Blockchain: Research and Applications 2 (2021) 100016

Contents lists available at ScienceDirect

Blockchain: Research and Applications

journal homepage: www.journals.elsevier.com/blockchain-research-and-applications

Blockchain-based prosumer incentivization for peak mitigation through temporal aggregation and contextual clustering

Nikita Karandikar [a,*], Rockey Abhishek [b], Nishant Saurabh [c], Zhiming Zhao [d], Alexander Lercher [c], Ninoslav Marina [e], Radu Prodan [c], Chunming Rong [a], Antorweep Chakravorty [a]

[a] Department of Electrical Engineering and Computer Science, University of Stavanger, Stavanger, 8600, 4036, Norway
[b] Department of Energy and Petroleum Engineering, University of Stavanger, Stavanger, 8600, 4036, Norway
[c] Institute of Information Technology, University of Klagenfurt, Klagenfurt am Wörthersee, 9020, Austria
[d] Multiscale Networked Systems, University of Amsterdam, Amsterdam 1012 WX, Netherlands
[e] University of Information Science and Technology, Ohrid, North Macedonia

ARTICLE INFO

Keywords:
Peak shaving
Aggregation analysis
Contextual clustering
Blockchain
Incentivization

ABSTRACT

Peak mitigation is of interest to power companies as peak periods may require the operator to over provision supply in order to meet the peak demand. Flattening the usage curve can result in cost savings, both for the power companies and the end users. Integration of renewable energy into the energy infrastructure presents an opportunity to use excess renewable generation to supplement supply and alleviate peaks. In addition, demand side management can shift the usage from peak to off-peak times and reduce the magnitude of peaks. In this work, we present a data driven approach for incentive-based peak mitigation. Understanding user energy profiles is an essential step in this process. We begin by analysing a popular energy research dataset published by the Ausgrid corporation. Extracting aggregated user energy behavior in temporal contexts and semantic linking and contextual clustering give us insight into consumption and rooftop solar generation patterns. We implement, and performance test a blockchain-based prosumer incentivization system. The smart contract logic is based on our analysis of the Ausgrid dataset. Our implementation is capable of supporting 792,540 customers with a reasonably low infrastructure footprint.

1. Introduction

Integration of renewable energy, especially solar energy into energy infrastructure is on the rise, driven in part by the economic benefits such as government incentives and money saved on energy bills and in part due to rising awareness of the environmental benefits [1]. Prosumers are a category of consumers who generate part of the energy they need through their on site micro-generation devices and buy the remainder from the energy grid as needed [2]. Several prosumers living in close proximity to one another can form prosumer communities or microgrids [3]. Prosumers can use the generated solar energy for their own needs or if there is a surplus, sell it to the grid or other customers.

Peak periods [4] are periods when the demand for energy is the highest in a given time frame. Peak demand is rising as a result of an increasing number of retail users [5]. Maintaining grid stability in presence of variation in demand, especially during peak demand periods is an important task for the grid operator [6]. If peak demand approaches the available grid capacity, grid operators must take measures in order to maintain grid stability and reliable supply. This can be accomplished either by increasing available supply or reducing the peak load. Increasing available supply to match the projected peak usage value requires the operator to over-provision generation capacity, which can be expensive. This additional capacity is only used during peak periods and often takes the form of peaker plants [7] that are often coal or diesel powered and thus polluting. Moreover, the operation and maintenance costs of these peaker plants that are only used some of the time increase the price per kWh of energy, a cost that is usually passed on to the consumer.

The process of reducing the magnitude of the peak is called peak shaving. Peak shaving is of interest to grid operators and customers as it offers the potential for cost reduction by either deferring or avoiding investments in additional capacity. Surplus solar energy if sold back to

* Corresponding author.
E-mail address: nikita.r.karandikar@uis.no (N. Karandikar).

https://doi.org/10.1016/j.bcra.2021.100016
Received in revised form 19 April 2021; Accepted 9 June 2021

the grid presents an opportunity to supplement energy supply during peak energy demand periods. Surplus solar energy that is generated during off-peak periods, can be stored in a battery infrastructure to discharge as needed. Pilot studies have been conducted on using grid connected battery systems [8] to reduce peaks. Local solar energy production has the advantage of being co-located with the consumption sites, thus reducing transmission losses inherent in transporting electricity over large distances [9].

Another important peak shaving strategy is demand response or demand side management [10]. Traditionally, user demand was considered inelastic and supply was largely structured around demand. Now, while demand certainly continues to drive supply, there is a value found in regulating demand in order to shift some of the peak time usage to off-peak times, in order to flatten the usage curve and reduce the required capacity of energy infrastructure. Demand side response to peak consumption can take the form of increased prices to disincentivize consumers from running shiftable or non-urgent appliances during peak times. Another approach is the use of incentivization tokens which is a scheme under which the prosumer can earn tangible benefits for reducing usage at peak times.

In this paper, we present a data driven approach for incentive-based peak shaving by using a two-pronged strategy.

1) Firstly, energy consumers who record consumption below a calculated threshold during identified peak consumption periods are awarded reward tokens.
2) Secondly, the top surplus producing prosumers in the network are rewarded according to the amount of surplus they produce.

Net production or surplus production in a given time period is defined as:

$$\text{Surplus} = \text{Production} - \text{Consumption} \tag{1}$$

We chose for our analysis, the Ausgrid dataset [11] published by the Ausgrid corporation, which offers one year of energy generation and consumption data collected from smart meters installed on site for 300 random and anonymized customers in their network. The prosumer reward management system is implemented on a Hyperledger Fabric [12] blockchain in order to be transparent and decentralized. Further, the system is performance tested under varying loads using Hyperledger Caliper [13].

The work has the following structure:

1) Section 2 presents the design rationale and the salient building blocks of the solution.
2) An aggregation analysis of temporal energy behavior presented in section 3: a) identifies periods of peak usage and high variation b) identifies thresholds for categorizing net producers based on surplus values.
3) A semantic analysis of energy behavior in order to identify thresholds for low, medium, and high categories for energy consumption is discussed in section 4.
4) The system is implemented on a blockchain and sections 3 and 4 inform its logic which is encoded in smart contracts. The design, implementation, and performance characterization of the incentivization system are discussed in section 5.
5) Section 6 discusses how our solution builds upon the state of the art, while section 7 presents the salient conclusions of this study.

2. Proposed system

2.1. System participants and requirements

This system involves three distinct entities or organizations. First, the user platform is composed of prosumer representatives and perhaps a government agency to ensure legal compliance. The second organization is the power company that monitors the user generation and production and offers the rewards. The third organization is the grid battery, where the energy generated by the prosumers is stored and quality checked before sending it to its destination. Pilot studies [14] are currently underway to study peak shaving through the use of community batteries. Prosumers can connect through the grid network and store their surplus in the community battery owned and maintained by the Power company. By participating in a community battery infrastructure, prosumers have the opportunity to earn credits towards their electricity bills and thus get more value out of their solar investment without needing to own and maintain individual battery systems. The power company and the user platform must be in agreement about the logic for calculation of rewards. As the user platform represents the prosumers, it must also have the opportunity to check and approve all the transactions against its own calculations as shown in Fig. 1. Moreover, the storage provider is responsible for storing the generated energy and hence must verify that the amount of energy generated shown by the smart meter is actually generated and available for use. Thus, all three organizations must agree upon the business logic and approve each transaction, and the reward system must be transparent to all parties involved.

As the ecosystem consists of a number of small actors with their own distributed generation devices, there is a push towards decentralized management in order to cut out intermediaries and prevent the management from being concentrated in the hands of a third party. We expect many small scale producers to join this system and we must evaluate how many users it can support.

Blockchain, a decentralized ledger, fulfils the requirements outlined above, as it prevents the decision making from being concentrated in the hands of a single party. Moreover, its inherent features of immutability and robustness due to its decentralized nature and the cryptographic linking of transaction blocks fit our use case well. Such a decentralized system must also be secure and only open to authenticated users. It is therefore necessary to restrict membership only to members of the community and to authenticate all users. This also reduces the operation cost and computational complexity inherent in an open decentalized ledger system, also known as a public blockchain system.

We built our implementation using the Hyperledger Fabric, which is one of the most popular enterprise grade permissioned blockchain platforms. It is open source, free to use and has a modular architecture, allowing the operator to tailor the implementation components to their needs. This blockchain implementation is the underlying transaction infrastructure of the reward system that processes and records the

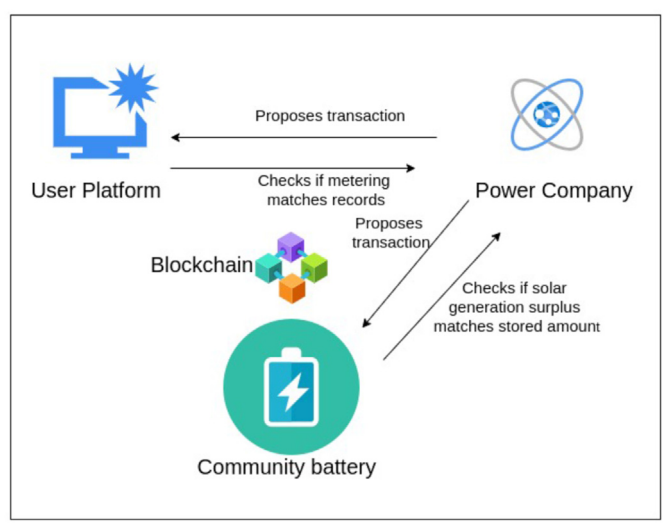


Fig. 1. Participants of the system.

transactions. In order to performance benchmark our implementation, we ran benchmarks using varying loads to evaluate the implementation based on latency and throughput. We implemented our benchmarking experiments in Hyperledger Caliper which interacts with the Hyperledger Fabric implementation and submits transactions as per the configured parameters.

