IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 1, 2021, accepted February 11, 2021, date of publication February 22, 2021, date of current version March 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3060871

Intelligent Consumer Flexibility Management With Neural Network-Based Planning and Control

JUSSI KILJANDER, ROOPE SARALA, JARI REHU, DANIEL PAKKALA, PEKKA PÄÄKKÖNEN, JANNE TAKALO-MATTILA, AND KLAUS KÄNSÄLÄ

VTT Technical Research Centre of Finland, 90570 Oulu, Finland

Corresponding author: Jussi Kiljander (jussi.kiljander@vtt.fi)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under Agreement 824418 and Agreement 957670, and in part by Business Finland.

ABSTRACT Optimal management of demand-side flexibility in buildings is important for properly integrating large amounts of intermittent generation from windmills and photovoltaics. This paper proposes a novel Energy Management Agent (EMA) concept that can optimize building's energy costs with respect to external prices while at the same time allow building's flexibility to be used via explicit demand response. The EMA combines Artificial Neural Networks (ANN) and model predictive control for modelling and optimization of building's flexibility. It continuously manages building's flexibility with respect to external prices and provides forecasts of the load and available flexibility for a defined time window. A proof-of-concept (PoC) of the EMA is implemented for controlling a heat pump in an apartment, located in Oulu, Finland. Two ANN-based models were implemented for modelling the energy consumption of the heat pump and the indoor temperature of the apartment. Monte Carlo Tree Search based planning and control was implemented for finding optimal control policies with the ANNs. The EMA PoC was evaluated in 16-week period between 11 November 2019 - 1 March, 2020. When compared to a fixed setpoint control strategy, the EMA achieved 14.8 % lower costs under Nord Pool spot prices for Finland. At the same time, it was also able to accurately follow the 24h load plans (NRMSE was 0.050) and activate the offered flexibilities (NRMSE was 0.074).

INDEX TERMS Demand response, artificial neural network (ANN), optimal control, model predictive control (MPC), optimization of HVAC system.

I. INTRODUCTION

The power generation of renewable energy sources (RES) such as photovoltaics (PV) and windmills is volatile and cannot be controlled in the same way as in traditional power plants. The growing penetration of these type of RES makes demand-side an essential part of power grid management. In this situation, residential consumers have an important role as they control a large share of flexible resources that can be used for balancing the power grid.

Demand response (DR) programs focusing on residential consumers have been studied extensively [1]–[6], but there is still a lack of solutions that properly integrate small-scale consumers and prosumers as core components of smart grid management. There are three key challenges that need to be properly addressed by the consumer flexibility management systems. First, the consumer flexibility management needs to be fully automated and it needs to adapt to end-user behavior and preferences so that they do not have to be bothered with the daily operation. Second, the DR solutions addressing consumers and prosumers need to be more predictable at finer level of granularity in order to properly manage distributed energy resources (DER) within distribution networks. Third, demand-side flexibility management solutions need to support both implicit and explicit demand response programs at the same time. This is because fluctuations in the generation and demand are typically only visible in the global electricity market prices and optimizing flexibility only based on these prices (i.e., implicit demand response) can cause bottlenecks within the distribution network. Therefore, the demand-side

The associate editor coordinating the review of this manuscript and approving it for publication was Vahid Vahidinasab.

flexibility management system should provide aggregators and Distribution System Operators (DSO) with means to directly activate the available flexibility (i.e., explicit demand response) as long as the consumer is properly compensated.

Buildings are major consumers of energy (constitute roughly 40% of the total energy consumption in the EU[1]) and typically have large thermal mass that can be used for storing energy for short time periods. Therefore, buildings and their heating, ventilation and air conditioning (HVAC) systems are good source for demand-side flexibility. However, buildings' HVAC systems have complex non-linear dynamics with long feedback cycles caused by the thermal mass of the building. This makes it challenging to design controllers that can utilize the available flexibility in an optimal way.

Different type of approaches for automated and optimal energy management in buildings have been proposed in the literature. A popular method for learning optimal control policies in buildings is model-free reinforcement learning (RL), which has been demonstrated to improve energy efficiency and reduce costs when compared to traditional rule-based control strategies [7]–[9]. However, model-free RL has two significant limitations, which make it non-ideal solution for consumer flexibility management. First, RL is sample inefficient making it difficult to apply in the real world as it requires a lot of trial and error learning. Second, model-free RL methods do not natively provide means for explicit demand response, because without a model of the system there is no way for forecasting the building's load profile and available flexibility nor predicting the response of HVAC systems to explicit demand response events.

A model-based approach such as optimal control can, at least in theory, properly address the above-mentioned challenges. Optimal control requires an accurate model of the building and associated energy systems. There are tools for creating very accurate physical models of buildings such as Energy Plus[2] and Revit.[3] However, these types of models can be impractical due to the modelling effort and their unsuitability for real-time optimization.

To address these limitations there is a need for more lightweight approaches that can learn models automatically from data (or learn some of the model parameters of an otherwise physics-based model). These type of approaches could be classified either as optimal control with learned model dynamics or model-based reinforcement learning as they nicely bring together the science from optimal control theory and machine learning communities.

Neural networks (NN) are powerful function approximators, which have been shown to provide state of the art results in building energy modelling and load forecasting [10], [11]. Neural network based model-predictive control (MPC) have also been demonstrate to provide good results on energy-efficient control of HVAC systems [12]–[14]. Edge architectures to support ANN-based MPC (i.e., ANN-MPC) have been also proposed in the literature [15]. However, there are still some limitations in the existing work. In particular, the current work on ANN-MPC focuses on energy-efficiency and implicit demand response, and to the best of our knowledge, there are no ANN-MPC approaches that provide both implicit and explicit DR capability.

In this paper, we propose a novel approach for intelligent agent based energy and flexibility management in buildings. A central concept in the approach is Energy Management Agent (EMA) that automates and optimizes consumer flexibility management. EMA is not only designed to optimize energy with respect to external signals (i.e., implicit demand response), but also contribute to power grid management in two ways. First, EMA provides interface for relevant stakeholders (e.g. e.g. DSOs, TSOs, aggregators) to receive information about the load profile it plans to follow. EMA will also provide estimate of the building's flexibility at different time periods in the future and execute explicit DR actions by activating these flexibilities.

In addition to the general EMA framework, a key contribution of the paper is a proof-of-concept (PoC) implementation of the EMA for an apartment located in Oulu, Finland. We use neural networks to learn dynamics of the apartment heating and utilize Monte Carlo Tree Search (MCTS) for planning and control. MCTS is a simulation-based tree search technique that has become popular in game playing. Most well-known examples are the AlphaGo [16], and AlphaZero [17] that combine MCTS with deep reinforcement learning to provide superhuman and state-of-the-art performances in Go, Shogi and Chess. We demonstrate the approach in context of heat pump control and show that it is possible to simultaneously optimize energy consumption with respect to external prices while providing accurate load forecasts and DR responses.

The rest of the paper is structure as follows. Section II represent the general EMA concept, including the functional architecture and the overall approach for modelling and control. Section III describes the PoC implementation of an EMA designed for optimizing heat pump control under spot prices. Section IV presents the validation of the PoC implementation. Section V concludes the paper.

II. ENERGY MANAGEMENT AGENT

A. CONTEXT VIEW

The Energy Management Agent optimizes energy within a site by controlling flexible resources to maximize consumer benefits. The consumer benefits are represented as a cost/reward[4] function that can be customized based on the end-user preferences. End-user comfort is typically represented as constrains (e.g. temperature limits), but can be also be included to the cost function. In section 3.C we

[1] https://www.odyssee-mure.eu/publications/policy-brief/buildings-energy-efficiency-trends.html

[2] https://energyplus.net/

[3] https://knowledge.autodesk.com/support/revit-products

[4] Whether reward or cost function is used depends on the optimization algorithm.

