Journal of Big Data

RESEARCH Open Access

Big data resolving using Apache Spark for load forecasting and demand response in smart grid: a case study of Low Carbon London Project

Hussien Ali El-Sayed Ali[1], M. H. Alham[1*] and Doaa Khalil Ibrahim[1]

*Correspondence:
moh.alham@eng.cu.edu.eg

[1] Electrical Power Engineering Department, Faculty of Engineering, Cairo University, Giza 12613, Cairo, Egypt

Abstract

Using recent information and communication technologies for monitoring and management initiates a revolution in the smart grid. These technologies generate massive data that can only be processed using big data tools. This paper emphasizes the role of big data in resolving load forecasting, renewable energy sources integration, and demand response as significant aspects of smart grids. Meters data from the Low Carbon London Project is investigated as a case study. Because of the immense stream of meters' readings and exogenous data added to load forecasting models, addressing the problem is in the context of big data. Descriptive analytics are developed using Spark SQL to get insights regarding household energy consumption. Spark MLlib is utilized for predictive analytics by building scalable machine learning models accommodating meters' data streams. Multivariate polynomial regression and decision tree models are preferred here based on the big data point of view and the literature that ensures they are accurate and interpretable. The results confirmed the descriptive analytics and data visualization capabilities to provide valuable insights, guide the feature selection process, and enhance load forecasting models' accuracy. Accordingly, proper evaluation of demand response programs and integration of renewable energy resources is accomplished using achieved load forecasting results.

Keywords: Apache Spark, Big data (BD), Demand response (DR), Load forecasting (LF), Smart grid (SG), Smart meters

Introduction

The electric grid changes to be smarter by deploying modern information and communication technologies like smart meters and digital relays. Smart meters are widely installed in the smart grid (SG) [1], and they are the primary source of data generated across it [2]. They provide a communication channel between the utility and consumers and offer online energy measurements [1–3]. Combining these measurements with exogenous data (weather, demographic, and holiday data) can be applied to understand

consumption patterns, make load forecasting (LF) [4], and enable demand response (DR) programs.

DR programs become more significant in the SG context because they can alter load profiles to be more flexible with grid conditions and maintain grid security against overloading and power failure [2, 5]. Applying DR programs requires accurate forecasting for the electric energy consumption and the available power on the generation side, which is more stochastic with the high penetration of renewable energy sources (RES). LF can help the utility provide reliable service by predicting load consumption and then using the predicted values as references for DR programs. Utilities also depend on LF to plan for network upgrading and maintenance activities. Dynamic Time-of-Use (DToU) tariff is one of the DR programs; where the electricity price has fixed rates at different times [5].

The main challenge to using measurements of smart meters is that these devices generate enormous amounts of data that cannot be handled using traditional tools and software [4, 6]. The ability to digest, store, and analyze this data can reveal new insights about load usage patterns and provide a database for more accurate and scalable Machine Learning (ML) models [1]- [3]. Using tools of big data (BD) becomes more relevant because the data in the SG has the same features as BD. These features are summarized by:

- Volume: Huge amounts of data are generated (*e.g.*, over a year and a 15-min resolution, 1 million smart meters generate data of 2920 TB volume [7]).
- Velocity: The data is streamed at a very high speed. Achieving fast processing and online decision-making (*e.g.*, dynamic demand response [7]) needs powerful tools.
- Value: The data generated has no value till usable knowledge is extracted [2].
- Variety: The data generated in SG may be:
  - Structured data (energy consumption data and measurements taken by digital relays),
  - Semi-structured data (data obtained from XML data or web services like weather data), or
  - Unstructured data (SMS notification about tariff and energy use) [7].

BD issues in SG

Researchers are enthusiastic about arguing sources, applications, and challenges of BD in SG which are reviewed in Table 1. As shown in Table 1, there are a lot of BD sources distributed across SG. These data sources capture high-resolution data that can be used in several applications. Applications in distribution systems are the most interesting ones regarding the electricity market, grid economics, and efficient operation. One main challenge that hinders these applications is the last one specified in Table 1, which is data processing and analysis with cost-effective tools and models. This study addresses this challenge by applying BD tools to handle BD sets and extract beneficial knowledge from them.

Over time, BD technologies have been developed. The need for tools that can harvest, store, process, and analyze the data generated rises to encounter the era of BD. Hadoop and Apache Spark are the most common and open-source BD ecosystems [9, 16]. In [9], a comprehensive explanation of the components of these ecosystems is provided.

Table 1 BD sources, applications, and challenges in SG

BD sources	Smart meters [1, 8–12]
	Weather data [1, 2, 12]
	Gas turbines and wind turbines [13]
	Sensors (*e.g.*, Internet of Things sensors, geographical information system data) [9, 12]
	Substation data collected from: - Phasor Measurement Units (PMUs) [9, 12], - Remote Terminal Units (RTUs) [9, 12], and - Digital relays
	SCADA [8, 9, 12]
BD applications	LF and RES forecasting [1, 2, 9, 12] to improve integration of RES [12]
	DR applications [8, 12, 14, 15]
	Asset management [2, 12]
	Preventive maintenance and health monitoring [9]
	Power quality monitoring [9]
BD challenges	BD management issues such as: - Data privacy and security [8, 12] - Data storage [12]
	Data processing and analysis [2, 16] with cost-effective solutions [12]