2.2. Data driven approach

Section 3 analyses and aggregates the user data and extracts user energy behavior based on seasons (winter, summer, autumn, and spring), day of week (weekday or weekend) and time of day, and uses this to identify peak consumption times. Also, the aggregation of surplus energy production gives thresholds for classifying prosumers according to the amount of surplus energy they produce.

Further in section 4 the dataset rows are semantically linked, clustered, and labelled based on contexts such as Solar Production and User Demand. Based upon the clustering analysis, low, medium, and high thresholds are identified for user demand and solar production contexts.

The blockchain system as described in section 5 encodes the business logic in smart contacts and implements the reward mechanism based on the peak consumption times, surplus production, and user demand thresholds identified in section 3 and section 4. This system is performance tested to identify the number of users that can be supported. Fig. 2 presents the schematic outline of the proposed system and how the different modules of this work are linked.

3. Ausgrid Dataset and analysis

The data used in this study is the solar home electricity dataset published by the Ausgrid Corporation in New South Wales (NSW), Australia [15]. We begin our analysis by presenting a brief overview of the dataset.

3.1. Ausgrid Dataset overview

The Ausgrid dataset includes data collected from installed electricity meters for 300 random customers in the Ausgrid network from July 1, 2012 to July 30, 2013 recorded at half-hour intervals (Δt=1/2 h). The Ausgrid dataset is popular among researchers investigating grid-defection [16], home energy management [17], and load forecasting using historical smart meter data [18].

The dataset includes solar PV production from roof top solar panels and residential load data for individual households. The residential load includes two distinct categories: (1) Energy Consumption and (2) Heating Load. The first category records the general energy consumption in the household. The heating load refers to energy consumption for heating water in the household. The utility provider controls this load by operating electric water heating during specific periods of the day. This is done with the aim of reducing overall network load during peak times and providing financial incentives to the customer. This is an optional feature and in the Ausgrid dataset [15], 137 out 300 have opted for this feature. The individual customers are only identified by a serial list (1:300) of Customer ID which serve as aliases and their geographical context location is identified by post code.

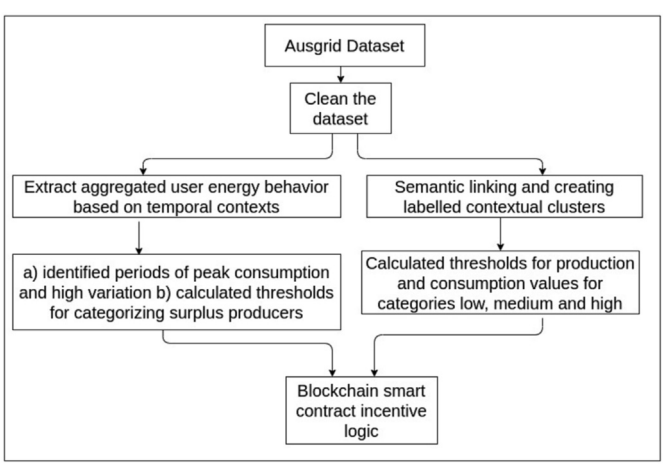


Fig. 2. Schematic outline of the system.

Table 1
Customer IDs in the cleaned Dataset.

ID	13, 14, 20, 33, 35, 38, 39, 56, 69, 73, 74, 75, 82, 87, 88, 101, 104, 106, 109, 110, 119, 124, 130, 137, 141, 144, 152, 153, 157, 161, 169, 176, 184, 188, 189, 193, 201, 202, 204, 206, 207, 210, 211, 212, 214, 218, 244, 246, 253, 256, 273, 276, 297

While, in principle the Ausgrid dataset consists of 1 year of electricity production and consumption data for 300 consumers at a resolution to 30 min, in practice there are several inconsistencies and anomalies in the data. Ratnam et al. [11] performed a detailed study on the Ausgrid dataset to identify these inconsistencies and identify a subset of customers to be included in the clean dataset with complete records. The current study uses this subset of customers, listed in Table 1. The geographic spread of these customers based on their postcodes is shown in Fig. 3. It can be seen in Fig. 3 that most of the customers in the clean dataset are located around Newcastle and Sydney metropolitan areas in NSW, Australia.

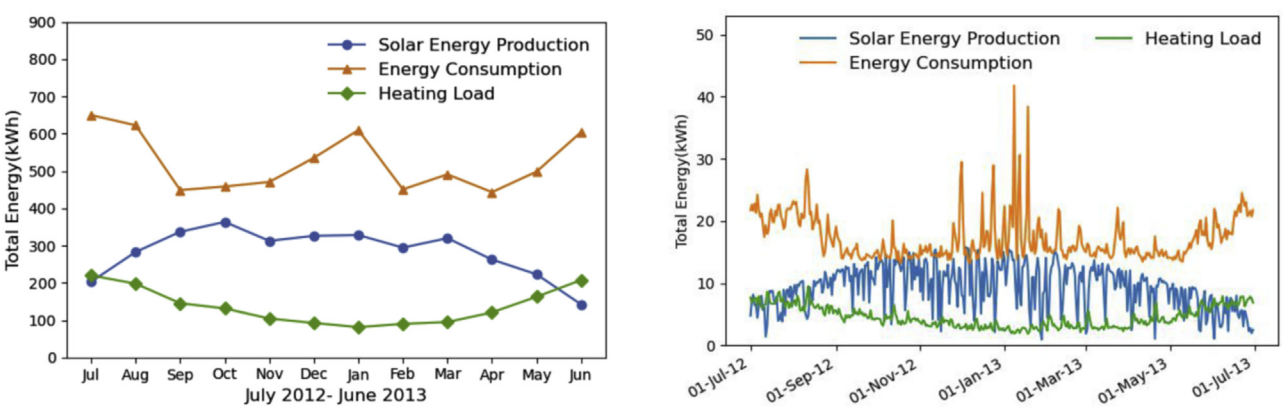
3.2. Aggregated energy profile for all customers

To gain an overview of the energy profiles of the customers as a group, we combined their energy profiles into an aggregate consumer. The energy profile of this aggregated consumer was cumulatively re-sampled on a monthly (Fig. 4(a)) and daily basis (Fig. 4(b)). The seasonal characterization of the months of a calendar year for Australia is outlined in Table 2. It can be seen from Fig. 4(a) and Table 2 that the energy profile of the consumers as a group shows significant seasonal variation. The peak in energy consumption corresponds to winter (June–August) and summer (December–February). This can be directly correlated with increased load due to climate control requirements during this period. While the overall monthly loads are similar for the summer and winter months, cumulative re-sampling on a daily basis shows higher peaks and greater variation during the summer months (December–February).

To better identify the days with high consumption during the summer months, we sorted aggregated household consumption in descending order, and the first 12 data-points were plotted chronologically along with the corresponding day of the week in Fig. 5. It can be seen in Fig. 5 that the highest aggregated daily consumption occurs on the weekends when the household members are home. However, there are two outliers of high consumption on weekdays: (1) Monday: 24-12-2012 and (2) Tuesday: 08-12-2013. The second outlier is especially interesting as it coincides with the peak summer temperatures and bush fire warnings across most of Australia and especially NSW [19]. It can be seen in Fig. 4 that solar energy production is highest in the summer months due to higher solar radiation after which it tapers off. Conversely, the heating load for electric water heating is highest during the winter months and it tapers off during the summer months.

In addition to the solar home electricity dataset [15], we obtained the historical Recommended Retail Price (RRP) for electricity in NSW and the total regional demand data for NSW, Australia between July 1, 2012 to June 30, 2013 at 30 min interval [20] corresponding to the timestamps in the Ausgrid Dataset. We computed the coefficient of variation for all

Fig. 3. Location of the customers.



(a) Aggregated monthly energy profile. (b) Aggregated daily energy profile.

Fig. 4. Energy profile aggregated across all customers.

the fields in the cleaned Ausgrid dataset (aggregated across all customers), the regional demand and RRP are shown in Fig. 6. It can be seen that both Solar energy and heating load show high variation in the range of 1.2–1.8. It can be observed that variation in solar energy production reduces during summer whereas the variation in heating load peaks during summer. Energy consumption shows much lower variation. However, we can see a comparatively higher variation in energy consumption during the summer. This agrees with the observations made earlier in Fig. 4. Fig. 6(d) shows that the regional demand remains quite stable with low variation except during the summer. Finally, Fig. 6(e) shows that the energy price in NSW shows low variation throughout the year with occasional sharp spikes throughout the year.

3.3. Seasonal energy profile for all customers

To better understand the seasonal variation in the energy profiles of the customers, we focused on the energy data for all the customers over a 1 month period in the middle of each season. The representative months for each season are listed in Table 2. Thereafter, we averaged the energy profile across all the customers for the 1 month period. This allowed us to compute a representative daily energy profile for an averaged customer during the four seasons. The results are shown in Fig. 7. It can be observed in Fig. 7 that the heating load is concentrated around midnight and early mornings i.e., during periods of low consumption. Solar energy production follows a clear bell curve. Energy production starts at around

Table 2
Seasons in Australia.

Season	Period	Representative month
Winter	June–August	July-2012
Spring	September–November	October-2012
Summer	December–February	January-2013
Autumn	March–May	April-2013

8 a.m. followed by a gradual rise and a peak at around 2 p.m. Thereafter it declines, finally stopping around 7 p.m. As expected, the highest overall production is observed in spring and summer followed by autumn and least in winter. The daily energy consumption profile shows significant seasonal variation. In the winter (Fig. 7(a)) we observe two peaks: one in the morning before the start of the working day and the second one in the evening at the end of the working day. This second peak can be observed persistently across all seasons in Fig. 7. This indicates consistent user demand in the evenings. In all the seasons except summer, we observe either a flat load curve between morning and evening or an inverse plateau in winter. However, in the summer season, we can observe an almost linear rise in load between morning and evening. This may be related to the rising electricity consumption associated with increased climate control as the temperature increases during the day. As the temperatures fall in the evening and the night, the load tapers off.