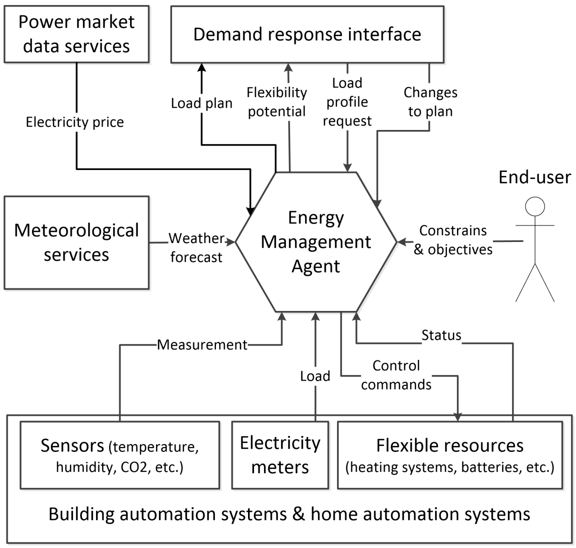

FIGURE 1. Context view of the energy management agent.

introduce the specific cost function used for the EMA PoC implementation.

The interaction between EMA and relevant external systems is presented in FIGURE 1. EMA follows the idea of bottom-up based flexibility management [18] where a building informs an aggregator both about the load profile it plans to follow and the flexibilities available at different time periods. This approach allows the aggregator to explicitly activate the flexibilities available at different time periods.

There is a wide variety of market structures, incentives, aggregation methods and DR programs between countries and geographical areas in the world. Energy sector is also under transition and new concepts such as Peer-to-Peer (P2P) trading and Virtual Power Plants (VPP) that change the dynamics of energy markets are becoming more popular. Because of this, EMA is designed to provide common functionality and interfaces that can be easily customized for different market structures and aggregation methods. Conceptually, this integration and customization is done within the *Demand response interface* component which provides external stakeholders such as utilities, retailers and aggregators with standard (e.g. IEC 62746 / OpenADR 2.0b) interface to interact with EMA. This component is also responsible for possible price forming, and trading activities required in the given setting. As presented in FIGURE 1, *Demand response interface* component is not part of EMA and is thus outside of the scope of this paper. A logical description of the flexibility management interface provided by EMA is presented in TABLE 1.

In addition to the *Demand response interface*, relevant external systems for EMA include *Building and home automation systems*, *Power market data services*, and *Meteorological services*. EMA interfaces with building and home automation systems in order to 1) collect necessary data about the building environment and energy consumption, 2) and to control flexible resources. Weather forecasts are fetched from

TABLE 1. EMA flexibility management interface.

NAME	TYPE	DESCRIPTION
Load plan	Output	A time series of forecast power loads for a configurable time window (typically 24h). This information can be used to calculate the baseline for DR events. It can be also sent to aggregators to improve their estimates of the overall loads within a certain area of the power grid. In P2P markets this information provides a baseline estimate for the energy that need to be acquired.
Flexibility potential	Output	A time series of up and down flexibility potential (i.e., delta compared to the load plan). Several flexibility levels can be provided for a single timeslot. The length of the time series is configurable.
Changes to load plan	Input	A time series of changes to the current load plan. This message is used for informing the Energy Management Agent about the activated flexibilities.
Load profile request	Input/ Output	This interface can be used to request the EMA to forecast the effects of a flexibility activation before the actual activation. The request message is identical to the Changes to load plan message. The response message includes the new load plan.

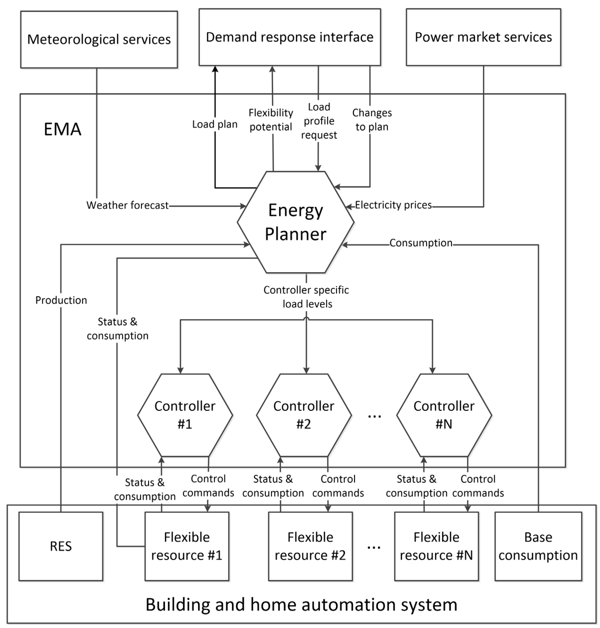

FIGURE 2. Functional view of the energy management agent.

meteorological services and electric price data from *Power market data services* such as the one offered by Nord Pool.

B. FUNCTIONAL VIEW

The Energy Management Agent consist of two type of functional components as illustrated in FIGURE 2.

The Energy Planner is responsible for planning and optimizing the energy usage within the site at all times. The functionality of the Energy Planner can be divided into five main parts:

1. Once a day, before the day-ahead market closes, the Energy Planner sends a *Load plan* message to the DR Interface. The details on how the load plan is optimized in practice is presented in section 3.C.
2. Continuously during the day, the Energy Planner provides the DR Interface with information about the flexibility potential of the site.
3. Whenever a *Load profile request* is received the Energy Planner optimizes a new load plan with the given constrains and returns it in a response message.
4. Whenever a *Change to load plan* message is received, it will perform the same operation as in step 3, but this time the new load profile is activated and sent to the Controllers.
5. Continuously, the Energy Planner monitors and plans the site overall load profile and assigns individual load profiles for each Controller. This is done continuously to be able to adapt to DR events and other unexpected changes in the planned load profile.

Logically, there is a Controller component for each flexible resource type within a site. Each Controller component is responsible for controlling a flexible resource according to the plan provided by the Energy Planner. The length of the time series is configurable and depends on the length of the overall load plan. Details on the Energy Planner and Controller are provided in section II.C.

C. NEURAL NETWORKS FOR PLANNING AND CONTROL

The approach for implementing the functional components of the EMA can be classified either as model-based reinforcement learning or optimal control where the models are learned from data with neural networks. FIGURE 3 presents a block diagram illustrating how neural networks and optimization methods are utilized in the EMA framework.

The Energy Planner utilizes ANN-based optimization for finding an optimal control policy for each flexible resource from the space of possible load profiles. The optimality of the load plan is measured by a reward or cost function, which varies depending on the end-user and incentive models. In addition to the flexible resource models, the Energy Planner can utilize models for inflexible loads and RES generation in the load plan optimization. In generic-level, the objective of the Energy Planner in making the load plan is presented in (1). A concrete example of this objective function, tied to spot-price optimization, is presented in section III.C (4).

$$\begin{aligned} &\max_{a_1,\ldots,a_T} \sum\nolimits_{t=1}^{T} r\left(s_t, a_t\right) \\ &s.t.\quad s_t = f_f\left(s_{t-1}, a_{t-1}\right) + f_g\left(s_{t-1,}\right) + f_d\left(s_{t-1,}\right) \\ &\qquad s_{min} \leq s_t \leq s_{max}, \end{aligned} \tag{1}$$

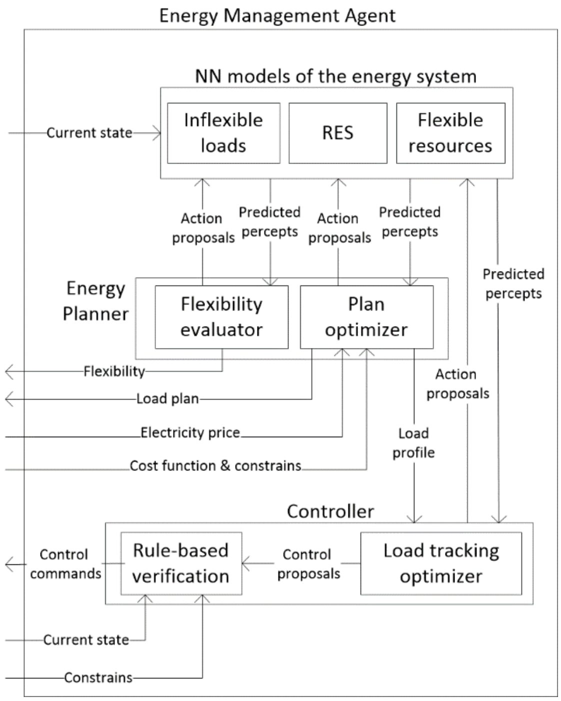

FIGURE 3. **Model predictive control for building flexibility management.**

where r is the reward function, s_t is the state of the system, and a_t is the action. The s_{max} and s_{min} represent possible constrains such as the minimum and maximum values for indoor temperature. The f_f, f_g and f_d represent ANN models for flexible resources, power generation and inflexible demands, respectively. In practice, each of the functions f_f, f_g and f_d can be represented with one or more ANN models and there is a wide variety of ANN architectures studied in this context, including Feed Forward Neural Networks (FFNN) [19], [20], Long short-term memory (LSTM) [21], [22], Factored Conditional Restricted Boltzmann Machine (FCRBM) [23], Convolution Neural Network (CNN) and Stacked Booster Network (SBN) [11], and Recurrent Inception Convolution Neural Network (RICNN) [24], to name a few.

