Green Planning of IoT Home Automation Workflows in Smart Buildings

SOTERIS CONSTANTINOU, University of Cyprus
ANDREAS KONSTANTINIDIS, Frederick University
PANOS K. CHRYSANTHIS, University of Pittsburgh
DEMETRIOS ZEINALIPOUR-YAZTI, University of Cyprus

The advancement of renewable energy infrastructure in smart buildings (e.g., photovoltaic) has highlighted the importance of energy self-consumption by energy-demanding IoT-enabled devices (e.g., heating/cooling, electromobility, and appliances), which refers to the process of intelligently consuming energy at the time it is available. This stabilizes the energy grid, minimizes energy dissipation on power lines but more importantly is good for the environment as energy from fossil sources with a high CO_2 footprint is minimized. On the other hand, user comfort levels expressed in the form of *Rule Automation Workflows (RAW)*, are usually not aligned with renewable production patterns. In this work, we propose an innovative framework, coined *IoT Meta-Control Firewall (IMCF$^+$)*, which aims to bridge this gap and balance the trade-off between comfort, energy consumption, and CO_2 emissions. The IMCF$^+$ framework incorporates an innovative *Green Planner (GP)* algorithm, which is an AI-inspired algorithm that schedules energy consumption with a variety of amortization strategies. We have implemented IMCF$^+$ and GP as part of a complete IoT ecosystem in openHAB and our extensive evaluation shows that we achieve a CO_2 reduction of 45–59% to satisfy the comfort of a variety of user groups with only a moderate ≈3% in reducing their comfort levels.

CCS Concepts: • **Information systems → Information systems applications**; • **Computing methodologies → Artificial intelligence**; • **Applied computing → Computers in other domains;**

Additional Key Words and Phrases: Smart IoT, green energy planning, renewable energy self-consumption

ACM Reference format:
Soteris Constantinou, Andreas Konstantinidis, Panos K. Chrysanthis, and Demetrios Zeinalipour-Yazti. 2022. Green Planning of IoT Home Automation Workflows in Smart Buildings. *ACM Trans. Internet Things* 3, 4, Article 29 (September 2022), 30 pages.
https://doi.org/10.1145/3549549

1 INTRODUCTION

The Paris Agreement within the United Nations Framework Convention on Climate Change, dealing with greenhouse-gas-emissions mitigation, adaptation, and finance, signed in New York City, on April 22, 2016, aimed to strengthen the global response to the threat of climate change, since we

Authors' addresses: P. K. Chrysanthis, University of Pittsburgh, Pittsburgh, PA 15260; email: panos@cs.pitt.edu; S. Constantinou and D. Zeinalipour-Yazti, University of Cyprus, Nicosia 1678, Cyprus; emails: sconst01@cs.ucy.ac.cy, dzeina@ucy.ac.cy; A. Konstantinidis, Frederick University, Nicosia 1036, Cyprus; email: com.ca@frederick.ac.cy.

2577-6207/2022/09-ART29 $15.00
https://doi.org/10.1145/3549549

are witnessing a steady increase in CO_2 since the Industrial Revolution (1760–1840 AD). Additionally, the cost of polluting, regarding power generation, has increased more than 140% in 2021[1] after a stricter environmental agenda in Europe was laid out, along with a sweeping rally in natural gas prices. Besides all that, natural gas will only reduce 1/3 of the terrible emission image.

In the long-term, high carbon prices could accelerate the investment in equipment, developing intelligent software to decrease the level of emissions, or switch to cleaner fuels. Technologies such as Green Hydrogen production from renewable energy or carbon capture and storage, become more economically viable in case that the carbon price remains at or above current levels. By 2030, the European's ambition is to produce 10 million tones of renewable hydrogen that is expected to significantly reduce the CO_2 emissions [1]. When the energy used to power electrolysis comes from renewable sources it is called Green Hydrogen, and this approach can be effectively used as a future step mainly anticipated to replace humongous mobile batteries (in airplanes, ships, large lorries, etc.).

On the other hand, self-consumption of renewable energy remains complementary to nowadays and future requirement for a cleaner environment. Particularly, it constitutes a distributed in-situ approach that does not require enormous infrastructure but rather only intelligent planning algorithms for the CO_2 reduction and is shown to achieve more than 70% for a domestic household. Consequently, minimizing the CO_2 pollution in spaces where the human is active (e.g., houses and offices) in which people spend 80–90% of their time, can positively impact the environment.

Given that energy is produced in a variety of manners (fossil, renewable, nuclear, etc.), the impact on the environment is typically measured in kg CO_2 emitted per kWh of energy produced.[2] In countries with a high kg CO_2 per kWh factor, this effectively reduces CO_2 pollution but also contributes to the stabilization of the energy grid. In Table 1, we can see the CO_2 emissions due to electricity generation supplied by the European Environment Agency (Eurostat). The CO_2 emission intensity (kg CO_2) is calculated as the ratio of CO_2 emissions from public electricity production (as a share of CO_2 emissions from public electricity and heat production related to electricity production), and gross electricity production. On average, we see that most countries have still a long way for becoming CO_2 neutral and that this is an exciting problem space to seek for novel contributions. In Table 2, we can see the CO_2 emissions produced by return flights according to figures from the German non-profit organization "Atmosfair" [2]. The figures are averages considering which aircraft models are typically used on flight routes, and the estimated occupancy of seats on board those planes.

A key driver for the control of CO_2 is the uptake of ***Internet of Things (IoT)***, which connect all the smart devices in the world that can "see", "hear", "think", "react", perform tasks as well as communicate with each other using open protocols [3–6], and thus, power consumption and CO_2 emissions controlled by IoT infrastructure can be brought under the same roof. IoT enables the development of smart applications in important domains, such as transportation, healthcare, industrial automation, emergency response, and business, having significant impact on the quality of people's life and the growth of the world's economy and security [5]. Studies showed that IoT connected devices worldwide is projected to amount to 30.9 billion units by 2025,[3] including smart cars, home devices, industrial equipment, and so on, and later on to 100 billion connected devices by 2030 [7].

In our previous works [8, 9], we presented the design and preliminary results of an innovative system, coined **IoT Meta-Control Firewall** (**IMCF**), which aims to schedule comfort preferences

[1]Bloomberg Green., URL: https://tinyurl.com/yxewwpzm.
[2]For the remainder of this work, we denote the more typical metric of kg CO_2-eq(uivalent) with only kg CO_2.
[3]Statista., URL: https://tinyurl.com/mw74ku2h.

Table 1. The European Environment Agency CO_2 Emission Intensity for EU-27 and USA-Average

COUNTRY	kg CO_2 per kWh
Sweden	0.013
Lithuania	0.018
France	0.059
Austria	0.085
Latvia	0.105
Finland	0.113
Slovakia	0.132
Denmark	0.166
Belgium	0.17
Croatia	0.21
Luxembourg	0.219
Slovenia	0.254
Italy	0.256
Hungary	0.26
Spain	0.265
Romania	0.306
Portugal	0.325
Ireland	0.425
Germany	0.441
Bulgaria	0.47
Netherlands	0.505
Czech Republic	0.513
Greece	0.623
Malta	0.648
Cyprus	0.677
Poland	0.773
Estonia	0.819
COUNTRY	**kg CO_2 per kWh**
EU-27 (average)	0.296
USA (average)	0.449

Table 2. The Carbon Dioxide (CO_2) Emissions Produced by Return Flights According to "Atmosfair" [2]

Flying From	Flying To	kg CO_2
Paris	Frankfurt	115
Frankfurt	London	138
Sydney	Melbourne	165
London	Rome	234
New York City	Paris	922
London	New York City	986
Moscow	Washington D.C.	1,383
Los Angeles	London	1,650
Brussels	Rio de Janeiro	1,756
Athens	Brasilia	1,783
Barcelona	Shanghai	1,844
Perth	Athens	2,530
London	Perth	3,153

of user in smart buildings (expressed in the form of so-called Rule Automation Workflows—RAW), such that long term energy objectives can be meet (e.g., consume less than 400 kWh in December). We presented the **Energy Planner** (**EP**) algorithm that takes care of the scheduling using primitive amortization strategies. Our previous work was however, unfortunately agnostic of the climate impact of the RAW automation process. In this work, we present the **IoT Meta-Control Firewall** (**IMCF$^+$**) framework, which is inspired by the advancement of renewable energy infrastructure in smart buildings (e.g., photovoltaic) that has highlighted the importance of energy

self-consumption by energy-demanding IoT-enabled devices (e.g., heating/cooling, electromobility, and appliances). Self-consumption refers to the process of intelligently consuming energy at the time it is available. This stabilizes the energy grid, minimizes energy dissipation on power lines but more importantly is good for the environment as energy from fossil sources with a high CO_2 footprint is minimized.

Particularly, in IMCF$^+$ a user (or group of users) start out by defining a vector of RAW rules, coined *MRT*, and an *Energy Consumption Profile*, coined *ECP* (see Figure 1). The high-level objective is to identify among all MRT rules the ones that must be dropped so that the user stays within the desired energy budget according to the *ECP* history. For this purpose, it utilizes an intelligent search algorithm, which goes over the exponentially large search space of $\sum_{r \leq n} r$-combinations (where $n = |MRT|$), quickly yielding the rules to be dropped. Particularly, IMCF$^+$ adopts an intelligent energy amortization process along with an AI-inspired **Green-Planner** (**GP**) algorithm we propose, to balance the trade-off between user comfort and CO_2 emissions subject to pre-specified energy consumption budget while satisfying the RAW pipelines of users. IMCF$^+$ adapts the RAW pipelines in a way that these do not collide with the long-term objectives of users (by dropping certain rules based on preference priority).

The RAW pipelines are distinguished in our discussion into the *comfort rules* and *necessity rules*. The *comfort rules* aim at promoting an individual's physical convenience (e.g., room temperature, ambient lighting, pre-heating of car, operation readiness of general appliances or whatever is considered tentative comfort to an individual), while the *necessity rules* are those rules that should always be executed regardless of whether the long-term target is met. For ease of exposition, consider only the comfort rules, sorted in order of importance, for the remainder of this work.

In respect to processing the RAW rules, one could ignore the RAW rules completely, obtaining in this way the best energy consumption (thus, low CO_2 emission) but the worst comfort (we call this the ***No Rule (NR)*** method–see Figure 1). In contrast, a user could obtain maximum comfort by having every single preference rule inside RAW executed that would obviously bring the highest comfort but at the same time also consume the highest amount of energy and consequently produce high CO_2 emissions (we call this the ***Meta-Rule (MR)*** method). We assume that a user sets, for example, the temperature/illumination to his/her most preferable level, for his/her maximum comfort. The IFTTT approach, in the absence of a detailed user preference profile MRT, being an arbitrary sequence of rule executions would then be somewhere in between these two borderline cases, while GP is a more well-rounded version of arbitrary IFTTT rules.

We claim that by consuming energy more intelligently (i.e., green-smart IoT actuations) can greatly contribute to the environmental impact of ICT enabling us to improve living conditions and respect the environment reaching agreed targets. We assume that IMCF$^+$ can be adopted in smart environments equipped with net-metering or net-billing PV or wind systems where self-consumption translates to an actuation in the physical space. Having different PV system sizes, means different production patterns. Thus, the historical user data incorporated in the algorithm and problem formulation will be proportional to the PV system. This also results to the proportional allocation and distribution regarding hourly upper bounds and energy consumption accordingly. To understand this desideratum, consider two separate examples in a green-home and green-dormitory setting, respectively.

Green-Home: As a first example consider a single family that has invested in photovoltaic technology to cover its heating/cooling, mobility, and other energy requirements. In our scenario, the family has a yearly budget of 8,500 kWh (i.e., yearly production of this household under a net-metering scheme, where energy excess on a sunny day can be used at later stages within a yearly cycle) and aims to spend this energy budget through a RAW that configures the energy consumption preferences of the family (e.g., room temperatures across the year in the house as well as

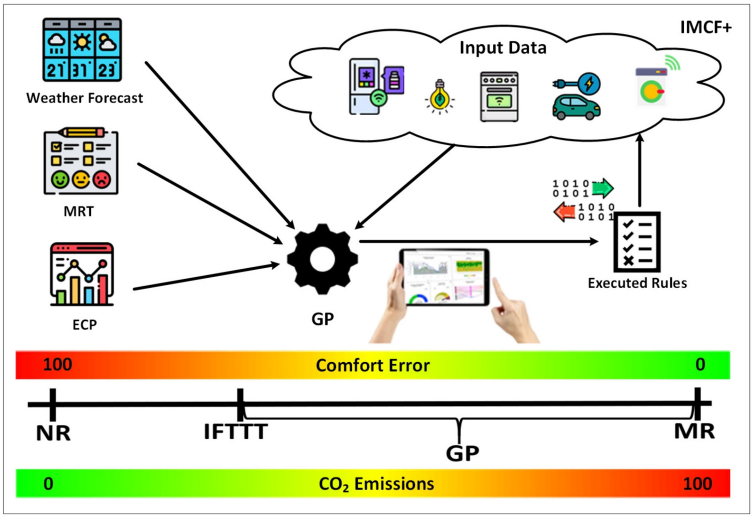

Fig. 1. The *GP* algorithm proposed in this work is an AI-inspired algorithm that finds the best possible energy consumption strategy with respect to user comfort and CO_2 emissions by only using a *MRT* profile, a *Weather Forecast,* and an ***Energy Consumption Profile (ECP)*** and without the necessity of a learning history used by ML methods.

auxiliary lighting). The family is willing to adapt its desired interior temperature preferences (e.g., adapting indoor temperature by 1°C, somebody can save 6% of its energy consumption[4]) according to production and consumption patterns but has no clue how the RAW pipelines contribute to the target of using only 8,500 kWh per year. Currently, they rely on manual guess-work and manual planning that is cumbersome and error-prone [10]. Section 2 overviews the related work extensively, showing that no other solution is available to the problem under investigation.

