IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 21, 2018, accepted January 5, 2019, date of publication January 14, 2019, date of current version February 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2892902

Sliding Time Window Electricity Consumption Optimization Algorithm for Communities in the Context of Big Data Processing

SIMONA VASILICA OPREA, ADELA BÂRA, AND VLAD DIACONITA
Department of Economic Informatics and Cybernetics, Bucharest University of Economic Studies, Bucharest 010374, Romania

Corresponding author: Vlad Diaconita (diaconita.vlad@ie.ase.ro)

This work was supported by the European Regional Development Fund through the Competitiveness Operational Programme (2014–2020), priority axis 1–research, technological development and innovation to support economic competitiveness and business development, Action 1.1.4-Attracting high-level personnel from abroad in order to enhance the RD capacity, under Contract P_37_418 (62/05.09.2016), for the Bucharest University of Economic Studies.

ABSTRACT Big data frameworks enable companies from various fields to build models that allow them to increase profit margins by improving decision making at different levels (middle management, senior management, and board) or by attempting to boost sales by customizing consumers' experience based on their history and feedback. Institutions and other entities also use big data coming from all kinds of sensors, data that can be used to detect, in real time or in retrospect, possible problems (e.g., frauds, malfunctions, and supply shortages), or to identify patterns and trends. In this paper, we organize large volumes of community electricity consumption data coming from smart meters, smart plugs, and other sensors, but also data regarding consumers' preferences in order to assist them to dynamically optimize their electricity consumption. In this regard, we develop a novel optimization approach that re-schedules every fifteen min the appliances for residential consumers to reduce both the consumption peaks and the payments at the community level. The consumers send their day-ahead schedule that is optimized and further implemented to some extent. Thus, we monitor the electricity consumption via sensors and smart meters and dynamically adjust the schedule in case the real consumption deviates from the optimized plan, considering appliances constraints and consumers' preferences. Every fifteen min, the algorithm evaluates the differences between the optimized schedule and the actual consumption and controls the operation of the interruptible appliances to stick with the day-ahead schedule as much as possible.

INDEX TERMS Day-ahead electricity and real-time consumption optimization algorithms, big data, stream processing, smart meters, sensors.

NOMENCLATURE

Symbol	Description
n	Number of appliances;
n_I	Number of appliances of type I;
n_S	Number of appliances of type S;
n_B	Number of appliances of type B;
I	Interruptible appliance;
S	Shiftable appliance;
B	Battery (car battery);
NP	Non-programmable appliance;
h	The time interval for an hour;
$h:m$	The time interval for hour: minute used in real-time optimization (15 minutes);
TIC^h	The vector of total initial consumption of all appliances for each hour h;
TNP^h	The vector of total initial and scheduled consumption of NP appliances for each hour h;
IC_i^h	Initial consumption of appliance i at hour h,$(\forall)i = 1..n$ and $(\forall)\,h = \overline{1,24}$, forming the matrix IC of initial consumption for I, S, B type appliances;
IC_t^h	Total initial consumption of all appliances of type I, S and B at hour h,$IC_t^h = \sum_{i=1}^{n_I+n_S+n_B} IC_i^h$;
SC_i^h	Day-ahead optimized scheduled consumption of appliance i at hour h, $(\forall)\ i = \overline{1, n_I + n_S + n_B}$ and $(\forall)h = \overline{1,24}$, forming the matrix SC of optimized scheduled consumption for I, S, B type appliances;

SC_t^h	Total optimized scheduled consumption of all appliances of type I, S and B at hour $h, SC_t^h = \sum_{i=1}^{n_I+n_S+n_B} IC_i^h$;
R_i^h	Operating constraint of appliance i at hour $h, R_i^h \in \{0, 1\}$, In case, the appliance i may operate at hour h, then $R_i^h = 1$, otherwise $R_i^h = 0$; These elements are forming the matrix of operation constraint R for S, I and B type appliances;
$AVIC$	The average initial consumption of all appliances;
pk	Peak hour for the initial consumption;
opk	Off-peak hour for the initial consumption;
C_t^{pk}	Total initial consumption at peak hour;
C_t^{opk}	Total initial consumption at off-peak hour;
IC_{move}^{pk}	Total consumption that can be moved from peak hour to off-peak hour;
n_h	Number of houses;
n_{hi}^m	Number of meters in-house hi;
CSV_{hi}^j	Comma-separated files that contain appliance power readings. The original data-set included a different file for each meter j of a house hi. The timestamp of the reading is on the first field of the file (index 0) and the power readings for each device are on the following fields (indexes *1..n*);
$FCSV_{hi}$	The resulting CSV file that contains the aggregated and concatenated data for all the meters and appliances in house hi;
T<t,IC_n >	A TreeMap structure where the map is sorted according to the natural ordering of its keys (i.e. LocalDateTime of the reading t_i) and where the values are a list of the consumptions of all appliances (IC_i) serviced by a meter at those timestamps;
TJOINED<t,IC_n >	A TreeMap structure that contains all the readings for all the meters in a house;
Δc_i	Maximum acceptable deviation of the real consumption from the day-ahead schedule for appliance i;
$RSU_i^{h:m}$	User option to re-scheduled appliance i at time interval $h : m$, $RSU_i^{h:m} \in \{0, 1\}$. In case the electricity consumer chooses to deviate from the day-ahead schedule and use the appliance i at time interval *h:m,* then $RSU_i^{h:m} = 1$. By default, $RSU_i^{h:m} = 0$, meaning that the electricity consumer respects the schedule;
$Nintr_i$	Number of maximum daily interruptions for each appliance i;
$intr_i^{h:m}$	Operation status of appliance i at $h : m$ time interval. In case the appliance i is switch-off at $h : m$ then $intr_i^{h:m} = 1$; in case the appliance respects the day-ahead schedule, then no intervention is required and $intr_i^{h:m} = 0$;
$switch_i^{h:m}$	The ON/OFF operation for appliance i at $h : m$ time interval;
$RC_i^{h:m}$	Real consumption of appliance i at $h : m$ time interval gathered from meters and sensors;
$RSC_i^{h:m}$	Re-scheduled consumption determined in real time by the optimization algorithm for each appliance i at $h : m$ time interval;
FI	Flattening index
PAR	Peak to average ratio
EP_C	Electricity payment
EP_C^{ToU}	Electricity payment with a ToU tariff
EP_C^{FT}	Electricity payment with a flat or standard tariff

I. INTRODUCTION

Until not so long ago, electricity was exclusively produced in large power plants, transmitted over power lines and distributed to the residential consumers. The balancing of generation and consumption was traditionally done by the transmission grid operators that dispatched the generating units to follow the demand requirement in real time. In the last ten years, variable renewable energy sources (RES) have been massively added into the mix, creating difficulties to the dispatching centers that have to cope not only with consumption fluctuations but also with less predictable and tempestuous RES operation.

Thus, we are facing a change of paradigm: distributed generation continues to grow, bringing power generation almost at the same place with consumption. Therefore, balancing can be done not only at the generation side that is challenged by the RES integration, but also at the consumption side involving residential consumers. Such consumers account for about 30% of total electricity consumption and have the potential via ICT for using smart technologies that can change the consumption pattern and allow remote control on some appliances, based on service contracts [1], in order to facilitate the peak curtailment. Interrupting continuously operating appliances, for short time periods (i.e. from 1 to 15 minutes), such as air conditioners, ventilation systems, boilers, refrigerators have been proved to have little impact on consumers' comfort, but for grid operators and suppliers can provide a particular demand response that is proportional with the community size, flexibility and the magnitude of interrupted power. The progress of sensors and actuators-based technologies allow the consumption monitoring and control of some appliances (such as interruptible or shiftable appliances) considering favorable operating schedules. Without a doubt, the day-ahead electricity consumption optimization is an important step to alleviate the peak, while in our approach the additional effective Demand Side Management (DSM) measures for adjusting the consumption are taken at every fifteen minutes very close to the real time so that last-minute changes are adequately handled.

Many optimization methods implement optimized schedules without monitoring and controlling the operation of the appliances, thus neglecting the real-time uncertainties due to the change in consumers' preferences that could appear next day and the opportunities to reschedule the operation of some devices in order to improve the results of the optimization process. Other optimization methods don't thoroughly discuss the impact of the vast volumes of data generated, especially in urban areas, by smart appliances. Even if the payloads of individual readings are usually small, sizing in the realm of a few kilobytes, in a grid with hundreds of thousands or even millions of smart meters and intelligent plugs, digital protection devices or other types of sensors, during a year, the data readings can add up to thousands of terra bytes [2]. This electric power big data has the 4V characteristics (volume, velocity, variety, and value) of big data as showed by Zhou *et al.* [3] and further demonstrated in this paper. Storing and manipulating such data and metadata is in most cases not fit to be organized in relational databases due to its size (scale of data), variety (different schemas associated with each meter), velocity (continuous streaming flow that require rapid analysis to extract useful information before it becomes stale) and veracity (detecting noise, abnormalities or even falsifications). Big data frameworks such as Hadoop and Spark or the multitude of NoSQL databases can handle the data storage and processing problem by distributing the workload in a cluster of computer nodes.

Considering the above-mentioned characteristics of electricity consumption data, the authors of this paper address two questions: Are the current day-ahead optimization mechanisms close to the real-time operation? How can such optimization mechanisms benefit by processing both streams and batches of data with Big Data frameworks?

II. LITERATURE REVIEW

A smart grid is an electric grid that can deliver electricity in a controlled way, from suppliers to consumers, the latter having the ability to modify their behavior according to information, incentives or disincentives [4]. A smart grid usually includes smart meters, smart appliances, sensors, and plugs, together with dynamic tariffs and bidirectional communications that enable the integration of RES.

In a smart grid, the electricity consumption and lifespan of smart meters and other intelligent appliances are critical factors in applications because faulty readings from meters which exceeded their lifespan (sometimes prematurely due to environment conditions) may produce unreliable decisions. Smart meters continuously generate a large volume of data with high velocity and with various schemas that usually need to be stored before schemas can be defined. A brief introduction regarding big data and its application in the smart grid context is given by recent studies [5], presenting typical big data applications and pointing out the future challenges in electricity consumption recorded by smart meters and sensors.

Research in power big data covers a whole range of topics including power generation side management, microgrid, and RES management, asset management and collaborative operations or demand-side response.

Demand-side response (DSR) aims at reducing load burden during peak periods. Kwac and Rajagopal [6] proposed a method that uses a combinatorial optimization involving predicted electricity consumers responses and that was tested on smart meter readings from more than 58.000 households from 4 climate zones. Depending on the climate, up to 50% of the additional critical peak electricity consumption of residences consumption is generated by appliances with limited flexibility in terms of rescheduling (i.e., air conditioners, refrigerators, freezers, dehumidifiers, heaters, etc. that typically operate in the background). Nonetheless, many of these devices can be adjusted with a negligible impact on consumers' perceived comfort [7], [8], [9].

The consumption peaks can be regulated, considering priorities and without exceeding the available energy, by controlling the priority of domestic appliances, spreading at best the energy. Marah and El Hibaoui [10] attempt to solve this problem by proposing an algorithm with two phases, a priority management phase, and a branch and bound phase, relevant for addressing the underlying knapsack (rucksack) problem.