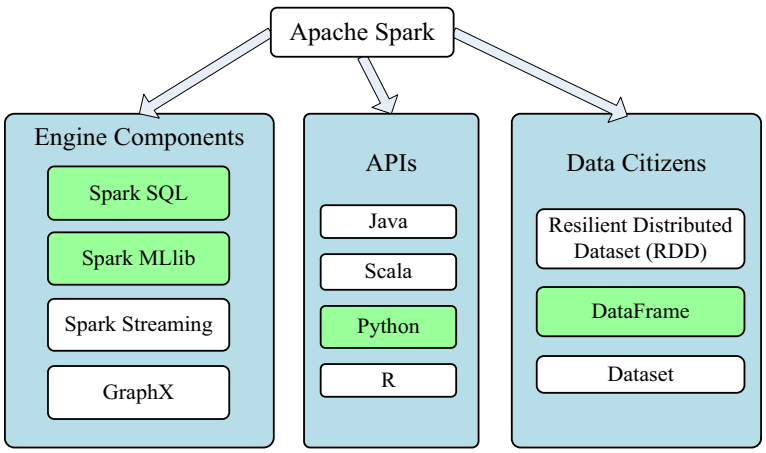

Fig. 1 Apache Spark engine components, APIs, and data citizens

Therefore, the advantages of the Spark ecosystem over Hadoop are summarized as follows:

- Speed: Spark performs computations in memory and not on the desk. This feature saves the time of writing on and retrieving data from the desk.
- Ease of use: Spark supports many application programming interfaces (APIs): Python, R, Scala, and Java.
- Capability: Spark supports streaming applications, machine learning, and SQL (like querying).

Figure 1 illustrates the components of the Apache Spark engine, Spark APIs, and Spark data citizens. As presented in Fig. 1, Spark has four distinctive engines. This research

utilizes Spark SQL and Spark MLlib. Spark SQL is used for descriptive analytics while Spark MLlib is used for predictive analytics to build scalable ML models using the power of distributed clusters.

Besides the four components of the Spark engine displayed in Fig. 1, Spark uses Hadoop Distributed File System (HDFS) for data storage. Currently, the DataFrame is the most popular data citizen supported by Spark, which is applied in this work. Python, one of the supported APIs in Spark, is used in this work.

LF and ML models

Because of the large volume of smart meters data and the exogenous data considered in LF problems, LF in SG becomes a BD problem. Exogenous data (like weather data, social information, and geographic information system data) increases the predictability and accuracy of the model [7]. In this research work, short-term LF for households is applied to enable and assess the DR programs since the households' energy consumption accounts for a large portion of the total energy consumption in the SG [7].

Some LF techniques have evolved from statistical-based methods, such as Auto Regressive Moving Average (ARMA), Auto Regressive Integrated Moving Average (ARIMA) [17], and Regression models [18, 19]. Some other techniques are ML and deep learning algorithms [3, 20]. This section presents a literature review of the ML models used for LF tasks in the BD context.

Different predictive models used for LF tasks are discussed in [20]. According to such review, Artificial Neural Networks (ANN) are gaining more confidence in the predictive modeling task as they can capture nonlinear relationships and thus deliver more accurate results. A detailed LF methodology, starting from the data gathering phase to finally predict the demand, is proposed in [21] using seven different ML algorithms based on BD. Nevertheless, such algorithms were implemented in Python, even though Python is not adequate for BD problems.

Although many reported efforts have been exerted on the LF problem, fewer ones addressed the problem using BD tools. Spark and Prophet (an open-source forecasting procedure offered by Facebook) were utilized to make LF for simulated data [17]. Such work proved that adding meteorological data enhanced the accuracy of LF. In [22], different ML models implemented in Spark were compared against the same models in the Python Sklearn library using residential smart meters data offered by the Low Carbon London (LCL) project. The results indicated that Sklearn could not handle BD sets, and employing Spark was mandatory for this case. Also, the results displayed that ML models of Spark were more accurate than those of the Sklearn library. Linear regression (LR) model was verified by such work to have the least testing and training times besides the best accuracy. One defect of LR model is that it requires high computational power due to matrix operations. By decreasing number of features and applying multi-core parallel processing in [18] made LR model faster and mitigated this problem. The abovementioned works were interested in the LF problem without further applications in SG.

Captured high-resolution data by smart meters can be used for energy consumption monitoring and DR applications. In [1], Spark is employed for the task of the Extract-Transform-Load (ETL) process, Tslearn library (a Python library) is used to make LF, and finally, Tableau is engaged for descriptive analytics. The accuracy of that model

was high, but the applied tools were not appropriate for BD applications. The role of BD in DR was emphasized in [2, 12]. Some solutions for DR were presented in [10, 14, 15] based on BD. Data from smart meters was utilized to predict customer eligibility for being recruited in DR programs [10]. An approximated optimized function was implemented in [14] for large-scale customer selection for DR programs based on the data of smart meters.

Research gap and contribution

From the literature review, it is noticed that few research works were interested in providing scalable solutions for SG applications. Most of the work was conducted from the computer science point of view, like testing and evaluating a suggested BD platform or comparing different predictive algorithms using a given dataset. Many works have used compound predictive models for LF tasks to enhance accuracy. Clustering is applied as a pre-step before the regression model to group customers and then fit a customized regression model for each group. Compound models require more computational burdens than single models, and these requirements rise exponentially for BD sets. Also from the literature, it is deduced that the contribution of BD analytics to RES integration in SG was limited. For that reason, the main contributions of this study are shortened to the following:

- Discuss the eligibility of different ML models suitable for BD applications.
- Address the LF task through a case study using actual smart meters data to build scalable, high-accuracy, and simple (*i.e.*, not compound) models. The models are established based on a concise feature selection using the descriptive analysis results.
- Compare the ML models applied (regarding their accuracy, fitting, and testing times), and then get a recommendation about which model to use.
- Utilize the LF results to put a framework for evaluating the DToU tariff and help to integrate RES efficiently. This framework would make the SG operation more economical and efficient.