Green-Dorms: As a second example consider the SAVES [11] project, which was an inter-dormitory energy-saving competition within the framework of the European Commission Intelligent Energy–Europe (IEE) program that took place between 2014–2016. The project aimed to achieve energy-saving habits by students at a key moment of change in their lives so that they can continue energy-saving actions throughout their private lives. SAVES aimed at delivering 8% average electricity savings in participating dormitories. In the case of University of Cyprus students, the task was undertaken with great excitement and passion that eventually led to a saving of only 4,44%. Even though students applied common sense and perseverance in achieving the energy reduction target, there was a lack of intelligent control to reach the higher desired target.

We expose how we have integrated IMCF$^+$ into the readily available openHAB IoT stack bringing in this way optimal integration and compatibility prospects (as the complete IoT ecosystem bindings are already readily available). In this article, we have the following contributions:

— We propose a novel notion of filtering RAW workflows using the IMCF$^+$ firewall that is formally defined. In this scope, we propose the design and implementation of the **Green Plan (GP)** algorithm that has the ability to handle the user's comfort profile by considering the user's energy budget and the carbon dioxide emissions.
— We present a complete system architecture of our IMCF$^+$ green energy management system implemented inside the openHAB stack.

[4]U.S. Dept. of Energy. http://tiny.cc/qbosuz.

— We evaluate our design with extensive experimentation on real datasets with weather forecast data from OpenWeatherMap, and measurements from a real residential apartment that comprises of a variety of sensors and approximately 5M readings (1.09 GB in total), showing that GP can be premise for energy-aware and CO_2-aware green actuations in the future. We finally also demonstrate the utility of our pioneer system.

The remaining of the article is organized as follows: Section 2 presents the background and other related work. Section 3 provides our system model and formulates the problem. Section 4 presents our proposed framework and its internal components. Section 5 presents our complete system architecture proposition while Section 6 presents our experimental methodology and results. Section 7 concludes the article.

2 BACKGROUND AND RELATED WORK

In this section, we provide background on the studied subject of IoT rule automation workflows and also provide the related work.

2.1 IoT Data Management

The uptake of IoT in recent years has brought a revived interest on data management and data engineering solutions, architectures and applications with a focus on data ingestion [12], analytic architectures for streaming data [13] as well as relevant benchmarking [14]. From the application perspective, a specific focus has been given to privacy [15], context awareness [16], temporal analytics [17], localization [18, 19], and telco big data [20]. Green Data Management has been a complementary and related topic with intensive research over the years, particularly in data centers [21–25] and data warehouse design [26, 27], green-aware route planning in GIS systems [28–31], smart grids [32–36], but the focus on green IoT actuation application frameworks has been overlooked over the years.

2.2 Rule Automation Workflows (RAW)

In this subsection, we cover complementary work of RAW pipelines and the competing approaches to achieve the exploration of the RAW search space.

Preamble: Beside data collection, many IoT devices also enable the execution of ***Rule Automation Workflows (RAW)***, which span from simple predicate statements to procedural workflows aiming to capture a smart actuation pipeline in tools like IFTTT [37], which controls Philips Hue lights, BMW i3 EVs or Daikin A/C units [38, 39], Apilio.io, or Apple Automation [40]. RAW aim to meet the convenience level of users under specific conditions (e.g., *"warm house to 22°C if cold or preheat EV when approaching"*). In the simplest case, a user expresses preferences manually through a vendor-specific smartphone app or an integrated app. This process requires attention by custodians, exposing distraction hazards, calling for more automated (i.e., *"smarter"*) approaches.

One of the most straightforward approaches to achieve a smarter RAW is to do so with the so-called *trigger-action* model. Users control the behavior of an IoT by specifying triggers (e.g., *"if it is sunny outside"*) and their resultant actions (e.g., *"turn off the lights"*). Because of its conceptual simplicity, the trigger-action model has attracted significant attention with IFTTT [37] (*"If This Then That"*) becoming one of the first large-scale deployments. Services like Apilio [41] expanded the expressiveness of the RAW with Boolean predicates (e.g., conjunctions) and Apple Automation [40] even introduced procedural programming constructs, like variables, while loops, if statements and functions to bring RAW smart actuations to new levels.

Real-time IFTTT: Heo et al. [42] implemented RT-IFTTT, a real-time IoT language and its framework that uses trigger condition-aware flexible sensor polling intervals. The RT-IFTTT

language extends the existing IFTTT syntax and allows users to specify real-time constraints for their applets. Again, this system does not enable long term energy planning and it does not also allow green planning.

RAW Informed Search Methods: are generally characterized by a *utility* in scanning the solution space to reach a goal. These algorithms utilize an evaluation function that greedily assesses some distance of the current state to the target state (e.g., in the case of Best-first search) and the least cost incurred to reach the current state (e.g., in the case of *A* heuristic* search). Unfortunately, A*-search always requires some evaluation function that is not available in our case as we really do not know the comfort target of a user within the agreed energy budget. As such, we must rely on stochastic informed search algorithms (e.g., *simulated annealing* and *hill climbing*), which probabilistically carry out a similar task but without requiring a rigid target function. The GP algorithm proposed in this work, is founded on *simulated annealing* space exploration method that deploys a user-controlled energy amortization strategy and domain heuristics to bring forward the expected result.

RAW Machine Learning and Privacy: Another important point is that the GP algorithm does not deploy **Machine Learning** (**ML**) techniques, such as ***Artificial Neural Networks (ANNs)*** or variants [43], given that these methods require a lot of training data that is not available in our case due to privacy reasons. Training data for user habits and preferences can be privacy-sensitive. We feel that this can pose a serious imminent privacy threat, given that smart-environment controllers like IMCF are many times private enterprises that are less controlled, thus they might be tempted to exploit the "big" behavior data of their customers, by either selling it to advertising companies or by linking it to other sensitive data.

In summary, none of the above RAW technologies enables individuals or group of users to express their comfort preferences while achieving some long-term energy objective while curbing emissions.

2.3 Smart Energy Management Systems

In this subsection, we overview energy management systems for three different contexts, stemming both from the industrial and academic sectors.

Photovoltaic Home Energy Management: The Sunny **Home Manager** [44] (**HM**) controllers by SMA monitors power flows, particularly the production of AC power from the inverters and the consumption of AC power from the households (recorded by an energy meter). HM then manages the power consumption workloads accordingly (e.g., when to operate a washing machine or smart car charger so that solar energy self-consumption is optimized). This is achieved with its open ***Simple Energy Management Protocol (SEMP)*** or the industry-wide adopted EEBUS [45] protocols with its KEO reference implementation. However, these protocols are geared for load management inside smart buildings rather than for enabling users achieve some long-term energy (energy consumption) targets and restricting CO_2 emissions as we do in our work. As such, these energy HMs have a complementary role to the energy planning propositions we present in this article.

Smart Thermostats: The Nest.com Learning Thermostat is a programmable and self-learning Wi-Fi-enabled thermostat that optimizes cooling and heating to conserve energy. However, there are the following differences with IMCF$^+$: (i) these thermostats do not enable the adaptation of comfort preferences to meet the long-term energy planning targets of individuals or group of individuals (see examples in Section 1) considering at the same time the carbon dioxide emissions; and (ii) these require learning data from users (e.g., location) that might be a privacy concern.

Smart Homes Energy Predictions: There is general research in the sphere of energy prediction of smart homes. Particularly, in [46] the authors predict user behaviors and designed a protection method to avoid privacy threads. Yang et al. [47] proposed an intelligent smart home

Table 3. Notation used Throughout this Work

Notation	Description
j, D, e_j	IoT device j, Set of all j, energy consumption of j
γ, Γ_j	CO_2 emission intensity per country, CO_2 emission of device j
MR_i^j, MRT, N	MR i for j, Set of all MR_i^j, $N = \|MR\|$
I, I_i^j	Input data, attribute value of input data
O, O_i^j	Output data, attribute value of output
p, t	Execution Period, time granularity
ECP	Energy Consumption Profile

energy management scheme which supports context-aware service that allows users to do customized configurations and offers energy usage modes, like general mode, power-saving mode, and economic mode to save the energy efficiently. The work in [48] describes an ongoing attempt in creating a smart IoT desk that can improve the occupant's satisfaction with the environment, their health, and productivity by personalizing the environment based on their monitored preferences.

Smart zoning: The approach to dynamically regulate the set points of thermostats in every room at different levels according to geometry, orientation, and interaction among rooms caused by occupancy patterns, refers to smart zoning. The research conducted in [49, 50], frames the problem of load management with smart zoning into a multiple-mode feedback-based optimal control problem, which refers to embedding multiple behaviors (triggered by building-occupant dynamic interaction) into the optimization problem, with closed-loop control strategies using information stemming from building and weather states. The authors' framework makes it is possible to save more than 15% energy consumption, with 25% increased thermal comfort.

3 SYSTEM MODEL AND PROBLEM FORMULATION

This section formalizes our system model, assumptions and problem. To exemplify our terminology, we use examples from a smart green-home setting. Table 3 summarizes our notation.

3.1 System Model

Consider a smart green-home composed of D IoT devices (e.g., Daikin A/C split units) monitored by a cloud-based, vendor-specific controller *Service* (e.g., Daikin Cloud Service) for defining their behavior based on a particular context (e.g., temperature, humidity). The IoT devices j consume e_j energy each time they operate under a particular mode of operation. For example, a split unit consumes around $e_j = 833$ watts (i.e., 0.833 kWh) when started but on average not more than $e_j = 318$ watts (i.e., 0.318 kWh), which according to Table 1 for the European Union with an emission intensity $\gamma = 0.296$ kg CO_2/kWh, it produces about $\Gamma = e_j \times \gamma = 0.95$ kg CO_2 emissions, on average. We assume, however, that the smart green-home is equipped with a net-metering photovoltaic system that allows the resident to consume the generated energy in the house, and request energy from the grid (and therefore produce CO_2 emissions) only when no energy is generated from the photovoltaic system (i.e., when there is no sunlight).

We also assume that a user has identified a set of MRs MR_i^j for each device $j = 1, \ldots, D$, which is recorded with a *Meta-Service*, such as the IMCF$^+$ service we propose in this work. Particularly, all MRs are stored centrally on a **meta-rule table** (MRT) $MRT = \{MR_i^j | i \geq 0\}$, and *Meta-Service* takes care to periodically, i.e., every t time steps in an overall execution period p, execute these rules on the IoT devices through *Service*. Each MR MR_i^j obviously relies on a particular input context

Table 4. The *MRT*

Meta-rule Description	Input Values	Output Values
MR_1^1: If outdoor temp. > 35°C then switch-on AC in room	$I_1^1 \in \Re$	$O_1^1 \in \{0,1\}$
MR_2^1: If summer then set AC in room at 20°C	$I_2^1 \in \{0,1\}$	$O_2^1 \in \Re$
MR_3^2: If daytime turn lights off	$I_3^2 \in \Re$	$O_3^2 \in \{0,1\}$
Constraint: My monthly CO_2 emission to NOT exceed 118kg (i.e., ≈400kWh)	$I_2 =$ *consumption*()	$O_2 = \{O_1^1, \ldots, O_N^M\}$

(e.g., temperature from weather channel, or indoor temperature of the A/C split unit, or outdoor temperature of the split unit fan, or user location, to name a few examples) and we will coin these I_i^j. The MR_i^j rule execution generates at every discrete time point an output O_i^j, which defines the action to be executed on device j.

Table 4 exemplifies some $MR_i^j \in MRT$ that satisfy a user's preference rules along with the long-term objective. For example, the constraint states *"Keep my monthly CO_2 emission below 118 kg"*, which is approximately ≈100 euros or ≈400 kWh (e.g., 1 kWh costs around 0.296 kg CO_2 in EU-27 and around 0.449 kg CO_2 in USA—see Table 1—so energy conversion to CO_2 emission is carried out directly). The incorporation of multiple rules may cause several deficiencies, such as rules competing or throwing a clash with each other, rules becoming infeasible to be satisfied and/or rules that their behavior depends on the output of other rules. This is mainly due to the inability of current controllers to autonomously track and monitor a high number of rules that may be set by the user in different periods, under different circumstances.

3.2 Research Goal

Design an intelligent algorithm that enables some user to find an energy-efficient plan for the execution of a set of actuation rules encoded in MRT and a tentative energy consumption history ECP, satisfying several objectives subject to a specific CO_2 emission constraint.

The efficiency of the proposed techniques to achieve the above research goal is measured by the following metrics: (i) the *Comfort Error*; (ii) the *Energy Consumption* required for finding a near-optimal plan of MRs; and (iii) *CO_2 Emission* required to execute the research goal.

Definition 3.2.1. **Comfort Error** $CE_j(MR_i)$ is the difference between the desired output value $\Omega_i^j \in \Re$ of a rule set by a user and the actual value $O_i^j \in \Re$ set by the controller, given by: $ce = |\Omega_i^j| - |O_i^j|$. The comfort can be defined individually by rules related to temperature, humidity, and illumination or a combination of those. Please note that this research study has adopted particular smart sensors for the sake of experimentation, however, the proposed framework can easily adopt any kind of smart sensor, which corresponds to comfort, with minor modifications.