Another approach that was put forward by Duan [11] focuses on the day-ahead stage, integrating price-elastic demand bids in order to reduce the demand to average demand ratio. A Smart Home Controller strategy which provides efficient management of electric energy in a domestic environment is presented in [12]. The problem is formalized by the authors as an event-driven binary linear programming problem and takes into consideration the real power threshold, the forecast of consumption from loads that cannot be planned and various electricity tariffs and outputs the best time to run the appliances. The best time to run of smart household appliances is determined, under a virtual power threshold constraint, considering the real power threshold and the forecast of consumption from uncontrolled loads. The study reveals the consumer benefits from using local energy management systems and shows the relevance of automated demand side management. The best savings of 21.07% are obtained on a low number of loads that can be scheduled in the off-peak interval with the best tariffs.

Another approach that focuses on computing day-ahead tariffs and on estimating and refining consumers' reaction to the tariffs is proposed by Soares *et al.* [13]. The approach allows the electricity supplier to adjust dynamically the offered tariffs based on consumers' behavior.

Various demand response programs have been introduced to help the Independent System Operator (ISO) in mitigating the demand fluctuation. Current market mechanisms in several countries enable retailers or aggregators to make bilateral agreements with their consumers who would like to benefit from consumption optimization. Vuelvas et al. [14] proposed a mathematically proved incentive-based demand response program that doesn't require sending energy preferences, a program in which the aggregator randomly chooses a user to perform the energy reduction. They demonstrated how integrating price-elastic demand bids into day-ahead scheduling can effectively reduce the demand to average demand ratio.

A variation on minimizing the electricity supplier's cost takes into consideration the electricity consumers' willingness to shift their appliances usages. Kwac et al. developed an algorithm that computes day-ahead prices and another algorithm that estimates the reaction of different user classes to these prices in order to refine the provider's estimates of user behavior [15]. They investigated a household electricity segmentation methodology that uses an encoding system with a pre-processed load shape dictionary and concluded that different consumer usually might require tailored forecasting approaches. Although the primary purpose of mechanisms such as time-varying prices is to encourage consumers to reduce their consumption during high electricity demand, it is usually a hassle to residential consumers to manually adjust their loads in response to dynamic electricity prices.

Pan *et al.* [16] collected every 15 minutes, multidimensional smart meters data from 2 million households for three years in order to propose a model that starts from classifying consumers based on benchmark distributions and their price differences at peak vs. non-peak times. The approach uses a sublinear amount of the collected data, nonetheless guaranteeing small error bounds with given confidence (e.g., 95%). The model computes the theoretical profit gains of utility companies by differentiating user service accordingly.

Time-of-use tariff (time-varying prices) is the simplest and the most traditional DSM approach to minimize peak load in the system. But in a smart grid context, the existence of dynamic tariffs and bidirectional communications simultaneously allow and require an active role from the end-user concerning consumption optimization [17] enabling suppliers to create a control-feedback loop using time-dependent pricing. Such strategies make consumption optimization an essential tool in balancing markets. Large consumers who act in day-ahead and intra-day markets have to declare their consumption schedule in hourly or 15 minutes basis (depending on the settling time period of the market). They are responsible and pay if they cannot meet their schedule. For small consumers like households, retailer companies or aggregators (i.e., load serving entities (LSEs)) take this role.

The question of "How will demand response aggregators affect electricity markets?" is explicitly addressed by Chen *et al.* [18] analyzing the market effects of demand response program based on load flexibility. To some extent, the consumers can serve to aggregators as "dispatching units" for certain periods to reshape the load curve by postponing the start times or by controlling the operation time. The authors proposed an approach for such a service in which consumers can self-report their baseline and reduction capacity, given a payment scheme that includes cost of electricity, incentive price, and penalty caused by any deviation between self-reported and actual energy consumption. Different from the classic solutions, the participant agent does not require reporting energy preference, and only announces information in terms of energy. Such mechanisms proposed in the literature may open new opportunities to players in the market including consumers and aggregators.

Fuzzy logic is also addressed as a solution in the scheduling and the controlling of electric loads in residential buildings [7] where a Swarm Optimization Fuzzy Sugeno (SOFS) based energy management controller is compared with a Swarm Optimization Fuzzy Mamdani (SOFM) based controller. Moreover, in order to manage the load in a compelling fashion, a load curtailment strategy is formulated using fuzzy logic for the seasonally used electric loads.

Another important topic of research in this area deals with privacy concerns and the problems sharing refusals can pose to data collection and implicitly to DSR. Data collected from consumers' smart can reveal personal insights like the number of inhabitants in a domicile, their daily schedules, the appliances they use or even TV or multimedia content preferences [19]. Yassine *et al.* [20] propose a negotiation game theoretic mechanism between the data aggregator and the data analyst that balances benefits and privacy levels and where consumers are rewarded for their participation in the data market. In another approach [21], the privacy-cost trade-off is formulated as a convex optimization problem, and a low-complexity backward water-filling algorithm is proposed to calculate the optimal energy management policy.

Big data and clustering techniques using a voting system to select the optimal number of clusters can be used to identify electricity consumption patterns. Such an approach was put forward by Pérez-Chacón *et al.* [22] and takes advantage of the distributed version of the k-means clustering algorithm embedded in the Apache Spark's Machine Learning Library.

Not all consumers are eligible for a demand response program, so, selecting electricity consumers for demand response programs utilizing data and big data available from individual-level smart meters is a challenging problem [23]. The results of the researches show that the prediction accuracy can go to up to 90% to identify the electricity consumers that are appropriate targets for demand response program participation according to the consumption behavior and lifestyle.

Large volumes of electricity consumption data are already generated by smart meters and sensors in different cities (e.g., Shanghai) [24], requiring identification and correction of outliers, advanced analyses of consumers' pattern, the correlation between different variables and applying big data techniques for smart grid data.

Electricity consumption is a critical factor in smart grid applications. To optimize the electricity consumption and to improve the lifespan of smart meters, a knowledge-based approach for smart meters is proposed by Agnetis *et al.* [25]. The results show that the proposed method can enhance the lifespan of smart meters by up to 72% and optimizes the electricity consumption with 21%.

The smart grid concept involves sensors and smart meters that continuously generates large volume of data with high velocity and variety, strategies, methods and real-time data processing requirements. A brief introduction about energy big data applications in smart grid, mentioning recent studies, developments and security issues, is pointed out in [26], also addressing some future challenges of big data in the energy domain.

Keshtkar *et al.* [27] and later Javaid *et al.* [28] addressed problems pertinent to the lack of energy management systems. The authors proposed a flexible autonomous energy management solution for residential Heating, Ventilation, and Air Conditioning systems. Moreover, an autonomous thermostat utilizing a synergy of Supervised Fuzzy Logic Learning, wireless sensors capabilities, and dynamic electricity pricing is developed. The thermostat proposed by the authors can adjust the set point temperatures of the day without any interaction while saving energy and thereby cost, without jeopardizing users' thermal comfort. The results of these studies demonstrate that if any change occurs to user's schedules and preferences, the Adaptive Fuzzy Logic Models can effectively learn and adapt to new changes while considering energy conservation issues.

The operation of the appliances is modeled by Barker *et al.* [29] with the goal of minimizing electricity cost considering the consumers' preferences and requirements as constraints. The authors propose an electricity consumption optimization algorithm that offers a schedule for home appliances by using a mixed-integer programming technique. They proved that by adding a PV system, the electricity bill is reduced.

Scheduling a set of appliances at the electricity consumer' premises is also studied by Qayyum *et al.* [30]. The authors consider a scenario with electricity tariff variability and a photovoltaic (PV) panel functioning as a power-producing appliance that acts as a micro-grid. The problem is solved using mixed-integer linear programming without affecting climatic confort.

In this paper, we focus on managing large volumes of community consumption data to schedule the appliances for the next day optimally y and evaluate the differences between the day-ahead optimized program and the real consumption in order to assist consumers to readjust the consumption schedule and reduce costs. After investigating many approaches that were put forward by other researchers, to the best of our knowledge, there are no other studies employing a similar approach.

The paper is structured as follows: in the current section, the importance of the proposed approach and literature review are presented; in the second section, we describe the big data processing and sliding time window electricity consumption optimization algorithm, providing a flowchart that shows the steps of the methodology; in the third section, simulations are performed emphasizing the results, and in the fourth section, the conclusion is drawn.

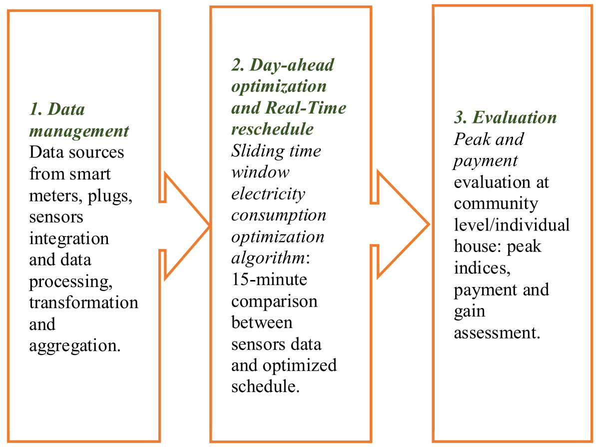

FIGURE 1. Flowchart of the methodology.

III. METHODS AND MODEL

Addressing the two questions mentioned in the introduction section, we propose a methodology described as a flowchart (shown in FIGURE 1) that consists of three steps:

- Identifying the electricity consumption *data sources* for the targeted consumers. In this study, real electricity consumption data for 2016 from 11 complex houses with numerous appliances are used [31]. The data sources consist of smart meters, plugs, and sensors that allow the monitoring and control in real time of the appliances;
- Processing and transforming the data generated by the different smart devices. This large volume of data is continuously flowing at different time resolution from various data sources. To handle it, we suggest an Elasticsearch centered architecture where the data streams are captured and transformed using Logstash and Beats;
- Considering that monitoring and controlling the appliances in smart homes by means of sensors networks and actuators are great opportunities to optimize the electricity consumption and decrease the load peak, we propose an optimization algorithm for the day-ahead schedule that takes in account the appliances based on the category they belong to. Thus, we classify the appliances into some categories without revealing their brand/model and other intrusive characteristics. The purpose of the algorithm is to minimize the consumption peak by shifting and controlling the operation of the programmable appliances that will also reduce the electricity payment by implementing different time-of-use (ToU) tariffs [32] and stimulating the consumption at off-peak hours. As a result, the day-ahead schedule is provided to the electricity consumers for next day implementation;

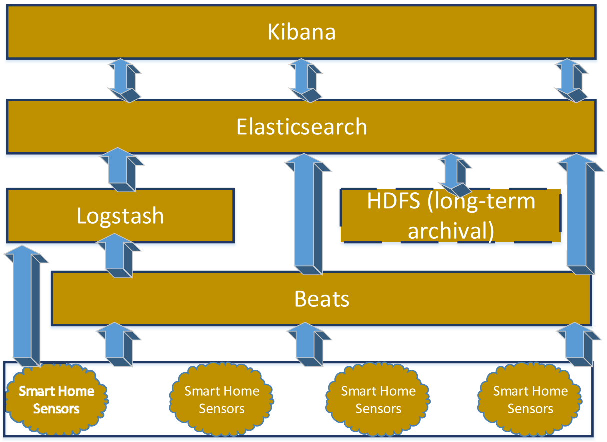

FIGURE 2. **Data management architecture.**

- In real-time operation, the electricity consumption deviates from the day-ahead optimized schedule. By monitoring and controlling the operation of the appliances to follow the schedule and to achieve the objective function, we propose a sliding time window optimization process that closely follows the consumption and corrects deviations from the day-ahead schedule program.
- In order to evaluate the efficiency of the proposed optimization approach (day-ahead and real-time optimization algorithms), we calculate two consumption peak indices: flattening index and peak to average ratio, and two relevant electricity payment gains considering four ToU tariffs.

