurons	clf_nn_201
4	NN	1 hidden layer/5 neurons	clf_nn_51
5	NN	2 hidden layer/5 neurons each	clf_nn_52
6	SVM	-	clf_svm
7	LR	-	clf_logreg
8	RF	100 estimators	clf_randomforest100
9	RF	250 estimators	clf_randomforest250
10	RF	500 estimators	clf_randomforest500
11	RF	1000 estimators	clf_randomforest1000
12	EMPC	AL forecast	y_al_solution

C. Machine Learning based Mode Selection Algorithm

The algorithm is divided into two parts. The first part includes an offline mechanism for training the supervised learning model. The second part focuses on the control of the ESD. The supervised learning model will predict the mode that will be used in the next time step.

Step 1: Data Acquisition and Offline Learning

In practice, a mode-based control algorithm with average load forecast can be used to operate the system until a sufficient amount of data is recorded to train the supervised learning model. The inputs at this stage are the electricity prices, the temperature, the load, and the PV production. This task will last until enough data has been acquired to perform the offline training. If the customer has enough historical data to train the ML based algorithm, the offline training can start immediately.

Once the historical data is made available, a mode-based algorithm with a perfect forecaster will be run to obtain the optimal mode and battery features (i.e. SOC, $E_{\max}-E(i)$, $E(i)-E_{\min}$) and the forecaster features (i.e. P^{f}_{sol}, $\sum_{i=1}^{24} P_{\mathrm{sol}}(i)$) at each time step. This process is described in Algorithm 1.

After the optimal modes for each time step are obtained for the historical data set, an offline training of the ML algorithm will start. First, the training data will be standardized; then, the training will be performed on the standardized training set. The standardization parameters as well as the trained ML algorithm will be transmitted to the agent. The standardization parameters will be used on the testing set.

To summarize this first step, the following tasks are performed in sequence: 1) acquire sufficient historical data (mainly load, PV, and electricity prices); 2) use a mode-based control with perfect forecast and an equivalent battery model to generate the optimal mode for each time step, the forecasting based features, and the battery features; and 3) train the model with the historic data and generated features. After this step, the ML algorithm is ready to be used for control.

Algorithm 1 Data Recording and First Training

```
while $i <$ threshold for first training do
    Record data from previous time step
    Obtain the forecasted data: $P^{f}_{\text{sol}}$, $\sum_{i=1}^{24} P_{\text{sol}}(i)$, $B_{\text{import}}$, $B_{\text{export}}$, $temperature$
    Obtain the data from the battery: $SOC$, $E_{\max} - E(i)$, $E(i) - E_{\min}$
    Find mode for next time step based on EMPC+ average load forecast algorithm
    Use the mode to control the ESD in real-time
end while
start training process
for $i$ in number of time steps recorded do
    Use EMPC + perfect forecast to find the optimal mode for time step i
end for
Create a $X_{train}$ vector containing the $K \in [1, N_{\text{feature}}]$ features identified, where $N_{\text{feature}} = 14$
Create a $y_{train}$ vector with the optimal mode corresponding to each row of $X_{train}$
Initialize a ML model
Train the ML model using $X_{train}$ and $y_{train}$
```

Step 2: Control, Mode Selection and Online Algorithm

Once the training on the historical data is completed, the ML model will replace the mode-based control algorithm. The agent will perform the following three tasks: using the ML model to predict the mode for the next time step, operating the ESD based on the mode predicted, updating the model every few days or weeks. This process is illustrated in Algorithm 2.

To keep updating the ML model, the load, PV, date, and temperature will be recorded at each time step. Then, as described in the previous section, a mode-based control algorithm with a perfect forecaster and an equivalent battery model will be run on the data to find the optimal mode at each time step, and populate the training sets. Once this training set is created, the ML algorithm can be updated.

For real-time control, the agent will generate at each time step a X_{test} vector of a length equal to the number of features corresponding to the data needed to predict the mode for the next time step. To populate this vector, the agent will need data from the smart meter, the ESD, and different APIs (e.g. temperature, PV forecast, and electricity prices). This vector will be sent to the ML algorithm in order to predict the mode for the next time step. Then, the mode is sent to the ESD for the next time step operation. This set of actions will be repeated at each time step.