It can also be seen in Fig. 7 that between 8 a.m. and 7 p.m. there are periods where solar production is higher than the consumption in the household leading to surplus energy production which can be fed back into the grid or be utilized for charging the community level battery storage. The surplus energy produced by the individual customers at each half-hour interval was calculated by subtracting the energy consumption and controlled heating load from the solar production. The resulting column was used to filter the data set by dropping the rows with negative surplus as they represent the intervals where the total consummation in the household exceeds the solar production. The summary statistics for the resulting dataset are listed in Table 3.

4. Smart energy transaction analytic

The semi-structured dataset includes information about timestamps, involved actor identifiers, etc. Analysis of this dataset can help to understand the behavior of actors and their associated contextual activities. For example, in the case of smart energy systems, the energy dataset can be studied in depth to understand the energy demand and solar production. To achieve this, we extract information from the dataset rows and transform it into a more expressive structured representation. Each row of the dataset is considered to be a transaction. If the proposed blockchain-based incentivization system is implemented, each dataset transaction along with additional reward fields would be added to the blockchain through a blockchain transaction as explained in section 5. The procedural workflow for analyzing Ausgrid Dataset follows four steps:

1) Split the dataset transactions into data nodes based on context, where a context represents an event category such as heating consumption, solar production, etc;
2) Semantically link data nodes occurring within multiple context;
3) Cluster data nodes for each context based on their contextual similarity, such as similar energy consumption value ranges. The identified value ranges are used as labels for each unique cluster;
4) Compute cross-contextual similarity to understand the energy demand and solar production.

Fig. 8 shows the workflow for smart energy transaction analytics. Initially, we take semi-structured and smart energy dataset transactions and split them into unique contexts (e.g. solar production, heating consumption, energy consumption, total demand per region, price, etc), and semantically link each transaction through diverse contexts. Next, we cluster data nodes for each individual context, where each cluster represents data nodes with similar properties. Further, we label each cluster with unique properties. Afterwards, we compute the pairwise similarity between clusters inside a single layer by calculating their Euclidean distance related to clusters of all other layers, where a lower distance means a higher similarity. This process allows us to not only understand the changing behavioral patterns of transactions in a single context but also across contexts. In this paper, we primarily analyse similarity for cluster of transactions within user demand layer with respect to other contextualised layers.

Based on the presented system design in section 4, we implement smart energy transaction analytic in the following stages.

4.1. Semantic linking and contextualisation

Semantic linking and contextualisation in complex datasets are required to analyse actors' transactions. In general, datasets are represented as monolayer graphs to visualise the relationships between the actors, where graph nodes denote different actors and edges represent the interactions between actors. However, monolayer graphs often fail to capture the dynamically changing structural contexts of actors and their corresponding evolving relationships. Hence, we adopt a multilayer graph theory that defines a set of context-based layers.

Fig. 9 shows an example of such a multilayer representation of smart energy transactions arranged in five layers, where each layer denotes a context. In this example, a multilayer network has a set of nodes like a normal network (i.e. a monolayer graph) but a distinct set of layers. Each

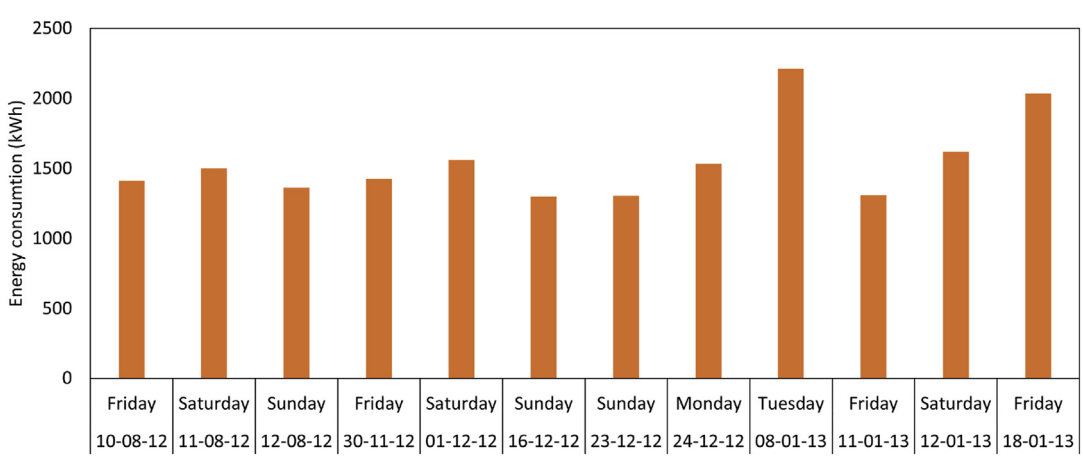


Fig. 5. Peak daily energy consumption in summer.

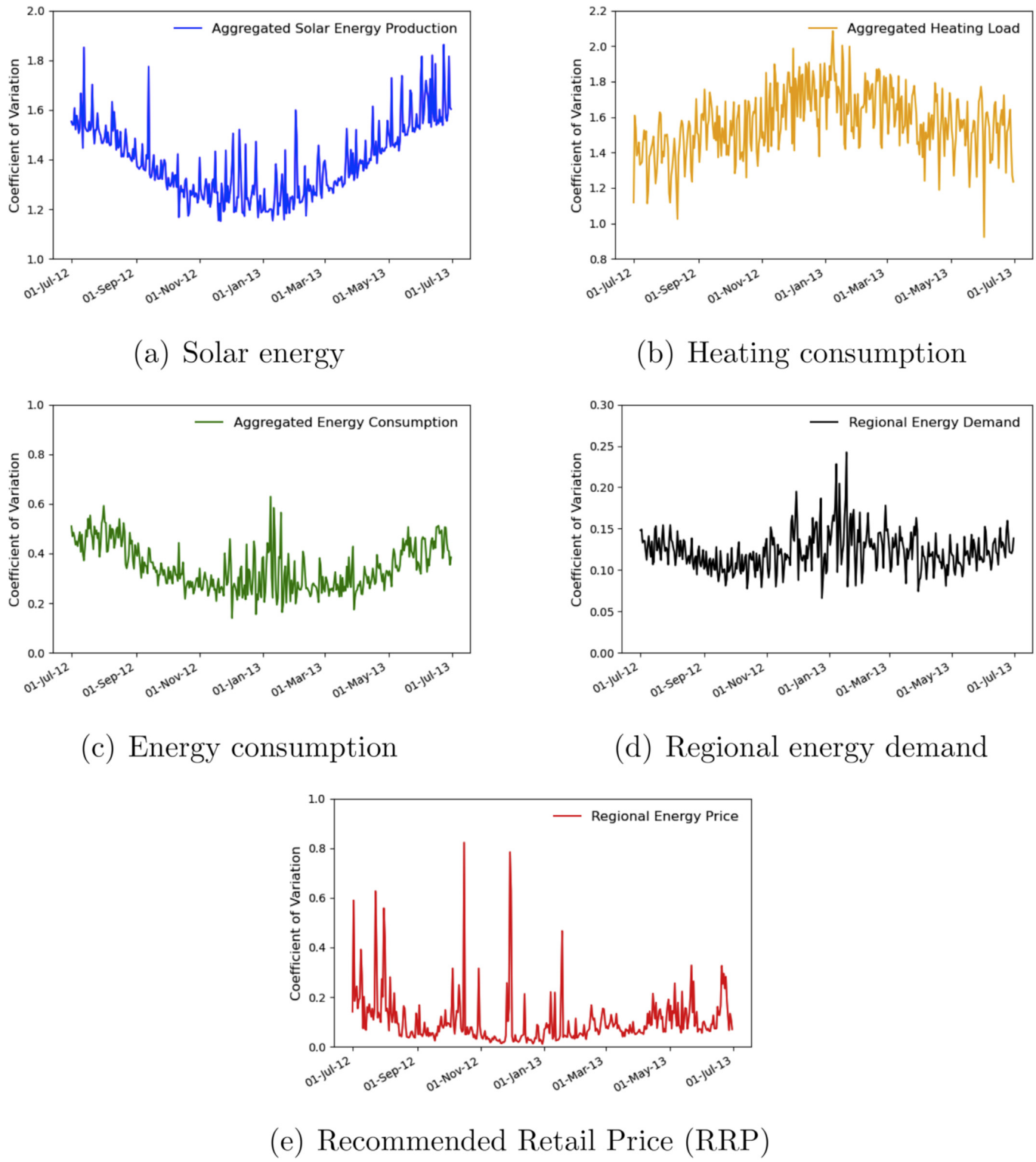


Fig. 6. Coefficient of Variation on a daily basis for fields in the Ausgrid Dataset, regional energy demand and Recommended Retail Price.

layer in the multi-layered representation represents a diverse context (e.g. solar production, heating consumption, energy consumption, regional price, regional demand) and various relationships of entities [21]. A multilayer network representation allows to link edges between same entities across multiple layers and provides a cross-context view of smart energy transactions. This improves the understanding of different interactions among the entities within complex systems across multiple and cross-contextual viewpoints.