A. DATA MANAGEMENT

To collect, store and analyze data, we favor an Elasticsearch centered architecture as described in FIGURE 2. The elastic stack can be an excellent solution to process sensor data, telemetry, and other metrics. Such data are fined-grained (usually collected at second resolution), annotated with metadata and require aggregations and roll-ups.

This architecture makes good use of the power of Elasticsearch, which is a horizontally scalable NoSQL database, being at the same time, more comfortable to implement compared to a Hadoop or Spark centered solution, and in most cases can do operations such as aggregations faster on similar clusters. Nonetheless, if needed, with the help of ES-Hadoop [33], data can be moved in both directions between Elasticsearch and Hadoop and Spark can be used for data processing or preprocessing. In such a scenario, Hadoop Distributed File System (HDFS) can function as a repository for long-term archival.

Data is captured from different smart meters, smart devices or other IoT devices. A SmartThings hub or an IoT Gateway [34] can help such devices to communicate with each other by collecting and translating different protocols and can be a point of integration from which Beats or Logstash can capture the data from a TCP socket or from a log file. There are many commercially SmartThings hubs (e.g. [35]), but then again such a device can also be built using, for example using a Raspberry Pi board, an SD card, a Zigbee USB adapter, a Wifi adapter and the Mozilla IoT Gateway [36].

For data ingestion, the architecture includes Beats, a lightweight agent that can be run on different servers or devices and Logstash that is suitable for Extract, Transform and Load (ETL) -like jobs and for connecting and accessing multiple data sources. Elastic provides Beats agents to capture Audit data (Auditbeat), Log files (Filebeat), Availability of services (Heartbeat), Metrics (Metricbeat), Network traffic (Packetbeat) and Windows event logs (Winlogbeat). The data can be sent from Beats to Logstash for further processing or directly to Elasticsearch. There is also an extensive list of community-sourced Beats [37]. The central component, Elasticsearch is a horizontally scalable solution for data storing, data searching and data analyses with built-in time series aggregating facilities, including nested aggregations searches. Kibana connects to Elasticsearch and focuses on data management and visualization.

Elasticsearch stores data in uniquely identified documents expressed in JSON. Documents reside in indexes which are collections of documents with similar characteristics (e.g., consumers, appliances, readings, etc.). Starting with version 6.0 there can be only one mapping type per index that determines how the documents are indexed. A mapping type has meta-fields and fields [38]. Large indexes can be divided into shards that can be distributed across the nodes of a cluster. For fault-tolerance and search performance, indexes can be replicated leading to primary shards and replica shards.

We stored all the real-time readings data inside an index and all the processed data inside another index. We used a three-node cluster with 5 primary shards and one complete replica. Having more shards induces overhead but can be necessary as one shard can only hold up to 2,147,483,519 documents [39]. In our case, each reading is stored in an individual JSON document.

In order to load electricity consumption data from sensors and meters into the Elasticsearch documents, a preprocessing stage is required to assure data consistency and validation checks against missing or invalid values. Another important aspect is to transform data recorded at different time intervals to a common time resolution for aggregation and optimization purposes. In our scenario, there are 11 houses with smart meters and sensors that record data in .csv files at different time resolution that varies from 1 minute up to 60 minutes. In order to run the proposed optimization algorithms, the time resolution for these records should be adjusted and set to 15 minutes. Therefore, a batch preprocessing approach is proposed that reads every .csv file sent by sensors/meters of a house (CSV_{hi}^{j}) and transforms the time resolution of the records to 15 minutes, as illustrated in Algorithm 1. We choose this approach for batch processing to have a reasonably consistent pre-processing plan for all the datasets that had different time resolutions (1 minute, 15 minutes, 30 minutes and 1 hour).

In case of one-minute time intervals, we only kept the maximum values for every quarter of the hour (0, 15, 30 and 45 minutes past the hour). In the case of 30-minute time intervals (0 and 30 past the hour), we calculated the average of two consecutive intervals to generate appropriate values for 15 and 45 pass the hour. In the rare cases when we only had data for every hour, we calculated the averages three times (once for 30 and once for 15 and 45 past the hour). For every data set, we encountered two particular cases, the change to daylight saving time in spring when for an hour (2:00-3:00 am) we had no readings and the switch back to standard time in the fall when we had two sets of recordings for one hour. In case of two sets of values for the same timestamp, we kept the first one, and in case we had no readings for an hour, we applied the average method described above.

This algorithm is also useful in the data pre-processing step for filling in missing values, resolve inconsistencies and dealing with outliers. The algorithm is implemented in Java which has TreeMap, an implementation of the NavigableMap interface where the map is sorted according to the natural order of the keys. In our case, the map was sorted according to the timestamp of the readings. With some adjustments, it can be implemented in other languages that don't implement a TreeMap structure but implement map, hash table or dictionary structures.

As a result, the records for each house are stored in one .csv file ($FCSV_{hi}$), having the time resolution set to 15 minutes. The final .csv files for all houses are then loaded into Elasticsearch as indexed documents and are used in the optimization process.

B. SLIDING TIME WINDOW ELECTRICITY CONSUMPTION OPTIMIZATION ALGORITHM

For the optimization process (step 2 of the flowchart), different types of appliances are considered with their operation constraints provided by the electricity consumers for the next day.

C. TYPOLOGY OF APPLIANCES

For day-ahead and real-time electricity consumption optimization approach (FIGURE 3), we classify the devices into three main categories:

a) **NP** representing the non-programmable appliances that include TV, laptops, lights, etc. Although most of these appliances are interactive and have little scheduling flexibilities, they are involved in the optimization process since they are a significant part of the total consumption.

b) **S** representing shiftable or programmable appliances without interruption that admit flexible delays in operation, such as washing machines, dishwashers, bread machines, etc. Their main characteristic is that they could be switched on almost any time, but the interruption of operation is not feasible (once they start, they should operate without disruption).

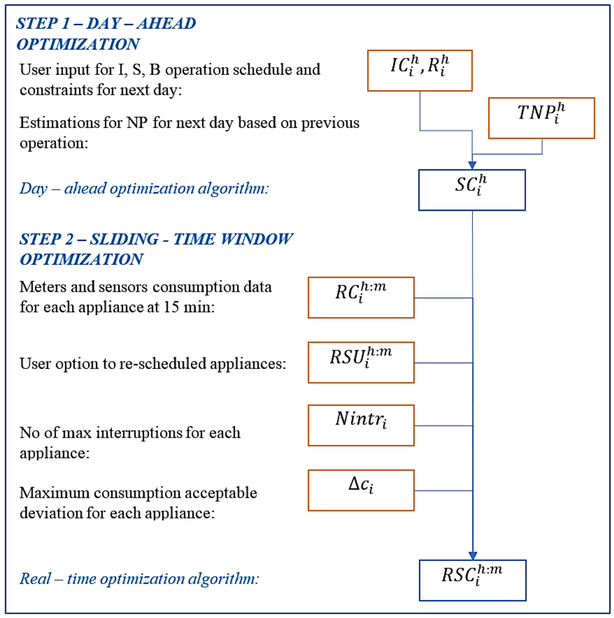

FIGURE 3. Optimization process for day-ahead and sliding time window interval.

Thus, they can be scheduled for the day-ahead optimization stage and significantly influence the daily load curve.

c) **I** - representing interruptible appliances (e.g., ACs, freezers, and refrigerators) that can be either ON with fixed consumption, OFF for shorter time intervals or dimmable according to the intrinsic characteristics of each appliance. Some I appliances (e.g., water heaters, furnaces, etc.) can be interrupted and shifted in order to meet the objective function of the day-ahead / real-time optimization process. Thus, the appliances can be remotely monitored and controlled by the supplier or other service providers based on service contracts in which consumers are compensated for partially giving up control over certain appliances. However, their ON/OFF cycle duration is based upon consumers' preference setting. This type of appliance is used in both day-ahead and real-time optimization due to their high operating flexibility.

d) **B** appliances are for storage use (batteries), electric household appliances with batteries (such as vacuum cleaners) or electric vehicles batteries that can be charged from the grid. They can be scheduled for both day-ahead and real-time optimization stages. However, they are not used as storage capacities at the community level since multiple charging/discharging cycles lead to a tremendous lifetime decrease.

For NP appliances, some estimations of the total consumption based on their previous operation are considered in the optimization algorithm, while for programmable appliances the preferences of the consumers prevail, and for controlled appliances, the preferences of the supplier in terms of consumption peak are also taken into account. Since non-programmable devices are numerous (sending preferences for them would hassle the consumers) and do not contribute to the improvement of the schedule, their operation for the following time interval is scheduled as the current real consumption as in naïve forecasting approaches.

For day-ahead optimization, the objective function is peak minimization by shifting programmable appliances (type I, S, B) from peak to off-peak hours considering their operation constraints and the consumers' preferences. Therefore, the initial hourly consumption of each appliance (IC_i^h) and the appliance constraints (R_i^h) for the next day are sent by the consumer via web-interface and stored in the Elasticsearch indexes. In the case of non-programmable appliances, the electricity supplier may estimate the total consumption (TNP^h) for the next day based on the previous day. The optimization algorithm takes these elements as inputs, performs peak minimization and provides the hourly schedule for each appliance (SC_i^h). As a result, this rescheduled program is sent back to the consumers for the next day implementation. The electricity consumers may choose to follow the schedule considering incentives and contract regulations. The electricity consumers may change this schedule in case of some unexpected events, in this case, they will set $RSU_i^{h:m} = 1$.

For real-time optimization, the real consumption of each appliance $RC_i^{h:m}$recorded by smart meters is compared with its schedule and major deviations greater than an accepted value (Δc_i) are corrected by interrupting that appliance. An individual appliance will not be interrupted for two consecutive time intervals or more than its maximum allowed interruptions per day ($Nintr_i$). As a result, reschedule consumption is calculated for each appliance ($RSC_i^{h:m}$).

1) DAY-AHEAD OPTIMIZATION ALGORITHM

The day-ahead optimization algorithm schedules the appliances based on their operating constraints set by the consumer by moving I, S and B appliances from peak hours to off-peak hours, thus flattening the consumption curve. First, the proposed algorithm evaluates the total hourly consumption of the appliances (TIC^h) based on the initial schedule sent by consumers for day-ahead.

$$TIC^h = TNP^h + IC_t^h, \quad (\forall)h = \overline{1, 24} \tag{1}$$

In order to flatten the consumption curve, the peak should be decreased as much as possible to the average consumption by shifting the operation of I, S, B appliances to the off-peak hours. To facilitate this, the average, peak and off-peak consumption are determined.

$$AVIC = \frac{1}{24}\sum_{h=1}^{24} TIC^h \tag{2}$$

$$IC_t^{pk} = Max(IC_t^h), \quad (\forall)h = \overline{1, 24} \tag{3}$$

$$IC_t^{opk} = Min(IC_t^h), \quad (\forall)h = \overline{1, 24} \tag{4}$$

For the peak hour, the algorithm determines the total consumption of appliances of type I, S and B that can be shifted to the off-peak hour by evaluating the operating constraints of these appliances.

$$IC_{move}^{pk} = \sum_j IC_j^{pk}, \quad (\forall)\, j \in \{I, S, B\} \tag{5}$$

For each appliance j that will be shifted, the algorithm will check the following set of conditions:

C1 – the initial peak consumption of appliance j should be greater than zero: $IC_j^{pk} > 0$;

C2 – the appliance can operate at off-peak hour: $R_j^{opk} = 1$

C3 – the appliance should not already operate at off-peak hour: $IC_j^{opk} = 0$;

C4 – the initial peak consumption minus the total moved consumption should not be less than the average consumption: $IC_t^{pk} - IC_{move}^{pk} \geq AVIC$;

C5 – the newly scheduled consumption at off-peak hour should be less than the initial peak consumption: $IC_t^{opk} + IC_{move}^{pk} < IC_t^{pk}$.