Therefore, this paper employs Apache Spark to analyze the massive data released by the smart meters to address the research question of how big data analytics can provide new insights to the Distribution Network Operator (DNO) to help in RES integration and implementation of DR programs. Also, this work investigates how to provide accurate LF results with low computational efforts in ML models. Applying BD tools in SG is postulated to provide valuable insights that would help the DNO understand consumption behaviors and provide scalable ML models for LF tasks. LF results are also proposed to be the baseline for applying DR programs and issuing DToU tariffs.

The rest of the paper is organized as follows: "Description of LCL Project Data" section introduces the LCL project and data used in this research work. "ML Models for Scalable Models" section briefly argues the ML models from the BD point of view and justifies the selection of LR and decision tree regression (DTR) models. Then, a brief introduction to the two selected models is presented in "Introduction to LR, DTR Models, and Evaluation" section. The methodology, platform, and framework used for evaluating customer response to DToU tariffs and for integrating RES are revealed in "Proposed

Methodology" section. Results for descriptive and predictive analytics are discussed in "Results and Discussion" section. Conclusions are drawn in the last section.

Description of LCL project data

This paper has used the data of the LCL project as a case study. The utilized data is the energy consumption captured by smart meters of 5,567 households in London between November 2011 and February 2014 with a 30-min interval. In 2013, 1044 households were selected for the DToU experiment for one year. The total readings captured over the project were over 167 million records [23]. Such data is open for reuse on the website of UK Power Networks [24]. Utilities in the UK apply a customer segmentation tool called ACORN that segments customers based on demographic conditions. Figure 2 illustrates the count of houses enrolled in the LCL project grouped by their ACORN category (Affluent, Comfortable, or Adversity) and the type of program they were enrolled in (either standard or DToU program). Based on [23, 24], two different types of DToU tariff events are proposed for the LCL project:

- Constraint Management (CM): It is used when the predicted load exceeds the grid capability limits. This type is critical for the distribution network because it prevents the grid from overloading and saves money spent to reinforce the grid. CM events were established with a Low–High-Low (LHL) price pattern. As a result, customers can shift their unnecessary loads to the low tariff periods.
- Supply Following (SF): A combination of high and low DToU prices are used to encourage customers to shift load away from periods where there is a shortfall in the supply of electric power to periods with surplus power supply and high RES generation.

As energy consumption depends on weather data and demographic conditions, such data was added to enrich our analysis and ensure more precise forecasting. The weather data was involved from Darksky API.

It is worth mentioning that LCL project data is utilized by other prior works such as [1, 4, 22, 25]. Table 2 displays a quick comparison among such works. It summarizes the scope, platform used, ML model, and target variable of such investigations. Since

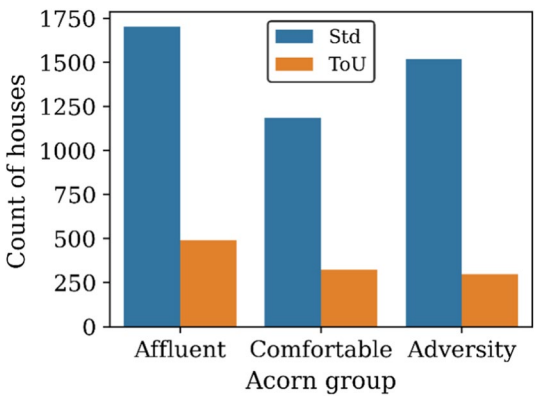

Fig. 2 Houses enrolled in the LCL project grouped by ACORN category

Table 2 Some research works used LCL project data

Refs.	Scope	Platform used (Programming)	ML model	Target Variable	Concerns	Contribution to		
						ML	BD	SG
[1]	- BD solutions for smart meters were reviewed from past publications - LCL data was used as a case study	- Spark for ETL and analysis, - Tableau for visualization - Tslearn (a Python library) for ML model	Clustering + nonlinear autoregressive exogenous model (NARX) model	Hourly load consumption for each house and each cluster (customer group)	- Tslearn and Tableu cannot be used for BD applications - The accuracy of the ML model was high because of the clustering model that divides customers into custom groups	Yes	No	Yes
[4]	BD platform is proposed for smart meter data analytics and compared with reported results of other competitive platforms like Google Cloud and IBM	- Hadoop - HDFS (Java)	Not applied	Not applied	- Results indicated that the proposed platform is competitive with other tested platforms	No	Yes	No
[22]	The performance of different regression models on Sklearn and Pyspark are compared	- Spark MLlib - Sklearn (a Python library)	All models are implemented in Pyspark	Aggregated energy consumption per day per household	- Results revealed that Spark MLlib models were more accurate and scalable than Sklearn models	Yes	Yes	No
[25]	Data is used to test the proposed multiple cycles self-boosted neural network framework for household LF (not aggregated)	Not mentioned	Multiple cycles self-boosted neural network framework	Hourly load consumption for each household	- Framework results were better than other ML models presented in the literature - It did not address the problem in the BD context	Yes	No	No

utilizing BD for SG applications is the main issue of this study, the main concerns for previous work in [1, 4, 22, 25]; regarding the contribution of BD and SG are also highlighted in the last columns of Table 2.

ML models for scalable models

This study considers Spark MLlib for its advantages noted before and meanwhile, Apache Spark supports data streaming and querying that serve applying ML models. In the beginning, brief guidelines for different regression models and their eligibility for BD applications are discussed in this section.

ANNs are the most popular ML models but are not implemented in Spark MLlib. Furthermore, ANNs are non-self-interpretable models, require high computational power, and are prone to overfitting [18]. K-nearest neighbor and support vector regression models are not implemented in Spark Mllib. K-nearest neighbor is not suitable for BD sets. It requires vast memory to save all data points, and the prediction time is very long since the computations are repeated each time a new reading comes, which is not acceptable for fast streaming data [26]. Because of the kernel trick that the support vector machine models apply, such models are not suitable for BD sets since the kernel trick needs much memory and processing. On the other hand, the regression models implemented in Spark MLlib are investigated in [22]. According to [22], random forest, LR, and DTR models have the best-evaluating indices (*e.g.*, accuracy, testing time, and training time). Ensemble learning is another promising domain in the ML field, and it is proven to provide accurate models. The idea of ensemble learning is to use different models to obtain better performance. It relies on the concept of the wisdom of the crowd. However, this type suffers from extraordinary computational costs when it fits large-scale datasets [27]. The ensemble learning models in Spark MLlib are random forest and gradient-boosted trees based on DTR models.