EMA approach does not enforce any particular non-linear optimization methods to be used for making the load plan. In fact, it is still an open research question to find the most suitable methods for ANN-based optimal control. The optimization methods can be roughly classified either as derivative-based methods or gradient-free methods. Derivative-based methods are more efficient, but it has been traditionally difficult to apply them in ANN-based control due to exploding and vanishing gradient problems [25]. These problems can be mitigated with computationally heavy methods exploiting second derivatives such as the Newton-Raphson method [26] and the Interior-point optimization [14], [27]. Additionally, there is recent work that shows how more lightweight first order methods based on Tensorflow can be used for ANN-based planning, but the approach does not support additional constrains making it infeasible for typical EMA applications [28]. Gradient-free methods such as Genetic Algorithms (GA) [29], [30] and Particle Swarm Optimization (PSO) [31], [32] have been so far more popular for ANN-based energy optimization in buildings. The PoC implementation of the EMA, presented in section 3, utilizes also a gradient-free method, called Monte Carlo Tree Search (MCTS) [33] for planning and control.

The Energy Planner can utilize either an open loop or closed loop control such as MPC for making the load plan by adjusting how often a new load plan is made. In MPC new load plan is made at every time step whereas in open loop control a plan is made once per length of the load plan. This design choice is a trade-off between the accuracy of the load plan and optimality of the energy management that minimizes the cost function. That is, with MPC a plan is optimized at every time step with the latest information which leads to more optimal control but less predictable long term plans when compared open loop control.

It should also be noted, that optimization can be also utilized for planning the flexibilities (i.e., optimizing a maximum flexibility for a period where it is likely that the price for flexibility is the highest). However, the current approach for evaluating flexibilities is based on forecasting the loads with maximum and minimum control values for every single time period separately and assumes that all of the other time periods are executed according to the current load plan.

As can be seen from FIGURE 3, the Controller also utilizes ANN-based model predictive control. In contrast to the Energy Planner where it is possible to configure between MPC and open loop control, the Controller will always utilize MPC since there is no tradeoff to be made in this case. As presented in section II.B, the objective of the Controller is to follow the individual load plan assigned by the Energy Planner. The objective of the Controller can be thus presented as follows.

$$\begin{aligned} \min_{a_1,\ldots,a_T} \quad & \sum\nolimits_{t=1}^{N} \left(E_t - \hat{E}_t\right)^2 \\ s.t. \quad & s_t = f_f\left(s_{t-1}, a_{t-1}\right) \\ & s_{min} \le s_t \le s_{max} \\ & \hat{E}_t \in s_t, \end{aligned} \tag{2}$$

where E_t is the energy in the load plan, $\hat{E}_t$ is the energy consumption predicted by the model f_f, and s_t is the state of the system including the energy consumption and user comfort.

Controller only requires model(s) of the flexible resource it is controlling. Same model as used by the Energy Planner can be typically used for control. It should be also noted, that if the flexible resource dynamics are simple and there are no long delays it is also possible to utilize rule based logic or proportional-integral-derivative (PID) controller instead of NN-MPC. However, NN-MPC is the preferred option with flexible resource that have non-linear dynamics and/or delays as it has been shown to outperform these more classical control strategies in building HVAC control [12], [14].

III. PROOF-OF-CONCEPT IMPLEMENTATION

The Energy Management Agent PoC implementation was implemented with Python. Tensorflow 2.0 with Keras API was used for implementing the ANNs for heat pump and heating dynamics modelling. The interface between EMA and the DR interface, was implemented on top of MQTT with Eclipse Paho Python client. The messages, presented in TABLE 1, are serialized with JSON. Weather forecast data is read from Finnish Meteorological Institute (FMI) open APIs[5] and electricity price data from Nord Pool Power Data Service.[6]

Section III.A briefly introduces the apartment where the EMA was deployed and describes the relevant interfaces. Section III.B presents how the apartment's flexible resource (i.e., a heat pump) is modelled with ANNs and section III.C describes how the planning and control were implemented with MCTS.

A. INSTANTIATION AT APARTMENT HEATING

The EMA instance was developed and deployed for test apartment located in Oulu, Finland. The apartment's floor area is $72m^2$ and it has three rooms and sauna. The apartment is equipped with a variety of sensors (e.g. temperature, humidity air pressure and CO2) of which the temperature sensors located in the living room, kitchen and outside of the building are used for this case study. Sensor information is received via Bluetooth Low Energy (BLE) in 1-minute resolution. The apartment is also equipped with energy meters to monitor the energy consumption of the whole apartment and various submetering points such as the heat pump in 1-minute resolution.

The apartment is equipped with a heat pump (Toshiba Digital RAV-SM307KRTP-E indoor unit and RAV-SM304ATP-E outdoor unit) that acts flexible resources for the case study. The output temperature of the heat pump can be controlled by modifying a setpoint via a Modbus[7] interface. We did not have access to the internal sensors and control logic of the heat pump, which makes it challenging to accurately control it. Moreover, the requested setpoints do not fully match with the temperatures measured within the apartment (i.e., the room temperatures are typically 1-2 °C higher than the requested setpoint). For this reason, a simple wrapper layer was implemented that controls the heat pumps so that the requested setpoints better reflect the actual temperatures within the apartment. The simple logic of the control wrapper is presented below as python-like pseudocode.

[5]https://en.ilmatieteenlaitos.fi/open-data-manual-time-series-data
[6] https://www.nordpoolgroup.com/historical-market-data/
[7]https://modbus.org/

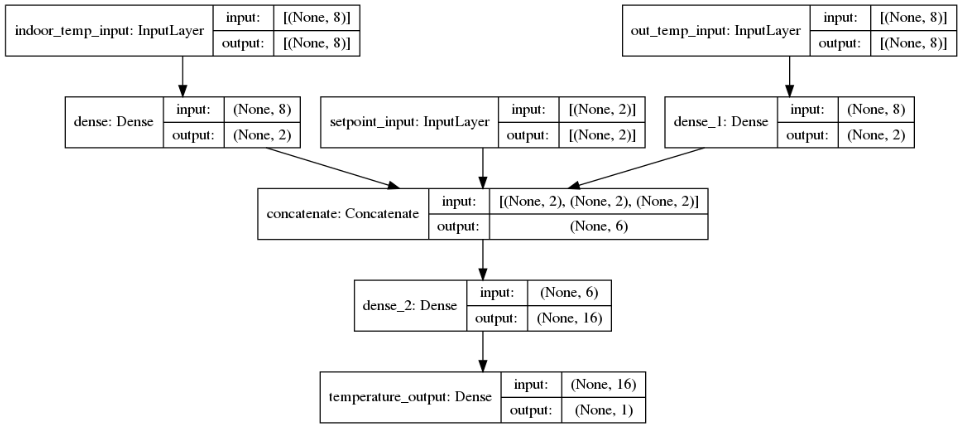

FIGURE 4. Feed forward neural network for temperature dynamics modelling.