The appliances are shifted from peak to off-peak hours in the ascending order of their operating constraints, so the matrix R is sorted in the ascending order of the total restrictions for each appliance. Then, the average consumption is computed according to equation (2) and the algorithm iterates from equation (3) until $IC_{move}^{pk} = 0$ for every hour considered as peak/off-peak. The pseudo-code is detailed in Algorithm 2.

The flowchart of the day-ahead optimization algorithm is represented in FIGURE 4.

The algorithm will set the optimized values for electricity consumption for each appliance to SC array and discard the changes of IC array. This is done in order to preserve the initial consumption values to further compare the electricity payment between the initial schedule and the real-time consumption. Afterwards,the schedule for day-ahead consumption of each appliance is sent to electricity consumers for next day implementation.

2) REAL-TIME OPTIMIZATION ALGORITHM

The algorithm receives from meters the real consumption of each appliance for every time frame interval (15 minutes) and compares it with the scheduled consumption. The objective of the real-time algorithm is to control the operation of appliances in order to minimize their deviation from the day-ahead schedule. Thus, the appliances that are under the supplier control (type I and B) will be switched on in case they are not already operating according to the schedule and the appliances that are operating at unplanned hours or are consuming more than the schedule will be switched off.

In real time conditions, the operation of the appliances may slightly deviate from the day-ahead schedule, so the algorithm should consider an acceptable rate of deviation for each appliance (Δc_i). Also, due to some unplanned events, the electricity consumers may choose to use some appliances that were not considered in the initial schedule or were scheduled at a different hour. In this case, the consumer option to re-schedule appliance i at time interval h:m is represented by the variable $RSU_i^{h:m}$ set to 1. In this case, the operation of appliance i will not be changed.

Another important aspect that must be considered in switching off the appliances is their maximum number of interruptions over the last 24 hours. So, according to the

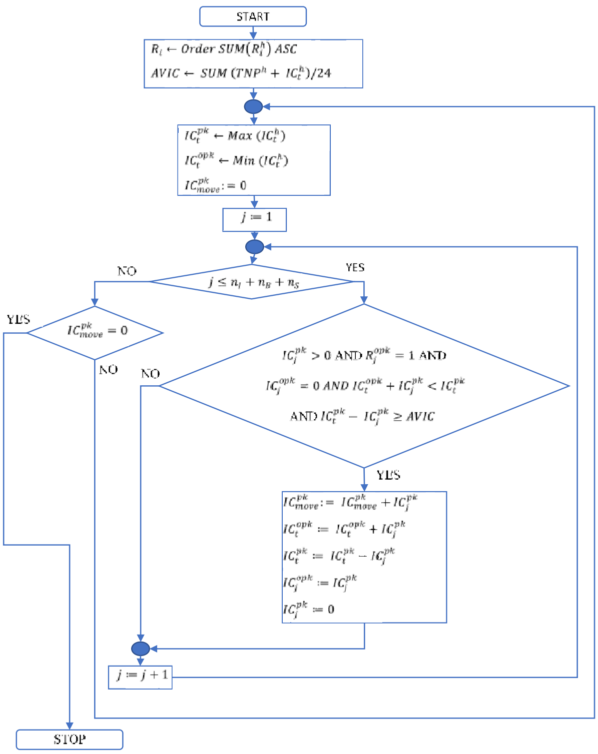

FIGURE 4. **The flowchart of the day-ahead optimization algorithm.**

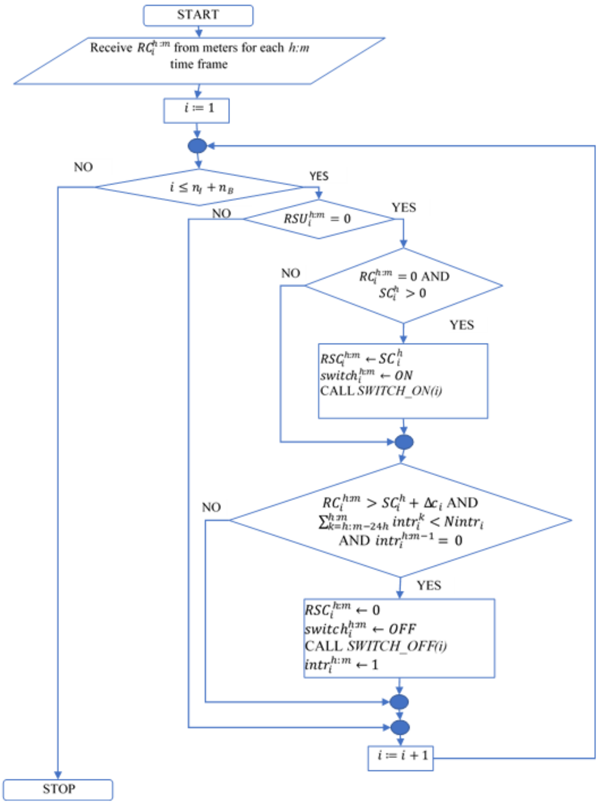

FIGURE 5. **The flowchart of the real-time optimization algorithm.**

operating conditions of each appliance, the value of $Nintr_i$ is checked. Also, an appliance will not be switched-off at two consecutive time intervals. Synthetizing, the above conditions can be formalized as follows:

C1 – check if the appliance is not re-scheduled by the electricity consumer: $RSU_i^{h:m} = 0$;

C2 – check if the appliance is scheduled for operating and its real consumption is zero: $RC_i^{h:m} = 0$ AND $SC_i^h > 0$. In this case, the appliance is switched-on and its consumption is set to the schedule:

$$RSC_i^{h:m} \leftarrow SC_i^h \tag{6}$$

$$switch_i^{h:m} \leftarrow ON \tag{7}$$

C3 – check if the real consumption of the appliance is greater than the accepted deviation from the scheduled: $RC_i^{h:m} > SC_i^h + \Delta c_i$;

C4 – in case the condition C3 is true, check if the maximum number of interruptions over the last 24 hours has not exceeded and if it was not interrupted in the previous time frame: $\sum_{k=h:m-24h}^{h:m} intr_i^k < Nintr_i$ AND $intr_i^{h:m-1} = 0$. In this case, the appliance is switched-off and the value of $intr_i^{h:m}$ is set to 1:

$$RSC_i^{h:m} \leftarrow 0 \tag{8}$$

$$switch_i^{h:m} \leftarrow OFF \tag{9}$$

$$intr_i^{h:m} \leftarrow 1 \tag{10}$$

Variable $switch_i^{h:m}$ is set to on/off in case the deviation from the schedule is greater than Δc_i and the Algorithm 3 calls routines for switching on/off the appliances.

The flowchart of the real-time optimization algorithm is represented in FIGURE 5.

IV. EVALUATION

In order to evaluate the performance of the day-ahead and real-time optimization algorithms in terms of peak reduction, the following indices are calculated: flattening index (FI) and peak to average ratio (PAR). FI represents the ratio between the average consumption and the total peak consumption; it varies from 0 to 1, increasing to 1 as the daily load flattens, while PAR is the ratio between the total peak consumption square and the average consumption square.

$$FI = \frac{AVG(C)}{C_t^{pk}} \tag{11}$$

$$PAR = \frac{{C_t^{pk}}^2}{AVG(C)^2} \tag{12}$$

These two indices can be calculated for specific days or for periods of time (monthly or annually) and give an indication of the performance of the optimization algorithms. They can be determined and compared at different stages (initial consumption, day-ahead optimization, consumption, real-time

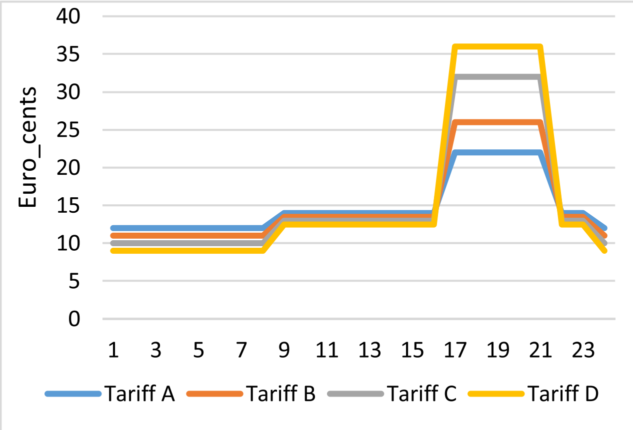

FIGURE 6. ToU tariffs proposed for payment evaluation.

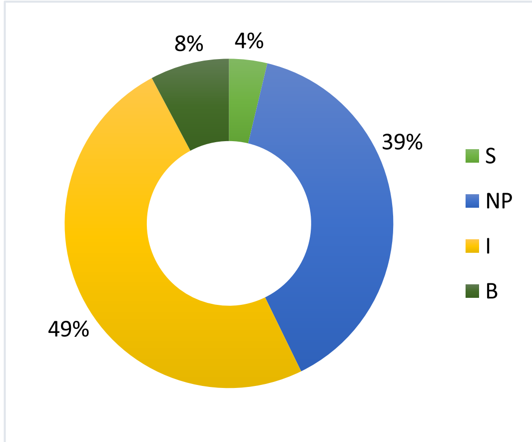

FIGURE 7. Appliances types belonging to the 11 houses.

optimization or re-scheduling) to identify the demand potential to reduce the peak.

For electricity payment evaluation, four ToU tariffs (as in FIGURE 6) are applied to estimate the electricity consumption expenses with day-ahead and real-time optimization in comparison with a flat standard fare of 16.2 Euro cents, adapted after [32].

The ToU peak rates (between hours 17 and 22) are higher than the standard tariff; A and B tariffs are milder in terms of peak rates compared with tariffs C and D, but the other rates of tariffs A and B (off-peak and day rates) are higher, compensating the difference. In other words, C and D discourage the consumption at peak hours and better reward the consumption at off-peak hours.

For tariffs evaluation and to measure the effect of day-ahead and real-time schedule on the electricity payment, we propose the subsequent gains.