Based on these fundamentals, LR and DTR models were selected to be applied in this study. LR and DTR are self-interpretable models. It means that their results can be easily understood and communicated with stakeholders, and ML engineers can inspect the model and troubleshoot any errors after revising the results. Nevertheless, LR requires massive computational resources due to heavy matrix operations; this work makes a concise feature selection, as discussed in [18], to reduce the matrix size so the model computation effort and time are optimized.

In summary, this paper considers regularized LR and DTR models. Regularized LR models often reduce model over-fitting and remove the least effective model coefficients. Besides, eliminating some model coefficients reduces the computation effort for LR. The following section briefly introduces the LR model with its regularized versions and the DTR model.

Introduction to LR, DTR models, and evaluation

Introduction to the LR model

The linear regression (LR) model is the most elementary regression model that assumes a linear relationship between the target and dependent variables [18]. Equation (1) displays

this relationship where y is the target variable (load consumption in this study), X is the matrix of independent variables, β is the matrix of coefficients and C is the intercept.

$$y = \beta \cdot X + C \tag{1}$$

Multivariate Polynomial Regression (MPR) is an extension of LR models by increasing the model degree. MPR can capture nonlinear relationships between the target and dependent variables. MPR chooses the coefficients that minimize the cost function for each training point. Regularized models (such as the Ridge, Lasso, and Elastic net models) penalize model coefficients to avoid over-fitting, improve model generalization, and reduce model fitting and testing times [28].

Cost functions for Ridge, Lasso, and Elastic net models are expressed in Eq. (2), (3), and (4), respectively, where J is the cost function, β_j are model coefficients, $Y_{predicted(i)}$ is the predicted energy consumption, and $Y_{observed(i)}$ is the true observed value for energy consumption. Finally, n denotes the total number of observations or rows, and M describes the number of columns (*e.g.*, features) in the training dataset. Equation (2) and Eq. (3) display that the Ridge model penalizes the model factors by their squared values while Lasso penalizes them by their absolute values. Elastic net models are a combination of Lasso and Ridge models [28]. The term α refers to the mixing parameter that mixes Ridge and Lasso penalties. The parameters λ and α are used interchangeably in ML references.

$$J_{Ridge} = \frac{1}{2n}\sum_{i=1}^{n}(Y_{predicted(i)} - Y_{observed(i)})^2 + \lambda\sum_{j=1}^{M}{\beta_j}^2 \tag{2}$$

$$J_{Lasso} = \frac{1}{2n}\sum_{i=1}^{n}(Y_{predicted(i)} - Y_{observed(i)})^2 + \lambda\sum_{j=1}^{M}\mid \beta_j \mid \tag{3}$$

$$J_{Elasticnet} = \frac{1}{2n}\sum_{i=1}^{n}(Y_{predicted(i)} - Y_{observed(i)})^2 + \alpha\left[\lambda\sum_{j=1}^{M}\mid \beta_j \mid +(\frac{1-\lambda}{2})\sum_{j=1}^{M}{\beta_j}^2\right] \tag{4}$$

Introduction to the DTR model

DTR models are widely used since they are easy to interpret, handle categorical features, do not require feature scaling, and can capture nonlinearities and feature interactions. The implementation of DTR models in Spark MLlib has been optimized to make the model fast and scalable [29]. The algorithm tests the different features to split data on and then selects the feature that achieves the minimum weighted variance [30]. The value of variance is calculated according to Eq. (5). The weighted variance is the corrected value of variance based on the ratio of points on each split.

$$Variance = \frac{1}{n}\sum_{i=1}^{n}(Y_{predicted(i)} - Y_{observed(i)})^2 \tag{5}$$

The result of the regression tree is the mean of all data points included in the final node (*i.e.*, leaf). Splitting data in the tree stops if the maximum depth of the tree is reached, the reduction in the weighted variance is less than a threshold, or the number of points in a node is less than a threshold.

Evaluating ML models

Table 3 summarizes the general characteristics of both DT and MPR models. Spark MLlib provides various evaluation metrics as described in Eqs. (6–9), where *MSE* is the mean squared error, *RMSE* is the root mean squared error, *MAE* is the mean absolute error, and *R*2 is the coefficient of determination (where $\bar{y}$ denotes the average of all true values of the target variable y). These metrics are reported for evaluating and comparing the models. Nevertheless, the *R*2 evaluation metric was used only for the hyperparameter tuning process. When the value of such metrics increases, it indicates that the model has an increased error except for the *R*2 metric. The value of the *R*2 metric ranges between 0 and 1, when its value increases (or approaches 1), it means the model is more accurate.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_{predicted(i)} - y_{observed(i)}\right)^2 \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{predicted(i)} - y_{oberved(i)}\right)^2} \tag{7}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_{predicted(i)} - y_{observed(i)}\right| \tag{8}$$

$$R2 = 1 - \frac{MSE}{\sum_{i=1}^{n}\left(y_{observed(i)} - \bar{y}\right)^2} \tag{9}$$

Table 3 Comparison between MPR and DTR models

Model	Pros	Cons	Over-fitting mitigation
MPR (LR)	- Self-Interpretable - Simple to implement	- Requires feature preprocessing and feature scaling - High computation effort in the case of BD sets due to heavy matrix operations	Model regularization by Ridge, Lasso, or Elastic net
DTR	- Self-Interpretable - Does not require features preprocessing - Handles categorical features	- Tends to be over-fitting - Most likely used for classification problems - Not suitable for high-dimensional data	Pruning (*e.g.,* decreasing the tree depth)