To provide contextualised semantic linking and exploit semantic enrichment of complex datasets, we adopt a multilayer approach based on transaction attributes and topological structure. For this purpose, we define a set of layers showing different contexts. Additionally, we jointly consider the context of all networked entities and their network similarity strength. To define a semantic link, we consider different semantic labels for edges across cross-contextual layers, while similar semantic labels for edges stay within a single

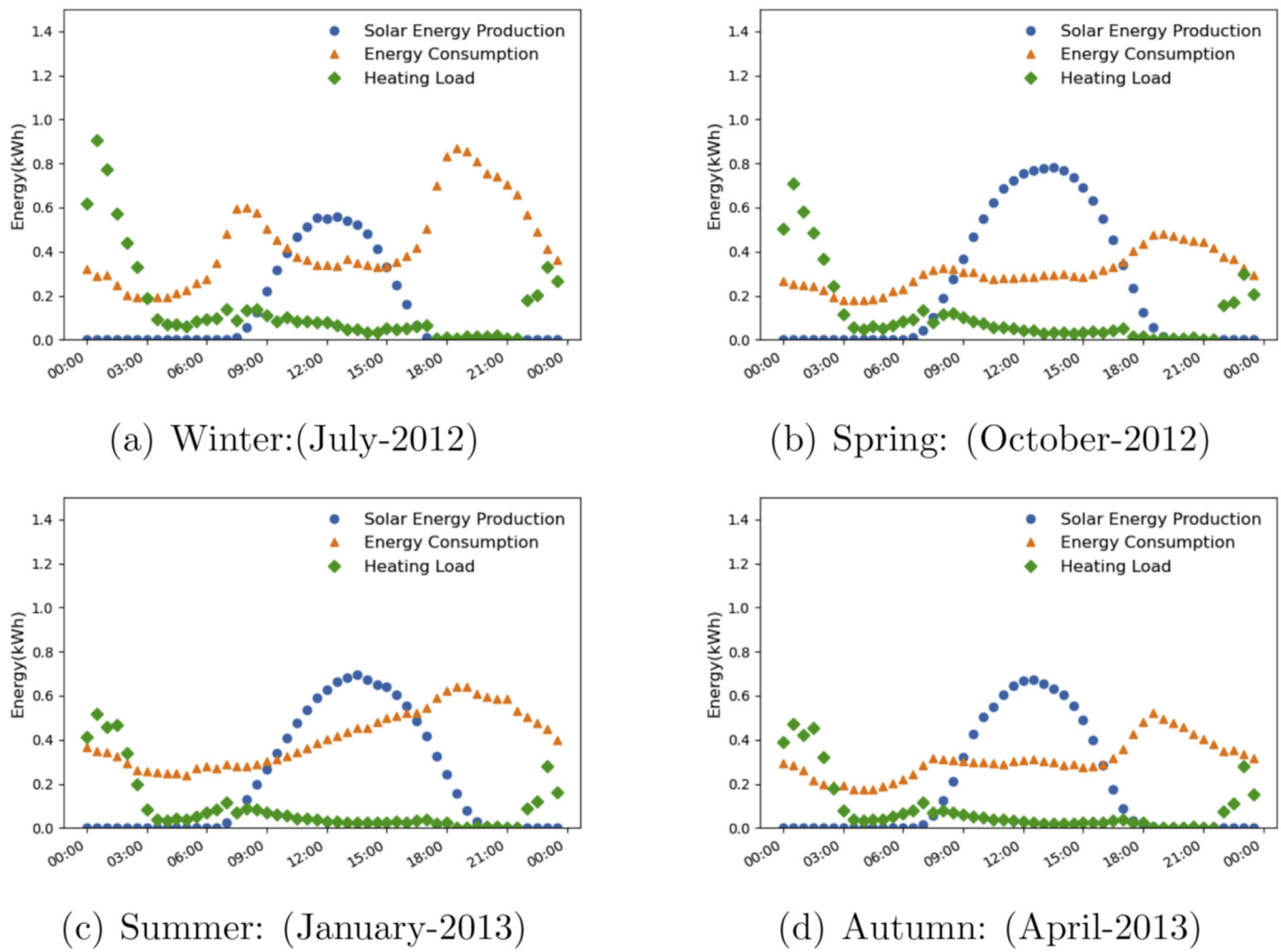


Fig. 7. Seasonal variation in daily energy profile aggregated across all customers.

layer. Our multi-layered approach provides a fully interconnected network where all layers contain all nodes and follows a diagonal coupling model in which inter-layer edges only exist between nodes and their counterparts. Our model also adopts a categorical coupling model in which inter-layer edges are present between any pair of layers and links between pairs in each layer describe similarity strengths. We implement three steps for constructing a semantic multi-layered network.

4.1.1. Step 1

Takes dataset transactions as input and extracts each row of data schema as context (e.g. energy consumption, price ranges, location), facts, attributes, concepts, and events to enable the accurate analysis of unstructured data.

Table 3
Summary statistics during periods of surplus energy production.

Metric	Surplus Energy (kWh)
Min	0.001000
Mean	0.397672
Max	4.095000
25%	0.134000
50%	0.284000
75%	0.477000

4.1.2. Step 2

Creates an attributed multi-layered network embedding M using the extracted activities pertaining to different transactions. Contrary to monolayer networks, a multilayer network $M=(V_M, E_M, V, L)$ has an underlying set V of N physical nodes that manifests on layers in L constructed from elementary layer sets (i.e $L_1, L_2, \cdots, L_d$, where d is the number of contexts). The set of node-layer tuples in M is $V_M \subseteq V \times L_1 \times \cdots \times L_d$, and the set of multilayer edges is $E_M \subseteq V_M \times V_M$. The edge $((i, \alpha), (j, \beta)) \in E_M$ indicates that there is an edge from node i on layer α to node j on layer β (and vice versa, if M is undirected).

4.1.3. Step 3

Creates a semantic link between layers in the multilayer network M by utilising attribute similarity between nodes in the same layer. We link nodes with similar attributes in each layer and measure the similarity of nodes using the Euclidean distance as a measure of similarity to compare the pairwise affinity of nodes. The higher the Euclidean distance score, the lower is the similarity score and vice-versa. We further construct a weighted similarity graph in each layer by measuring the similarity between nodes based on their contextual activity in each specific layer. If the similarity score of two nodes is higher than a threshold, it creates a link between them across layers. The threshold is use case dependent and determined based on the input transaction dataset schema. Finally, we assign a weight to each link between two similar nodes based on their similarity score, where the higher weight corresponds to links between more similar nodes.

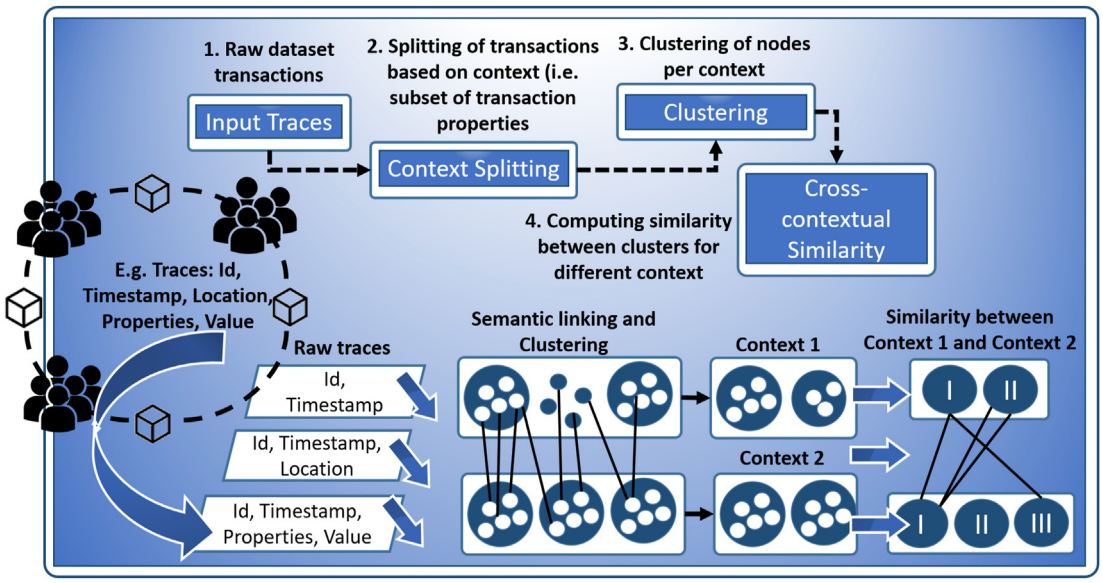


Fig. 8. Transaction analytic workflow.

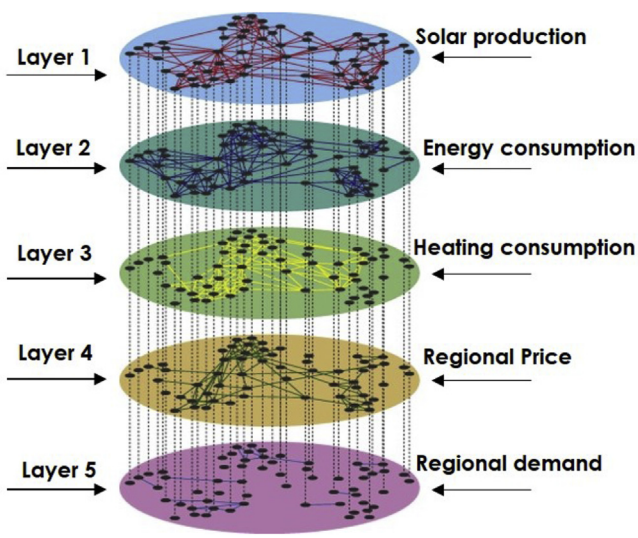


Fig. 9. Multilayered contextualisation of smart energy transactions.

Based on the smart energy dataset presented in section 3.1, we construct a multilayered network with six identified layers.