Definition 3.2.2. **Energy Consumption** $E_j(MR_i)$ is the energy consumption of device j given the action defined by output O_i^j of MR MR_i, given by:

$$E_j = \begin{cases} e_j, & \text{if } O_i^j \text{ is executed} \\ 0, & \text{otherwise} \end{cases},$$

where e_j is the energy cost of device j for MR MR_i.

Definition 3.2.3. **CO_2 Emission** $\Gamma_j(E_j, \gamma)$ is the CO_2 emission produced by the actuation of device j given the energy consumption E_j as well as the CO_2 emission intensity γ of a particular country, given by:

$$\Gamma_j = \begin{cases} E_j \times \gamma, & \text{if device } j \text{ operates} \\ 0, & \text{otherwise} \end{cases},$$

where E_j is given by Definition 3.2.

Both the comfort $ce_j(MR_i)$, the energy consumption $E_j(MR_i)$ and the CO_2 emission $\Gamma_j(E_j, \gamma)$ functions, are repeated every t seconds (e.g., hourly, daily, monthly, and yearly preference) over a time period p (i.e., the complete duration of the execution). Our research goal can be expressed as follows:

$$\min F_{CE} = \sum_{k=1}^{t} \left(\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{D} CE_j(MR_i) \right) \tag{1}$$

subject to $F_E \leq E_p$, and $F_\Gamma \leq \Gamma_p$ where:

$$F_E = \sum_{k=1}^{t} \left(\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{D} E_j(MR_i) \right), \tag{2}$$

E_p is total available energy budget for the complete period p during which the execution of our algorithm takes place,

$$F_\Gamma = \sum_{k=1}^{t} \left(\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{D} \Gamma_j(E_j(MR_i, \gamma)) \right) \tag{3}$$

and Γ_p is the maximum desired CO_2 emission for the complete period p during which the execution of our algorithm takes place.

3.3 Baseline Approaches

There are two extreme scenarios that can be considered as the baselines for our proposition, the one guaranteeing minimum CO_2 emission and maximum comfort error, and the other minimum comfort error and maximum CO_2 emission. *NR*: takes into consideration NRs and therefore it does not modify the behavior of the autonomous devices. This conflicts with the user's comfort level. Consequently, the energy consumption and consequently the CO_2 emission of this approach is minimum and the comfort error is maximum. *MR*: is a greedy approach that ignores the CO_2 emission and triggers all actions for satisfying all MRs set by the user. Consequently, the energy consumption and consequently the CO_2 emission of this approach is maximum and the comfort error is minimum.

4 THE IOT META-CONTROL FIREWALL ($IMCF^+$)

In this section, we detail the internal phases of the $IMCF^+$ framework, followed by an example of its operation and analysis.

4.1 Outline of Operation

The $IMCF^+$ framework (presented in Algorithm 1) is composed of two subroutines: (i) the ***Amortization Plan (AP)***; and (ii) the *GP*. The combination of the two, forms the energy management process, and the operation of each is described in the following subsections. The AP is responsible for calculating the maximum energy budget constraint (coined E_p) and the CO_2 emission constraint

(F_Γ) through a pre-selected amortization formula. Then an AI approach is executed for generating a GP solution s^* for optimizing the comfort error $F_{CE}(s^*)$ subject to satisfying the $F_E(s^*) \leq E_p$ and $F_\Gamma(s^*) \leq \Gamma_p$ constraints. In this article, we have adopted a simulated annealing heuristic, which does not require a learning history (like respective ML techniques), does not require a target function (e.g., like A*), it does not get stuck in local optima (e.g., like traditional hill-climbing) and it is straightforward to be implemented in a resource-constraint setting like local smart controllers (e.g., Raspberry).

In our research study, the real monthly consumption patterns indicated on Table 5 are considered as the maximum energy consumption of a household per month within a year. This max monthly energy is then transformed to hourly max energy consumption based on each month's overall consumption and a particular amortization formula. Consequently, the energy will not be equally shared in every hour of the year, since there is a different energy consumption pattern per month. Therefore, there will be a difference between the available hourly energy consumption for each month. The APs, e.g., the Linear Amortization formula, provide upper bounds (constraints the max energy consumption) that can be consumed hourly at each month, but it does not set the actual energy that will be consumed. The latter is the GP algorithm's responsibility to decide how much energy will be consumed, which should be less than the maximum budget set by the amortization formulas. By satisfying the constraint (that is, not consuming more energy hourly than the max calculated by the amortization formulas) will consequently guarantee that the monthly energy consumption and consequently the yearly energy consumption will be less than the real consumption patterns of Table 5.

4.2 Amortization Plan (AP) Algorithm

The *AP*() subroutine is initially executed for calculating the energy budget constraint E_p subject to a monthly residence *Energy Consumption Profile ECP*, such as the one exemplified in Table 5 for a flat. The hourly energy budget constraint is referring to the maximum upper bounds per hour, therefore it does not necessarily mean that for every hour we will be using the entire available budget. The AP() is responsible to set the hourly upper consumption bounds (hourly energy budget constraint), and later the GP() will handle the actual hourly energy allocation. Moreover, the AP() algorithm brought forward in this work does not simply translate kWh to CO_2 emissions, but rather uses a weather dataset along with heuristics to make consumption as clean as possible. There are several amortization strategies that can be used, such as the following:

(i) Linear Amortization Formula (LAF): In this case, the total energy consumption TE can be linearly allocated throughout a pre-specified period p of duration time t, which can be set as yearly, monthly, daily, hourly, and so on, giving the energy budget constraint:

$$E_p = \frac{TE}{t}, \tag{4}$$

where TE is the total energy allocated for the complete period p. In our *ECP* example of Table 5, the flat consumes a total energy TE = 3,666 kWh yearly, on average. In this case, if an hourly energy budget period is selected by the user, then the energy budget constraint E_h will be calculated as $E_h = 3666/8928 = 0.742$ kWh, for a duration $t = 12 \times 31 \times 24 = 8{,}928$, indicating the hourly available budget for the whole year.

(ii) Balloon Linear Amortization Formula (BLAF): In this case, the user saves a percentage π of energy from total energy TE for a period of time $\lambda < t$, the so-called balloon σ, which is used in the remaining period $\lambda' = t - \lambda$ that the energy consumption is higher. The energy budget

Table 5. Energy Consumption Profile (ECP) of a Flat

Months	kWh per month	kWh per day	kg CO_2 (EU-27)	kg CO_2 (USA)
January	775.50	25.0	7.40	11.23
February	528.75	17.06	5.05	7.66
March	246.75	7.96	2.36	3.57
April	141.00	4.55	1.35	2.04
May	176.25	5.69	1.68	2.55
June	211.50	6.82	2.02	3.06
July	246.75	7.96	2.36	3.57
August	317.25	10.23	3.03	4.60
September	211.50	6.82	2.02	3.06
October	176.25	5.69	1.68	2.55
November	211.50	6.82	2.02	3.06
December	423.00	13.65	4.04	6.13
Total	3666.00	–	–	–

constraint E_p for a period p of duration t is calculated as follows:

$$E_p = \begin{cases} \frac{TE}{t} - \frac{\sigma}{\lambda}, & \text{for } \lambda \text{ period} \\ \frac{TE}{t} + \frac{\sigma}{\lambda}, & \text{for } \lambda' \text{ period} \end{cases}, \quad \text{where } \sigma = \left(\frac{TE}{t} \times \lambda\right) \times \pi. \tag{5}$$

In our example, if the user desires to save $\pi = 30\%$ of the total energy consumption $TE = 3{,}666$ kWh, for $\lambda = 7$ months (e.g., for April to October) that the consumption is lower than the remaining $\lambda' = 5$ months (i.e., November to March) then $\sigma = (305.5 \times 7) \times 0.3 = 641.55$ kWh. Therefore, the energy consumption for seven months, between April to October, will be $E_p = 397.15$ and for five months, between November to March will be $E_p = 213.85$ kW. The corresponding hourly energy budget constraint of this formula will be $E_h = 397.15/(31 \times 24) = 0.53$ kWh and $E_h = 213.85/(31 \times 24) = 0.28$ kWh, accordingly.

(iii) *ECP*-**based Amortization Formula (EAF):** In this case, a set of weights is calculated using the *Energy Consumption Profile ECP* vector (e.g., see Table 5). The weights are then used to define the energy budget constraints for a user-defined period over an available energy budget E:

$$E_p = \left\{ \frac{w_i \times E}{t/|ECP|} \right\}, for \quad i = 1, \ldots, |ECP|, \quad \text{where } w_i = \frac{TE}{ECP_i} \text{ and } \sum_{i=1}^{|ECP|} w_i = 1, \tag{6}$$

TE is the total energy consumption derived from the ECP, E is the user-specified available energy budget, $|ECP|$ is the size of the *Energy Consumption Profile* vector and $t/|ECP|$ normalizes the energy budget based on the time granularity duration t. Clearly, t could have taken a different granularity (e.g., day, hour, or even minute), given that this information is typically available in energy monitoring systems. For example, let's assume an hourly energy budget period and an available yearly budget $E = 3{,}500$ kWh selected by a user of a flat with an ECP indicated in the left column of Table 5. The total energy consumption derived from the ECP set is $TE = 3{,}666$ kWh and $|ECP| = 12$. Therefore, $w_1 = 0.211, w_2 = 0.144$, and so on until $w_{12} = 0.115$. The hourly energy consumption per month can be calculated as $\{\frac{w_i \times 3500}{31 \times 24}\}$. In all cases, the max CO_2 emission Γ_p is

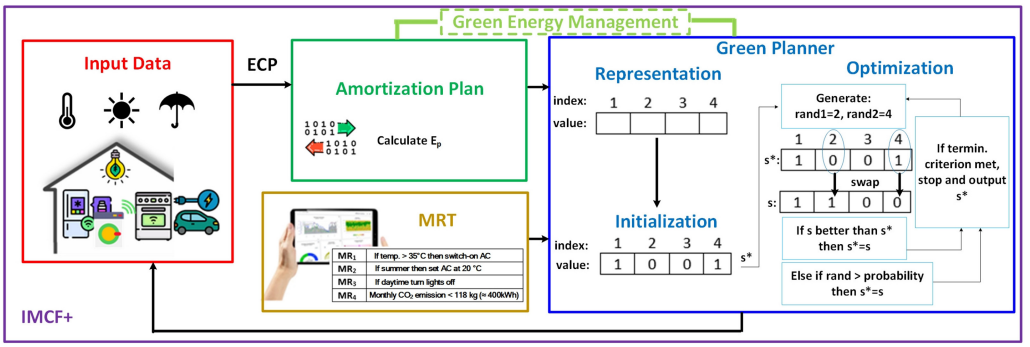

Fig. 2. Example execution of the IMCF$^+$ framework planner.

calculated as:

$$\Gamma_p = E_p \times \gamma$$

and the constraint F_Γ is directly calculated using Equation (3).

4.3 Green Plan (GP) Algorithm

In this subsection, we discuss our GP algorithm and its related components and parameters.

Solution Representation: A GP solution is a vector $s = < s_1, \ldots, s_N >$ of size $N = |MRT|$. A vector component s_i represents a MR in table MRT, where $s_i = 0$ means ignoring MR at position i of table MRT and $s_i = 1$ means adopting MR at position i.

Initialization: At the beginning, an initial solution s^* is developed in line 8 that will specify the initial state of the algorithm. An initial solution can be generated randomly or deterministically. In the latter, a deterministic solution for the GP can be to set all vector components to 1, meaning that all MRs will be greedily triggered, favoring in this way the comfort error objective, but having a high probability of violating the Γ_p constraint. In the case of a random initialization, the values of all vector components are uniformly randomly selected.

Optimization: For the optimization step, a simulated annealing heuristic is utilized to transform the current state's solution s^* to a new state's solution s by uniformly randomly selecting and swapping up to k components of s^*. The probability of making the transition from the current state s^* to a candidate new state s is specified by an acceptance probability function $P(e(s^*), e(s), T)$, where $e(s^*)$ and $e(s)$ is the energy (or fitness evaluation) of the solutions generate in states s^* and s, respectively, and T is a global time-varying parameter, called the temperature. The temperature T plays a crucial role in controlling the evolution of the state s, since a large T favors coarse variations between the states, where a small T favors more fine grain variations. Parameter T can be either set through a pre-specified annealing schedule or calculated in respect to the number of iterations passed (i.e., $T = \frac{\tau+1}{\tau_{max}}$). In any case, a gradual reduction of T is required as the simulation proceeds. A transition is always accepted when s is better than s^* and is probabilistically accepted, with a probability $P(e(s^*), e(s), T)$ when s is worse than s^*, allowing in this way the heuristic to avoid local optima. Here, it is important to note that any heuristic or meta-heuristic approach can be utilized in the GP optimization step.

Evaluation: Each solution s is evaluated using the performance metrics F_{CE}, F_E and F_Γ of Equations (1)–(3) in lines 10 and 13. A solution s is considered better and replaces the current best solution s^* (i.e., transition is accepted) if $(F_E(s) \leq E_p)$ && $(F_{CE}(s) < F_{CE}(s^*))$ && $(F_\Gamma(s) < F_\Gamma(s^*))$. Otherwise, if s is worse than s^*, the transition is accepted only if a uniformly random generated value $rand < P(e(s^*), e(s), T)$, where $rand$ is in the range of $[0, 1]$.