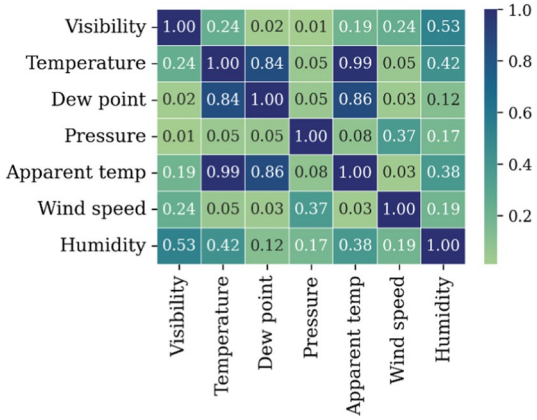

Fig. 3 Absolute correlations among weather variables

Proposed methodology

As discussed before, analyzing LCL project data necessitates the usage of a BD platform because of the large volume of the Dataset and the need for fast processing. So, Python data analysis libraries such as Pandas would crash if used for analyzing such a Dataset. Accordingly, descriptive analytics were performed in this work via the Spark SQL module. Also, the Spark SQL module was used to make summary statistics on the entire Dataset. Matplotlib and Seaborn are used to plot the summarized results obtained from Spark SQL. Sklearn library is not a decent candidate for predictive analytics in this case study because it cannot handle BD sets [22]. On the other hand, Spark MLlib can train scalable and optimized models and thus it was used on this study for predictive analytics.

Feature selection

The number of features is condensed as discussed in [18] to mitigate the heavy matrix computations for the MPR model and decrease the DTR depth and complexity. Reducing the number of input features makes the applied ML models faster, more robust, more interpretable, and more scalable for BD applications. For determining the most effective weather parameters for the energy consumption prediction process, the heat maps for Pearson correlations are plotted. Figure 3 displays the absolute correlations among different weather variables, while Fig. 4 demonstrates the absolute correlations between energy consumption and weather variables. It is worth highlighting that the correlations shown in Fig. 4 denoted by 'Energy' belonging to an individual household energy consumption (one house), while the aggregated sum of energy consumption for one settlement block of houses (*i.e.*, around 50 houses) is denoted by 'Energy sum'.

From Fig. 4, it is inferred that individual household consumption has little correlation with weather parameters (in the range of 0.01 to 0.06). However, the correlation is much more significant for the aggregated sum of energy consumption for the whole block for temperature, apparent temperature, and dew point (of values 0.43, 0.45, and 0.44, respectively). From Fig. 3, strong correlations are observed among temperature,

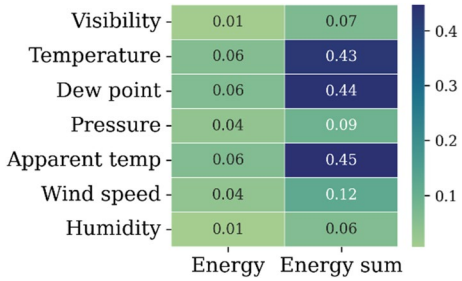

Fig. 4 Absolute correlations between energy consumption and weather variables

apparent temperature, and dew point (with values ranging between 0.84 and 0.99). To avoid multicollinearity, the apparent temperature is only considered since it has the highest correlation with the energy sum variable (0.45).

Fitting regression models

Based on the above analysis, the target variable is designated as "the sum of the energy consumption of all residential houses"; since predicting aggregated energy consumption is much easier than predicting the energy consumption of a single household, where the slight variations of individual household consumption are canceled out. Also, the DNO is more interested in the total substation or feeder load.

Spark MLlib was used for training regularized MPR and DTR models for the sum of load consumption of one settlement block (*i.e.*, Block # 0) and again for all households in the LCL project. To assess the customer response for the DToU program, the sum of load consumption for blocks (# 2, 4, 6, 8, 46, and 48) was also forecasted as these blocks had the same CM events at the same time. It is noteworthy that building regression models for one settlement block is beneficial since DNOs can have insights into the corresponding feeder or the distribution transformer load. Also, utilities need insights into the predicted load for each housing block to target customers; who are more engaged in DR programs.

Table 4 displays a sample of the DatatFrame fed to the ML model predicting the sum of energy consumption for Block # 0 after accomplishing the feature selection process. The meaning of these variables is demonstrated in Table 5. In the results section, other variables that affect the prediction process are investigated. The parameter count (LCLid) is included as there were a lot of missing data records at some time instances. This variable is valuable in real applications to compensate

Table 4 A sample of Spark DataFrame (for Block # 0) after feature selection

Count (LCLid)	Temperature	Hour	Weekend	Sum (energy)
44	8.78	19	1	40.748
47	21.84	8	0	13.408
43	7.41	11	1	27.39
5	6.44	4	0	1.302

Table 5 Features description

Feature Name	Type	Description
Count (LCLid)	Feature variable	The number of houses registered at this time instance because there were many house records missed at some times
Temperature	Feature variable	The ambient temperature at this time (in Celsius)
Hour	Feature variable	The hour at which the energy is recorded
Weekend	Feature variable	Whether the day is a public holiday and weekend or a normal working day (Boolean variable)
Sum (energy)	Target variable	The sum of energy of all houses at the given time instance

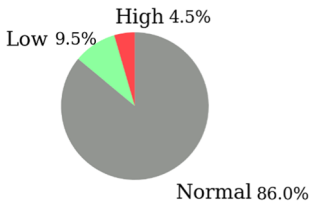

Fig. 5 Tariffs applied in the LCL project

houses with power outages. For example, with a power outage in some part of the network, the affected houses would consume no power.