- *L1-Solar production* represents the amount of energy produced by different residents;
- *L2-Energy consumption* represents the amount of energy consumption pertaining to each resident;
- *L3-Heating consumption* is the heating load across diverse residents;
- *L4-Regional price* shows the price paid by each resident for their energy usage at the given time across a geographical location;
- *L5-Regional demand* represents the overall energy demand across a specific geographical region;
- *L6-User demand* is the overall demand per resident. While, user demand is not represented in the dataset, we represent it as an additional layer, where user demand per resident is the sum of energy consumption and heating consumption per resident.

4.2. Contextual clustering and labelling

The multi-layer network constructed using steps in section 4.1 contains raw, but contextualised information. Hence, the next step is to find similar nodes based on the characteristics of each layer. The smart energy transaction analytics model does not consider links or edges for clustering. Instead, we utilize feature values of each row of the dataset, where a feature is represented by the column values for each transaction. To achieve this, we applied a clustering technique that tags similar nodes based on their feature values with the same arbitrary but fixed label. Initially, we fetch the multilayered contextualised graphs across temporal stages as input for clustering and labelling. Further, we utilize OPTICS [22], an augmented cluster-ordering algorithm to find similarities among different nodes in each layered context through a distance function and a minimum number of neighbors required as a unique cluster.

We implemented the OPTICS clustering technique in python using scikit-learn[1] library. In the current implementation, we configured clustering hyper-parameter *MinPts*=15, and *Eps* is set to infinity. The hyper-parameter *Eps* is the radius of clustering neighborhood around a node point, while *MinPts* represents the minimum number of neighbors around a radius *Eps*. We used Minkowski [23] distance metric to compute the Euclidean distance between different node points with unique feature values. Fig. 10 shows the clusters obtained for L6-User demand layer, while Fig. 11 shows the clusters obtained for L1-Solar production layer for all customers. Each point in Figs. 10 and 11 with similar colors represents the customers belonging to the same cluster with similar transactions, while black colored points represent the noise. In both cases, we compute clusters with four different values of *MinPts*=15, 25, 50, 100 respectively. We observe that with *MinPts*=15, our clustering approach finds clusters closest to the original dataset, but with low density. In general, increasing *MinPts* value increases the density of clusters. Hence, in the current implementation, we fixed the value of *MinPts*=15. However, for future works, we will consider dynamically estimating the value of *MinPts* depending upon the individual layer size and structure. Fig. 12 shows clustering for a subset of customers within *L6-User demand* layer. Similarly as in Figs. 10 and 11, Fig. 12 shows clustering with four *MinPts* values, but represents cluster for a subset of customers within *L6-User demand* layer. We observe that the same customers in different transactions can correspond to varied demand values represented in different clusters.

[1] https://scikit-learn.org/.

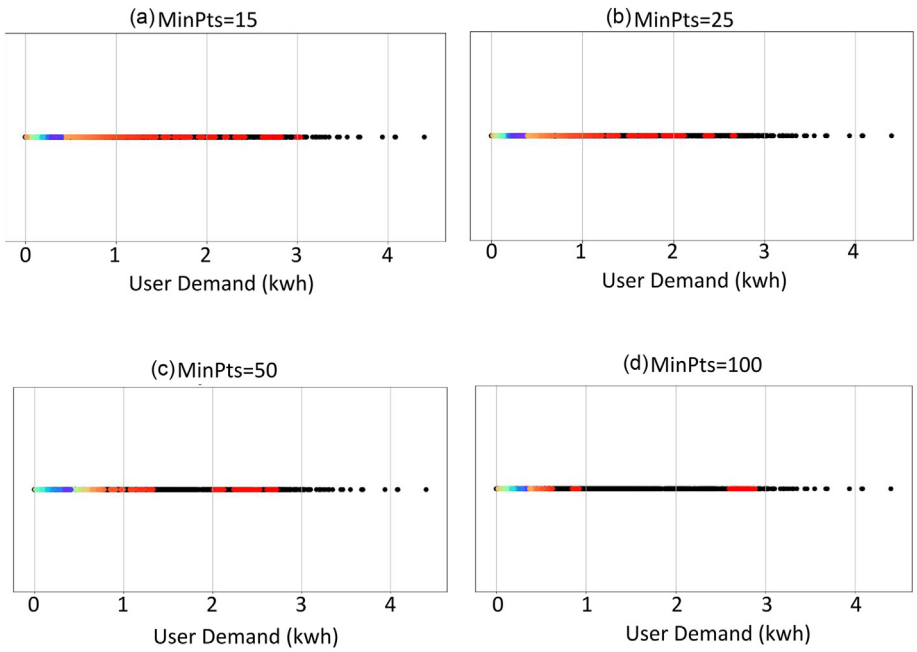


Fig. 10. User demand clusters for all customers.

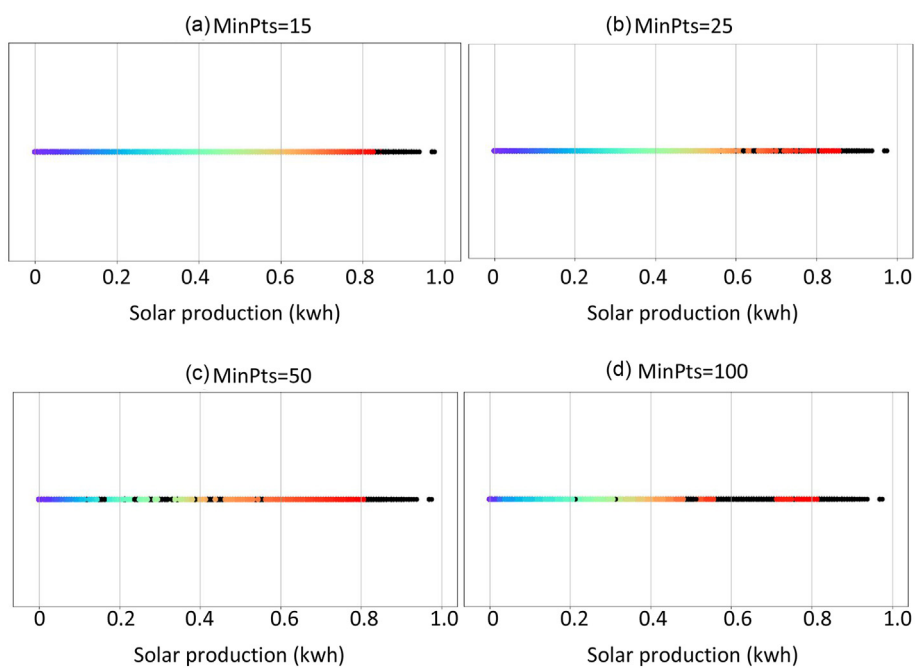


Fig. 11. Solar production for all customers.

After identifying the clusters for each layer, we implicitly know the transaction activity of different customers represented as a node within a cluster. Hence, the next step is to label the clusters. In this work, we primarily focus on analysing energy production and consumption per transaction, hence we label only the clusters in *L1-solar production* and *L6-User demand* layers. To this end, we encapsulate each cluster with either of k label categories. In this paper, we define k=3 namely: High, Medium, and Low label categories. Initially, we define a feature associated to each cluster $\in C$, where a feature is a tuple of length n=2 with the highest and lowest values of all nodes in the same cluster, and C is the number of clusters obtained in a specific layer. Further, we compute the average *MEAN* of these feature values of each cluster. Henceforth, we identify the lowest (X) and the highest (Y) *MEAN* value of all clusters $\in C$ and compute the distance between the lowest and the highest mean X and Y as $Z=Y-X$. Next, we map k ranges to k labels using Z/k representing the value range for the labels. The value range enables us to define the two global thresholds t_1 and t_2, where $t_1=X+Z/k$ and $t_2=Y-Z/k$. Based on the thresholds, finally we assign labels to each cluster. A cluster is labelled High: if $MEAN>t_2$, while we label a cluster Medium: if $t_1<MEAN\leq t_2$. Finally, a cluster is labelled Low: if $MEAN\leq t_1$. In this work, we primarily

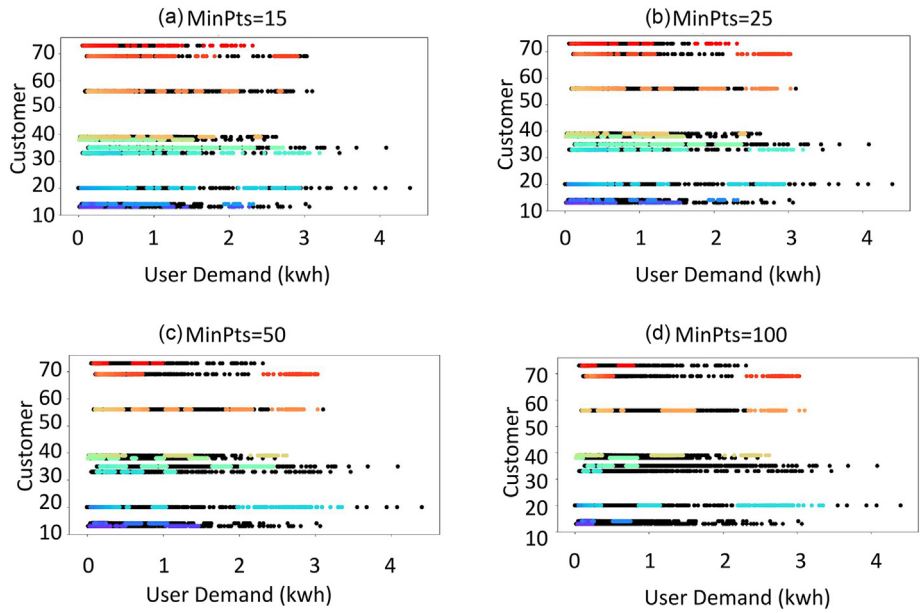


Fig. 12. User demand cluster distribution per customer.

label clusters obtained in solar production and user demand layers. For *L1-solar production* layer, the computed thresholds t_1 and t_2 are 1.43016 and 2.8603, respectively. While for the user demand layer, the thresholds t_1 and t_2 are 1.8306 and 3.6608.