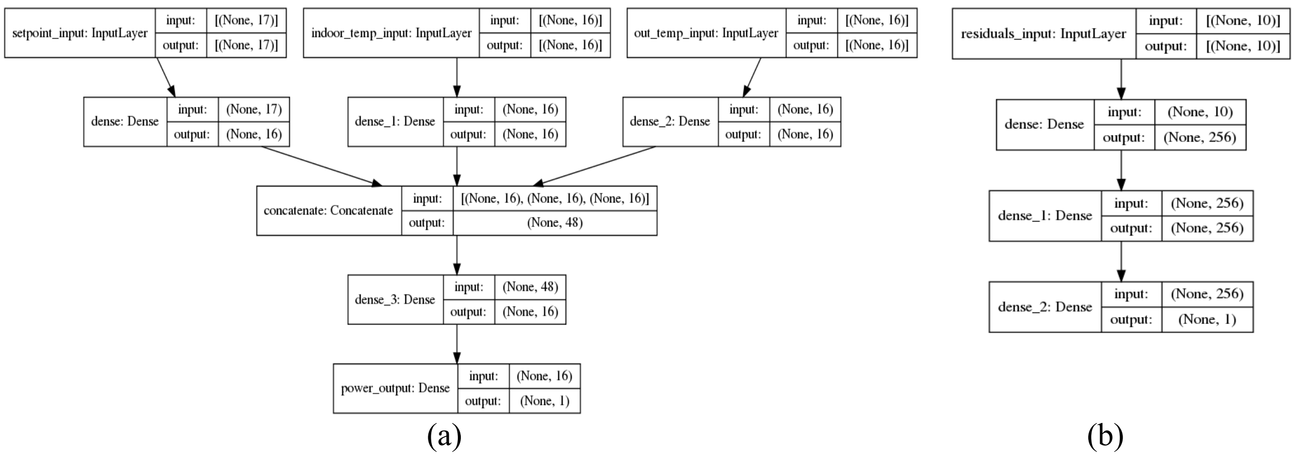

FIGURE 5. Feed forward neural networks for heat pump energy consumption modelling.

```
def control_wrapper(setpoint,
                    temperature,
                    last_setpoint,
                    threshold = 0.25):
      if setpoint-threshold <= temperature < set_point + threshold:
         setpoint = last_set_point
      elif temperature < set_point − threshold:
         setpoint = setpoint
      else:
         setpoint = 18.0
      return setpoint
```

B. HEATING MODELLING WITH NEURAL NETWORKS

The apartment's heating dynamics and the heat pump power consumption are modelled separately with Feed Forward Neural Networks. The sampling time for the modelling is chosen to be 15-minute since the temperature of the apartment does not change rapidly and there can be long delays in the heat pump control. The selected sampling time is also fine enough granularity for energy management and demand response since the current markets operate at 60-minute resolution.

The first model, architecture presented in FIGURE 4, forecasts the indoor temperature of the living area (average of the living room and kitchen temperature measurements is used) over the next sampling time. The inputs of the model include indoor temperatures from eight past sampling periods, outdoor temperature for eight past sampling periods (weather forecast are used for long horizon forecast in online mode), the setpoint of the last sampling period, and the next sampling period's setpoint (i.e., the period for which the temperature forecast is being made). Longer than 15-minute forecasts are performed by calling the model iteratively and using the indoor temperature forecast of the previous iteration as input for the next.

The second model, presented in FIGURE 5, forecasts the energy consumption of the heat pump for the next sampling period. The model consist of two FFNNs, which are trained separately: the main model and the residual model. The inputs of the main model include past 16 indoor temperature, outdoor temperature and setpoint samples, and a setpoint for the next sampling period. It should be noted that the future values for indoor temperature are provided by the temperature model presented in FIGURE 4. The residual model (b) was developed to correct the forecast of the main model based on residual information obtained from past forecasts. It utilizes 10 past residuals for correcting a forecast. When the model (a) is called iteratively to perform a longer forecast, the same correction provided by the model (b) is used in all iterations.

In all three FFNNs, Rectifier Liner Unit (RELU) activation function is used for all neurons except the outputs which are linear. Adam [34] is used as the optimization algorithm in training of the models. All figures of the FFNNs were created with Keras and the *None* in the input shape indicates that the batch size is not fixed.

The models were trained and evaluated with a total of 72 days of data collected during 2019. It should be noted, that the 72 days were not consecutive, but instead spread between 18th of October and 10th of November 2019. The training set consisted of 43 days collected between 18th of October and 10th of November. The test set consisted of 29 days collected between 11th of November and 28th of February 2020 (there were some longer caps caused by problems in the data collection).

Root-mean-square error RMSE (3) was used as the error metrics in the model validation. The one-step prediction errors for the temperature and energy models were 0.103 Celsius degree and 0.023 kWh, respectively. When normalized with respect to maximum and minimum values the NRMSE (4) are 0.032 and 0.077 for the temperature and energy consumption models, respectively.

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{3}$$

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \tag{4}$$

Figure 6 illustrates the autocorrelation function (ACF) plots for the one-step ahead residuals. The dashed line and the solid line correspond to the 99% and 95% confidence intervals, respectively. The energy model residuals do not show any significant correlation, but there are still some lag values that go above the 95% and 99% confidence bands. The temperature model residuals show stronger correlation that cannot be fully explained as white noise. However, the correlation is in general weak and modelling the residuals did not help to improve the accuracy nor reduce the correlations.

C. PLANNING AND CONTROL WITH MONTE CARLO TREE SEARCH

As presented in section II.C, EMA control is performed at two levels. The Energy Planner makes a load plan for the whole site, which includes load plans for individual flexible resources. The role of the Controller(s) is then to follow the plan (i.e., to minimize (2)). The control window N in (2) was selected to be 4 (i.e., 60 min) and the size of the search space was thus 5^4. We selected energy cost reduction as the optimization target for the EMA PoC implementation under hourly changing electricity spot prices. Comfort (i.e., indoor

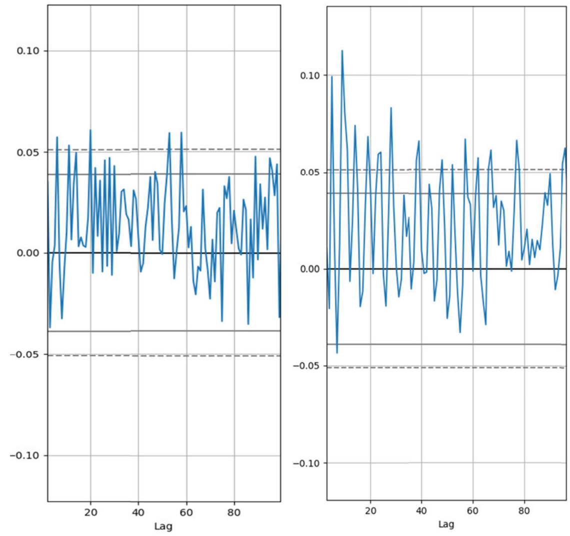

FIGURE 6. Autocorrelation function plots for energy (left) and temperature (right) one-step ahead forecasts errors.

temperature in this case) is included as a constrain to the optimization problem presented as follows:

$$\begin{aligned}
&min \ \sum\nolimits_{t=0}^{N-1} P_t E_t \\
&sbt.\ E_t = f_E\left(a_t, a_{t-1}, \ldots, a_{t-\delta_S}, T_t, T_{t-1}, \ldots, T_{t-\delta_E}, O_t,\right. \\
&\qquad\qquad \left. O_{t-1}, \ldots, O_{t-\delta_O}\right) \\
&\qquad T_t = f_t\left(T_{t-1}, \ldots, T_{t-\delta_E}, a_t, a_{t-1}, \ldots,\right. \\
&\qquad\qquad \left. a_{t-\delta_S}, O_t, O_{t-1}, \ldots, O_{t-\delta_O}\right) \\
&\qquad T_{ref} - \epsilon_1 \leq T_t \leq T_{ref} - \epsilon_2 \\
&\qquad \forall a_t \in \{21, 22, 23, 24, 25\} \\
&\qquad \forall t \in \{0, 1, \ldots, N-1\},
\end{aligned} \tag{5}$$

where P_t is the hourly changing energy price, f_E is the energy consumption model, and f_t is the indoor temperature model. T_{ref} is the indoor temperature reference value and the slack variables ϵ_1 and ϵ_2 specify how much the temperature can deviate from the reference setpoint.

The complexity of the planning was reduced in two ways. First, the planning was done at 60 min resolution instead of 15 min. This is reasonable approach in this case since the Nord Pool day-ahead prices change hourly. Second, the 24 h load plan was made in 24 parts and the length of the sliding planning window was selected to be 8 hours. Thus the search space of potential scenarios for each planning window was 5^8.