Concerning the regression model for forecasting the sum of energy of all houses, the features are the same as demonstrated in Table 4 except for the variable count (LCLid). It decomposes into three variables that exhibit the number of houses per ACORN category. This decomposition was not done for the regression model run for one block since the houses of one block belong to the same ACORN category. It is also worth clarifying that the regression models would not consider the applied tariff. As revealed in Fig. 5, the portion involving low and high tariffs was limited, so regression models cannot be trained on these features. However, it can be utilized to evaluate customer responsiveness to DToU tariffs and create an automated framework for tariff design and RES integration.

Assessing customer responsiveness to DToU tariffs

For assessing the customer responsiveness to the DToU tariff, the difference between the predicted energy consumption (from the ML model) and the actual energy consumption is estimated as expressed in Eq. (10), where $E_{predicted}$ and E_{Actual} are the forecast and actual energy consumptions, respectively.

$$\Delta E = E_{predicted} - E_{Actual} \tag{10}$$

By neglecting the error of the ML model itself, there should be some deviation between these two values because ML models do not consider the applied tariff. By applying the low tariffs, it is anticipated that ΔE deviates more to negative values as per the actual consumption increases motivated by the low price of electricity.

Framework for RES integration and issuing DToU tariffs

In the proposed framework, the LF results obtained from the ML model (aligned with RES forecasting) were used to generate proper DToU tariffs and also evaluate customer response to these tariffs. The flowchart of this framework is illustrated simply in Fig. 6. Firstly, the framework guarantees a secure operation for the SG by verifying that the forecasted load does not exceed the grid capability limits (*i.e.,* ratings of downstream feeders) by applying a CM event. Secondly, the framework computes the total RES share, and a supply following (SF) event is issued based on the RES share in load demand. When the predicted RES share is high, a low tariff is suggested, and vice versa for high tariffs. SF events guarantee efficient operation for the SG. Customer response to CM and SF events is evaluated; as described in "Assessing Customer Responsiveness to DToU

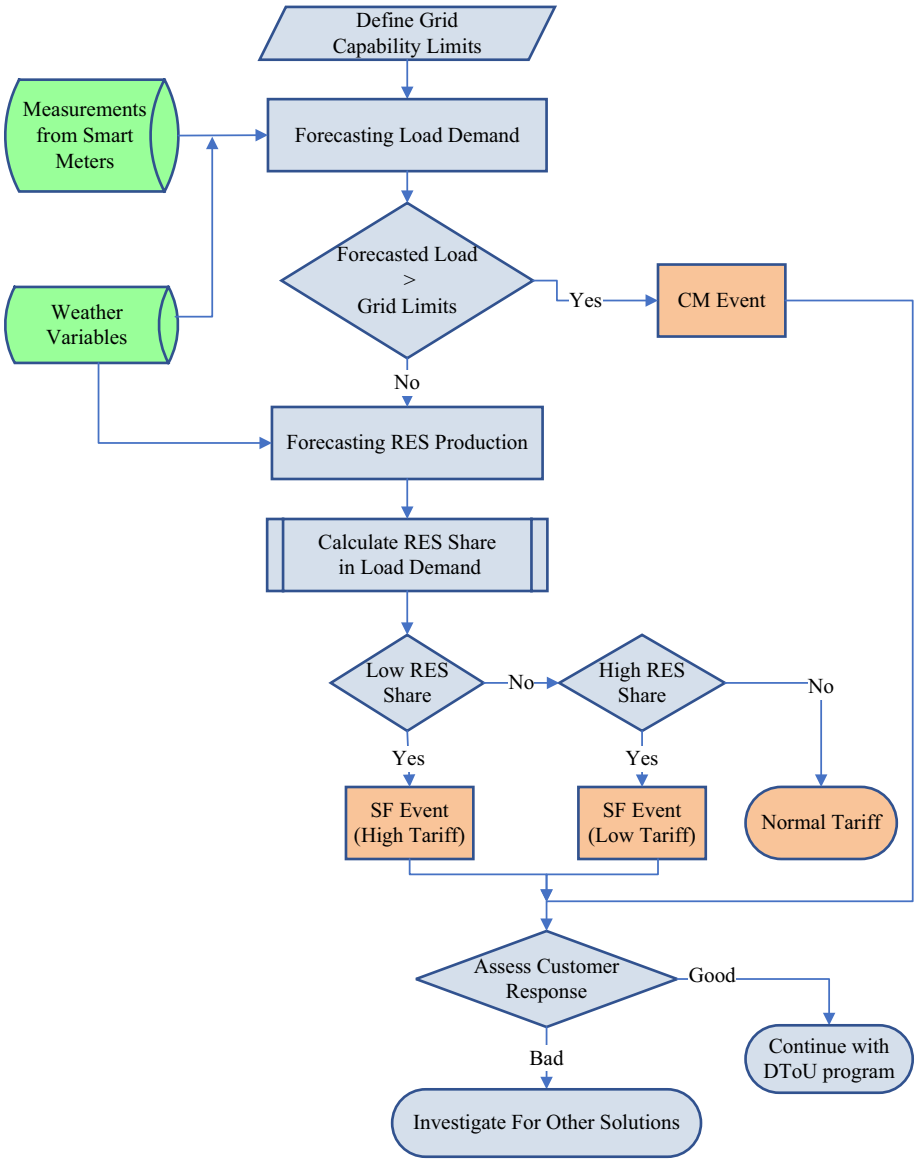

Fig. 6 Flowchart of the proposed framework for DToU tariff generation and RES integration using predictive analytics results

Tariffs" section. In case the response is not satisfactory, other solutions should be considered; such as:

- Increasing the number of customers in the DToU program.
- Organizing campaigns to increase customer awareness about DR programs and the sustainable operation of the SG.
- Investing in energy storage systems (*e.g.*, batteries and electric vehicles).
- Reinforcing the grid to withstand peak loading.