Gain 1 represents the benefit if the consumers would follow the day-ahead schedule (*SC*) or real-time schedule (*RS*) in comparison with real consumption.

$$Gain1SC \rightarrow RC = 100 - \frac{EP_{SC}}{EP_{RC}} \times 100 \tag{13}$$

$$Gain1RS \rightarrow RC = 100 - \frac{EP_{RS}}{EP_{RC}} \times 100 \tag{14}$$

Gain 2 is the benefit if the consumers would follow the day-ahead schedule (*SC*), real-time schedule (*RS*) or real consumption (*RC*) with different ToU tariffs instead of the flat tariff.

$$Gain2SC = 100 - \frac{EP_{SC}^{ToU}}{EP_{SC}^{FT}} \times 100 \tag{15}$$

$$Gain2RC = 100 - \frac{EP_{RC}^{ToU}}{EP_{RC}^{FT}} \times 100 \tag{16}$$

$$Gain2RS = 100 - \frac{EP_{RS}^{ToU}}{EP_{RS}^{FT}} \times 100 \tag{17}$$

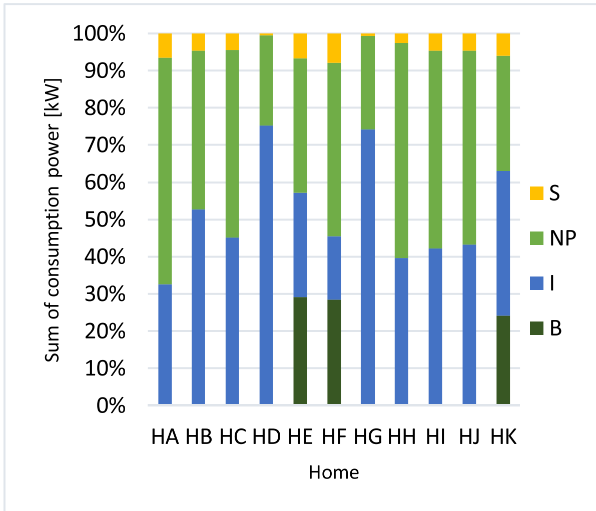

FIGURE 8. Distribution of appliances at house level based on the type of appliances.

V. INPUT DATA DEPICTION

The dataset contains consumption records of 314 appliances measured in 2016 that belong to 11 modern houses that form a community. Based on the operating particularities and consumers' requirements, appliances are classified into several categories: Non-programmable (NP), Interruptible (I), Shiftable (S) and Batteries (B). The number and percentage of appliances according to their types (NP, I, S, B) for each house are depicted in Table 1 and FIGURE 7.

As depicted in FIGURE 7, NP appliances represent 39%, I appliances represent 49%, while B and S appliances represent only 8% and 4% respectively of the total consumption at the community level. Basically, the fixed consumption of NP appliances represents 39%, while the variable consumption (S, I, B appliances) represents 61% of the total consumption.

FIGURE 8 shows the potential of each house to contribute to the optimization process. Only 3 houses (HE, HF and HK)

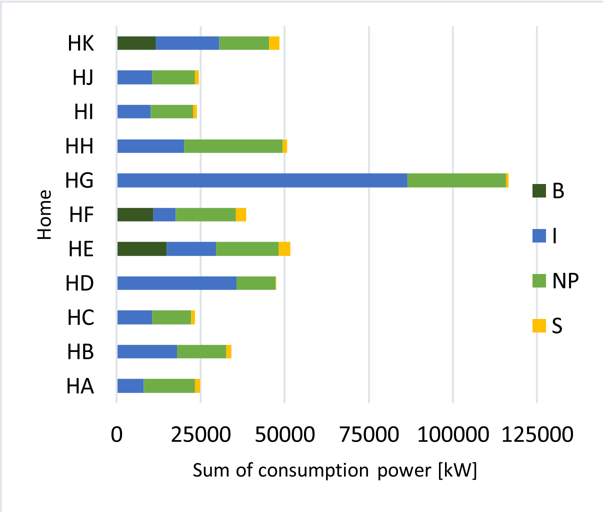

FIGURE 9. Sum of consumption power for each type of appliance/house.

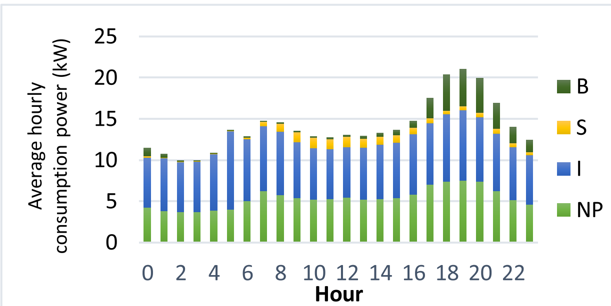

FIGURE 10. Daily load profile at the community level.

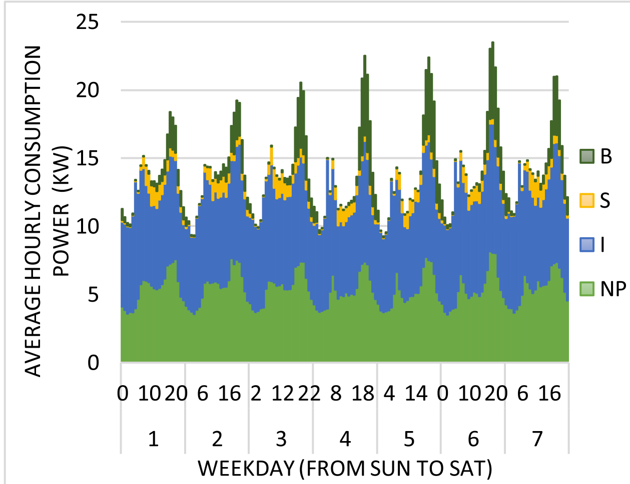

FIGURE 11. Load profiles for each weekday at the community level.

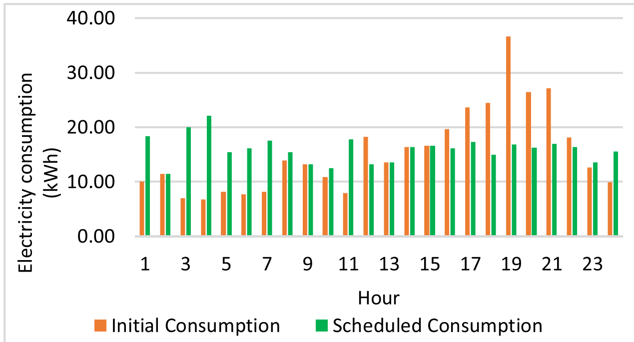

FIGURE 12. Hourly electricity consumption before/after day-ahead optimization (working day).

have all four components (S, NP, I appliances and B), while the other houses have NP, I and S appliances.

FIGURE 9 shows the level of consumption power in kW for each house recorded in 2016. House G had the highest consumption, while the lowest consumptions were recorded for HA, HB, HC, HI, and HJ.

The daily load profile of the entire community represents the average hourly consumption of each appliance type, as shown in FIGURE 10. The fixed or NP consumption (in light green) is positioned at the bottom of the curve forming the consumption baseline that cannot be altered. The curve described in FIGURE 10 shows two consumption peaks: morning (around 7:00) and evening (around 19:00) peaks and two consumption off-peaks: night (around 3:00) and afternoon (around 11:00) peaks, providing enough space for shifting the operation of the appliances to the valleys of the curve. The proportion of NP and I appliances is almost equally distributed for the 24-hour intervals. The batteries (B) are charging mostly during the evening, while S appliances are operating mainly during the day (especially in the morning and afternoon).

Batteries could be assimilated to the S or I appliances rather than considering them as storage capacities due to the lifetime reduction issue in case of daily repeated charging/discharging cycles. The load profiles representing the average hourly consumption by appliances for each weekday (where 1 is for Sundays and 7 for Saturdays) at the community level are illustrated in FIGURE 11.

The highest consumption was recorded on Fridays, while the lowest consumption was recorded on Sundays. Also, the shape of consumption curves reflects the weekday consumers' activities.

VI. SIMULATIONS AND RESULTS

A. DATA SOURCES DESCRIPTION AND MANAGEMENT

The data is recorded by smart meters and sensors into separate .csv files. Each house has more than one smart meter/sensors, so the data is split into 2 or more flat files depicting reads from several meters servicing various appliances within that house. The number of attributes and the number of lines are different, even for data coming from the meters of the same house. The schemas are also different (i.e., the number of attributes) because each meter serviced a variable number of appliances and the number of records also varied due to different resolutions. To exemplify, for a meter from house A we have data for 8 devices for the interval

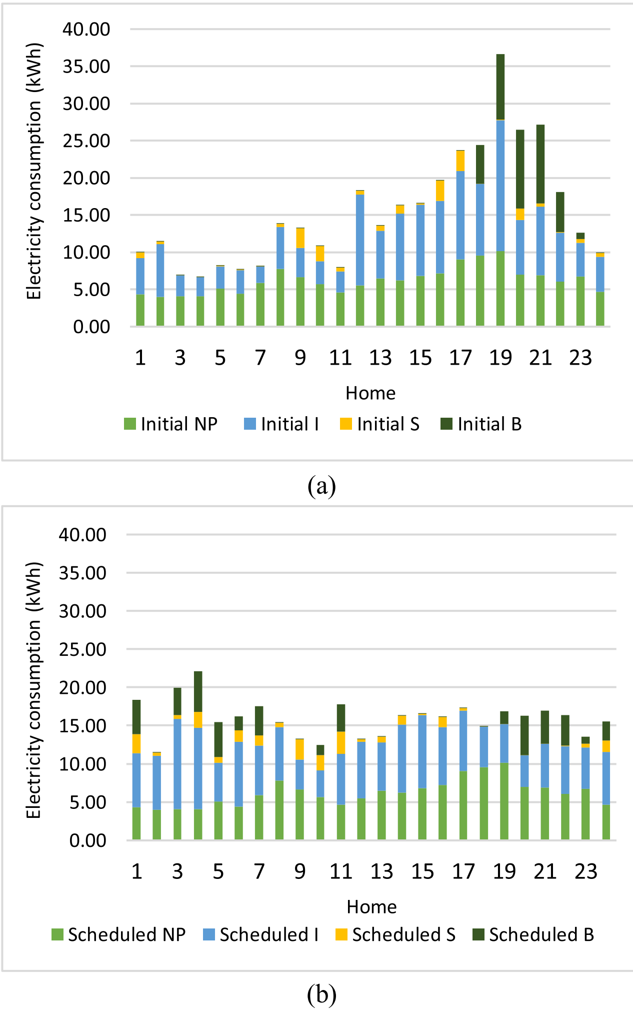

FIGURE 13. Consumption before (a)/after day-ahead optimization (b) based on the type of appliances.

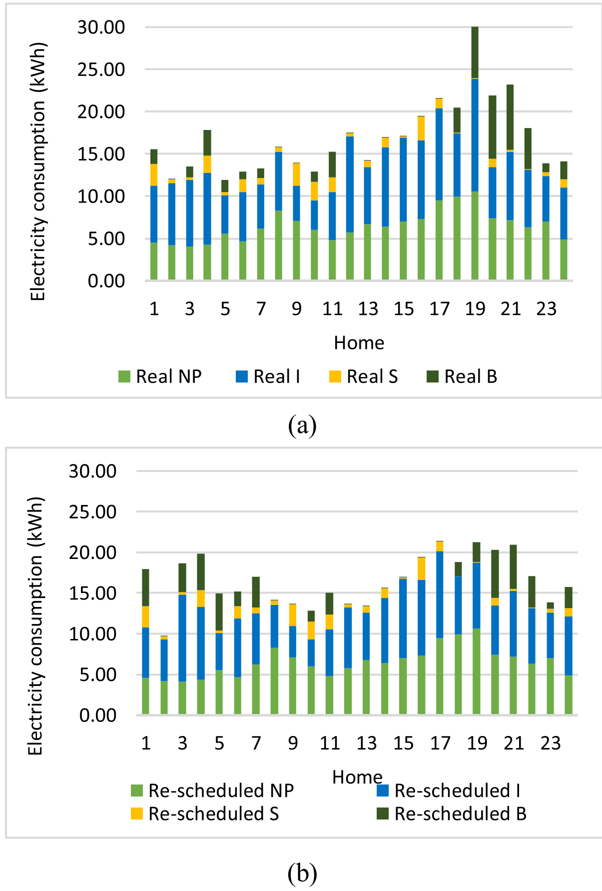

FIGURE 14. Real (a) and rescheduled electricity consumption (b).