4.3. Cross-contextual similarity

After obtaining the clusters and corresponding cluster labels in each layer, the next step is to analyse the similarity of clusters in one layer with respect to all clusters of another layer. This allows us to understand the changes in behavioral activity of transactions in one context with regards to another context. To achieve this, we compute pairwise similarity of two clusters in one layer (e.g. cluster (C_i, C_j) in *L6-User demand* layer with all clusters of *L1-solar production* layer). Henceforth, we compute the similarity of cluster pair (C_i, C_j) $\in$ *L6-User demand* with cluster $C_k \in$ *L1-solar production* layer using Euclidean distance E as follows:

$$E = \sqrt{\sum_{k=1}^{N} (D_i^k - D_j^k)^2} \tag{2}$$

where, N is the total number of clusters in *L1-solar production* layer, D_i^k and D_j^k are the number of edges from cluster C_k to clusters C_i and C_j respectively. We perform similarity computation for all the possible combinations of clusters in *L6-User demand* with all the clusters of other layers. The final output of this operation is a collection of all suitable pairs of clusters with their associated similarity regarding each layer. The key to the dictionary is a pair of clusters labels and the value is another dictionary of the computed similarity in regard to each layer, where the key is the name of the layer and the value is the computed Euclidean Distance. As a future work, we also plan to use this pairwise similarity computation for proactive prediction of cluster size and structure.

5. Blockchain based reward system

In this section we use the insights garnered from section 3 and section 4 to design the smart contract that will process prosumer rewards. Each row from the dataset (1 transaction) is modelled as a key value pair token on the blockchain, created through a blockchain transaction, where the token's value field is composed of the values from the row being processed. In addition, we add six new values in the value field of each token, RSeasonal, RWeekend, RPeakTime, RMiniProducer, RProducer, and RMegaProducer which are boolean fields, which will be set to true if the conditions for a reward of that type are met for a particular transaction. A token is created for each customer at each half-hour interval and consists of customer data from the dataset as well as the reward fields mentioned.

Section 3, identified several energy generation and usage patterns. Of these, we focus on the observations listed below for creating our smart contract. Alongside we mention, the reward field that will be updated based on that observation. The smart contract is written in Golang v1.16 [24] and creates the token shown in Fig. 13 for each transaction.

1) The Highest consumption was observed in summer and winter seasons, which can be attributed to climate control needs. Moreover, the highest variation in energy consumption was observed in summer. So, for low consumption during summer and winter the customer can win a reward associated with the field RSeasonal.
2) Peak loads were seen on weekends when household members are home. If a customer's consumption is low on the weekend, they will gain a reward associated with the field RWeekend.
3) Peaks were twice a day, once in the morning before the start of the workday and once in the evening at the end of the workday. So, in the case of low consumption during identified peak times the customer can get a reward associated with the field RPeakTime.
4) The identified thresholds for 25th percentile, 50th percentile and 75th percentile surplus production are t_a, t_b and t_c. If surplus solar production for any customer is greater than or equal to t_a, then the field set to true is RMiniProducer. Similarly, the fields RProducer and RMegaProducer are set to true for transactions when surplus generation is greater than or equal to t_b and t_c respectively. Thus, a transaction with surplus production in the 75th percentile will have RMiniProducer, RProducer, and RMegaProducer, all set to true.

Moreover, in section 4, of the several observations made on user generation and usage behavior, we focus on the thresholds identified for user demand. The thresholds identified are used for labelling clusters. In

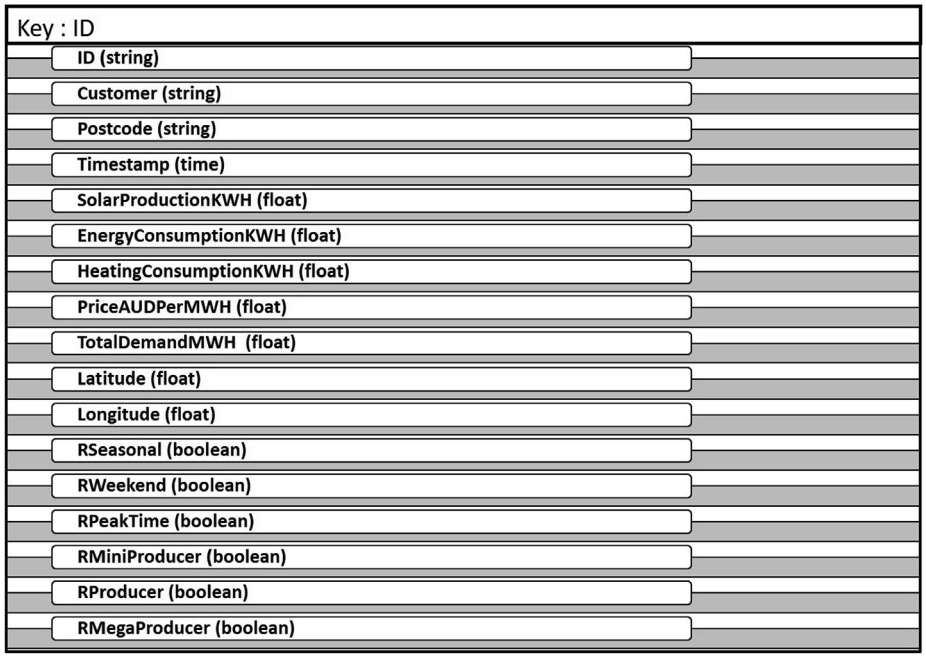


Fig. 13. Structure of transaction token.

the case of user demand, if the demand in kWh associated with a given transaction is less than t_1 then it is labelled as low, while if it is more than t_2, it is labelled as high. Demand between t_1 and t_2 is labelled as medium.

5) For user demand, the thresholds t_1=1.8306 kWh and t_2=3.6608 kWh were used to create labelled clusters of low, medium, and high consumption levels.

5.1. Smart contract

The smart contract encodes the business logic of the reward system. When this smart contract is deployed, we set an endorsement policy that stipulates that all organizations in the network must endorse each transaction. This is due to the reasons identified in section 2.2. The algorithm of interest is Algorithm 2, while Algorithm 1 is a helper algorithm.

Algorithm 1. TokenExists

Algorithm 1 TokenExists

```
function TokenExists(idval string)
    token ← worldstate(idval)
    if token=nil then
        return false
        else return true
    end if
end function
```

In Algorithm 2, first the identity of the invoker organization is found from the identity of the invoker client. If the client is not a member of the Power Company this attempt to create a transaction is rejected. As the Power company is the one that awards the tokens and has access to both creation and generation data for all customers, it is the only organization with the privilege of initiating transactions. However, it cannot actually process transactions without the endorsement of the other two organizations. Then, the provided timestamp string is converted into an integer array by removing special characters and by type conversion. This is done in order to work with the individual parts of the timestamp such as month and date. The key to the token to be created is checked against the world state to make sure that there is no collision between the new key and an existing key. This can only happen if a transaction token for a given customer and timestamp has already been created and is now being recreated. Each customer will have transaction tokens with timestamps corresponding to each half-hour period of the day and thus will have only one token per half-hour. If the token for a given half-hour period for a customer exists already, the transaction will be rejected. Now, the conditions for the various rewards are checked and the relevant field is updated if the customer will get that particular reward. Thus, if the customer's consumption is low and the season calculated from the month part of the timestamp is summer or winter, which are known to be high consumption periods, then the RSeasonal field gets updated. Similarly, for low consumption on the weekend or during identified morning and evening peak times, the fields RWeekend and RPeakTime respectively will be updated. Also, if the transaction shows a net production, the value of the surplus will be checked against the percentile thresholds identified in Table 3 in order to update the fields RMiniProducer, RProducer and RMegaProducer. After updating the rewards, the token will be saved to the world state with the key customer_timestamp.

5.2. Implementation

Our test infrastructure included 5 Virtual Machines (VM) on a Cloud environment as shown in Fig. 14. Each VM has Ubuntu 20.04 installed and features 32 GB RAM, 4 dedicated virtual CPUs, and a 100 GB SSD disk. Each VM uses Docker version 19.03, Docker Compose version 1.26, Hyperledger Caliper version 0.4.2 and Hyperledger Fabric 2.3.0, which are all the latest stable versions available at the time of writing. The Ordering service uses the RAFT [25] consensus algorithm with 3 Ordering Service Nodes (OSN) as recommended in the Hyperledger Fabric documentation. We chose LevelDB as the state database as it is the