Algorithm 2 Machine Learning Based Control and Online Training

```
Obtain the forecasted data: $P^{f}_{\text{sol}}$, $\sum_{i=1}^{24} P_{\text{sol}}(i)$, $C_{\text{import}}$, $temperature$
Obtain the data from the battery: $SOC$, $E_{\max} - E(i)$, $E(i) - E_{\min}$
create a $X_{test}$ vector containing the $K \in [1, N_{\text{feature}}]$ features identified, where $N_{\text{feature}} = 14$.
Use the train ML-model to predict the mode for the next time step real-time operation
Send the predicted mode to the ESD for the next time step real-time operation
if Amount of new data recorded $>$ New training Threshold then
    Run EMPC mode based algorithm + perfect forecast on new historical data to derive optimal mode for each time step
    generate $X_{train}$ and $y_{train}$ for online training
    Use this new $X_{train}$ and $y_{train}$ to update the ML-algorithm training
end if
```

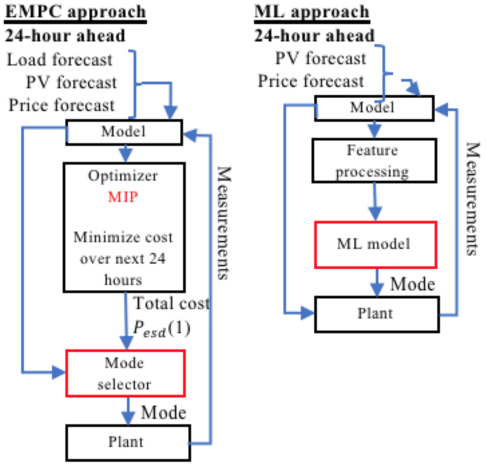

Fig. 2: Mode-based control algorithms, on the left the EMPC approach, on the right the machine learning approach

In Fig. 2 we compare the real-time process of the EMPC-based and ML-based approaches for the mode prediction and the mode-based control. In the EMPC approach, we need to run multiple optimization in the optimizer and mode selector blocks. In the ML approach, the ML algorithm replaces the optimization and mode selector blocks. The ML approach requires a training process to prepare the K features. Then, the features can be used for predicting the best operation modes for controlling the ESD. The biggest advantage of the ML approach is that it does not require load forecast.

IV. Simulation Results

The load data used in the simulation is from the Pecan Street dataset [23] as discussed in Section III-B. The characteristics of the houses are summarized in Table III. The electricity tariff is from HECO in Hawaii, as described in Table IV. This tariff encourages customers to consume their solar generation and does not valorize backfeed to the grid. The ESD used is assumed to be the same for all houses. The ESD is a 7kWh@3.3kW battery with a round-trip efficiency of 90%. We select January, April, July and November to the the testing month, which correspondingly represents winter, spring, summer and fall.

Table III: Statistical description of 50 houses selected

	Annual Payment ($)	Annual Load (kWh)	Annual PV Generation (kWh)	Ratio PV/load
Mean	1,660	11,407	6,608	0.64
Standard deviation	676	3,990	1,505	0.22
Maximum	3,426	22,916	11,102	1.28
Minimum	630	4,921	3,882	0.27
Median	1,555	10,726	6,713	0.64

Table IV: Time-of-use Rate in HECO (Hawaii Utility)

	Price (cent/kWh)	Hour weekday	Hour weekend
Off peak	18.21	9 PM -7 AM	9 PM - 5 PM
Shoulder	23.71	7 AM - 5 PM	5 PM - 9 PM
Peak	26.71	5 PM - 9 PM	-

Two metrics are used to evaluate the algorithm performance. The first metric is the $accuracy$ of the mode selection. $length$ represents the test duration. In this paper, length is 30 days (i.e. 720 hours). $M_{predicted}$ represents the modes predicted by the ML algorithm and $M_{optimal}$ represents the optimal modes.