The load planning and control in the EMA PoC implementation is based on the Monte Carlo Tree Search. MCTS has not been yet widely studied in building energy optimization or demand response. However, it provides a lightweight alternative to derivative-based methods for planning and control with ANN models. It is also a natural selection over derivative-based methods when the control inputs (i.e., the heat pump setpoints in this case) are not continuous. There are many variants for MCTS. In our PoC implementation, the algorithm consist of four steps: selection, expansion, simulation, and backpropagation. Next these steps and how they are implemented in EMA are briefly presented.

1) SELECTION

Starting from the root node (i.e., current state of the system) select successive child nodes (i.e., possible states that can be reached from the current state) until a leaf node is reached. A leaf is any node that has a child node from which no simulation has been executed. Upper Confidence Bound applied to Trees (UCT) [35] (6) is used as the evaluation function to balance exploration-exploitation trade off.

$$V_i = x_i + C\sqrt{\frac{\ln(n_p)}{n_i}}, \tag{6}$$

where x_i is the empirical mean value of the node i, C is a constant used for balancing between exploitation and exploration, and n_p and n_i are the number of times the parent of node i and the node i have been visited, respectively.

2) EXPANSION

Unless the leaf node is the last time step on the planning or control window, create new child node and choose one of them by taking a valid action from the leaf node state. Child nodes are possible states that can be reached from the leaf node by taking a valid action (i.e., setpoint allowed with respect to the forecast indoor temperature of the state). In practice, the action is taken by predicting the next (child) state with the ANN models presented in section III.A.

3) SIMULATION

Complete a rollout from the selected child node until the end of the planning or control window is reached. The purpose of the simulation is to evaluate the value of the current state. Typically, random policy is used but EMA uses fixed policy instead where the simulation is performed with a fixed setpoint (i.e., the desired temperature of the apartment). Fixed setpoint policy is used as the simulation policy, because it is more likely policy than random sampling in this setting and provides thus more accurate estimate for the value of the state. Similarly to the expansion phase, the simulation is performed by utilizing the ANN models presented in FIGURE 4 and in FIGURE 5.

4) BACKPROPAGATION

Use the result of the simulation to update the values of all nodes on the path from the child node to the root node (i.e., current state). The value of a node is the total reward that has been reached from a node divided by the amount of visits to the node. The equations (2) and (4) cannot be directly applied in MCTS as the score function however. This is because the MCTS is typically applied in two player games, where scores are -1, 0, 1 for loss, draw and win, respectively. This means the score of a node is therefore typically within $[-1, 1]$. The problem here is that if the score values differs significantly from this range, the typical value (i.e., $\frac{1}{\sqrt{2}}$) for the constant C in (6) is not feasible. A solution to this issue is to modify (2) and (4) so that score values remain close the range $[-1, 1]$. Another option would be to find a value for C so that it is feasible for the new score range. We used the first approach in our case study. In practice, the minimum and maximum values needed for scaling the rewards were first evaluated by running a fixed setpoint policy with historical data and updated with better estimates during the simulations.

IV. EVALUATION

The goal of the EMA concept is to realize an intelligent software agent that can optimize consumer's energy locally (e.g. with respect to external price signals) while at the same time contribute to power grid balancing with more predictable load profiles and flexibility offers. To this end, the aim of the evaluation was to answer following questions:

1. How much EMA is able to reduce costs when compared to a baseline control strategy?
2. How accurately EMA is able to follow the load plans during and outside of explicit DR events?

The evaluation was divided into two scenarios. The first scenario, presented in section IV.A, targeted to answer the first question by comparing EMA to a fixed setpoint (FSP) controller in local energy optimization with hourly changing electricity prices (i.e., implicit demand response). The second scenario, presented in section IV.B, extends the first scenario with explicit DR events and evaluates how well the EMA PoC is able to follow the day-ahead load profile both during and outside of the DR events.

A challenge in the evaluation was to credibly compare the control strategies. This is not straightforward since weather influences the energy consumption and the energy prices vary dynamically, making it impossible to replicate the exact same conditions for EMA and the baseline control. Moreover, it is also important to collect data from long enough period. In the literature simulations have been typically used in similar settings [12], [13], [30]. However, if we would directly simulate with the models represented in section III, the results would be too optimistic as the accuracy of the models used in planning and control would be perfect. To tackle this problem and make the setting as realistic and fair as possible, errors were sampled from the empirical error distributions obtained during the validation of the models. As presented in section III.B, the ACF showed only were weak correlations and it is thus possible to sample the errors independently from each other. With temperature model there is also no dependency between the error and the model inputs or outputs. With the energy model there was a natural correlation between the errors and the measured energy consumption. This correlation was taken into account in the error sampling by restricting the error so that the energy consumption remains between zero and maximum power of the heat pump. This way the models used by EMA have the same accuracy as with real apartment and

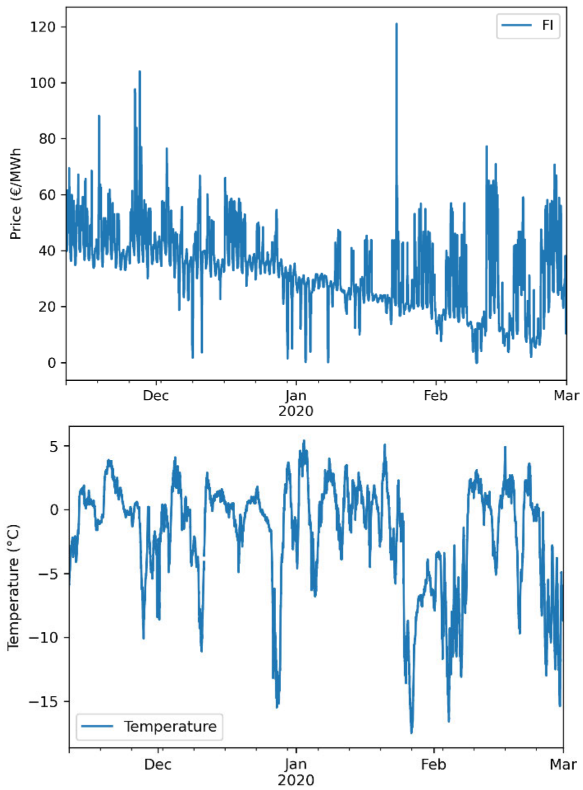

FIGURE 7. **Measured outside air temperature and corresponding electricity spot price for the evaluation period.**

the whole concept can be evaluated in as realistic setting as possible.

The evaluation period was 16 weeks and it was executed between 11 November 2019 - 1 March, 2020. FIGURE 7 presents the outside air temperatures and the Nord Pool Elspot electricity prices for Finland in the evaluation period. The simulations were conducted on Intel Core i5 with 16 GB RAM.

A. IMPLICIT DEMAND RESPONSE

In this scenario, EMA is provided with the next day electricity prices and it optimizes energy costs by controlling the heat pump. The performance of EMA is compared to a baseline controller with fixed setpoint at 23.0 °C. The slack variables ϵ_1 and ϵ_2 in (5) were set to 0.5 and 2.0, respectively.

Since the electricity costs are reported at one hour resolution the planning window for the EMA was 24 time steps long. On average, the Planner used 4 minutes (roughly 10 seconds for each hour) to make a load plan. As presented in section II, the Controller follows the original plan provided by the Energy Planner and tries to maximize (2). Control was executed at 15-minute intervals and the control window was 60 minutes (4 time steps) long. One-second time was given to the Controller for searching the optimal control at each time step.

FIGURE 8 and FIGURE 9 present the daily and cumulative electricity costs for the baseline (i.e., FSP) and EMA controllers.