ALGORITHM 1: IMCF$^+$: Generates a Green Plan

Input: *MRT*: Meta-Rule Table; k: components to be modified; τ_{max}: max iterations; t: time granularity; *apl*: amortization plan; *ECP*: Energy Consumption Profile; T:Temperature
Output: An green plan solution $s^* = (s_1, \ldots, s_N)$

AP(apl, p, ECP) ▷ **Amortization Plan Routine**
  switch (apl)
  a: $E_p \leftarrow LAF(t, ECP)$ ▷ use linear Equation (4)
  b: $E_p \leftarrow BLAF(t, ECP)$ ▷ use balloon Equation (5)
  c: $E_p \leftarrow EAF(t, ECP)$ ▷ use *ECP*-based Equation (6)
  return $\Gamma_p = E_p \times \gamma$ ▷ max CO_2 emission for γ CO_2 emission intensity

GP$(MRT, k, \tau_{max}, i, \Gamma_p)$ ▷ **Green Plan Routine**
  $s^* \leftarrow init_i(MRT)$ ▷ s^*: initial solution for time i
  $(F_{CE}, F_E, F_\Gamma) \leftarrow evaluate(s^*)$ ▷ with Equations (1)–(3)
  While $\tau < \tau_{max}$ **do** ▷ τ: current iteration
    $s \leftarrow optimization(s^*)$ ▷ randomly select k positions and swap their binary value
  $(F_{CE}, F_E, F_\Gamma) \leftarrow evaluate(s)$ ▷ with Equations (1)–(3)
    If $(F_E(s) \leq E_p)$ && $(F_{CE}(s) < F_{CE}(s^*))$ && $(F_\Gamma(s) < \Gamma_p)$ **then**
      $s^* \leftarrow s$ ▷ Set s as the current solution s^*
    Else If $rand(0,1) \leq P(F_{CE}(s^*), F_{CE}(s), T)$ && $(F_{CE}(s) < F_{CE}(s^*))$ && $(F_\Gamma(s) < \Gamma_p)$ **then**
      $s^* \leftarrow s$ ▷ Set s as the current solution s^*
    EndIf
    $\tau + +$ ▷ Increase iterations
  EndWhile
  return s^* ▷ Return the final green plan solution

$\Gamma_p \leftarrow AP(apl, t, ECP)$;
return $(\forall_i^t\ GP(MRT, k, \tau_{max}, i, \Gamma_p))$

Termination criterion: The GP stops when τ_{max} iterations are completed. Alternatively, the algorithm can iterate until $\nexists s | F_{CE}(s) < F_{CE}(s^*)$. However, in the absence of any knowledge on the optimal solution this may result in an infinite loop.

Example: Consider the simplified scenario of Figure 2 in which a user sets four MRs MR_i^j in the *MRT* for a four-room residence, which along with some input data from the house's sensors as well as some online web services (e.g., weather forecasting website) are forwarded to the *IMCF*$^+$. *IMCF*$^+$ initially runs the AP subroutine using a pre-selected amortization formula as well as the *ECP* to calculate an energy budget constraint E_p and consequently a CO_2 emission constraint Γ_p. Then it converts the *MRT* to a binary vector, in which each index of the vector represents a MR in the *MRT*. A random initialization process generates the first solution $s^* = < 1, 0, 0, 1 >$, which means that MRs 1 and 4 will be triggered and MRs 2 and 3 will be ignored. Solution s^* is evaluated using the performance metrics F_{CE}, F_E and F_Γ of Equations (1)–(3). During the optimization, $k = 2$ vector components are modified using a uniform random generator. In this example, the value of vector component 2 is swapped from 0 to 1 and the value of component 4 is swapped from 1 to 0. The newly generated solution $s = < 1, 1, 0, 0 >$ is again evaluated using Equations (1)–(3) and compared with the current best solution s^*. At each iteration, when s is better than s^* then s becomes the s^*, otherwise of s is worse than s^* then a uniformly random generator picks and returns a value *rand* in the range [0, 1] and if $rand \leq P(F_{CE}(s^*), F_{CE}(s), T)$ then s becomes s^*. The algorithm stops when the termination criterion is met.

4.4 Performance Analysis

We analytically derive the performance of $IMCF^+$ with respect to the estimated comfort error CE and CO_2 emission Γ. We adopt a worst-case analysis as it provides a bound for all input. Our experimental evaluation in Section 6, shows that under realistic and real datasets our approach performs more efficiently than the projected worst case. The analysis is based on our system model and ignores any energy not directly associated with the MRT MRT.

Lemma 1. *Our IMCF$^+$ approach has a comfort error of* $F_{CE} = \frac{1}{n}\sum_{i=1}^{D}\sum_j ce_j(MR_i), i = 1,\ldots,n$, *where* $n > 0$ *is the number of MRs that will be executed.*

Proof. The GP will select at least $n > 0$ MRs to be executed satisfying in this way the energy budget constraint. In the worst case scenario and for an energy budget equal to zero, IMCF$^+$ will act as the NR approach providing $F_{CE} = 1$. On the other hand, the MR approach by greedily executing all MRs in the MRT will offer an $F_{CE} = 0$. □

Lemma 2. *Our IMCF$^+$ approach has a CO_2 emission of* $F_\Gamma = \frac{1}{n}\sum_{i=1}^{D}\sum_j \Gamma_j(E_j(MR_i), \gamma), i = 1,\ldots,n$, *where* $n \leq N$ *is the number of MRs that will be executed.*

Proof. The GP will select at most $n \leq N$ MRs to be executed satisfying in this way the CO_2 emissions constraint. In the worst case scenario and by ignoring the constraint, IMCF$^+$ will act as the MR approach providing $F_\Gamma = 1$. On the other hand, the NR approach by not executing any MR of the MRT will offer an $F_\Gamma = 0$. □

5 THE IMCF$^+$ SYSTEM ARCHITECTURE

In this section, we describe an integrated system we have developed for IMCF$^+$ using the open Home Automation Bus (OpenHAB),[5] the Linux crontab daemon, as well as the Laravel PHP web framework following the model–view–controller architectural pattern. We start out with a discussion of the system architecture and then describe the Graphical User Interface we have developed that integrates directly into OpenHAB's mobile and web Panel view for both interactive management of IoT and automated management of Energy-aware and CO_2-aware MRT rule pipelines using the GP described in this work.

5.1 System Architecture

Our system architecture comprises the following components: (i) a full-fledge local controller implemented inside the openHAB stack, which is a smart home management software; and (ii) IMCF$^+$, which is the software system that encapsulates the complete application logic of the energy management stack we propose in this work along with the respective user interfaces.

Local Controller (LC): is a java-based system installed on a micro device, like a Linux Raspberry PI, running on the local network of a user. The LC will be in direct communication with the IoT devices (i.e., *Things (TG)*) to instruct them based on the preferences registered by a user (see Figure 3). A user will typically download the openHAB smartphone *application (APP)*, for iOS or Android, and interact with TG through LC. For the implementation of LC we decided to extend the openHAB stack, which is a vendor and technology agnostic open source automation software for smart home that provides a rich ecosystem of bridges through which a user can interact directly with IoT devices (e.g., Daikin Smart A/C, Phillips HUE lights) both locally and remotely. This gives us the benefit to achieve maximum IoT market compatibility as the integration of IoT is always an immense challenge.

[5]OpenHAB, https://openhab.org.

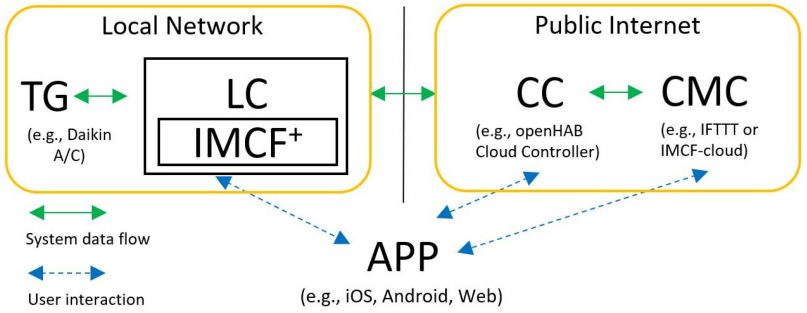

Fig. 3. Overview of the IMCF+ system implementation.

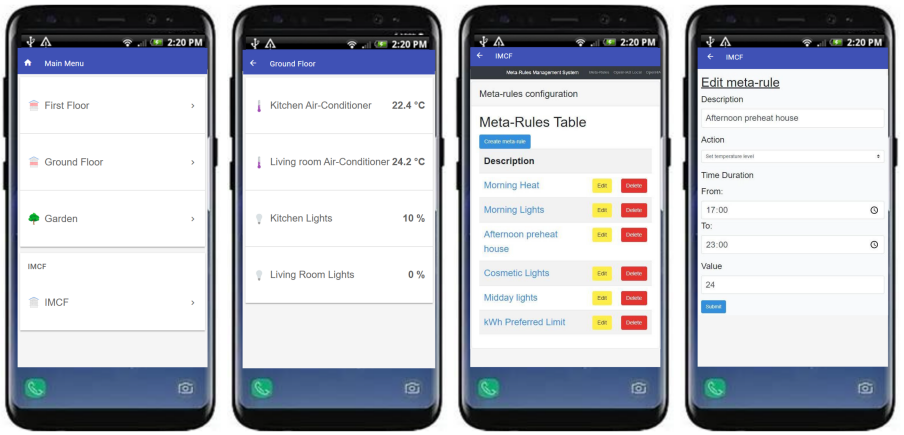

Fig. 4. IMCF+ Graphical User Interface: Integration of the IMCF+ Software Library in the openHAB Home Automation Stack. From left to right: (a) Interactive and Automated Menu; (b) Dashboard for smart space current state; (c) MRT Configurator; and (d) MRT data entry form.

The IMCF+ system is protected by the authentication provided by openHAB and Laravel framework. The LC is also located on the user's local network and is protected by the Ubuntu operating system firewall. Therefore, the system is quite secure, as someone with malicious intent will have to break through the security offered by openHAB and Laravel to be able to infect files or penetrate the firewall.

To realize the operation of LC, consider, for example, a user inside his smart space that selects through an APP to increase the temperature of an A/C from 21 to 25 degrees Celsius (see Figure 4(a) and (b)). This manual interaction goes to LC that communicates this directly to TG (on older units this is typically unencrypted http communication channels, either http query string or in some cases JSON web 2.0 interactions). When a user's APP is outside a smart space, the network firewall and **Network Address Translation** (**NAT**) will obviously not let this user interact with LC. As such, the user's APP connects to the ***Cloud Controller (CC)***, which is a server on the public Internet that communicates and controls LC remotely.

The complete picture can tentatively be complemented by a ***Cloud Meta-Controller (CMC)***, like IFTTT, which can enable the user to configure and run various custom rules. CMC would in this case interact with CC that would in turn interact with LC that would eventually interact with TG, all under the *manual* control of the user APP.

All above clearly do not enable the adaptation of comfort preferences to meet the long-term energy planning targets of individuals or group of individuals as we do in this work (see examples in Section 1) and describe technically next. In the future, we also plan to investigate the so called IMCF$^+$-Cloud extensions that will enable IMCF$^+$ to operate as a CMC controller in the cloud.

The IMCF$^+$ Component is a software extension to LC we have implemented that encapsulates the implementation of the GP algorithm but also the **Graphical User Interfaces** (**GUI**) and storage necessary to allow the user to interact with the system. The GP algorithm is implemented as a JAVA library which takes the user configurations from a local MariaDB persistency layer. The storage layer is populated by the user using the APP, which has been configured in a way to integrate seamlessly the MRT rule definition process through a web-based GUI (see Figure 4(c) and (d)). The GUI code is written in the Laravel PHP web framework following the model–view–controller architectural pattern as well as JavaScript and HTML. Our complete code is approximately 2,500 **lines-of-code** (**LOC**) plus 3,000 LOC going to the GUI.

For the GUI code execution, we rely directly on the NGINX web web-server available on Raspberry PI, while for the IMCF$^+$ GP library we invoke the cron job daemon that reliably executes the Green Planning every few minutes. In case devices have to be turned on or off, the IMCF$^+$ system has the following options in our system:

— *Binding-mode*, where IMCF$^+$ exploits the rich ecosystem of bridges available on the openHAB open source project to interact with local devices. We use this as the default mode, as it allows our platform to scale to a very wide spectrum of IoT devices.

Example[6]**:**

daikin.things: `daikin:ac_unit:living_room_ac [ host="192.168.0.5" ]`

daikin.items: `Switch DaikinACUnit_Power`
`channel = "daikin:ac_unit:living_room_ac:power"`
`Number:Temperature DaikinACUnit_SetPoint`
`channel = "daikin:ac_unit:living_room_ac:settemp"`

— *Extended mode*, where IMCF$^+$ implements locally the custom instructions for enabling and disabling the various TG devices in the smart space of a user. An example of this mode is the following command:

Example: Setting Daikin in Cool Mode 25 degrees.[7]

http://192.168.0.5/aircon/set_control_info?
`pow=1&mode=3&stemp=25&shum=0`

Given that many of the IoT communications are unencrypted, this can easily be captured by deep packet analyzers like Wireshark. Additionally, to avoid any additional CMC, CC or LC interactions with the Daikin TG, we also configure the LC network firewall with the `iptables` command to disable TCP flows to designated TG devices on the local network. In this case, IMCF$^+$ works as a real network firewall by blocking all outgoing traffic from LC to TG.