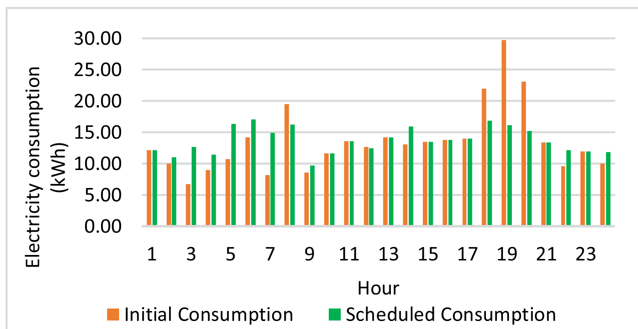

FIGURE 15. Hourly electricity consumption before/after day-ahead optimization for a weekend day.

1-January-2016 00:00 to 15-December-2016 20:59 at a one-minute resolution (503.820 records in total). For another meter of the same house we have data for 11 devices at a thirty-minute resolution between 1-January-2016 00:00 and 20-July-2016 11.30 and at a one-minute resolution and from 20-July-2016 11.31 and 31-December-2016 23:59 (246.639 records). For the third meter of the same house, we have data for 12 devices at a fifteen-minute resolution between 1-January-2016 00:00 and 19-January-2016 1:45 and at a one-minute resolution between 19-January-2016 2:00 and 31-December-2016 23:59 (502.743 records in total). For other houses, in rare cases, we have data at a one-hour resolution. Given the fact that our algorithms evaluates the differences between the optimized schedule and the actual consumption and suggests a new re-optimized program, we pre-processed the data and transformed it to a fifteen-minute time resolution.

In case of real-time processing of the data streams, we calculated in Elasticsearch the average consumption every fifteen minutes for each interruptible (I) and shiftable (S) appliances of each house (as in Query 1).

For real-time optimization, the query performance is a very important aspect that should be considered. Therefore, apart from the existing data set, to test the real-time processing of data, after loading the initial data into an Elasticsearch cluster (3 nodes), we generated streams of data, every second, from 10 different computers using multiple threads, simulating continuous readings from a total of 50 devices. To ensure that the stream values are realistic, we used a Monte Carlo simulation to randomly sample data from the actual readings of a corresponding appliance (*j*) for each of the 50 simulated appliances, as shown in algorithm 4 implemented in Java.

Query 1 Aggregate Data in Elasticsearch

```
{
    "query": {
    "bool": {
        "must": {"terms": {"type": ["i", "s"]}},
        "filter": {"range": {"date_time": {"gte": "now-
15m/m","lt": "now/m"}}}
        }
},
    "aggs": {
    "group_by_day": {
      "date_histogram": {
        "field": "date_time",
        "interval": "15m"
      },
    "aggs": {
        "group_by_Type": {
          "terms": {
          "field": "appliance"
          }
       ,
        "aggs": {
        "type": {
         "terms": {
         "field": "type"},
         "aggs": {"avg_rc" :{"avg" : {"field" : "rc"
}},
                "avg_sc" :{"avg" : {"field" : "sc" }}
                } } } }}}}}
```

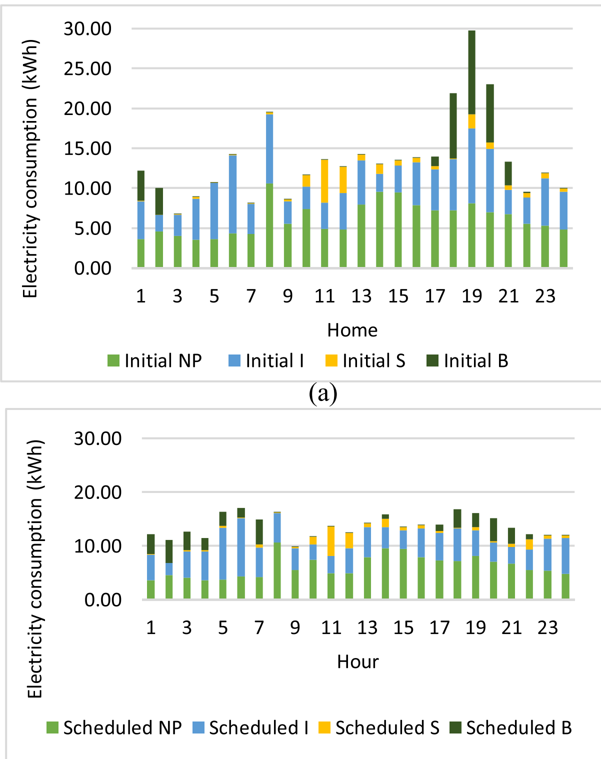

FIGURE 16. Consumption before (a)/after day-ahead optimization (b) based on the type of appliances.

The Logstash TCP/IP plugin [40] was set up to listen on the port the messages are written by Algorithm 2. As a filter, we used the Dissect plugin [41] to separate the data and output it to Elasticsearch. We didn't experience any noticeable lags in data ingestion and in processing aggregating queries such as the one in Query 1.

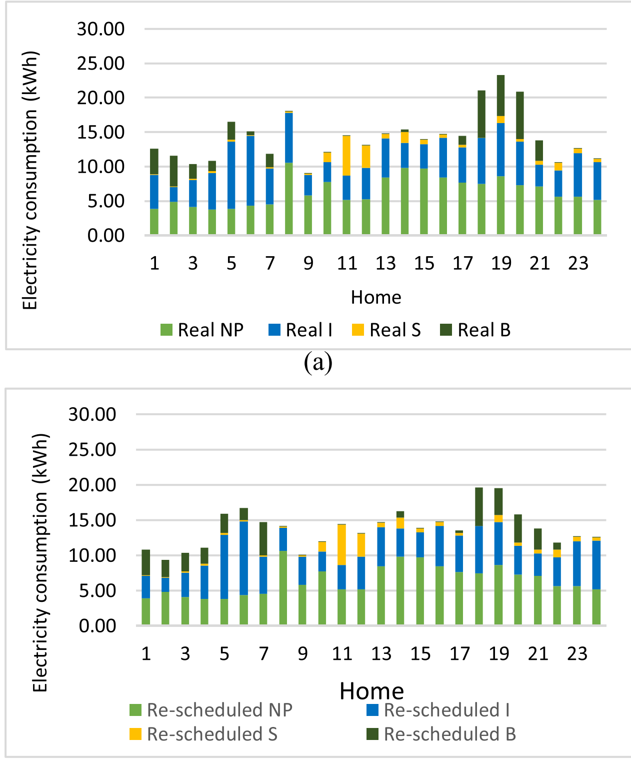

FIGURE 17. Real (a)/rescheduled electricity consumption (b).

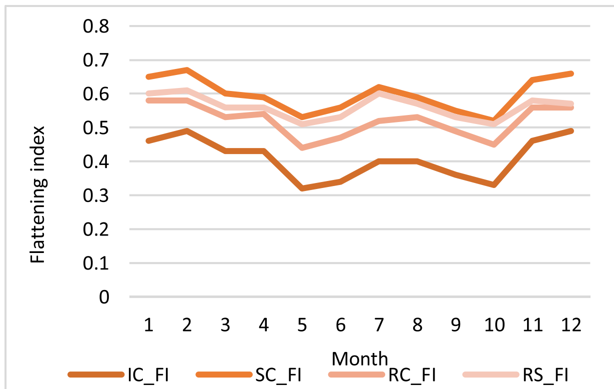

FIGURE 18. Monthly flattening index.

B. OPTIMIZATION ALGORITHMS

The proposed day-ahead and real-time optimization algorithms described in section 3 are implemented in Python and the aggregated data from Elasticsearch is retrieved into Pandas DataFrames for processing. Based on the available data sets, we perform the optimization process for each day of the entire period (year 2016). To evaluate its performance, we randomly choose two individual examples for a working day (1-July-2016) and a weekend day (1-October-2016) which are discussed in the following paragraphs. In FIGURE 12, the initial (according to consumers' schedule) and scheduled hourly electricity consumption (based on day-ahead optimization algorithm) for 1-July-2016 is shown.

FIGURE 19. Monthly peak to average ratio.

TABLE 1. Number of appliances.

No.	House	NP	I	S	B	Total
1	HA	24	4	3	0	31
2	HB	20	4	3	0	27
3	HC	14	3	2	0	19
4	HD	48	14	3	0	65
5	HE	11	6	3	1	21
6	HF	11	6	3	1	21
7	HG	28	13	3	0	44
8	HH	19	5	3	0	27
9	HI	14	3	2	0	19
10	HJ	14	3	2	0	19
11	HK	11	6	3	1	21
12	Total	214	67	30	3	314

The hourly electricity consumption before optimization based on the type of appliances shows a higher peak of around 36 kWh at 19:00 that decreased after optimization to around 22 kWh (as in FIGURE 13).

In real time, the electricity consumers may not stick to the schedule, and for some appliances, significant deviations are encountered. Thus, the actual consumption as in FIGURE 14 (a) deviates from the day-ahead optimized consumption being necessary to reschedule the consumption as in FIGURE 14 (b) based on day-ahead schedule and the changes from the previous 24 hours that the consumers may impose.

In FIGURE 14 (a) real consumption on 1-July-2016 is depicted, while in (b) the re-scheduled consumption at 15 minutes is simulated sticking as much as possible with the day-ahead schedule and considering the consumers' preferences expressed between the day-ahead optimization and the real consumption (i.e., when they change the initial preferences).

In Table 2, the optimization performance indices for 1-July-2016 are shown.

In FIGURE 15, the initial and scheduled hourly electricity consumption for 1-October-2016 is shown.

Algorithm 1 Preprocess the Data in Batch

```
DEFINE TreeMap<LocalDateTime, List<Float>> T
FOR hi=1 TO n_h
   FOR j=1 TO n^m_hi
    FOR RECORD     : CSV^j_hi
     IFRECORD.getMinute()in (0, 15, 30, 45) THEN
        T < t,IC_hi> .setKey() ← RECORD [0]
        T < t,IC_hi> .setValue() ← RECORD[1..n]
     END IF
    END FOR
    FOR ENTRY      : T < t,IC_hi>.entrySet()
       t_1 = ENTRY.getKey()
       t_2 = ENTRY.next().getKey()
       IF t_1.getMinute()−t_2.getMinute()IN(30, −30) THEN
         IC_j = ENTRY.getValue()
        FOR j=1 TO IC_n.Size()
         T.set(t_1+15 minutes, AVG (T.get (t_1) [j], T.get(t_2)[j])
        END FOR
        END IF
       IFt_1.getMinute() = 0 ANDt_2.getMinute() = 0 THEN
        FOR j=1 TO IC_n.Size()LOOP
        T.set(t_1+30 minutes, AVG (T.get (t_1) [j], T.get(t_2)[j])
        T.set(t_1 + 15 minutes, AVG (T.get(t_1)[j],
        T.get(t_1+30 minutes)[j])
           T.set(t_1 + 15 minutes, AVG
       (T.get (t_1+30 minutes) [j], T.get(t_2)[j])
        END FOR
        END IF
    END FOR
    TJOINED<t, IC_n >=TJOINED<t, IC_n >⋈TJOINED.t
     =T.t T<t, IC_n >
    FOR ENTRY      : TJOINED < t,IC_hi>.entrySet()
     t_1 = ENTRY.getKey()
     t_2 = ENTRY.next().getKey()
    IF t_2.getMinute()<>t_1.getMinute()+15minutes THEN
       RAISE ERROR
    END IF
    FCSV_hi ← TJOINED<t,IC_n >
   END FOR
  END FOR
END FOR
```

TABLE 2. PEAK indices calculated for working day 1-July-2016.