$$accuracy = \frac{\sum_{i=1}^{length} M_{predicted}}{length} \tag{29}$$

At each time step i, we have

$$M_{predicted}(i) = \begin{cases} 1, & \text{if } M_{predicted}(i) = M_{optimal}(i) \\ 0, & \text{otherwise} \end{cases} \tag{30}$$

The second metric is the percentage of the maximum savings achieved for each house, $PMSA$, which is calculated as

$$PMSA = \frac{C_{base} - C_{simu}}{C_{base} - C_{optimal}} \tag{31}$$

C_{base} represents the base cost without using ESDs; $C_{optimal}$ is the optimal cost when using ESDs calculated by the EMPC algorithm assuming that the forecast of PV and load is perfect; C_{simu} is the calculated cost. Two simulation cases are studied: EMPC with the average load forecaster and the ML-based algorithm.

The ML models are simulated using the Scikit-Learn library [27] in Python; the optimization problem for the EMPC-based algorithm is formulated using the Pyomo library [28] and solved by GLPK. To train the NN we use the Adam algorithm [29]. Adam implements a gradient descent algorithm with an adaptive learning rate. This method can be used for both offline and online training. It is particularly effective for large datasets.

A. Feature Selection

The ANOVA test has been run on each house. If a feature is among the K best in one house, we consider it is one count. As obtained in [22], the ANOVA test is performed on 149 houses and with K being comprised between one and 14. We considered the same ranking of the features going from 149 houses to 50 houses in this paper. Table V shows the ranking of the feature from the most important (price to import electricity) to the least (day of the month) based on the ANOVA test.

Table V: Rank of Best Feature Selected, Based on ANOVA Test

Rank	Feature
1	Price import
2	Next hour PV production
3	Capacity to Charge
4	Previous hour PV production
5	Day of the week
6	State of Charge
7	Hour
8	Previous Load
9	Temperature
10	Sum of the PV forecast for the next 24 hours
11	Capacity to Discharge
12	Month
13	Weekday or weekend
14	Day of the month

Based on the ranking of each feature, we then run different combinations of features for each algorithm and compare the accuracy of mode selection. From Fig. 3, we can make the following observations: 1) NN and RF seem to have the best accuracy across the four seasons; 2) RF seem to have strong performance with at least 10 features while NN seem to have better performances with 14; and 3) the best performance seem to be in the Fall while the worse performance seem to be in the Spring. The ML approach outperforms the EMPC approach in every scenario. In summer, we need to select at least seven features but in other seasons, we can select less than seven features.

B. Length of the Training Data

To analyze the sensitivity of the accuracy with respect to the length of the training data, we consider a case with 14 features. From Fig 4, we can observe that with more data in the training set, we can achieve better accuracy. We can also observe in the summer the accuracy plateau with at least 270 days in the training set, while not plateau in the winter and spring, and plateau at 300 days in the fall. However we can observe strong performance in the winter with only a few months of data, while requiring at 180 days in the summer. From this analysis, the optimal length of the training set for can be obtained.

From these different simulations, we can observe that NN (two layers of 10 neurons and one layer of 20 neurons) and RF vastly outperform SVM and logistic regression. Therefore, we select the NN with one layer of 20 neurons, two layers of ten neurons along with the random forests (RF) cases for the final performance comparison.