Example:

```
iptables -A OUTPUT -s 192.168.0.5 -j DROP
```

5.2 Graphical User Interface (GUI)

Our prototype GUI provides all the functionalities for a user participating in IMCF$^+$. The GUI is divided into a MRT interface and the OpenHAB Rules Table, respectively, as shown in Figure 4(d).

[6]OpenHAB Daikin Binding. https://www.openhab.org/addons/bindings/daikin/.
[7]Daikin Control. https://github.com/ael-code/daikin-control.

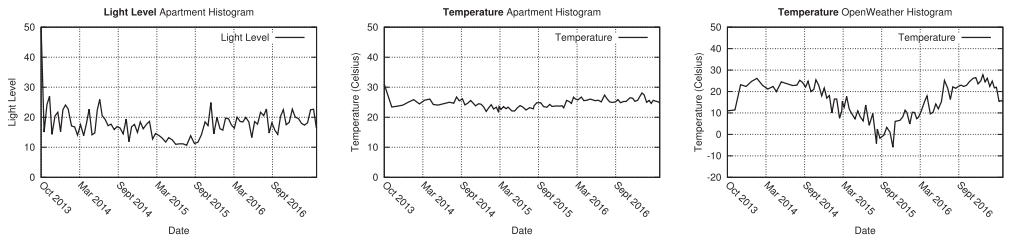

Fig. 5. Histograms of Light-Level and Temperature of the apartment dataset, and the outdoors Temperature retrieved from the OpenWeather forecast dataset.

The MRs interface prompts users to define kWh preferred limit, temperature, and light values for any configured time slots. The OpenHAB Rules Table records are retrieved through the OpenHAB Rest API system consisted of smart device sensor measurements installed and pre-configured in a building. These rule combinations are used by the AI GP algorithm to satisfy the user needs keeping the balance between comfort and energy consumption.

At a high level, our GUI enables the following functions: (i) record OpenHAB item measurements/values on local storage and present those on a table; (ii) configure various MRs in regards of kWh limit, temperature, and light values; and (iii) operate IMCF$^+$ framework and get an efficient execution considering user satisfaction along with balanced comfort error and energy consumption.

6 EXPERIMENTAL METHODOLOGY AND EVALUATION

This section presents an experimental evaluation of our proposed framework. We start-out with the experimental methodology and setup in Section 6.1, followed by various evaluation studies in different smart environments in Section 6.2, and finally a number of experiments that expose the core benefits of our IMCF$^+$ framework and its internal GP algorithm compared to baseline techniques. The experimentation was conducted based on a micro-benchmarking study for parameters of the GP algorithm and concludes with an Energy Conservation Study as defined in Section 6.3.

6.1 Methodology

This section provides details regarding the algorithms, metrics, and datasets used for evaluating the performance of the proposed approach.

Datasets: We have adopted a trace-driven experimental methodology in which real datasets are fed into our simulator executed on the testbed. This allows repeatable execution of workloads under different control parameters. Our evaluation is carried out on an Ubuntu 18.04 VMWare server image, featuring 8 GB of RAM with 2 virtual CPUs (@ 2.40 GHz). The image utilizes fast local 10K RPM RAID-5 LSILogic SCSI disks, formatted with VMFS 6 (1MB block size). We utilize anonymized measurements from a real residential apartment that comprises a variety of sensors, sub-meters and approximately 5,668,878 readings (1.09 GB in total). Our real datasets of residential data are collected by the "**Center for Advanced Studies in Adaptive Systems**" (**CASAS**) [51] at Washington State University in the School of Electrical Engineering and Computer Science. CASAS serves to meet research needs around testing of the technologies using real data through the use of a smart homes environment located on the WSU Pullman campus. The real weather forecast dataset was acquired using Weather API on the OpenWeatherMap website and contains ~5 years (2012–2017) of high temporal resolution data based on hourly measurements of various weather attributes, such as temperature, humidity, air pressure, weather description, wind direction, and speed (see Figure 5).

Table 6. MRT and IFTTT Configurations for Flat Experiments

Description	Time/Duration	Action	Value
Night Heat	01:00–07:00	Set Temper.	25
Morning Lights	04:00–09:00	Set Light	40
Day Heat	08:00–16:00	Set Temper.	22
Midday Lights	10:00–17:00	Set Light	30
Afternoon Preheat	17:00–24:00	Set Temper.	24
Cosmetic Lights	18:00–24:00	Set Light	40
Energy Flat	for three years	Set kWh Limit	11,000
Energy House	for three years	Set kWh Limit	25,500
Energy Dorms	for three years	Set kWh Limit	480,000

IF	THIS	THEN	THAT
Season	Summer	Set Temper.	25
Season	Winter	Set Temper.	20
Weather	Sunny	Set Temper.	20
Weather	Cloudy	Set Temper.	22
Weather	Sunny	Set Light	0
Weather	Cloudy	Set Light	40
Temperature	>30	Set Temper.	23
Temperature	<10	Set Temper.	24
Light Level	>15	Set Light	9
Door	Open	Set Light	0

— **Temperature Dataset:** The 700 MB dataset contains 3,555,238 readings on a second basis between October 2013 and December 2016. The readings, which are recorded at a residential apartment of a volunteer adult, include temperature and door/window sensor measurements.
— **Light Dataset:** The 416 MB dataset contains 2,113,640 readings on a second basis between October 2013 and December 2016. The readings, which are recorded at a residential apartment of a volunteer adult, include light measurements.
— **Weather Forecast Dataset:** The 71,23 MB dataset contains 271,561 readings on an hourly basis between October 2012 and November 2017. The historical weather data is available for 30 US and Canadian Cities, as well as 6 Israeli cities, including temperature, humidity, air pressure, weather description, wind direction, and speed.

To evaluate the scalability of our propositions for residential buildings of various scales, we have generated three realistic datasets by replicating the above onto various building sizes. The resulting datasets are the following:

— **Flat Dataset:** A single user flat/apartment dataset consisted of one bedroom, a bathroom, and a kitchen. The apartment has a single split unit to warm/cool an area size of 50 m^2. It has a size of 1.09 GB.
— **House Dataset:** A residential house dataset generated by replicating, mixing up the readings, and multiplying the real dataset by the factor of four. It has three bedrooms and four split units used by four residents. The area size is approximately 200 m^2. It has a size of 4.50 GBs.
— **Dorms Dataset:** A University Campus dataset (dorms) generated synthetically from the bootstrap datasets. We have generated 50 dorm apartments consisting of two bedrooms (10 m^2/room) with a shared bathroom, a kitchen, and two split units. The total area size of the dorms is approximately 2000 m^2 and has a size of 20 GBs.

Metrics: Our cost metrics are *Energy Consumption* (F_E), CO_2 Emission (F_Γ), and *Comfort Error* (F_{CE}) as defined in Section 3 as well as *CPU Time* (F_T) for the comparison in the performance study. The *CPU Time* (F_T) is the processing time required by the controller for running the optimization function and calculating the output for all MRs. The mean and standard deviation of the results is shown with error bars in all experimental studies that follow, based on ten repetitions.

Algorithms: Here we provide a concise overview of the compared methods and algorithms considering the MRT Table 6, which is inspired from real preferences recorded by users.

— **No-Rule (NR):** This method ignores all rules in the MRT (see Table 6 used in the flat dataset, the rest use uniformly random variations of the same table). F_E is obviously always 0 since no IoT device is turned on. On the other hand, F_{CE} is measured as a percentage of comfort a user will had he executed all rules (Def. 3.1) and (F_T) is only the cost of doing the simulation.

— **If-This-Then-That (IFTTT):** This executes the IFTTT preferences (see Table 6) used in the flat dataset. The dataset was collected from the official IFTTT website. For the evaluation we measure F_{CE}, i.e., percentage of comfort a user will get from executing the IFTTT rules against all rules (recorded in the MRT table).
— **Meta-Rule (MR) method:** This method executes all rules in the MRT (again, see Table 6 used in the flat dataset, the rest use uniformly random variations of the same table). F_E is obviously maximum here and F_{CE} is minimum as IoT devices will operate maximally.
— **Green Planner (GP) algorithm:** This is the algorithm we propose in Section 4. For the construction of the GP algorithm we have set the number of rules activation/deactivation in each iteration (k), a savings percentage amount (s), and the number of iterations (τ_{max}) and detailed evaluation follows for these parameters in Sections 6.3.2 and 6.3.3.

Evaluation Plan: We split our experimental evaluation into two basic series: (i) Evaluation Studies in Section 6.2, during which we carry out an evaluation in a variety of environments using a pre-configured instance of our GP algorithm according to the parameters we identify in the subsequent micro-benchmarking section; (ii) Micro-benchmarking Study in 6.3, during which we carry out a detailed evaluation of all parameters according to the following table:

Table 7. Micro-benchmark Study Configuration Table

Section	Name	k	s	τ_{max}	Initialization
6.3.1	Performance Evaluation	4	0%	15	random
6.3.2	k-opt Evaluation	2, 3, 4	0%	15	random
6.3.3	Initialization Evaluation	4	0%	15	all-1s, random, all-0s
6.3.4	Energy Conservation Study	4	0%–40%	15	random

6.2 IMCF$^+$ Evaluation Studies

In this section, we carry out an extensive evaluation study in various smart environments equipped with net-metering photovoltaic systems. Particularly, we examine IMCF$^+$ in scenarios such as a household, university campus, and hotel apartments, using different time frames and rules as well as diverse number of users. Note that the financial benefit of using our GP is not significant in a net-metering system, however, the environmental impact is very significant.

6.2.1 Household Evaluation Study. For this experimental series, we deployed an instance of our real system for a family of three persons for one week (see Figure 6). Particularly, we allowed each person to configure their personal preferences using the Mobile APP that interacts with an IMCF$^+$-LC node on a Linux VM on our datacenter described earlier. Particularly, each individual resident entered approximately three different MRs according to their personal preferences. The weekly energy consumption (kWh) limit was set by one of the residents to 165 kWh. This results in configuration data of approximately 65 bytes/user stored in the MariaDB persistency layer. To measure the environmental parameters (i.e., temperature, light) we use data from the open weather API. We measure again the performance of the proposed GP framework in regard to F_E, F_Γ, and F_{CE}.

The F_E, F_Γ, and F_{CE} results for our evaluation using real Weather Forecast data are summarized in Table 8. In respect to F_{CE}, our observation is that GP is indeed an efficient approach for retrieving great user satisfaction, as it performs in 4 seconds on average with an Average Comfort Error ≈3.45%. Table 9 demonstrates for each individual resident their own Average Comfort Error values in respect with their configured MRs, showing both a consistent and high satisfaction close to 98.9% for all residents. The system behaves correspondingly to what we observed in the simulations,

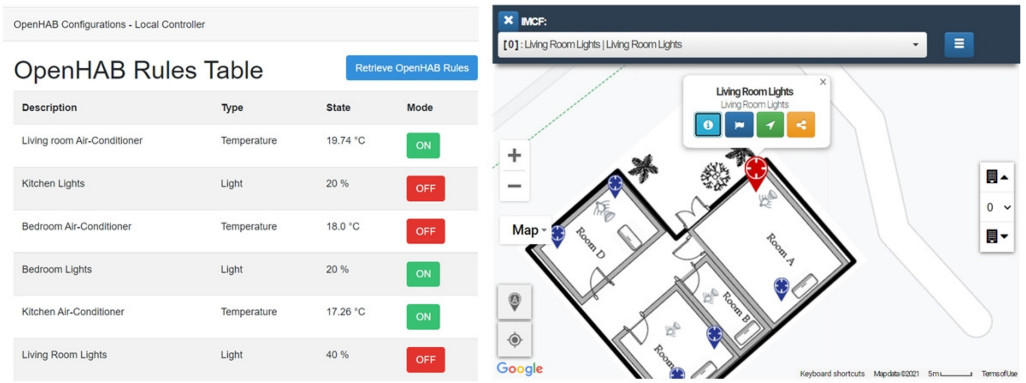

Fig. 6. Household evaluation study.

Table 8. Evaluating the Household System with Respect to F_E, F_Γ and F_{CE}

	MR algorithm	**GP algorithm**
Time Duration	Week	Week
Energy Consumption (F_E)	210.85 kWh	132.15 kWh
Comfort Error (F_{CE})	0%	3.45%
kg CO_2 Emission (F_Γ)	84,15 kg CO_2	50.79 kg CO_2
Energy Consumption (F_E) **with CO_2 emissions**	185.0 kWh	114.85 kWh
Energy Consumption (F_E) **without CO_2 emissions**	25.85 kWh	17.3 kWh

Table 9. Individual Resident Comfort Error (F_{CE})

Users	**Comfort Error** (F_{CE})
Father	1.2806%
Mother	1.1500%
Daughter	1.0234%

therefore the $F_E \approx 130.64$ kWh is within the preferred budget limit as pre-configured by the user, and with a reasonable $F_\Gamma \approx 50.79$ kg CO_2. This is approximately a one-way flight trip from Paris to Frankfurt, since according to the figures in Table 2 from the German non-profit organization called "Atmosfair", a flight from Paris to Frankfurt and back generates about 115 kg of CO_2 per passenger. As presented in the results section, the kilograms of carbon dioxide emissions reduction while using the GP algorithm is ≈59% (33 kg CO_2) less than the MR algorithm.