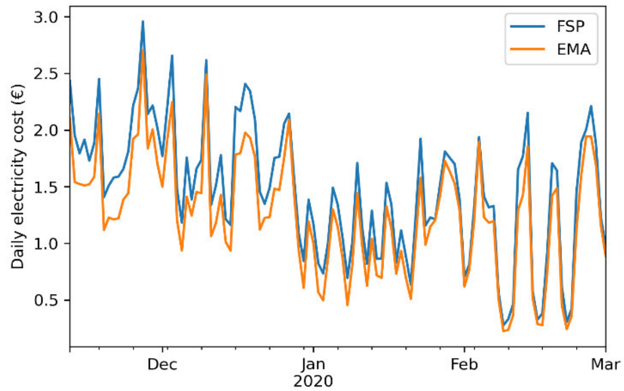

FIGURE 8. **Daily electricity costs for baseline (blue) and MCTS (orange) based control.**

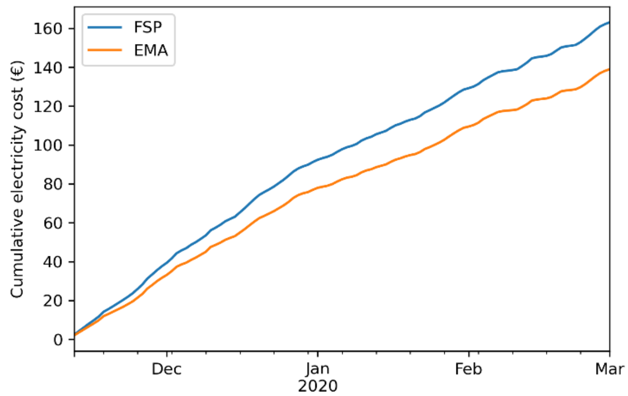

FIGURE 9. **Cumulative electricity costs for baseline (blue) and MCTS (orange) based control.**

During the 16-week validation period the electricity cost of the EMA controller was 14.8% lower than with the baseline controller with a fixed setpoint. It should be noted that roughly half of the costs reduction originated from reduced energy consumption instead to spot price optimization. Although the main goal was not to reduce energy consumption, EMA found a way to reduce the costs by keeping the temperature closer to the minimum value (i.e., 22.5 Celsius degree). It should be also emphasized that evaluation focused only to the electricity prices, which constitute roughly 1/3 of the total price in Finland (the other 2/3 comes from the network free and taxes).

FIGURE 10 and FIGURE 11 illustrate snapshots of heat pump loads and indoor temperatures for the EMA controller during the validation period. The data is represented in 15-minute resolution. FIGURE 10 illustrates a period between 2019-12-29T16:00 and 2019-12-31T:06:00 with two clear drops in the electricity price during the nights. As can be seen from the figure, the EMA exploits these low price periods for extra heating. FIGURE 11 shows a high peak in the electricity spot price at 2020-01-23T:07:00. As can be seen, the EMA pre-heats the apartment in order to avoid heating during the high price periods. In general, the EMA controller also used much shorter heating cycles than the FSP in order to keep the temperature close to the minimum value.

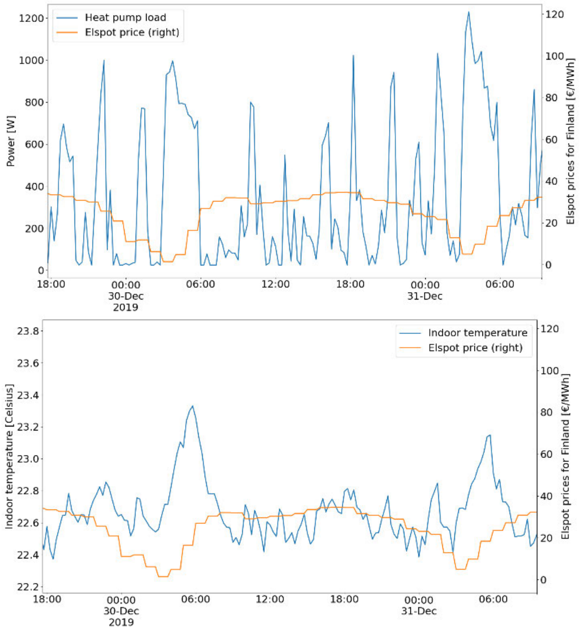

FIGURE 10. Heat pump loads (above) and indoor temperatures (below) between 18:00 2019-12-29T16:00 and 2019-12-31T:06:00.

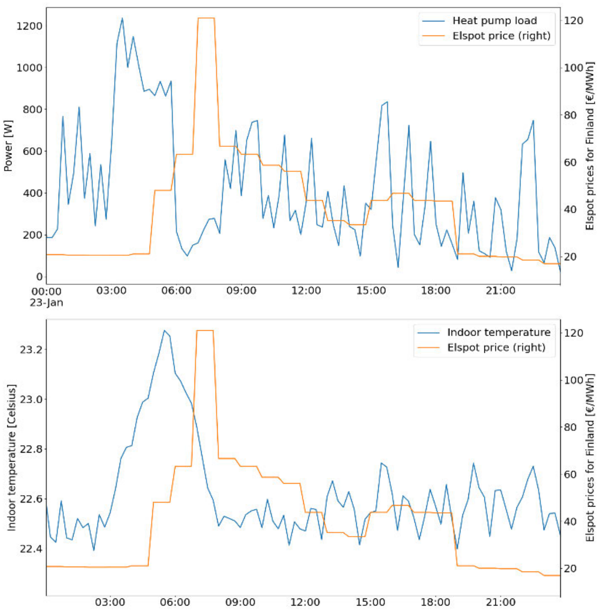

FIGURE 11. Heat pump loads (above) and indoor temperatures (below) between at 2020-01-23.

B. COMBINED IMPLICIT AND EXPLICIT DEMAND RESPONSE

In the second part of the validation, the implicit demand response optimization (i.e., scenario 1) is extended with daily demand response events to evaluate how well the EMA implementation can support explicit and implicit demand response at the same time.

There can be many reasons for aggregators and utilities for utilizing explicit DR in order to deviate the consumer from their normal load profile. In this evaluation, the explicit DR events were selected to occur during low price periods in order to simulate local bottlenecks in the distribution network. That is, the explicit DR events were selected to occur during the hour of the lowest price for each day to follow the assumption that majority of the consumers would be optimizing energy within a substation causing a peak load in that area. The duration of each DR event was 60 minutes and a total of 112 DR events were executed during the validation period (i.e., one event per day).

In similar way as in scenario 1, EMA makes the initial load plan by optimizing the heating with respect to Finnish Elspot prices and sends the *Load plan* message to the *Demand response interface*. The *Load plan* message contains the estimated load of the heat pump for every hour of the day. At 60-minute intervals EMA also forecasts the up and down flexibility for each remaining hour in the current market window (i.e., current day) and sends the *Flexibility Potential* message to the *Demand response interface*. Only the first flexibility offer of the day (i.e., 24h forecast) was used in this case study. From the first *Flexibility potential* message of the day, the down flexibility for the hour with the minimum price was activated by sending the *Changes to load plan* message to the EMA. EMA then made a new plan taking the activated flexibility into account. It should be noted, that exactly the same control strategy as in scenario 1 was executed and the only difference was that the flexibility was activated for the highest price period.

Following two indicators were used for validating the suitability of the EMA PoC for explicit DR:

1) **Accuracy of the original load plan without DR events:** The accuracy of the original load plan measures how well EMA is able to follow the planned load profile. In practice, it is measured by calculating the RMSE (eq. 3) between the *Load plan* and the actual measured load. This is an important metric for two reasons. First, the load plan provides the aggregator a forecast on the load profile which the aggregator will use for planning the DR actions. Second, all flexibility is compared to this baseline so errors in this baseline make it also more difficult to validate the flexibility.
2) **Accuracy of the original load plan during the DR periods:** This indicator measures how accurately the EMA is able to activate the offered flexibility. In practice, it consists of following interlinked tasks which cannot be measured separately: accuracy of the flexibility forecasts and ability of the controller to follow the adjusted load profile. RMSE (eq. 3) is used as the metric for the accuracy of the load plan during the DR event. The RMSE is calculated for the hours (112 events in total) where explicit DR events where executed by comparing the modified load plan (i.e., the original *Load plan* modified with the flexibility activated from *Flexibility potential* message) to the measured load.

TABLE 2 presents the RMSE and NRMSE metrics for the abovementioned indicators obtained during the validation.