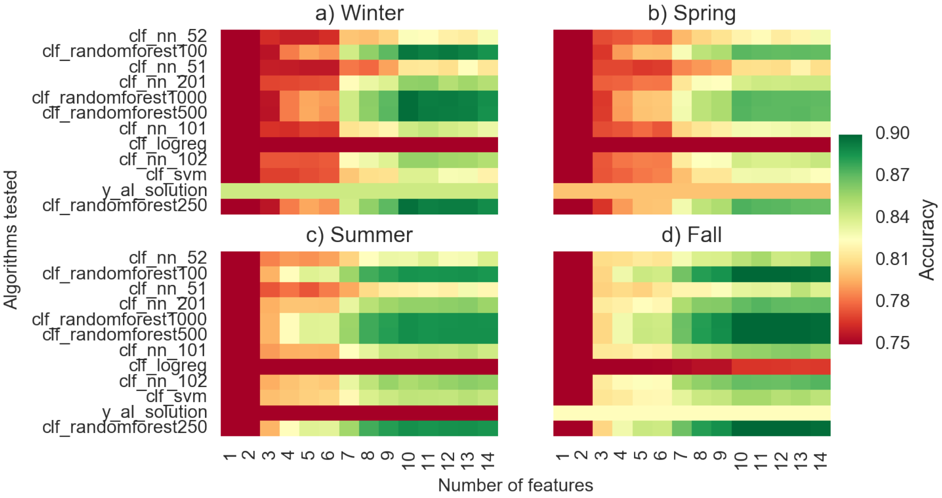

Fig. 3: Sensitivity analysis over the number of features and in different seasons

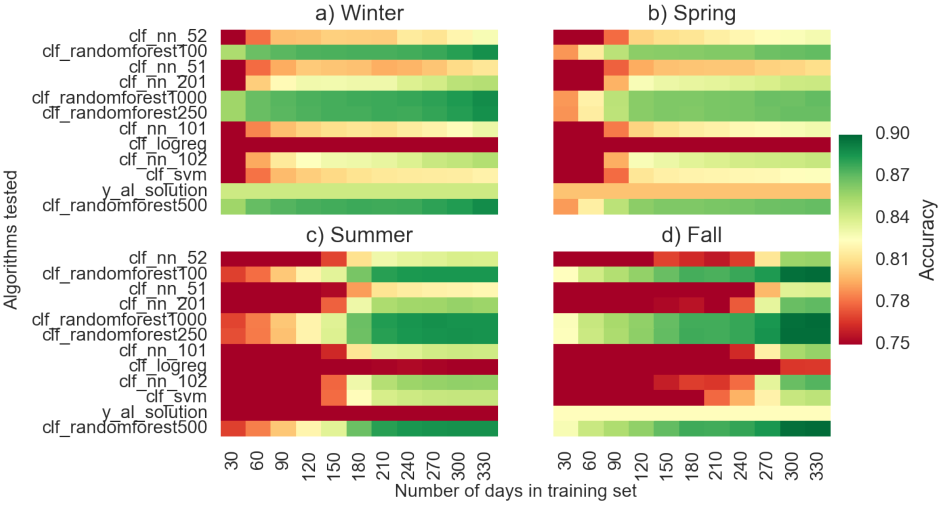

Fig. 4: Sensitivity analysis over the training length, in a case with 14 features, and in different seasons

C. Simulation

Table VI, we show the mean savings of the 50 houses selected along with their PMSA. We can observe that all ML algorithms, except the NN 102, perform better than EMPC with average load forecast. In Fig. 5, showing the distribution of PMSA in our dataset across the different seasons, we can see that RF-based algorithm performs better than NN while confirming the observation from Table VI. This figure also shows that the ML approach outperforms the MPC-based approach with five out of the six ML algorithms proposed. For the remaining part of this section we will focus on NN 201 and RF 1000 for a more detailed analysis.ALF is used for ”Average Load Forecast”, using the mean load of the year as a constant load forecaster, PF is used for ”Perfect Forecast”.

Table VI: Savings results for the different cases using mode based control

Case	Mean cost ($)	Cost standard deviation ($)	Mean PMSA (%)	PMSA standard deviation (%)
EMPC with PF	423	328	100	3.5
EMPC with ALF	450	332	83.6	8.0
NN 102	451	331	81.8	6.8
NN 201	447	332	84.5	6.3
RF 100	445	332	85.6	5.9
RF 250	444	332	86.1	5.8
RF 500	444	332	86.2	6.0
RF 1000	444	332	86.2	5.7

We plotted the PMSA function of PV/Load ratio (sum of the PV generation over the year divided by the sum of the

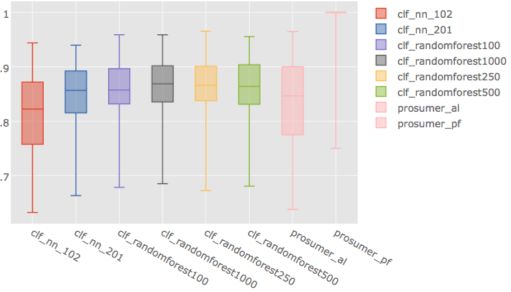

Fig. 5: Boxplot representing the PMSA for the 4 months tested

load over the year). From Fig. 6 we can observe that the three different algorithms studied have similar performance. We can also observe that the performance of the algorithm is consistent with different PV/Load ratios.