6.2.2 University Campus Evaluation Study. In view of this experimental series, we have simulated our system at the University of Cyprus for the timespan of a year using the kg/CO_2 readings of Cyprus (see Table 1). The campus consists of 10 hall blocks, a sports center, laundry stations, a library, a parking lot, a restaurant, and the accommodation service office (see Figure 7). The administration officer configured approximately 25 MRs/preferences for the entire university campus through the Mobile APP directly interacting with the IMCF$^+$ Local Controller node located on our datacenter. The administration officer is responsible to manage and act as a core to the setting system considering the smooth functionality of all the campus premises. Additionally, the moderator has the privilege to monitor the state and mode of the IoT devices though the GUI console. The annual energy consumption (kWh) limit was set to 500,000 kWh. Similarly, to the previous case,

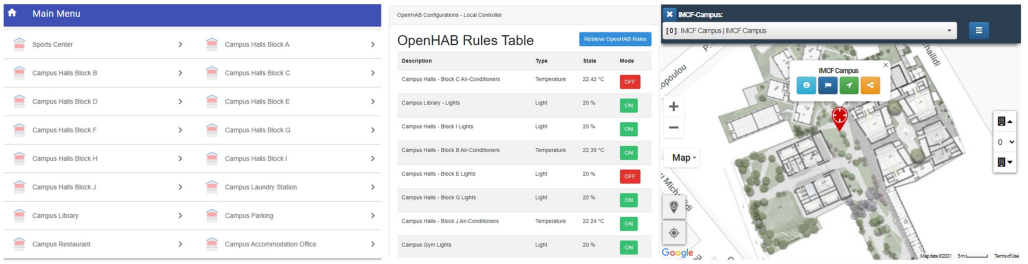

Fig. 7. University campus evaluation study.

Table 10. Evaluating the University Campus System with Respect to F_E, F_Γ, and F_{CE}

	MR algorithm	GP algorithm
Time Duration	Year	Year
Energy Consumption (F_E)	652,641 kWh	414,548 kWh
Comfort Error (F_{CE})	0%	3.20%
kg CO_2 Emission (F_Γ)	355,719.50 kg CO_2	166,956.03 kg CO_2
Energy Consumption (F_E) **with CO_2 emissions**	580,655.40 kWh	368,832.85 kWh
Energy Consumption (F_E) **without CO_2 emissions**	71,985.60 kWh	45,715.15 kWh

data from the Open Weather API have been used for the environmental parameters (i.e., temperature, light) measurement, thus the GP framework strictly considers the performance of the F_E, F_Γ, and F_{CE}, respectively.

Table 10 summarizes the F_E, F_Γ, and F_{CE} results for our university campus evaluation study. In respect to F_{CE}, our observation is that GP behaves proficiently while retrieving a great satisfaction, as it performs with an Average Comfort Error ≈3.20%. This indicates that the more rules are configured to the system, the better results are produced by the proposed framework. The algorithm acts correspondingly to what we observed in the simulations, therefore the $F_E \approx$ 414,548 kWh is within the preferred budget limit as pre-configured by the administration officer, and also with a reasonable $F_\Gamma \approx$ 166,956.03 kg CO_2. This is approximately 53 return flight trips from London to Perth, since according to the figures of Table 2 from "Atmosfair", flying from London to Perth and back generates about 3,153 kg of CO_2 per passenger. As the results clearly present, the kilograms of carbon dioxide emissions reduction while using the GP algorithm is ≈47% (188,763 kg CO_2) less than the MR algorithm.

6.2.3 Hotel Apartments Evaluation Study. In the final experimental series, we have deployed an instance of our framework for a hotel's apartments under an annual operation. The hotel consists of approximately 50 apartments, each one with its own kitchen, a bathroom, a living room, one-three bedrooms, and a balcony (see Figure 8). For each apartment tenants were prompted to set up their personal MR preferences through a smart mobile device. In contrast with the previous cases, each tenant had the privilege to configure its rented apartment's smart devices in regard to temperature and light level, but only the hotel's administrative landlord could exclusively configure the annual energy consumption (kWh) limit for all hotel apartments, which in this case was set to 375,000 kWh. All the users can log into the IMCF$^+$ platform and observe their personal comfort level. Moreover, Open Weather API data have been used for the measurement of the environmental parameters, and then the GP framework was executed and evaluated upon the F_E, F_Γ, and F_{CE}.

Table 11 summarizes the F_E, F_Γ, and F_{CE} results for our hotel apartments evaluation study. In respect to F_{CE}, our observation is that GP operates effectively as it performs with an Average Comfort Error ≈2.45%, showing an excellent satisfaction level for the entire group of users. The

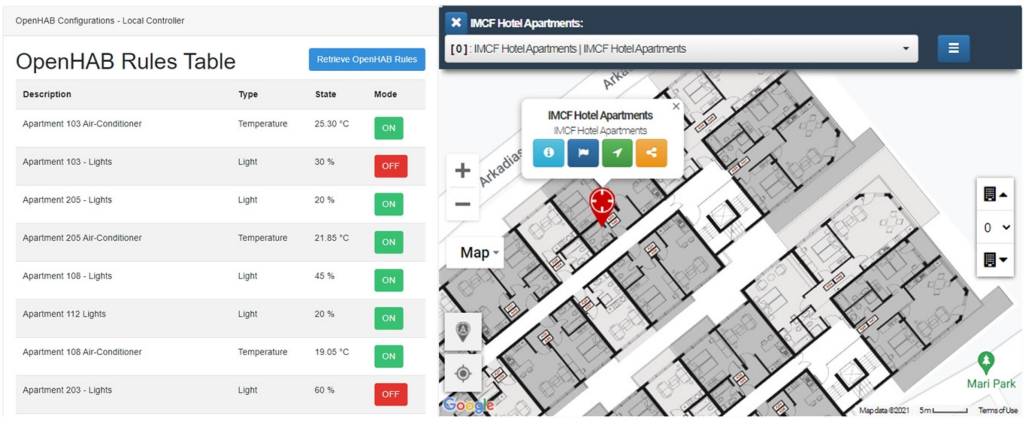

Fig. 8. Hotel apartments evaluation study.

Table 11. Evaluating the Hotel Apartments System with Respect to F_E, F_Γ, and F_{CE}

	MR algorithm	GP algorithm
Time Duration	Year	Year
Energy Consumption (F_E)	509,339 kWh	322,411 kWh
Comfort Error (F_{CE})	0%	2.45%
kg CO_2 Emission (F_Γ)	271,962.35 kg CO_2	126,125.29 kg CO_2
Energy Consumption (F_E) **with CO_2 emissions**	452,988.50 kWh	280,902.65 kWh
Energy Consumption (F_E) **without CO_2 emissions**	56,350.50 kWh	41,508.53 kWh

framework behaves accordingly to what we observed in the simulations, therefore the $F_E \approx 322{,}411$ kWh is within the preferred budget limit as pre-configured by the hotel's administrative landlord, and also with a reasonable $F_\Gamma \approx 126{,}125.29$ kg CO_2. This is approximately as 49 return flight trips from Perth to Athens, since according to the figures in Table 2 from "Atmosfair", flying from Perth to Athens and back generates about 2,530 kg of CO_2 per passenger. As clearly demonstrated in the results section, the kilograms of carbon dioxide emissions reduction while using the GP algorithm is ≈45% (145,837 kg CO_2) less than the MR algorithm.

6.3 Micro-benchmarking Series

In this section, we carry out an extensive micro-benchmarking study for various parameters. Particularly, we examine the number of rules activation/deactivation in each iteration (k), the savings percentage amount (s), different initialization methods, and the number of iterations (τ_{max}) as it is summarized in Table 7. This study guided us to tune and select the right combination of configuration settings for the optimal operation of the GP algorithm.

6.3.1 Series-1: Performance Evaluation. In this experimental series, we evaluate the performance of the proposed GP framework against all algorithms over all datasets introduced, with respect to F_E, F_Γ, and F_{CE}. Figure 9 demonstrates the trade-off between the Energy Consumption, the Comfort Error, the CO_2 Emission, and the CPU Execution Time between all approaches. The NR approach obtained the worst F_{CE} = 62–72% of the whole dataset, and the best F_E = 0 kWh along with the best F_Γ = 0 kg CO_2. The GP algorithm obtained a reasonable F_{CE} around 6.5–8%, the second lowest F_E and the less CO_2 Emission intensity. The IFTTT and MR algorithms are greedy in regards of Energy Consumption, thus their kWh consumed and CO_2 Emissions are very high. The main difference between the two is that IFTTT has F_{CE} = 26% in the residential flat case,

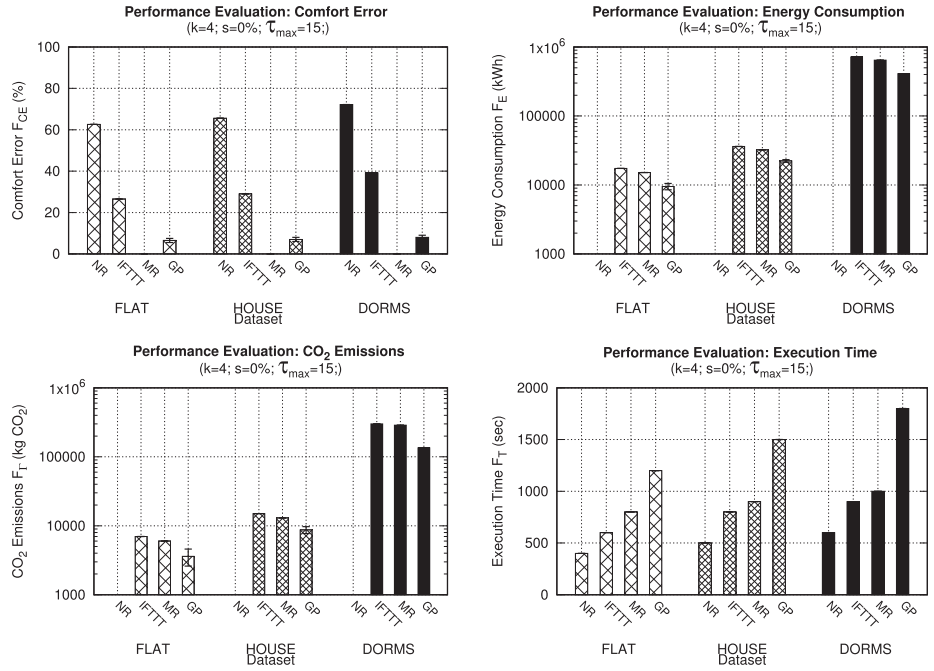

Fig. 9. Performance Evaluation: Evaluation in terms of the comfort error (F_{CE}), the *Energy Consumption* (F_E), the CO_2 emission (F_Γ), and the CPU execution time (F_T) in all datasets.

F_{CE} = 29% in the case of a house, and F_{CE} = 39% in the dorms case, while the MR satisfies all the MRs, thus its F_{CE} is 0%, which is the best possible obtained.

In the residential flat case, the preferred energy budget was configured to 11,000 *kWh* for all three years, and the GP managed to save up to 10% of energy, which is approximately 9,500 *kWh*, with a reasonable F_{CE} around 6.5–7%, and ≈3,600 kg CO_2 Emission. In the case of a house, the preferred energy budget was configured to 25,500 *kWh* for all three years, and the GP managed to achieve approximately 22,300 *kWh*, with a reasonable F_{CE} around 7–7.5%, and ≈8700 kg CO_2 Emission. In dorms case, the preferred energy budget was configured to 480,000 *kWh* for all three years, and the GP managed to achieve approximately 410,000 *kWh*, with a reasonable F_{CE} around 7.5–8%, and ≈135,000 kg CO_2 Emission. It is important to notice, that the difference between the MR and the GP in regard to energy consumption, is relatively high and particularly ≈5,000 kWh for the flat dataset, ≈10,000 kWh for the house dataset, and ≈150,000 kWh for the dorms dataset.

The fastest execution time (F_T) was achieved by the NR method since it simply calculates the error without applying any rules on the imported datasets. The GP algorithm is using the hill climbing approach by searching an optimal solution for the user taking into account the preferred allowed energy budget constraint and curbing the CO_2 emissions, thus it is the most time-consuming method. The MR greedy approach focuses only on minimizing the Comfort Error, which means executing all MRs without any iterative processes or calculations, since F_{CE} = 0%. The GP has been also applied across different countries around the world based on a house scenario as shown on Table 12, demonstrating the CO_2 emission intensity (kg CO_2 per kWh) for each case, respectively.