The EMA was able to follow the load profile accurately both during and outside of the DR events. The NRMSE for

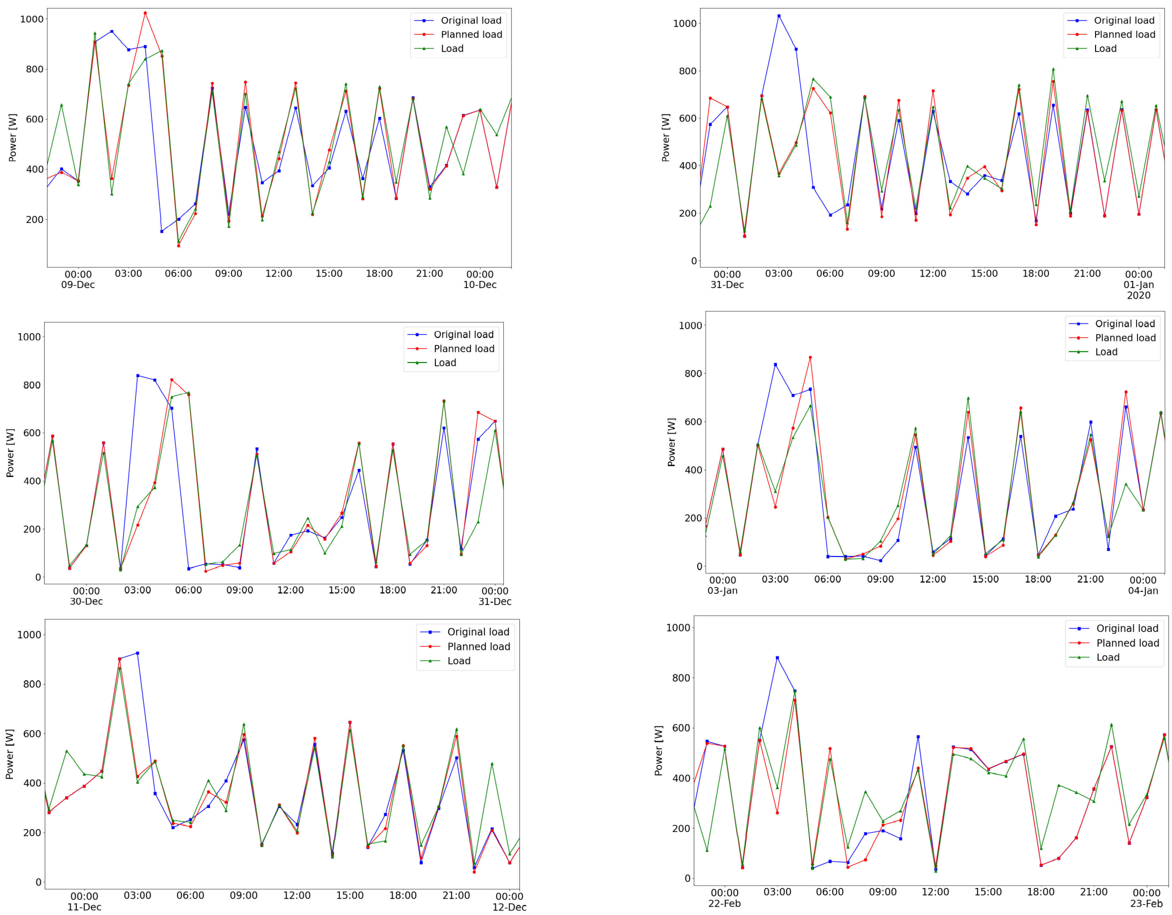

FIGURE 12. Examples of heat pump load profiles during the validation period. Blue line (with square markers) depicts the load plan before flexibility activation. Red line (with circle marker) presents the updated load plan after the flexibility offer for the hour with the lowest price was accepted. Green line (with triangle markers) depicts the actual measured load during for each hour. The data is represented in 60-minute resolution.

TABLE 2. Error metrics.

Indicator	*RMSE*	*NRMSE*
Accuracy of the load plan without DR events	75 Wh	0.050
Accuracy of the load plan during the DR events	110 Wh	0,074

the load plan outside of the DR events was 0,050 which is even lower than the one-step energy model accuracy (0.077) presented in section III.B. The main reasons for this are that the 1) RMSE is measured in 60-minute resolution instead of 15-minute, and 2) that the Controller tries to fulfill the daily load plan (i.e., forecast) by controlling the setpoint in 15-minute intervals. With NNRMSE of 0.074, the accuracy of the load plans is slightly worse during the DR events. A reason for this that the indoor temperature during these events drops typically closer to the minimum value and even small errors in the temperature model forecasts are significant for the load profile forecasting. Nevertheless, the error is still small and the validations show that the heating behavior of the apartment is highly predictable.

FIGURE 12 illustrates the original load plans (in blue with square markers), the actual load plans with flexibility activation for the lowest price hour (in red with circle markers) and the measured loads (in green with triangle markers). In contrast to the scenario 1, the load profile data is represented in 60-minute resolution, which is the actual resolution of the Nord Pool power markets. This resolution was selected instead of the control resolution (i.e., 15-minutes) to properly visualize how well the EMA controller is able to follow the load plan. The six days visualized in FIGURE 12 were selected randomly among 18 days with the lowest hourly price.

The DR events were activated for following time periods: 2019-12-09T02:00, 2019-12-11T03:00, 2019-12-30T03:00, 2019-12-31T03:00, 2020-01-03T03:00, 2020-02-23T03:00. As can be seen, there is a roughly a 500-600 Wh delta between the original (blue) and updated (red) load plans. The actual load (green) also follows the load plan (red) accurately during these DR events.

V. CONCLUSION

This paper presented a novel concept and implementation of a consumer flexibility management solution, called Energy

Management Agent. A key idea in EMA is to learn building's HVAC dynamics with ANNs and utilize model-predictive control for finding optimal control policies. This way EMA can fully automate consumer flexibility management, as well as, reduce the efforts needed for modelling. Another key idea in EMA is to provide support for implicit and explicit demand response at the same time. This is important so that the flexibility available in the consumer-side can be seamlessly utilized for the most critical need at any given time (e.g. a bottleneck in the distribution network) while making sure that the consumer is properly compensated. The EMA utilizes implicit DR for planning a load profile for a given period (e.g. day). It then follows the load profile by utilizing model predictive control with ANN-based models. On fixed intervals, it advertises the up and down flexibilities, which can be activated by an aggregator. This way the aggregator has a good view on the consumers load profile and available flexibilities, and has means to directly activate the flexibilities when needed.

To evaluate the EMA concept in implicit and explicit demand response scenarios, a prototype was designed and implemented for controlling a heat pump in a test-apartment located in Oulu, Finland. The heat pump control was executed by modifying a temperature setpoint via a Modbus interface. The energy consumption of the heat pump and the indoor temperature of the apartment were modelled with separate ANNs. MCTS-based planning and control was implemented for searching optimal control policies with the ANNs.

The validation period was 16 week long and was executed between 11 November 2019 - 1 March, 2020. The validation consisted of two scenarios that were simulated with the models validated against real data. The focus in the first scenario was on implicit demand response. EMA was compared to fixed setpoint control strategy and it achieved 14.8 % lower costs under Nord Pool spot prices. In the second scenario the setting was extended with explicit DR events and the accuracy of the load plans, as well as, flexibility offers and activations were measured. EMA was able to follow the 24h load plan accurately both outside (NRMSE was 0.050) and during (NRMSE was 0.074) the explicit DR events.

REFERENCES

[1] N. Shaukat, S. M. Ali, C. A. Mehmood, B. Khan, M. Jawad, U. Farid, Z. Ullah, S. M. Anwar, and M. Majid, "A survey on consumers empowerment, communication technologies, and renewable generation penetration within smart grid," *Renew. Sustain. Energy Rev.*, vol. 81, pp. 1453–1475, Jan. 2018.

[2] J. Wang, M. A. Biviji, and W. M. Wang, "Lessons learned from smart grid enabled pricing programs," in *Proc. IEEE Power Energy Conf. Illinois*, Feb. 2011, pp. 1–7.

[3] A. Mnatsakanyan and S. W. Kennedy, "A novel demand response model with an application for a virtual power plant," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 230–237, Jan. 2015.

[4] G. R. Newsham and B. G. Bowker, "The effect of utility time-varying pricing and load control strategies on residential summer peak electricity use: A review," *Energy Policy*, vol. 38, no. 7, pp. 3289–3296, Jul. 2010.

[5] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A survey on demand response programs in smart grids: Pricing methods and optimization algorithms," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 152–178, 1st Quart., 2015.

[6] J. Kiljander, D. Gabrijelcic, O. Werner-Kytola, A. Krpic, A. Savanovic, Z. Stepancic, V. Palacka, J. Takalo-Mattila, and M. Taumberger, "Residential flexibility management: A case study in distribution networks," *IEEE Access*, vol. 7, pp. 80902–80915, 2019.