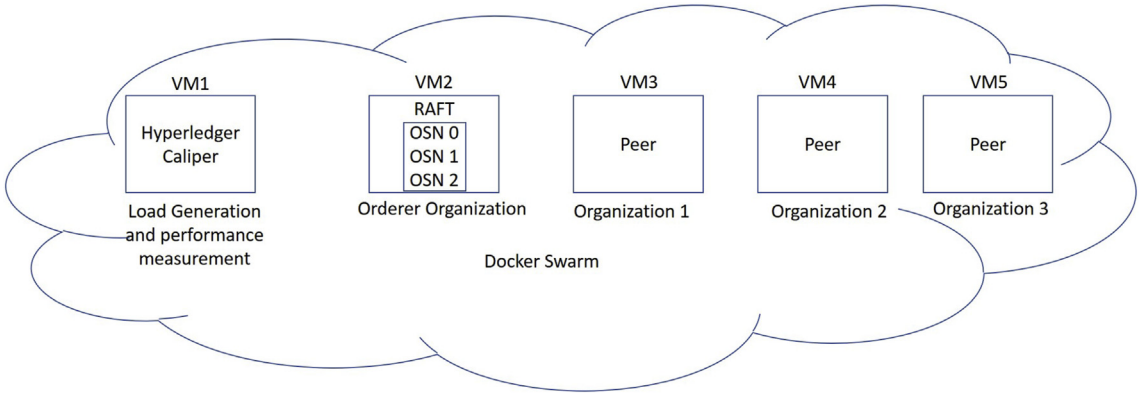


Fig. 14. Experimentation testbed. VM: virtual machine; OSN: ordering service nodes.

more performant choice [26]. Each node of our network runs as a Docker container and is connected in a Docker Swarm to ensure high availability. Our experiments feature an architecture of 3 Organizations. We run the load generator Hyperledger Caliper on a separate VM as it is resource intensive, in this case VM1. Each organization, including the orderer organization uses a separate VM to run its Docker containers. Our reason for this setup is twofold. Firstly, the orderer organization, or any of its OSN must not be in control of any of the member organizations as it performs a vital function. Moreover, putting the orderer organization on the same VM as an organization will consume resources of that VM and will skew the results for that organization negatively due to resource contention and perhaps positively due to proximity to the orderer. In a real world implementation, each organization would have its own infrastructure and therefore, we put each organization on a separate VM. Thus, we avoid resource contention arising from an increasing number of containers on the same infrastructure.

5.3. Results

As mentioned in section 5.2, we use Hyperledger Caliper to create and send transaction requests to the implemented blockchain network. Each transaction in our experiments runs the smart contract for creating a token which was described in Algorithm 2. Each transaction thus consists of one query and one created transaction. We ran 10,000 transactions for each data point shown in the graphs. Four worker processes were created to drive the load and we configured the fixed load rate control mechanism in our benchmark. The fixed load rate controller in Hyperledger Caliper starts with a configured send rate in transactions per second (TPS) and maintains a defined backlog of transactions in the network by modifying the send rate. We configured the starting send rate to 1000 TPS and varied the maximum limit of unfinished transactions to observe the effects on the request send rate as shown in Fig. 15. We found that the send rate achieved rose with the increase in the maximum permissible number of unfinished transactions configured. As shown in Fig. 16, with the rise in send rate, the throughput of the system increased. However, average latency per transaction and maximum latency per transaction rose as well. The value of average latency remained under 1 second even at a throughput of over 440.3 TPS which was achieved at a send rate of 443 TPS. The throughput could be increased further by increasing request send rates, but with diminishing returns due to the increase in transaction latency. The current dataset contains data for 300 customers updated every half an hour, which corresponds to a throughput of 0.167 TPS. So, extrapolating, we can say that the current implementation is able to support 792,540 customers with a reasonably low infrastructure footprint. This implementation can be scaled further by providing it with more resources through horizontal and vertical scaling as described in Thakkar and Nathan [27].

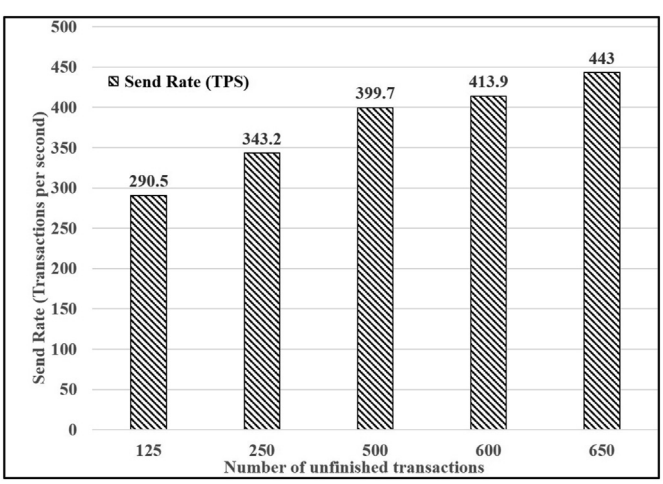


Fig. 15. Request send rate at varying limits of maximum unfinished transactions.

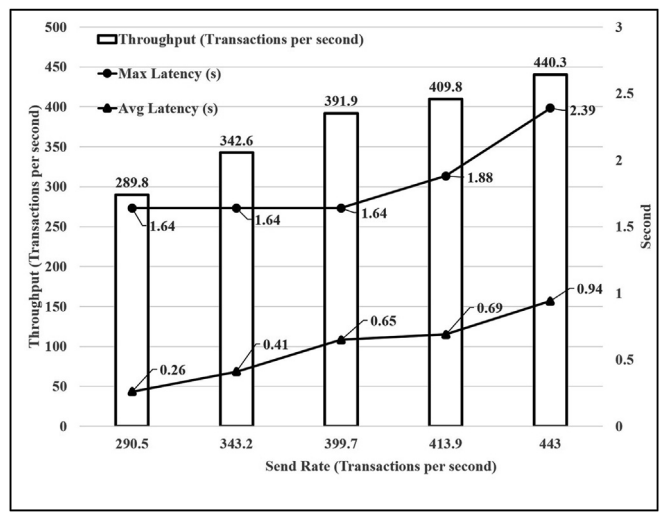


Fig. 16. Transaction throughput and latency at varying send rates.

Algorithm 2. Create transaction token

```
Algorithm 2 Create transaction token
function CREATETOKEN(Customer string, Postcode string, TimeStamp string, SolarProductionKWH float, EnergyConsumptionKWH float, PriceAUD-PerMWH float, TotalDemandMWH float, Latitude float, Longitude float)
    invokerorg ← organization of invoker client
    if invokerorg is not Power Company then
        Invalid invoker
    end if          ▷ Only the Power company can initiate the transaction

    Create integer array from string timestamp
    idval := customer + "_" + timestamp

    exists ← TokenExists(token idval)
    if exists = true then return error
    end if
    if energyconsumptionKWH <= t1 and timestamp month in (summer, winter) then RSeasonal =true
    end if
    if energyconsumptionKWH <= t1 and timestamp date is weekend then RWeekend =true
    end if
    if energyconsumptionKWH <= t1 and timestamp hour in (morning peak, evening peak) then RPeakTime =true
    end if
    if surplusproductionKWH >= ta then RMiniProducer =true
    end if
    if surplusproductionKWH >= tb then RProducer =true
    end if
    if surplusproductionKWH >= tc then RMegaProducer =true
    end if
    Putinworldstate(idval, token)                    ▷ Saving new token
end function
```

6. Related works

Several studies [28–31] focus on different aspects of blockchain-based energy trading among prosumers in a microgrid. Our work presents an approach for incentive-based peak shaving and does not discuss prosumer to prosumer trading.

Pop et al. [32] present an architecture for implementing demand response programs in microgrids. Their solution, implemented in Ethereum [33] takes the baseline values for each prosumer which is the average of their past values and then calculates the required flexibility per prosumer. Their results do not include an analysis of the performance of their blockchain implementation itself.

Di Silvestre et al. [34] also present a framework for demand side response by calculating baseline usage per customer, publishing the required reduction in usage with the reduction amount and time window, and monitoring compliance. Their system architecture considers two organizations, one composed of grid and market operators and the other composed of market operators and customers. This architecture reduces the agency that the users have in the implemented business logic. Also, the article does not characterize the performance of the blockchain implementation.

Guo et al. [35], Wang et al. [36] and Afzal et al. [37] also propose an individualized incentivization model. These articles also do not discuss the performance of their blockchain implementation. Moreover, Afzal et al. [37], implement their solution using Ethereum. In addition, Di Silvestre et al. [34], Guo et al. [35], Wang et al. [36] and Afzal et al. [37] do not use real world data to inform their incentivization logic, but Guo et al. [35] and Wang et al. [36] do use real world data to validate their model.

In our work, the logic of our incentivization system is informed by an in depth analysis of the Ausgrid dataset. The logic encoded in the smart contracts is based on an aggregation analysis and semantic clustering of all transactions so that all customers are subject to the same rules. Thus, it does not penalize good performers. Our system was implemented in Hyperledger Fabric, which is formed of authenticated nodes with clearly defined privileges in the network and provides identity management and provenance tracing. Hyperledger Fabric authentication can also help companies fulfil their legal obligations like Know your customer and Anti-money laundering which are imposed by governments of several countries [38]. Also, the consensus in Hyperledger Fabric is achieved based on an agreed endorsement policy and thus it does not rely on computation intensive and thus energy intensive mechanisms like Proof of Work to reach consensus. In our architecture, the user platform is a separate organization that gives the user representatives the opportunity to review and approve the business logic before it is updated on the platform. Further, we performance test our implementation under varying loads and present our findings in terms of transaction throughput and latency.

7. Conclusion

This work presents a data-centric approach for incentive-based peak shaving and demonstrates the implementation of a blockchain-based reward platform. First, we extracted from the Ausgrid dataset, aggregated user energy behavior in different temporal contexts such as seasons (summer, winter, spring, autumn), days of the week (weekday, weekend), and time of the day. Analysing the aggregated user profile gave us peak consumption times in seasonal, weekly, and daily contexts, as well as thresholds to categorize surplus production. Semantic linking and contextual clustering and labelling of this data gave us thresholds to categorize user demand. This analysis informed the logic of the smart contract in our blockchain implementation. Based on this study, we present the following conclusions.