6.3.2 Series-2: k-opt Evaluation. In the second experiment, we evaluate the performance of the proposed GP framework against different *k*s (rule modifications), with respect to Energy Consumption, CPU Execution Time, CO_2 Emission, and the Comfort Error. Figure 10 illustrates that

Table 12. The GP Algorithm has been Applied Across different Countries Demonstrating the CO_2 Emission Intensity (kg CO_2 per kWh) based on a House Scenario with an Approximate Energy Consumption of ≈22300 *kWh*

COUNTRY	kg CO_2 per kWh
Sweden	255.35
Lithuania	351.76
France	1143.53
Austria	1474.27
Latvia	1920.60
Finland	2349.90
Slovakia	2805.54
Denmark	3248.08
Belgium	3690.77

COUNTRY	kg CO_2 per kWh
Croatia	4148.76
Luxembourg	4590.48
Slovenia	5015.88
Italy	5125.36
Hungary	5300.80
Spain	5349.64
Romania	6578.41
Portugal	6800.32
Ireland	7500.01

COUNTRY	kg CO_2 per kWh
Germany	8214.16
Bulgaria	8951.33
Netherlands	9959.84
Czech Republic	10750.20
Greece	11562.38
Malta	12064.89
Cyprus	12905.67
Poland	13559.11
Estonia	15200.73

COUNTRY	kg CO_2 per kWh
EU-27 (average)	5651.82
USA (average)	8700.15

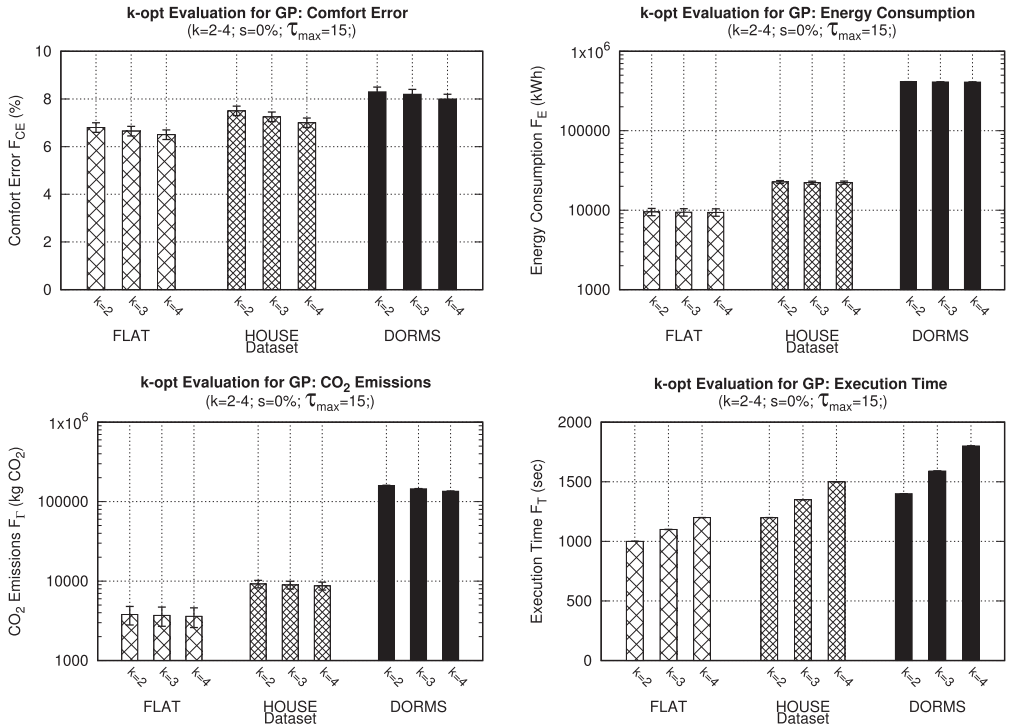

Fig. 10. k-opt Evaluation: Evaluation in terms of the comfort error (F_{CE}), the CO_2 emission (F_T), the CPU execution time (F_T), and the *Energy Consumption* (F_E) based on the number of modified rules (activated/deactivated), in all datasets.

by using four activation/deactivation rule modifications in each iteration we obtain the best F_{CE} with the best possible CO_2 Emission intensity. The worst F_{CE} occurred when we used two rule modifications. In the residential flat case, the energy consumed was in every case approximately the same and around 9,500 *kWh*. What actually made a difference was the F_{CE}, which decreased from 6.8% to 6.5% (≈3600 kg CO_2) in the flat case, from 7.5% to 7.0% (≈8700 kg CO_2) in the house

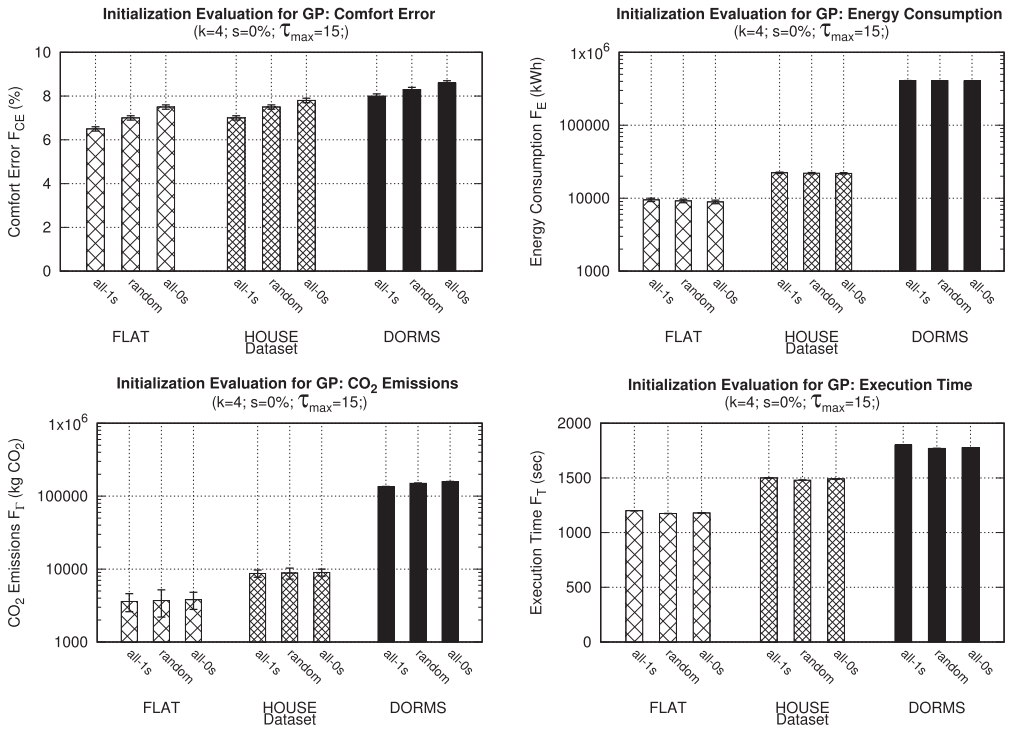

Fig. 11. Initialization Evaluation: Evaluation in terms of the comfort error (F_{CE}), the CO_2 emission (F_Γ), the CPU execution time (F_T), and the *Energy Consumption* (F_E) based on various initialization techniques, in all datasets.

scenario, and from 8.3% to 8.0% (≈135,000 kg CO_2) in the dorms scenario, as we increased the activation/deactivation rule modifications in each iteration. This is due to the hill climbing approach performing bigger "jumps" towards the local optimum at each step and thus searching the solution space more effectively. As the number of k rule modifications increases, the execution (F_T) takes more time to complete and the energy consumption is decreasing gradually.

6.3.3 Series-3: Initialization Evaluation. In the third experimental series, we evaluate the performance of the proposed GP framework using different initialization strategies, with respect to Energy Consumption, CO_2 Emission, CPU Execution Time, and the Comfort Error. In the first (all-1s) case, we have initially activated and applied all rules. In the second (random) case, we have uniformly randomly activated some rules and in the last (all-0s) case, we have initially deactivated all rules. Figure 11 presents the F_{CE} that increases by using the "all-deactivated" (i.e., all-0s) rules strategy, hence consuming less energy and slightly more CO_2 Emission in contrast to the "all-activated" (i.e., all-1s) and the "random" rule strategies. In the residential flat case, starting from all-1s, moving to random and finally to all-0s, we observe an increase on the F_{CE} from approximately 6.5% to 7.5% and on the F_Γ from ≈3600 kg CO_2 to ≈3800 kg CO_2, but, respectively, there is a decrease on the F_E from approximately 9500 kWh to 8900 kWh. In the house scenario, starting from all-1s, moving to random and finally to all-0s, we observe an increase on the F_{CE} from approximately 7.0% to 7.8% and on the F_Γ from ≈8700 kg CO_2 to ≈9000 kg CO_2, but, respectively, there is a decrease on the F_E from approximately 22300 kWh to 20500 kWh. In the dorms case, starting from all-1s, moving to random and finally to all-0s, we observe an increase on the F_{CE} from

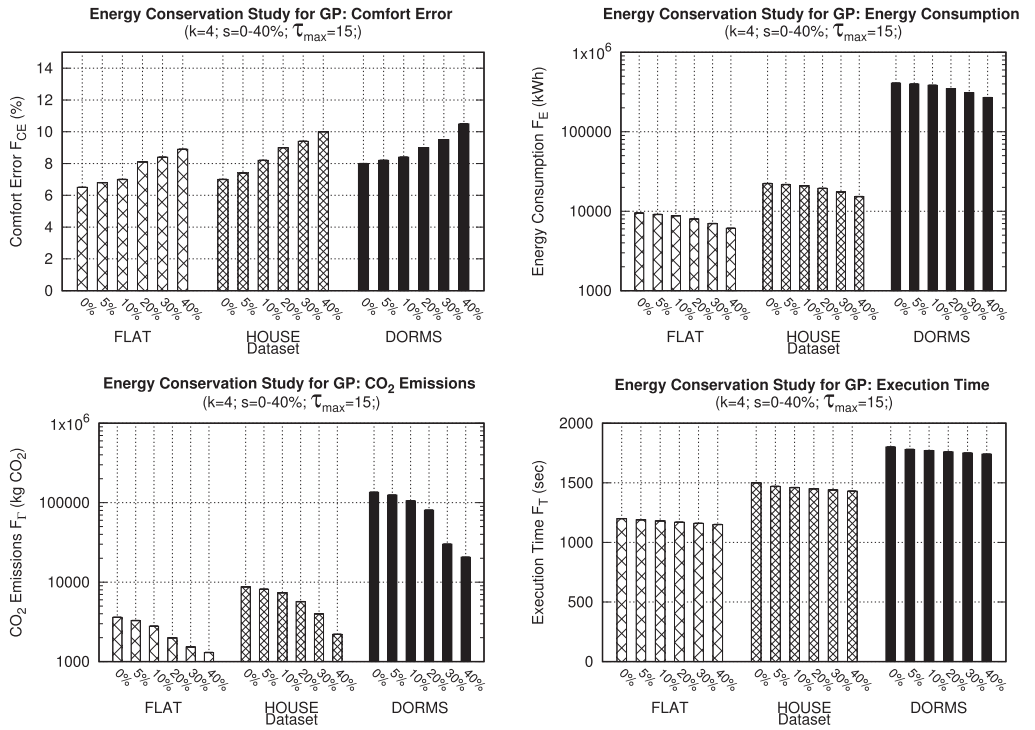

Fig. 12. Energy Conservation Study: Evaluation in terms of the comfort error (F_{CE}), the CO_2 emission (F_Γ), the CPU execution time (F_T), and the *Energy Consumption* (F_E) based on different saving values, in all datasets.

approximately 8.0% to 8.6% and on the F_Γ from ≈135,000 kg CO_2 to ≈158,000 kg CO_2, respectively, though there is a decrease on the F_E from approximately 410,000 *kWh* to 400,000 *kWh*. This is due to the hill climbing approach that needs to perform more iterations in the solution space to find the local optimum, and consequently an optimal GP, when all rules are deactivated.

6.3.4 Series-4: Energy Conservation Study. In the fourth experimental series, we evaluate the performance of the proposed GP approach over various savings percentages, with respect to Energy Consumption, CO_2 Emission, CPU Execution Time, and Comfort Error. This evaluation is inspired by the SAVES is an inter-dormitory energy-saving competition that took place on 2014–2016 and that we outlined in the introduction. SAVES aimed at delivering 8% average electricity savings in participating dormitories.

Figure 12 shows that by increasing the potential energy savings there is a slight increase on the F_{CE} and a decrease on the F_Γ clearly demonstrating the trade-off between those objectives. The trade-off ranges between 5–40% of energy savings (that is around 1,500 *kWh* and ≈500 kg CO_2 in the residential flat case) for 1–3% increase on the F_{CE} can be considered as a fair exchange.

7 CONCLUSIONS AND FUTURE WORK

In this work, we propose an innovative framework, coined *IMCF*$^+$, which aims to bridge this gap and balance the trade-off between comfort, energy consumption, and CO_2 emissions, while satisfying the RAW pipelines of users in smart environments. The IMCF$^+$ framework incorporates an innovative *GP* algorithm, which is an AI-inspired algorithm that schedules energy consumption

with a variety of amortization strategies. We have implemented IMCF$^+$ and GP as part of a complete IoT ecosystem in openHAB and our extensive evaluation on real traces from an apartment, a house, and a campus shows that we achieve a CO_2 reduction of 45–59% to satisfy the comfort of a variety of user groups with only a moderate ≈3% in reducing their comfort levels. We also found that the execution of GP is fast and efficient, carrying out the computation in about 6 seconds for the largest datasets. Given that our approach requires no training data and only a primitive MRT preference profile, this can easily integrate in low end edge smart actuations platforms, as we have demonstrated with our architecture.

In the future, we plan to investigate in further detail the interesting topic of multiple energy/GPs representing conflicting interests for the benefit of smart communities. We also aim to investigate the so-called IMCF-Cloud extensions that will enable IMCF$^+$ to operate as a CMC controller in the cloud and carry out large field studies. Moreover, the rule adaptation process is a feature that we will consider integrating in our framework in the future. Finally, we aim to investigate power workload identification methods for power-hungry devices (e.g., white devices, electric vehicles, and heating) and how to reschedule those workloads in a environmental friendly manner. Expanding the scope of IMCF$^+$ into other domains, beyond smart residences, is another interesting generalization direction of the architecture we propose in this work that we will consider for future work.

REFERENCES

[1] European Commission Energy. 2022. Hydrogen. Retrieved Jul. 1st, 2022 from https://energy.ec.europa.eu/topics/energy-system-integration/hydrogen_en.

[2] 2020. Atmosfair - Go climate conscious, promote green energy. Retrieved Jul. 1st, 2022 from https://www.atmosfair.de/en/.