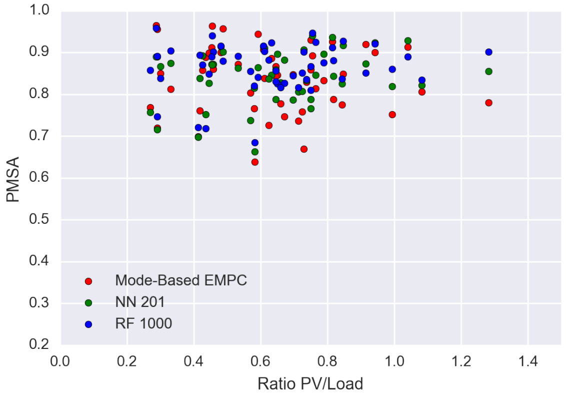

Fig. 6: Scatter plot representing the PMSA depending on the ratio PV/load, for the EMPC with ALF, the NN 201 and RF 1000

In Fig. 7 we show the sum of the cost for each season while in Fig. 8 we show the mean PMSA for each season. In both examples we compare four cases: the EMPC plus perfect forecast, the EMPC plus average load forecast, NN 201 and RF 1000. We can see that compared with the EMPC algorithm with an average forecaster, the machine learning algorithms' performance is slightly worse in winter and is better in summer, while in spring and fall, there is no obvious difference between the two algorithms. Because summer is the period with the highest energy consumption and solar generation, the ML algorithms will generate more savings compared with the EMPC algorithm. In summary, the machine learning based mode selection has consistent performance across different load and PV shapes.

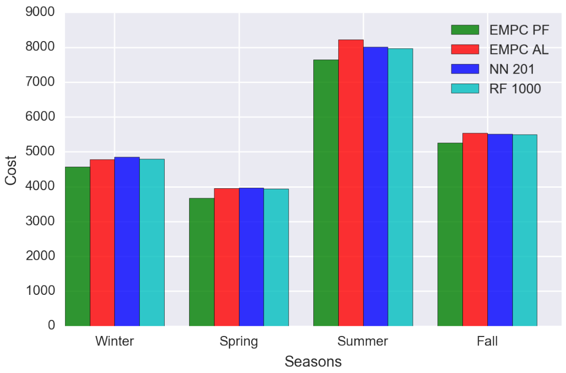

Fig. 7: Barchart representing the total cost of the 50 houses per season of test with three algorithms

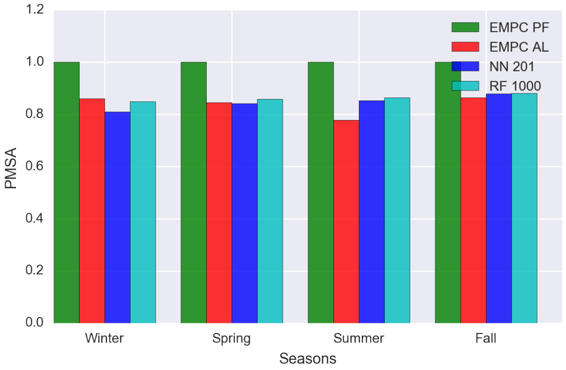

Fig. 8: Barchart representing the mean PMSA of the 50 houses per season of test with three algorithms

Overall, the following observations can be made regarding the machine learning approach performance:

- the logistic regression has the worst performance, as it is a linear classifier;
- the NN with 10 or 20 neurons in their layers outperform the ones with five, as they are better at capturing the non-linearity;
- the NN with 20 neurons in one layer as better simulation performance than the NN with 10 neurons in two layers;
- the RF models have similar performance whether the the number of classifiers is 250, 500 or 1000.