1) Seasonal peak consumption was observed in the summer and winter seasons. On a weekly basis, the highest consumption was seen on weekends. Moreover, there are two peaks observed daily, one before the start of the work day and one in the evening at the end of the workday. The highest variation in consumption was also seen in summer. Additionally, outliers associated with high consumption were linked to increased demand during heat waves in the region.
2) Aggregation analysis of surplus production gave us thresholds t_a, t_b and t_c for categorizing transactions into Mini Producer, Producer and Mega producer categories respectively. The values for t_a, t_b and t_c were 0.134000 kWh, 0.284000 kWh, and 0.477000 kWh, respectively.
3) Contextual clustering identified two thresholds for production t_1 and t_2 with values 1.43016 and 2.8603, respectively. Similarly, for user demand layer, the thresholds t_1 and t_2 with values 1.8306 and 3.6608 were identified. These threshold values enable us to characterize production and demand in low, medium, and high categories.
4) The implementation of the blockchain-based reward system encoded a smart contract with the business logic for earning rewards based on our analysis. If the conditions for a given reward are met, the smart contract processes the reward for the transaction. A given transaction can acquire multiple rewards.
5) While the Power company is the only organization allowed to initiate transactions, all three organizations must endorse each transaction in order to perform various checks and prevent the management from being concentrated with a particular entity. This implementation with a reasonably low infrastructure footprint was shown to be able to support 792,540 customers.

Author contributions

Nikita Karandikar: Conceptualization, methodology, software, validation, investigation, writing (original draft), writing (review & editing); **Rockey Abhishek:** Conceptualization, methodology, software, formal analysis, data curation, writing (original draft), writing (review & editing); **Nishant Saurabh:** Conceptualization, software, writing (original draft); **Zhiming Zhao:** Conceptualization, supervision; **Alexander Lercher:** Software; **Ninoslav Marina:** Conceptualization, supervision; **Radu Prodan:** Conceptualization, supervision; **Chunming Rong:** Conceptualization, supervision; **Antorweep Chakravorty:** Conceptualization, writing (review & editing), supervision.

Funding

This research was funded by the Project number 267967: Energix of NFR (Norwegian Research Council) and Grant number 825134: ARTICONF of European Union's Horizon 2020 program. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] X. Yang, L. He, Y. Xia, Y. Chen, Effect of government subsidies on renewable energy investments: the threshold effect, Energy Pol. 132 (2019) 156–166.
[2] L. Olkkonen, K. Korjonen-Kuusipuro, I. Grönberg, Redefining a stakeholder relation: Finnish energy "prosumers" as co-producers, Environmental Innovation and Societal Transitions 24 (2017) 57–66.
[3] A. Lantero, How Microgrids Work, vol. 17, US Department of Energy, 2014.
[4] K. Qian, C. Zhou, M. Allan, Y. Yuan, Modeling of load demand due to ev battery charging in distribution systems, IEEE Trans. Power Syst. 26 (2011) 802–810.
[5] M. Uddin, M.F. Romlie, M.F. Abdullah, S. Abd Halim, T.C. Kwang, et al., A review on peak load shaving strategies, Renew. Sustain. Energy Rev. 82 (2018) 3323–3332.
[6] V. Mehta, R. Mehta, Principles of Power System: Including Generation, Transmission, Distribution, Switchgear and Protection, S. Chand, 2005.
[7] M. Erol-Kantarci, H.T. Mouftah, Tou-aware energy management and wireless sensor networks for reducing peak load in smart grids, in: 2010 IEEE 72nd Vehicular Technology Conference-Fall, IEEE, 2010, pp. 1–5.
[8] Ausgrid, Ausgrid demand management newington grid battery trial, 2016. https://www.ausgrid.com.au/-/media/Documents/Demand-Mgmt/DMIA-research/Ausgrid_Newington_Grid_Battery_Report_Final.pdf. (Accessed 17 February 2021).
[9] S.I. Sulakov, The cross-border trade impact on the transmission losses, in: 2017 15th International Conference on Electrical Machines, Drives and Power Systems (ELMA), IEEE, 2017, pp. 115–118.
[10] W. Huang, N. Zhang, C. Kang, M. Li, M. Huo, From demand response to integrated demand response: review and prospect of research and application, Protection and Control of Modern Power Systems 4 (2019) 1–13.
[11] E.L. Ratnam, S.R. Weller, C.M. Kellett, A.T. Murray, Residential load and rooftop pv generation: an australian distribution network dataset, Int. J. Sustain. Energy 36 (2017) 787–806.
[12] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, 2018, pp. 1–15.
[13] H. Caliper, Hyperledger caliper architecture, Electronic Article, 2021, https://hyperledger. github. io/caliper/docs/2_Architecture. html (visited on 04/04/2021).
[14] Ausgrid, Community batteries. https://www.ausgrid.com.au/In-your-community/Community-Batteries, 2021. (Accessed 17 February 2021).
[15] Ausgrid, Solar home half-hour data - 1 july 2012 to 30 june 2013, 2013. https://www.ausgrid.com. au/-/media/Documents/Data-to-share/Solar-home-electricity-data/Solar-home-half-hour-data—1-July-2012-to-30-June-2013.zip.
[16] H. Liu, D. Azuatalam, A.C. Chapman, G. Verbič, Techno-economic feasibility assessment of grid-defection, http://www.sciencedirect.com/science/article/pii/S0142061518321136, Int. J. Electr. Power Energy Syst. 109 (2019) 403–412, https://doi.org/10.1016/j.ijepes.2019.01.045.
[17] B. Jeddi, Y. Mishra, G. Ledwich, Dynamic programming based home energy management unit incorporating pvs and batteries, in: 2017 IEEE Power Energy Society General Meeting, 2017, pp. 1–5, https://doi.org/10.1109/PESGM.2017.8273925.
[18] B. Yildiz, J.I. Bilbao, J. Dore, A. Sproul, Household electricity load forecasting using historical smart meter data with clustering and classification techniques, in: 2018 IEEE Innovative Smart Grid Technologies - Asia, ISGT Asia), 2018, pp. 873–879, https://doi.org/10.1109/ISGT-Asia.2018.8467837.
[19] Bureau of Meteorology, Special Climate Statement 43—extreme Heat in January 2013, A.G., 2013, 2013, http://www.bom.gov.au/climate/current/statements/scs43.pdf.
[20] AEMO, A.E.M.O., Aggregated price and demand data - historical. https://aemo.com.au/energy-systems/electricity/national-electricity-market-nem/data-nem/aggregated-data, 2013.
[21] M. Kivelä, A. Arenas, M. Barthelemy, J.P. Gleeson, Y. Moreno, M.A. Porter, Multilayer networks, Journal of complex networks 2 (2014) 203–271.
[22] M. Ankerst, M.M. Breunig, H.P. Kriegel, J. Sander, Optics: ordering points to identify the clustering structure, ACM Sigmod Record 28 (1999) 49–60.
[23] G. Crasta, A. Malusa, The distance function from the boundary in a minkowski space, Trans. Am. Math. Soc. 359 (2007) 5725–5759.
[24] Golang, The go programming language. https://golang.org/, 2021. (Accessed 17 February 2021).
[25] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm, in: 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14), 2014, pp. 305–319.
[26] P. Thakkar, S. Nathan, B. Viswanathan, Performance benchmarking and optimizing hyperledger fabric blockchain platform, in: 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), IEEE, 2018, pp. 264–276.

[27] P. Thakkar, S. Nathan, Scaling Hyperledger Fabric Using Pipelined Execution and Sparse Peers, arXiv preprint arXiv:2003.05113, 2020.
[28] N. Karandikar, A. Chakravorty, C. Rong, Transactive energy on hyperledger fabric, in: 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), IEEE., 2019, pp. 539–546.
[29] G. van Leeuwen, T. AlSkaif, M. Gibescu, W. van Sark, An integrated blockchain-based energy management platform with bilateral trading for microgrid communities, Appl. Energy 263 (2020), 114613.
[30] A. Kumari, R. Gupta, S. Tanwar, S. Tyagi, N. Kumar, When blockchain meets smart grid: secure energy trading in demand response management, IEEE Network 34 (2020) 299–305.
[31] N. Karandikar, A. Chakravorty, C. Rong, Renewledger: renewable energy management powered by hyperledger fabric, in: 2020 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2020, pp. 1–6.
[32] C. Pop, T. Cioara, M. Antal, I. Anghel, I. Salomie, M. Bertoncini, Blockchain based decentralized management of demand response programs in smart energy grids, Sensors 18 (2018) 162.
[33] Ethereum, Welcome to ethereum, 2021, https://ethereum.org/en/, 2021. (Accessed 17 February 2021).
[34] M.L. Di Silvestre, P. Gallo, E.R. Sanseverino, G. Sciumè, G. Zizzo, Aggregation and remuneration in demand response with a blockchain-based framework, IEEE Trans. Ind. Appl. 56 (2020) 4248–4257, https://doi.org/10.1109/TIA.2020.2992958.
[35] Z. Guo, Z. Ji, Q. Wang, Blockchain-enabled demand response scheme with individualized incentive pricing mode, Energies 13 (2020) 5213.
[36] J. Wang, H. Zhong, C. Wu, E. Du, Q. Xia, C. Kang, Incentivizing distributed energy resource aggregation in energy and capacity markets: an energy sharing scheme and mechanism design, Appl. Energy 252 (2019), 113471.
[37] M. Afzal, Q. Huang, W. Amin, K. Umer, A. Raza, M. Naeem, Blockchain enabled distributed demand side management in community energy system with smart homes, IEEE Access 8 (2020) 37428–37439, https://doi.org/10.1109/ACCESS2020.2975233.
[38] C. Bührer, I. Hubli, E. Marti, The regulatory burden in the swiss wealth management industry, Financ. Mark. Portfolio Manag. 19 (2005) 99–108.