Overall implementation and requirements

Dataproc service from Google Cloud Platform (GCP) is selected for this work. Cloud services provide scalable services that can scale up only when computing power is wanted. This feature lowers monthly fees. The free trial on GCP offers the platform shown in Fig. 7 with sufficient capabilities for analyzing the entire data of the LCL project. It comprises the following:

- One master node that has two virtual CPUs with 7.5 GB RAM,
- Three worker nodes, each has two virtual CPUs and 7.5 GB RAM.

The overall outline for data flow and procedures for Spark implementation in this study is demonstrated in Fig. 8. The following are five main phases in this work:

1) Data gathering from different databases and APIs: as mentioned, smart meters data is combined with ACORN category data, bank holidays data, and weather data to enrich our analysis and build accurate predictive models.
2) Data cleaning and validating: by eliminating null values and false data (like zero energy measurements that would confuse ML models. The zero values are wrong as they represent power outages or bad communications). The portion of that bad-

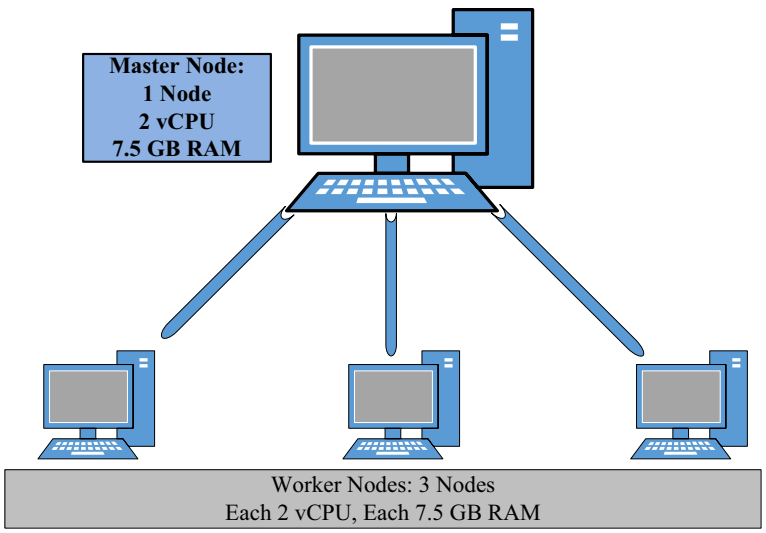

Fig. 7 GCP cluster features

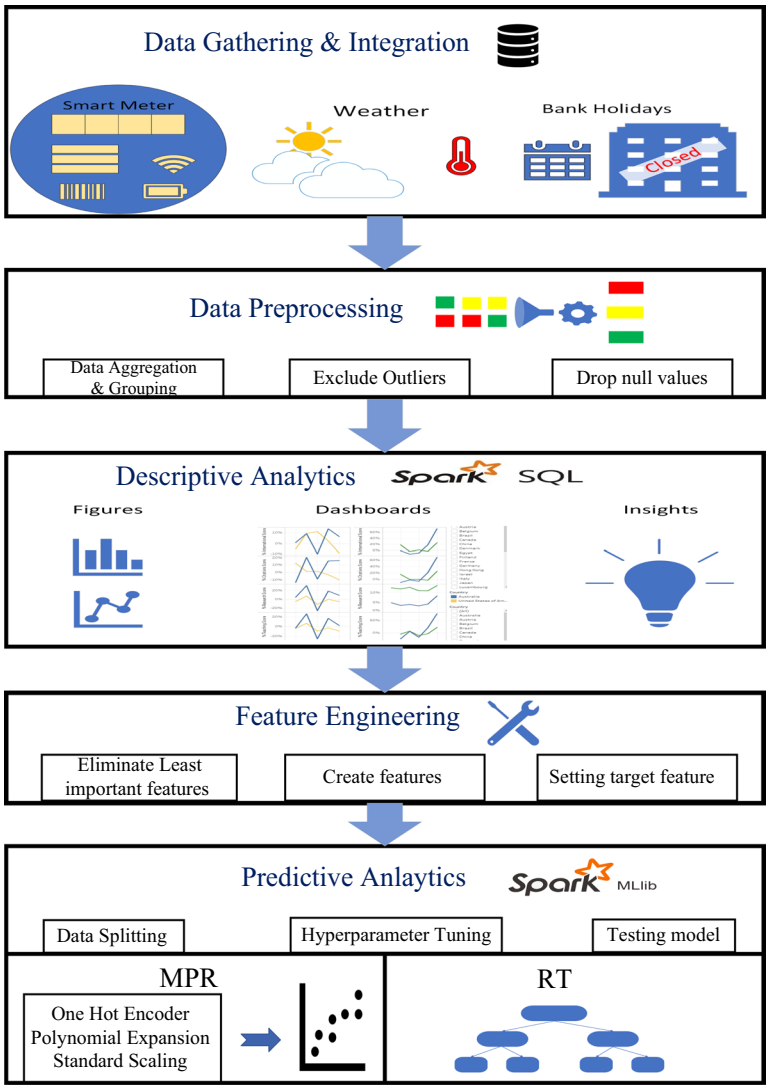

Fig. 8 Data sources and their applications in this work

unclean data is very limited compared to the overall data. Then, the data is grouped and aggregated to be ready for subsequent analysis.