[3] J. Chen, Y. Chen, Z. Chen, A. Ghazal, G. Li, S. Li, W. Ou, Y. Sun, M. Zhang, and M. Zhou. 2019. Data management at Huawei: Recent accomplishments and future challenges. In *Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 13–24.

[4] L. Yao, Q. Z. Sheng, and S. Dustdar. 2015. Web-based management of the Internet of Things. *IEEE Internet Computing* 19, 4 (2015), 60–67.

[5] A. Al Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. 2015. Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys and Tutorials* 17, 4 (2015), 2347–2376.

[6] Shancang Li, Li Da Xu, and Shanshan Zhao. 2015. The internet of things: A survey. *Information Systems Frontiers* 17, 2 (2015), 243–259.

[7] M. A. M. Albreem, A. A. El-Saleh, M. Isa, W. Salah, M. Jusoh, M. M. Azizan, and A. Ali. 2017. Green internet of things (IoT): An overview. In *Proceedings of the 2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application*. 1–6.

[8] S. Constantinou, A. Konstantinidis, D. Zeinalipour-Yazti, and P. K. Chrysanthis. 2021. The IoT meta-control firewall. In *Proceedings of the 37th IEEE International Conference on Data Engineering*.

[9] S. Constantinou, A. Vasileiou, A. Konstantinidis, D. Zeinalipour-Yazti, and P. K. Chrysanthis. 2021. IMCF: The IoT meta-control firewall for smart buildings. In *Proceedings of the 24th International Conference on Extending Database Technology*. DOI: http://dx.doi.org/10.5441/002/edbt.2021.77

[10] Will Brackenbury, Abhimanyu Deora, Jillian Ritchey, Jason Vallee, Weijia He, Guan Wang, Michael L. Littman, and Blase Ur. 2019. How users interpret bugs in trigger-action programming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. DOI: http://dx.doi.org/10.1145/3290605.3300782

[11] 2014. Erasmus student network (ESN). Retrieved Jul. 1st, 2022 from https://esn.org/saves.

[12] C. Costa, G. Chatzimilioudis, D. Zeinalipour-Yazti, and M. F. Mokbel. 2017. Efficient exploration of telco big data with compression and decaying. In *Proceedings of the 2017 IEEE 33rd International Conference on Data Engineering*. 1332–1343.

[13] Bonaventura Del Monte, Del Monte Zeuch, Tilmann Rabl, and Volker Markl. 2020. Rhino: Efficient management of very large distributed state for stream processing engines. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*.

[14] M. Poess, R. Nambiar, K. Kulkarni, C. Narasimhadevara, T. Rabl, and H. Jacobsen. 2018. Analysis of TPCx-IoT: The first industry standard benchmark for IoT gateway systems. In *Proceedings of the 2018 IEEE 34th International Conference on Data Engineering*. 1519–1530.

[15] Sameera Ghayyur, Yan Chen, Roberto Yus, Ashwin Machanavajjhala, Michael Hay, Gerome Miklau, and Sharad Mehrotra. 2018. IoT-detective: Analyzing IoT data under differential privacy. In *Proceedings of the 2018 International Conference on Management of Data*. ACM, 1725–1728. DOI: http://dx.doi.org/10.1145/3183713.3193571

[16] D. Lohani and D. Acharya. 2016. SmartVent: A context aware IoT system to measure indoor air quality and ventilation rate. In *Proceedings of the 2016 17th IEEE International Conference on Mobile Data Management*, Vol. 2. 64–69.

[17] A. Gal, A. Senderovich, and M. Weidlich. 2018. Online temporal analysis of complex systems using IoT data sensing. In *Proceedings of the 2018 IEEE 34th International Conference on Data Engineering*. 1727–1730.

[18] Andreas Konstantinidis, Panagiotis Irakleous, Zacharias Georgiou, Demetrios Zeinalipour-Yazti, and Panos K. Chrysanthis. 2018. IoT data prefetching in indoor navigation SOAs. *ACM Transactions on Internet Technology* 19, 1 (Nov. 2018), Article 10, 21 pages. DOI: http://dx.doi.org/10.1145/3177777

[19] Paschalis Mpeis, Thierry Roussel, Manish Kumar, Constantinos Costa, Christos Laoudias, Denis Capot-Ray, and Demetrios Zeinalipour-Yazti. 2020. The anyplace 4.0 IoT localization architecture. In *Proceedings of the 2020 IEEE 21st IEEE International Conference on Mobile Data Management*.

[20] C. Costa and D. Zeinalipour-Yazti. 2019. Telco big data research and open problems. In *Proceedings of the 2019 IEEE 35th International Conference on Data Engineering*. 2056–2059.

[21] Baris Aksanli, Jagannathan Venkatesh, Liuyi Zhang, and Tajana Rosing. 2011. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. In *Proceedings of the 4th Workshop on Power-Aware Computing and Systems*. ACM, Article 5, 5 pages. DOI: http://dx.doi.org/10.1145/2039252.2039257

[22] Sara Diouani and Hicham Medromi. 2019. Trade-off between performance and energy management in autonomic and green data centers. In *Proceedings of the 2nd International Conference on Networking, Information Systems and Security*. ACM, New York, NY, Article 6, 8 pages. DOI: http://dx.doi.org/10.1145/3320326.3320332

[23] Dusit Niyato, Sivadon Chaisiri, and Lee Bu Sung. 2009. Optimal power management for server farm to support green computing. In *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*. 84–91. DOI: http://dx.doi.org/10.1109/CCGRID.2009.89

[24] Zhenyu Wen, Saurabh Garg, Gagangeet Singh Aujla, Khaled Alwasel, Deepak Puthal, Schahram Dustdar, Albert Y. Zomaya, and Rajiv Ranjan. 2021. Running industrial workflow applications in a software-defined multicloud environment using green energy aware scheduling algorithm. *IEEE Transactions on Industrial Informatics* 17, 8 (2021), 5645–5656. DOI: http://dx.doi.org/10.1109/TII.2020.3045690

[25] Athanasios Tryfonos, Andreas Andreou, Nicholas Loulloudes, George Pallis, Marios D. Dikaiakos, Nikolas Chatzigeorgiou, and George E. Georghiou. 2018. ENEDI: Energy saving in datacenters. In *Proceedings of the 2018 IEEE Global Conference on Internet of Things*. 1–5. DOI: http://dx.doi.org/10.1109/GCIoT.2018.8620159

[26] M. Poess and R. O. Nambiar. 2010. Tuning servers, storage, and database for energy efficient data warehouses. In *Proceedings of the 2010 IEEE 26th International Conference on Data Engineering*. 1006–1017.

[27] Orlando Belo, Ricardo Gonçalves, and João Saraiva. 2015. Establishing energy consumption plans for green star-queries in data warehousing systems. In *Proceedings of the 2015 IEEE International Conference on Data Science and Data Intensive Systems*. 226–231. DOI: http://dx.doi.org/10.1109/DSDIS.2015.108

[28] Jilin Hu, Bin Yang, Christian Søndergaard Jensen, and Yu Ma. 2017. Enabling time-dependent uncertain eco-weights for road networks. *Geoinformatica* 21, 1 (2017), 57–88. DOI: http://dx.doi.org/10.1007/s10707-016-0272-z

[29] Ove Andersen, Christian S. Jensen, Kristian Torp, and Bin Yang. 2013. EcoTour: Reducing the environmental footprint of vehicles using eco-routes. In *Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management*, Vol. 1. 338–340. DOI: http://dx.doi.org/10.1109/MDM.2013.50

[30] Na Ta, Guoliang Li, Tianyu Zhao, Jianhua Feng, Hanchao Ma, and Zhiguo Gong. 2018. An efficient ride-sharing framework for maximizing shared routes. In *Proceedings of the 2018 IEEE 34th International Conference on Data Engineering*. 1795–1796. DOI: http://dx.doi.org/10.1109/ICDE.2018.00255

[31] Bin Yang, Chenjuan Guo, Christian S. Jensen, Manohar Kaul, and Shuo Shang. 2014. Stochastic skyline route planning under time-varying uncertainty. In *Proceedings of the 2014 IEEE 30th International Conference on Data Engineering*. 136–147. DOI: http://dx.doi.org/10.1109/ICDE.2014.6816646

[32] Samaresh Bera, Sudip Misra, and Mohammad S. Obaidat. 2014. Energy-efficient smart metering for green smart grid communication. In *Proceedings of the 2014 IEEE Global Communications Conference*. 2466–2471. DOI: http://dx.doi.org/10.1109/GLOCOM.2014.7037178

[33] Chang-Sic Choi, Jinsoo Han, Wan-Ki Park, Youn-Kwae Jeong, and Il-Woo Lee. 2011. Proactive energy management system architecture interworking with smart grid. In *Proceedings of the 2011 IEEE 15th International Symposium on Consumer Electronics*. 621–624. DOI: http://dx.doi.org/10.1109/ISCE.2011.5973905

[34] Hantao Huang, Hang Xu, Yuehua Cai, Rai Suleman Khalid, and Hao Yu. 2018. Distributed machine learning on smart-gateway network toward real-time smart-grid energy management with behavior cognition. *ACM Transaction on Design Automation of Electronic Systems* 23, 5 (Oct. 2018), Article 56, 26 pages. DOI: http://dx.doi.org/10.1145/3209888

[35] Srinivasan Keshav and Catherine Rosenberg. 2010. How internet concepts and technologies can help green and smarten the electrical grid. In *Proceedings of the 1st ACM SIGCOMM Workshop on Green Networking*. New York, NY, 35–40. DOI: http://dx.doi.org/10.1145/1851290.1851298

[36] Sarvapali D. Ramchurn, Perukrishnen Vytelingum, Alex Rogers, and Nicholas R. Jennings. 2011. Agent-based homeostatic control for green energy in the smart grid. *ACM Transactions on Intelligent Systems Technology* 2, 4 (July 2011), Article 35, 28 pages. DOI: http://dx.doi.org/10.1145/1989734.1989739

[37] Ovadia Steven. 2014. Automate the internet with "If This Then That". *Behavioral and Social Sciences Librarian* 33, 4 (2014), 208–211.

[38] Blase Ur, Melwyn Pak Yong Ho, Stephen Brawner, Jiyun Lee, Sarah Mennicken, Noah Picard, Diane Schulze, and Michael L. Littman. 2016. Trigger-action programming in the wild: An analysis of 200,000 IFTTT recipes. *In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*.

[39] Xianghang Mi, Feng Qian, Ying Zhang, and XiaoFeng Wang. 2017. An empirical characterization of IFTTT: Ecosystem, usage, and performance. In *Proceedings of the 2017 Internet Measurement Conference*. 398–404.

[40] X. Meng, W. Cong, H. Liang, and J. Li. 2018. Design and implementation of Apple Orchard Monitoring System based on wireless sensor network. In *Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation*. 200–204. DOI: http://dx.doi.org/10.1109/ICMA.2018.8484350

[41] 2019. Getting started with Apilio and IFTTT. Retrieved Jul. 1st, 2022 from https://community.apilio.com/t/getting-started-with-apilio-and-ifttt/195.

[42] S. Heo, S. Song, J. Kim, and H. Kim. 2017. RT-IFTTT: Real-time IoT framework with trigger condition-aware flexible polling intervals. In *Proceedings of the IEEE Real-Time Systems Symposium*.

[43] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah. 2019. Artificial neural networks-based machine learning for wireless networks: A tutorial. *IEEE Communication Surveys Tutorials* 21, 4 (2019), 3039–3071. DOI: http://dx.doi.org/10.1109/COMST.2019.2926625

[44] 2019. Sunny Home Manager 2.0. Retrieved Jul. 1st, 2022 from https://www.sma.de/en/products/monitoring-control/sunny-home-manager-20.html.

[45] Leonie Blume. 2015. SMA is opening its doors to third-party suppliers. Retrieved from http://tiny.cc/4jejjz.

[46] G. Kalogridis and S. Z. Denic. 2011. Data mining and privacy of personal behavior types in smart grid. In *Proceedings of the International Conference on Data Mining Workshops*.

[47] T. Yang, C. Yang, and T. Sung. 2015. An intelligent energy management scheme with monitoring and scheduling approach for IoT applications in smart home. In *Proceedings of the 3rd International Conference on Robot, Vision and Signal Processing*.

[48] A. Aryal, F. Anselmo, and B. Becerik-Gerber. 2018. Smart IoT desk for personalizing indoor environmental conditions. In *Proceedings of the 8th International Conference on the Internet of Things*.

[49] Simone Baldi, Christos D. Korkas, Maolong Lv, and Elias B. Kosmatopoulos. 2018. Automating occupant-building interaction via smart zoning of thermostatic loads: A switched self-tuning approach. *Applied Energy* 231 (2018), 1246–1258. https://www.sciencedirect.com/science/article/pii/S0306261918315095?via%3Dihub.

[50] Christos D. Korkas, Simone Baldi, and Elias B. Kosmatopoulos. 2018. Grid-connected microgrids: Demand management via distributed control and human-in-the-loop optimization. In *Proceedings of the Advances in Renewable Energies and Power Technologies*, Imene Yahyaoui (Ed.). Elsevier, 315–344. DOI: http://dx.doi.org/10.1016/B978-0-12-813185-5.00025-5

[51] Washington State University. 2020. CASAS: Center for advanced studies in adaptive systems. Retrieved Jul. 1st, 2022 from http://casas.wsu.edu/.

Received June 2021; revised April 2022; accepted June 2022