The Machine Learning for mode-based energy storage control has several limitations compared to an optimization-based approach. The first limitation is the need for historical data, as shown in Fig. 2 and Fig. 3, eight to nine months are required to achieve high accuracy. It might not be possible to always obtain this much data especially in places with high turnover of tenants (close to college campuses for example). A related limitation is the need for cloud computing infrastructure to train and maintain up-to-date algorithms. The optimization-based method does not need historical data nor a cloud infrastructure to operate. Another limitation would be to detect the houses for which the algorithm underperforms other control methods.

V. Conclusion

This paper builds on our previous work on mode-based control and presents a machine learning approach for real-time battery control to reduce the reliance on load forecasters. The ML-based algorithm does not use a load forecaster. Another advantage of the ML based approach is to reduce the computation needs in real-time operation; however, this approach would require a cloud operation to train the algorithms, which is a trade-off between embedded computation need and cloud based computation need. We tested the algorithm performance using actual data sets collected from 50 houses. The results show higher accuracy for predicting the operation mode. This higher accuracy translates in better overall performance, with a significant improvement during the summer period. We can also note that, like the EMPC with AL forecast, the ML algorithms have consistent performance across different load and PV patterns. In our future work, we will include a more accurate battery model, notably to improve the degradation model and to consider the non-linear pattern for charge and discharge operations. We will also expand this method to a shared-learning approach, with the goal of reducing the need of temporal data to achieve similar performances.

References

[1] P. Denholm and R. Margolis, “Energy Storage Requirements for Achieving 50 % Solar Photovoltaic Energy Penetration in California,” Tech. Rep. August, 2016.

[2] K. Ardani, E. O . Shaughnessy, R. Fu, C. Mcclurg, J. Huneycutt, and R. Margolis, “Installed Cost Benchmarks and Deployment Barriers for Residential Solar Photovoltaics with Energy Storage: Q1 2016,” Tech. Rep. February 2017, 2017. [Online]. Available: https://www.nrel.gov/docs/fy17osti/67474.pdf

[3] “PUC Decision and Order 32499,” 2014. [Online]. Available: https://www.hawaiianelectric.com/clean-energy-hawaii/producing-clean-energy/selling-power-to-the-utility/feed-in-tariff

[4] N. DiOrio, A. Dobos, and S. Janzou, “Economic Analysis Case Studies of Battery Energy Storage with SAM,” *National Renewable Energy Laboratory: Denver, CO, USA*, no. November, 2015.

[5] Zhi Wu, Xiao-Ping Zhang, J. Brandt, Su-Yang Zhou, and Jia-Ning Li, “Three Control Approaches for Optimized Energy Flow With Home Energy Management System,” *IEEE Power and Energy Technology Systems Journal*, vol. 2, no. 1, pp. 21–31, 2015.

[6] I. Lampropoulos, P. Garoufalis, P. P. Van den Bosch, and W. L. Kling, “Hierarchical predictive control scheme for distributed energy storage integrated with residential demand and photovoltaic generation,” *IET Generation, Transmission & Distribution*, vol. 9, no. September, pp. 1–9, 2015.

[7] L. Liu, Y. Zhou, Y. Liu, and S. Hu, “Dynamic programming based game theoretic algorithm for economical multi-user smart home scheduling,” *Midwest Symposium on Circuits and Systems*, pp. 362–365, 2014.

[8] B. Stephen, X. Tang, P. R. Harvey, S. Galloway, and K. I. Jennett, “Incorporating Practice Theory in Sub-Profile Models for Short Term Aggregated Residential Load Forecasting,” *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1591–1598, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7322269/

[9] Z. Genç, M. Oey, and F. Brazier, “Monitoring Stakeholder Behaviour for Adaptive Model Generation and Simulation: A Case Study in Residential Load Forecasting,” 2015.