	FI	PAR
Before optimization (initial schedule)	0.25	16.52
Day-ahead optimization (optimized schedule)	0.56	3.18
Real consumption	0.35	8.37
Re-scheduled consumption (real-time optimization)	0.58	2.99

Hourly electricity consumption before optimization based on the type of appliances shows a high peak of almost 30 kWh at 19:00 that decreased after optimization to around 16 kWh, as in FIGURE 16.

In FIGURE 16 (a), the real consumption for a weekend day is depicted, while in (b) the re-scheduled consumption is simulated sticking as much as possible with the day-ahead schedule and considering the consumers' preferences expressed between day-ahead optimization and real consumption when

Algorithm 2 The Day-Ahead Optimization Algorithm

$R_i \leftarrow OrderSUM\left(R_i^h\right) ASC$
$AVIC \leftarrow SUM\left(TNP^h + IC_t^h\right)/24$
REPEAT
  $IC_t^{pk} \leftarrow Max(IC_t^h)$
  $IC_t^{opk} \leftarrow Min(IC_t^h)$
  $IC_{move}^{pk} := 0$
    FOR $j=1$ TO $n_I + n_B + n_S$
      IF $IC_j^{pk} > 0$ AND $R_j^{opk} = 1$ AND $IC_j^{opk} = 0$ AND
      $IC_t^{opk} + IC_j^{pk} < IC_t^{pk}$ AND $IC_t^{pk} - IC_j^{pk} \geq AVIC$ THEN
        $IC_{move}^{pk} := IC_{move}^{pk} + IC_j^{pk}$
        $IC_t^{opk} := IC_t^{opk} + IC_j^{pk}$
        $IC_t^{pk} := IC_t^{pk} - IC_j^{pk}$
        $IC_j^{opk} := IC_j^{pk}$
        $IC_j^{pk} := 0$
      END IF
    END FOR
UNTIL $IC_{move}^{pk} = 0$
FOR $i=1$ TO n
    $SC_i^h \leftarrow IC_i^h$
END FOR

Algorithm 3 The Real-Time Optimization Algorithm

FOR every $h:m$ time interval
    FOR $i=1$ TO $n_I + n_B$
      IF $RSU_i^{h:m} = 0$ THEN
        IF $RC_i^{h:m} = 0$ AND $SC_i^h > 0$ THEN
          $RSC_i^{h:m} \leftarrow SC_i^h$
          $switch_i^{h:m} \leftarrow ON$
          CALL *SWITCH_ON(i)*
        ELSE
          IF $RC_i^{h:m} > SC_i^h + \Delta c_i$ AND $\sum_{k=h:m-24h}^{h:m}$
          $intr_i^k < Nintr_i$ AND $intr_i^{h:m-1} = 0$ THEN
            $RSC_i^{h:m} \leftarrow 0$
            $switch_i^{h:m} \leftarrow OFF$
            CALL *SWITCH_OFF(i)*
            $intr_i^{h:m} \leftarrow 1$
          END IF
        END IF
      END IF
    END FOR
END FOR

they change their announced preferences. In case of the weekend day, the actual consumption also deviates from the day-ahead optimized consumption being necessary to reschedule the consumption as in FIGURE 17 (b). This is accomplished based on the day-ahead schedule and the last 24-hour changes that consumers may impose.

In Table 3, the optimization performance indices for 1-October-2016 are shown.

For the entire period, at the monthly level, the optimization performance indices are calculated in Table 4.

In FIGURE 18 and FIGURE 19, monthly flattening index and peak to average ratio are represented for initial, day-ahead scheduled, real and rescheduled consumption.

On average, the FI increased from 0.41 to 0.6 for day-ahead optimization and to 0.56 for real-time optimization,

Algorithm 4 Generate a Data Stream for each Appliance j With a Given T Resolution

DEFINE sourceData List<Float>
DEFINE sourceDataIndexed TreeMap<Float,Float>
DEFINE echoSocket Socket(hostName, portNumber);
DEFINE PrintWriter out (echoSocket)
FOR RECORD $FCSV_x$
    sourceData < E > .add(*RECORD* [*j*])
END FOR
FOR i=1 TO sourceData.Size()
  sourceDataIndexed.put(i/sourceData.size(),
  sourceData(i))
END FOR
WHILE VALUES ARE NEEDED DO
    r $\leftarrow$ RANDOM(0, 1)
      FOR ENTRY : sourceDataIndexed < K, *V* >.entrySet()
          f = ENTRY.getKey()
          IF $f \geq r$ THEN
        Out.println(LocalDateTime.now(),
        ENTRY.getValue(), “DEVICEj”, “TYPEOFj”)
            BREAK
        END IF
      SLEEP(T)
      END FOR
END DO

TABLE 3. Peak indices calculated for weekend day 1-October-2016.

	FI	PAR
Before optimization (initial schedule)	0.31	10.22
Day-ahead optimization (optimized schedule)	0.57	3.13
Real consumption	0.48	4.35
Re-scheduled consumption (real-time optimization)	0.57	3.03

while PAR (peak to average ratio) fell from 7.1 to 3 for day-ahead optimization and to 3.37 in real-time rescheduling. The data from Table 5 indicates that tariff A is the most convenient for the residential consumers due to the lowest electricity payment, while tariff D rewards best the shifting of the appliances with almost 9% gain if the consumer would have scheduled their appliances based on the day-ahead optimization program (*SC*) and with almost 5% if the supplier reschedules the appliances based on real-time optimization program (*RS*).

Considering that the tariff A brings the highest savings at the community level, the electricity payment and gain evaluation for each house with tariff A are shown in Table 6.

The highest gain is obtained by HG, HE, HF, HK, and HB if the houses stick with the day-ahead schedule or by HK, HG, HE and HD in case the supplier reschedules the appliances of these houses in real time.

On average, the gain per year per house regardless the tariff is about 137 Euro if the consumers would follow the day-ahead optimization schedule and 81 Euro if the supplier reschedules the appliances in real time. Although the gain is not impressive, we should have in mind that the primary

TABLE 4. Peak indices for each month at the community level.

	Initial schedule (IC)		Day-ahead schedule (SC)		Real consumption (RC)		Re-scheduled consumption (RS)	
Month	**FI**	**PAR**	**FI**	**PAR**	**FI**	**PAR**	**FI**	**PAR**
1	0.46	4.84	0.65	2.45	0.58	3.14	0.6	2.93
2	0.49	4.4	0.67	2.28	0.58	3.07	0.61	2.78
3	0.43	5.83	0.6	2.91	0.53	3.74	0.56	3.28
4	0.43	5.86	0.59	2.97	0.54	3.68	0.56	3.28
5	0.32	11.48	0.53	3.71	0.44	5.87	0.51	4.1
6	0.34	9.74	0.56	3.42	0.47	4.84	0.53	3.74
7	0.4	6.94	0.62	2.8	0.52	3.98	0.6	2.99
8	0.4	6.98	0.59	3	0.53	3.75	0.57	3.19
9	0.36	8.73	0.55	3.53	0.49	4.53	0.53	3.81
10	0.33	10.86	0.52	3.98	0.45	5.71	0.51	4.12
11	0.46	5.18	0.64	2.58	0.56	3.36	0.58	3.01
12	0.49	4.34	0.66	2.4	0.56	3.24	0.57	3.17
Average	**0.41**	**7.1**	**0.6**	**3**	**0.52**	**4.08**	**0.56**	**3.37**

TABLE 5. Electricity payment and gain evaluation at the community level.

Tariffs	SC [€/year]	RC [€/year]	RS [€/year]	RC-SC [€/year]	RC-RS [€/year]	Gain1 SC-> RC [%]	Gain1 RS-> RC [%]	Gain2 SC [%]	Gain2 RC [%]	Gain2 RS [%]
Tariff A	18,571.81	19,864.17	19,070.87	1,292.36	793.30	6.51	3.99	6.81	5.23	5.80
Tariff B	18,998.63	20,483.02	19,608.05	1,484.39	874.97	7.25	4.27	4.67	2.28	3.15
Tariff C	19,989.28	21,764.40	20,760.00	1,775.12	1,004.40	8.16	4.61	-0.30	-3.84	-2.54
Tariff D	20,416.07	22,383.27	21,297.20	1,967.20	1,086.07	8.79	4.85	-2.44	-6.79	-5.19
Tariff S	19,929.95	20,960.16	20,245.93	1,030.21	714.23	4.92	3.41	0.00	0.00	0.00

TABLE 6. Electricity payment and gain evaluation for each house with tariff A.

House	IC [€/year]	SC [€/year]	RC [€/year]	RS [€/year]	RC-SC [€/year]	RC-RS [€/year]	Gain SC-> RC [%]	Gain RS> RC [%]
HA	948.19	940.58	994.81	980.65	54.23	14.16	5.45	1.42
HB	1,340.24	1,283.10	1,392.65	1,374.42	109.55	18.23	7.87	1.31
HC	891.41	888.74	936.02	909.72	47.28	26.30	5.05	2.81
HD	1,779.04	1,784.99	1,879.09	1,786.30	94.10	92.79	5.01	4.94
HE	2156.34	2076.27	2292.89	2168.44	216.62	124.45	9.45	5.43
HF	1612.08	1551.1	1685.08	1626.39	133.98	58.69	7.95	3.48
HG	4381.28	4359.62	4612.34	4435	252.72	177.34	5.48	3.84
HH	1960.8	1917.32	2044.05	1998.77	126.73	45.28	6.20	2.22
HI	923.55	919.71	968.57	940.71	48.86	27.86	5.04	2.88
HJ	940.85	937.27	986.69	960.55	49.42	26.14	5.01	2.65
HK	1989	1913.1	2072.12	1889.97	159.02	182.15	7.67	8.79

purpose is to reduce the peak as it definitely brings long-run advantages, such as grid investment reduction, RES integration, better environment, etc.

VII. CONCLUSION

Starting from massive volumes of electricity consumption data coming from smart meters, smart plugs, other sensors and consumers' preferences, we proposed a two-stage dynamic optimization approach to minimize the load peak and decrease the electricity payment based on the incentives of the ToU tariffs. A novel optimization approach that schedules the appliances for residential consumers for the next day and then re-schedules the appliances in real time at fifteen-minute time intervals are developed at the community level. For this approach, we processed a large volume of data from sensors and smart meters considering consumers' preferences. Every fifteen minutes, the algorithm evaluates the differences between the optimized schedule and the actual consumption and control the operation of the interruptible appliances to stick with the day-ahead schedule as much as possible. We showed that real-time processing of such electrical big data using the proposed algorithm takes advantage of the Elasticsearch centered architecture, manages to reduce the peaks and delivers financial gains for the consumers without harming their comfort.