[7] F. Ruelens, S. Iacovella, B. Claessens, and R. Belmans, "Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning," *Energies*, vol. 8, no. 8, pp. 8300–8318, Aug. 2015.

[8] E. Barrett and S. Linder, "Autonomous HVAC control, a reinforcement learning approach," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Cham, Switzerland: Springer, 2015, pp. 3–19. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-23461-8_1#citeas

[9] R. Jia, M. Jin, K. Sun, T. Hong, and C. Spanos, "Advanced building control via deep reinforcement learning," *Energy Procedia*, vol. 158, pp. 6158–6163, Feb. 2019.

[10] K. Amasyali and N. M. El-Gohary, "A review of data-driven building energy consumption prediction studies," *Renew. Sustain. Energy Rev.*, vol. 81, pp. 1192–1205, Jan. 2018.

[11] T. Salmi, J. Kiljander, and D. Pakkala, "Stacked boosters network architecture for short-term load forecasting in buildings," *Energies*, vol. 13, no. 9, p. 2370, May 2020.

[12] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, "Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system," *Energy Buildings*, vol. 141, pp. 96–113, Apr. 2017.

[13] Z. Zhang, A. Chong, Y. Pan, C. Zhang, S. Lu, and K. P. Lan, "A deep reinforcement learning approach to using whole building energy model for HVAC optimal control," in *Proc. Building Perform. Modeling Conf.*, 2018, pp. 1–9.

[14] A. Jain, F. Smarra, E. Reticcioli, A. D'Innocenzo, and M. Morari, "NeurOpt: Neural network based optimization for building energy management and climate control," in *Proc. 2nd Conf. Learn. Dyn. Control*, vol. 120, 2020, pp. 445–454.

[15] P. Pääkkönen, D. Pakkala, J. Kiljander, and R. Sarala, "Architecture for enabling edge inference via model transfer from cloud domain in a kubernetes environment," *Future Internet*, vol. 13, no. 1, p. 5, Dec. 2020.

[16] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, doi: 10.1038/nature16961.

[17] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, Shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018.

[18] K. Känsälä, "Control power matrix calculation: A new way to increase controllable resources in the grid," in *Proc. 19th Int. Conf. Energy Environ. Eng. (ICEEE)*, 2017.

[19] S. Karatasou, M. Santamouris, and V. Geros, "Modeling and predicting building's energy use with artificial neural networks: Methods and results," *Energy Buildings*, vol. 38, no. 8, pp. 949–958, Aug. 2006.

[20] J. Yang, H. Rivard, and R. Zmeureanu, "On-line building energy prediction using adaptive artificial neural networks," *Energy Buildings*, vol. 37, no. 12, pp. 1250–1259, Dec. 2005.

[21] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2016, pp. 7046–7051.

[22] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.

[23] E. Mocanu, P. H. Nguyen, M. Gibescu, and W. L. Kling, "Deep learning for estimating building energy consumption," *Sustain. Energy, Grids Netw.*, vol. 6, pp. 91–99, Jun. 2016.

[24] J. Kim, J. Moon, E. Hwang, and P. Kang, "Recurrent inception convolution neural network for multi short-term load forecasting," *Energy Buildings*, vol. 194, pp. 328–341, Jul. 2019.

[25] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 1310–1318.

[26] B. C. Ng, I. Z. M. Darus, H. Jamaluddin, and H. M. Kamar, "Application of adaptive neural predictive control for an automotive air conditioning system," *Appl. Thermal Eng.*, vol. 73, no. 1, pp. 1244–1254, Dec. 2014.

[27] A. Kusiak, G. Xu, and Z. Zhang, "Minimization of energy consumption in HVAC systems with data-driven models and an interior-point method," *Energy Convers. Manage.*, vol. 85, pp. 146–153, Sep. 2014.
[28] G. Wu, B. Say, and S. Sanner, "Scalable planning with deep neural network learned transition models," *J. Artif. Intell. Res.*, vol. 68, pp. 571–606, Jul. 2020.
[29] A. Garnier, J. Eynard, M. Caussanel, and S. Grieu, "Predictive control of multizone heating, ventilation and air-conditioning systems in non-residential buildings," *Appl. Soft Comput.*, vol. 37, pp. 847–862, Dec. 2015.
[30] W. Kim, Y. Jeon, and Y. Kim, "Simulation-based optimization of an integrated daylighting and HVAC system using the design of experiments method," *Appl. Energy*, vol. 162, pp. 666–674, Jan. 2016.
[31] A. Kusiak, M. Li, and F. Tang, "Modeling and optimization of HVAC energy consumption," *Appl. Energy*, vol. 87, no. 10, pp. 3092–3102, Oct. 2010.
[32] X. He, Z. Zhang, and A. Kusiak, "Performance optimization of HVAC systems with computational intelligence algorithms," *Energy Buildings*, vol. 81, pp. 371–380, Oct. 2014.
[33] R. Coulom, "Efficient selectivity and backup operators in Monte-Carlo tree search," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Berlin, Germany: Springer, 2007, pp. 72–83.
[34] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR) Conf. Track*, 2015, pp. 1–15.
[35] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo planning," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Berlin, Germany: Springer, 2006, pp. 282–293.

JUSSI KILJANDER received the M.Sc. (Technology) and Ph.D. (Technology) degrees from the University of Oulu, Oulu, Finland, in 2010 and 2016, respectively. He is currently a Senior Research Scientist with the VTT Technical Research Centre of Finland, Oulu. He has been with VTT, as a Research Trainee since 2009, a Research Scientist since 2010, and a Senior Research Scientist since 2019. He has been a project manager, technical coordinator, and work package leader in several EU and national projects. He has authored over 30 scientific publications. His current research deals with machine learning, autonomous systems, and smart grids.

ROOPE SARALA received the M.Sc. degree from the University of Jyväskylä, Finland, in 2016. He is currently a Research Scientist with the VTT Technical Research Centre of Finland, Oulu, Finland. Since 2019, he has been working with VTT, in various projects in the smart energy research area. His main expertise is in machine learning and optimal control.

JARI REHU received the B.E. degree in information technology and embedded systems from the Oulu University of Applied Sciences, Finland, in 1993. He is currently a Senior Research Engineer with the VTT Technical Research Centre of Finland, Oulu, Finland. He has been with VTT, as a Research Engineer since 1993 and as a Senior Research Engineer since 2005. He has been a system designer and a software and electronics designer in several EU and national projects. His current research deals with smart building automation, energy systems, and machine learning in building control.

DANIEL PAKKALA received the Ph.D. degree in information processing science. He is currently working as a Principal Scientist with the VTT Technical Research Centre of Finland. He has authored over 40 scientific publications, and his current research interests include design, architecture, and engineering of digital service and systems, machine intelligence, and value creation in digital transformation of industries.

PEKKA PÄÄKKÖNEN received the M.Sc. degree in information technology from the Department of Electrical and Information Engineering, University of Oulu, in 2002, and the second M.Sc. degree in economics and business administration from the Oulu Business School, University of Oulu, in 2015. He is currently working as a Senior Research Scientist with the Machine Intelligence Team, VTT Technical Research Centre of Finland. His research has focused on network mobility and ad hoc networking protocols, mobile application design and development, energy optimization techniques for mobile phones, big data architectures and related technologies, machine learning, quality of social media data, and cloud management architectures.

JANNE TAKALO-MATTILA received the M.Sc. (Technology) degree from the University of Oulu, in 2009. He is currently a Research Scientist with the VTT Technical Research Centre of Finland. His current research interests include software development and machine learning in the context of energy, process, and manufacturing industries.

KLAUS KÄNSÄLÄ was born in 1961. He received the M.Sc. (Tech.) and M.B.A. degrees from the University of Oulu. He has been working with the VTT Technical Research Centre of Finland, since 1989, where he is currently a Principal Scientist. He worked as a project engineer, expert, and product development manager in numerous domestic and international projects. He has gained a solid expertise on wireless metering, embedded systems, and onboard DSP solutions. He has also been developing wireless metering solutions for smart grid applications and advised corporations and public organizations on energy markets, energy technologies, demand-responsive concepts, and renewable production.