[10] V. P. Lonij, V. T. Jayadevan, A. E. Brooks, J. J. Rodriguez, K. Koch, M. Leuthold, and A. D. Cronin, “Forecasts of PV power output using power measurements of 80 residential PV installs,” *Conference Record of the IEEE Photovoltaic Specialists Conference*, pp. 3300–3305, 2012.

[11] T. Niimura, K. Ozawa, D. Yamashita, K. Yoshimi, and M. Osawa, “Profiling residential PV output based on weekly weather forecast for home energy management system,” *IEEE Power and Energy Society General Meeting*, pp. 1–5, 2012.

[12] M. C. Bozchalui, S. A. Hashmi, H. Hassen, C. A. Cañizares, and K. Bhattacharya, “Optimal operation of residential energy hubs in smart grids,” *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1755–1766, 2012.

[13] C. Chen, S. Duan, T. Cai, B. Liu, and G. Hu, “Smart energy management system for optimal microgrid economic operation,” *IET Renewable Power Generation*, vol. 5, no. 3, p. 258, 2011.

[14] Y. Wang, X. Lin, and M. Pedram, “Adaptive control for energy storage systems in households with photovoltaic modules,” *IEEE Transactions on Smart Grid*, 2014.

[15] S. Teleke, M. E. Baran, S. Bhattacharya, and A. Q. Huang, “Rule-based control of battery energy storage for dispatching intermittent renewable sources,” *IEEE Transactions on Sustainable Energy*, vol. 1, no. 3, pp. 117–124, 2010.

[16] F. Ruelens, B. J. Claessens, S. Vandael, B. De Schutter, R. Babuska, and R. Belmans, “Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning,” *IEEE Transactions on Smart Grid*, no. 1, pp. 1–9, 2016.

[17] D. Zhang, S. Li, M. Sun, and Z. O’Neill, “An Optimal and Learning-Based Demand Response and Home Energy Management System,” *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 1790–1801, 2016.

[18] Y. Wang, S. Member, X. Lin, S. Member, M. Pedram, and A. Integrating, “A Near-Optimal Model-Based Control Algorithm for Households Equipped With Residential Photovoltaic Power Generation and Energy Storage Systems,” *IEEE Transaction on Sustainable Energy*, vol. 7, no. 1, pp. 1–10, 2015.

[19] Q. Wei, D. Liu, and G. Shi, “A Novel Dual Iterative Q-Learning Method for Optimal Battery Management in Smart Residential Environments,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2509–2518, 2015.

[20] G. Henri, N. Lu, and C. Carrejo, “Design of a Novel Mode-based Energy Storage Controller for Residential PV Systems,” in *IEEE PES Innovative Smart Grid Technologies, Europe*, 2017.

[21] J. B. Rawlings, D. Angeli, and C. N. Bates, “Fundamentals of economic model predictive control,” *Proceedings of 51st IEEE Conference on Decision and Control (CDC)*, pp. 3851–3861, 2012.

[22] G. Henri, N. Lu, and C. Carrejo, “A Machine Learning Approach for Real-time Battery Optimal Operation Mode Prediction and Control,” in *IEEE PES Transmission & Distribution Conference*, 2018.

[23] “Pecan Street Project.” [Online]. Available: www.pecanstreet.org

[24] J. Y. Kim, G. Li, X. Lu, V. L. Sprenkle, J. P. Lemmon, Z. Yang, and C. A. Coyle, “Distributed energy services management system,” 2013.

[25] I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” *Journal of Machine Learning Research (JMLR)*, vol. 3, no. 3, pp. 1157–1182, 2003.

[26] S. Kotsiantis, I. Zaharakis, and P. Pintelas, “Supervised machine learning: A review of classification techniques,” *Informatica*, vol. 31, pp. 249–268, 2007.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and . Duchesnay, “Scikit-learn: Machine Learning in Python,” vol. 12, pp. 2825–2830, 2012.

[28] W. E. Hart, “Python optimization modeling objects (Pyomo),” *Operations Research/ Computer Science Interfaces Series*, vol. 47, pp. 3–19, 2009.

[29] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *Iclr*, pp. 1–15, 2015.