Nonetheless, the algorithm is based on the readiness of the consumers to send their day-ahead schedule and to make small adjustments to their habits based on service contracts. They can be further stimulated to submit their consumption patterns by using additional incentives such as gamification, i.e., the use of game elements such as rewards for accomplishing tasks. These type of approaches are heavily employed by companies like Uber [42] to stimulate their drivers in order to take more rides and could be translated into the field of electricity consumption optimization. For example, a further reduction of the electricity bill or discounts to partner companies can be offered after the first 10 submitted consumption patterns or for completing a task that asks to reduce consumption 5 days in a row for a given time interval.

ACKNOWLEDGMENTS

This paper presents the scientific results of the project "Intelligent system for trading on wholesale electricity market" (SMARTRADE), co-financed by the European Regional Development Fund (ERDF), through the Competitiveness Operational Program (COP) 2014-2020, priority axis 1 - Research, technological development and innovation (RD&I) to support economic competitiveness and business development, Action 1.1.4 - Attracting high-level personnel from abroad in order to enhance the RD capacity, contract ID P_37_418, no. 62/05.09.2016, beneficiary The Bucharest University of Economic Studies.

REFERENCES

[1] L.-L. Richter and M. G. Pollitt, "Which smart electricity service contracts will consumers accept? The demand for compensation in a platform market," *Energy Econ.*, vol. 72, pp. 436–450, May 2018.

[2] L. Wen, K. Zhou, S. Yang, and L. Li, "Compression of smart meter big data: A survey," *Renew. Sustain. Energy Rev.*, vol. 91, pp. 59–69, Aug. 2018.

[3] K. Zhou, C. Fu, and S. Yang, "Big data driven smart energy management: From big data to big insights," *Renew. Sustain. Energy Rev.*, vol. 56, pp. 215–225, Apr. 2016.

[4] P. Siano, "Demand response and smart grids—A survey," *Renew. Sustain. Energy Rev.*, vol. 30, pp. 461–478, Feb. 2014.

[5] I. F. Siddiqui, S. U.-J. Lee, A. Abbas, and A. K. Bashir, "Optimizing lifespan and energy consumption by smart meters in green-cloud-based smart grids," *IEEE Access*, vol. 5, pp. 20934–20945, 2017.

[6] J. Kwac and R. Rajagopal, ''Data-driven targeting of customers for demand response,'' *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2199–2207, Sep. 2016.
[7] S. Javaid, N. Javaid, S. Iqbal, S. Aslam, and M. H. Rahim, ''Optimizing energy consumption of air-conditioning systems with the fuzzy logic controllers in residential buildings: Optimizing energy consumption of air-conditioning systems in residential buildings,'' in *Proc. Int. Conf. Comput., Math. Eng. Technol. (iCoMET)*, Mar. 2018, pp. 1–9.
[8] A. Mirakhorli and B. Dong, ''Occupancy behavior based model predictive control for building indoor climate—A critical review,'' *Energy Buildings*, vol. 129, pp. 499–513, Oct. 2016.
[9] A. Keshtkar and S. Arzanpour, ''An adaptive fuzzy logic system for residential energy management in smart grid environments,'' *Appl. Energy*, vol. 186, pp. 68–81, Jan. 2017.
[10] R. Marah and A. El Hibaoui, ''Algorithms for smart grid management,'' *Sustain. Cities Soc.*, vol. 38, pp. 627–635, Apr. 2018.
[11] Q. Duan, ''A price-based demand response scheduling model in day-ahead electricity market,'' in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Jul. 2016, pp. 1–5.
[12] A. Di Giorgio and L. Pimpinella, ''An event driven smart home controller enabling consumer economic saving and automated demand side management,'' *Appl. Energy*, vol. 96, pp. 92–103, Aug. 2012.
[13] A. Soares, Á. Gomes, and C. H. Antunes, ''Categorization of residential electricity consumption as a basis for the assessment of the impacts of demand response actions,'' *Renew. Sustain. Energy Rev.*, vol. 30, pp. 490–503, Feb. 2014.
[14] J. Vuelvas, F. Ruiz, and G. Gruosso, ''Limiting gaming opportunities on incentive-based demand response programs,'' *Appl. Energy*, vol. 225, pp. 668–681, Sep. 2018.
[15] J. Kwac, J. Flora, and R. Rajagopal, ''Household energy consumption segmentation using hourly data,'' *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 420–430, Jan. 2014.
[16] E. Pan, D. Wang, and Z. Han, ''Analyzing big smart metering data towards differentiated user services: A sublinear approach,'' *IEEE Trans. Big Data*, vol. 2, no. 3, pp. 249–261, Sep. 2016.
[17] S. Oprea, A. Bara, M. E. Cebeci, and O. B. Tör, ''Promoting peak shaving while minimizing electricity consumption payment for residential consumers by using storage devices,'' *Turkish J. Elect. Eng. Comput. Sci.*, vol. 25, no. 5, pp. 3725–3737, 2017.
[18] C. Chen, S. Kishore, Z. Wang, M. Alizadeh, and A. Scaglione, ''How will demand response aggregators affect electricity markets?—A cournot game analysis,'' in *Proc. 5th Int. Symp. Commun. Control Signal Process. (ISCCSP)*, May 2012, pp. 1–6.
[19] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, ''A survey on cyber security for smart grid communications,'' *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 998–1010, 4th Quart., 2012.
[20] A. Yassine, A. A. N. Shirehjini, and S. Shirmohammadi, ''Smart meters big data: Game theoretic model for fair data sharing in deregulated smart grids,'' *IEEE Access*, vol. 3, pp. 2743–2754, 2015.
[21] O. Tan, J. Gómez-Vilardebó, and D. Gündüz, ''Privacy-cost trade-offs in demand-side management with storage,'' *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 6, pp. 1458–1469, Jun. 2017.
[22] R. Pérez-Chacón, J. M. Luna-Romera, A. Troncoso, F. Martínez-Álvarez, and J. C. Riquelme, ''Big data analytics for discovering electricity consumption patterns in smart cities,'' *Energies*, vol. 11, no. 3, p. 683, 2018.
[23] J. Kwac and R. Rajagopal, ''Demand response targeting using big data analytics,'' in *Proc. IEEE Int. Conf. Big Data*, Oct. 2013, pp. 683–690.
[24] H. Qu, P. Ling, and L. Wu, ''Electricity consumption analysis and applications based on smart grid big data,'' in *Proc. IEEE 12th Int. Conf. Ubiquitous Intell. Comput., IEEE 12th Int. Conf. Autonomic Trusted Comput., IEEE 15th Int. Conf. Scalable Comput. Commun. Associated Workshops (UIC-ATC-ScalCom)*, Aug. 2015, pp. 923–928.
[25] A. Agnetis, G. Dellino, P. Detti, G. Innocenti, G. de Pascale, and A. Vicino, ''Appliance operation scheduling for electricity consumption optimization,'' in *Proc. IEEE Conf. Decis. Control*, Dec. 2011, pp. 5899–5904.
[26] H. Jiang, K. Wang, Y. Wang, M. Gao, and Y. Zhang, ''Energy big data: A survey,'' *IEEE Access*, vol. 4, pp. 3844–3861, 2016.
[27] A. Keshtkar, S. Arzanpour, and F. Keshtkar, ''An autonomous system via fuzzy logic for residential peak load management in smart grids,'' in *Proc. North Amer. Power Symp. (NAPS)*, Oct. 2015, pp. 1–6.
[28] S. Javaid, N. Javaid, S. Iqbal, M. Guizani, A. Al-Mogren, and A. Alamri, ''Energy management with a world-wide adaptive thermostat using fuzzy inference system,'' *IEEE Access*, vol. 6, pp. 33489–33502, 2018.
[29] S. Barker, A. Mishra, D. Irwin, P. Shenoy, and J. Albrecht, ''SmartCap: Flattening peak electricity demand in smart homes,'' in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2012, pp. 67–75.
[30] F. A. Qayyum, M. Naeem, A. S. Khwaja, A. Anpalagan, L. Guan, and B. Venkatesh, ''Appliance scheduling optimization in smart home networks,'' *IEEE Access*, vol. 3, pp. 2176–2190, 2015.
[31] *UMassTraceRepository, Smart Data Set for Sustainability*. Accessed: Sep. 26, 2018. [Online]. Available: http://traces.cs.umass.edu/index.php/Smart/Smart
[32] *Electricity Smart Metering Customer Behaviour Trials (CBT) Findings Report*. Accessed: Sep. 26, 2018. [Online]. Available: https://www.ucd.ie/t4cms/Electricity%20Smart%20Metering%20Technology%20Trials%20Findings%20Report.pdf
[33] *ES-Hadoop*. Accessed: Oct. 28, 2018. [Online]. Available: https://www.elastic.co/products/hadoop
[34] *Iot Gateways*. Accessed: Oct. 27, 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/internet-of-things/iot-gateways.html#~benefits
[35] *SmartThings Hub*. Accessed: Oct. 27, 2018. [Online]. Available: https://www.samsung.com/uk/smartthings/hub-f-hub-uk-v2/
[36] *Mozilla IoT Gateway*. Accessed: Oct. 27, 2018. [Online]. Available: https://iot.mozilla.org/gateway/
[37] *Community Beats*. Accessed: Oct. 26, 2018. [Online]. Available: https://www.elastic.co/guide/en/beats/libbeat/current/community-beats.html
[38] *Elasticseach Mapping*. Accessed: Dec. 2, 2018. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping.html#mapping-type
[39] *Elasticseach Concepts*. Accessed: Dec. 2, 2018. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/_basic_concepts.html
[40] *Logstash TCP/IP*. Accessed: Dec. 2, 2018. [Online]. Available: https://www.elastic.co/guide/en/logstash/current/plugins-inputs-tcp.html
[41] *Logstash Disect*. Accessed: Dec. 2, 2018. [Online]. Available: https://www.elastic.co/guide/en/logstash/6.5/plugins-filters-dissect.html
[42] A. Rosenblat and L. Stark, ''Algorithmic labor and information asymmetries: A case study of uber's drivers,'' *Int. J. Commun.*, vol. 10, p. 27, Nov. 2015.

SIMONA VASILICA OPREA received the M.Sc. degree through the Infrastructure Management Program from Yokohama National University, Japan, in 2007, the first Ph.D. degree in power system engineering from the Bucharest Polytechnic University, in 2009, and the second Ph.D. degree in economic informatics from the Bucharest University of Economic Studies, in 2017. She is currently an Assistant Professor with the Faculty of Cybernetics, Statistics, and Economic Informatics with the Bucharest Academy of Economic Studies, involved in several research projects.

ADELA BÂRA received the degree from the Faculty of Economic Cybernetics, in 2002, and the Ph.D. degree in economics, in 2007. She is currently a Professor with the Economic Informatics Department, Faculty of Cybernetics, Statistics and Economic Informatics, Bucharest Academy of Economic Studies. She has authored over 70 papers in international journals and conferences. She has coordinated three research and development projects. Her research interests include data science, analytics, databases, big data, data mining, and power systems.

VLAD DIACONITA received the Ph.D. degree in statistics and cybernetics from the Department of Economic Informatics and Cybernetics, Bucharest Academy of Economic Studies, where he is currently an Associate Professor. His research interests include databases, big data, system integration, decision support, cloud computing, and renewable energies. He has authored over 50 papers in international journals and conferences. He is a member of the IEEE Computer Society. He was a Postdoctoral Fellow of the Research Group on Agent-Based, Social and Interdisciplinary Applications, Complutense University of Madrid, under the coordination of Prof. J. Pavon.