3) Descriptive analytics: via Spark SQL to get insights and recognize consumption behavior.
4) Feature engineering: based on the descriptive analysis was done as explained in "Feature Selection" and "Fitting Regression Models" section. This stage aims to select the most important features and set the target variable of the ML models. New features were added to the model based on the exploratory analysis; like Count (LCLid) feature.
5) Predictive analytics: using Spark MLlib to build MPR and DTR models. DTR models require no data preprocessing. On the other hand, MPR models require some data preprocessing practices (such as feature scaling, polynomial expansion, and hot encoding [31]). Also, the models' hyperparameters are tuned using cross-validation [32].

Results and discussion

In this section, the results obtained from the descriptive and predictive analyses are demonstrated and discussed.

Results of descriptive analysis and discussion

Descriptive analytics assist the DNO in understanding consumer behaviors and patterns, allowing the DNO to make informed decisions for appropriate DR programs and planning preventive maintenance actions.

Regarding all consumers

As previously stated, the Spark SQL module is used to compute summary statistics, and the Matplotlib library is used to plot the summarized data. Figure 9 displays the average load consumption for all the consumers for each hour. On the other side, Fig. 10 demonstrates the average load consumption for each month along with the average apparent temperature recorded in each month. As shown, Fig. 9 agrees with the typical daily profile, and Fig. 10 ensures that the average load consumptions in colder months were higher than in hotter ones. Figure 11 demonstrates the relation between the apparent temperature and the average energy consumption for all time stamps. The regression

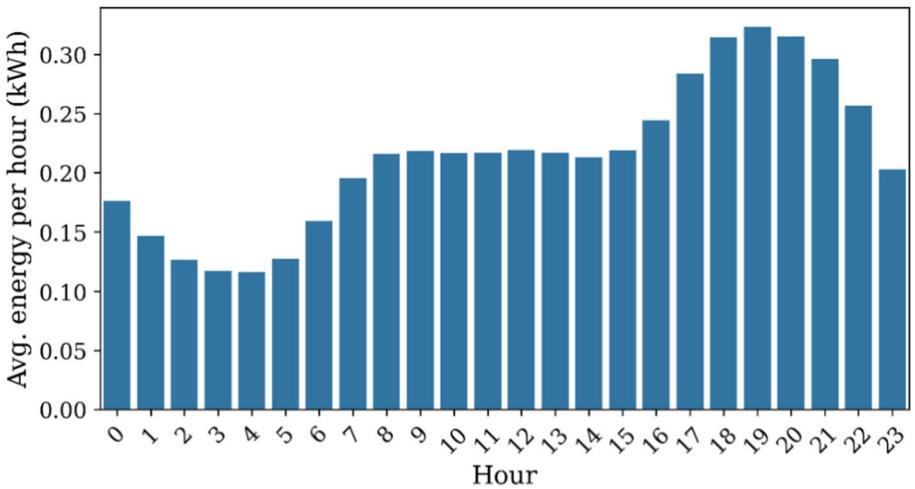

Fig. 9 Average energy consumption per hour

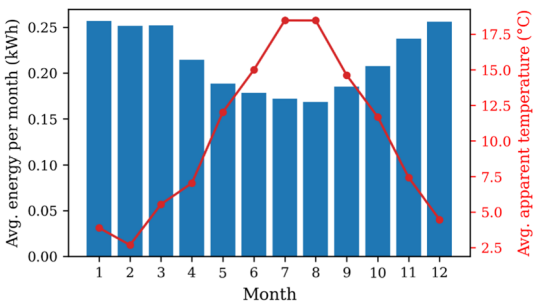

Fig. 10 Average energy consumption and apparent temperature per month

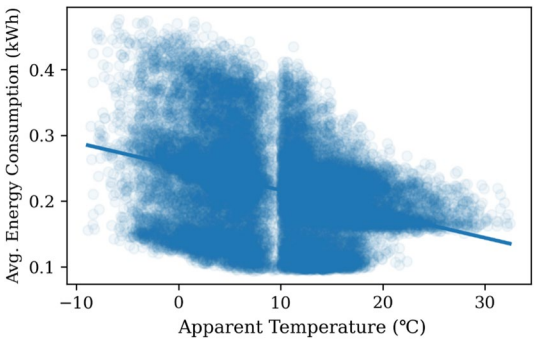

Fig. 11 Scatter plot for average energy consumption versus apparent temperature

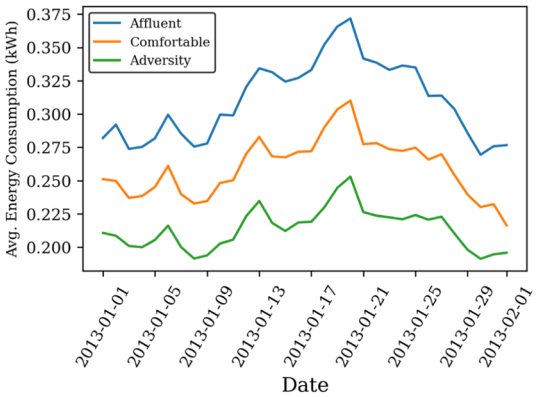

Fig. 12 Time-series plot for average energy consumption per day in January 2013 for each acorn category

line plotted in the graph ensures the negative correlation between the two variables emphasizing the conclusion of Fig. 10.

Regarding the acorn category and weekend feature

Figure 12 displays the average energy consumption for all days in January 2013 based on the acorn category. It highlights how demographic conditions influence the energy consumption pattern. To also investigate the effect of the weekend feature, the average energy consumption for working days is compared with the average consumption for weekends for the three acorn categories, as presented in Fig. 13. As revealed, the average consumption for weekends is slightly higher than normal working days, and the average consumption varies according to the acorn category.

Results from predictive analysis and discussion

Evaluating different tested models

As mentioned before, aggregating hundreds of millions of energy records is only possible using a distributed computing cluster and a BD tool like Spark. Table 6 summarizes the scores of evaluating metrics for different tested models and the value-tuned hyperparameter. The key outcomes of the tabulated scores can be summarized in the following:

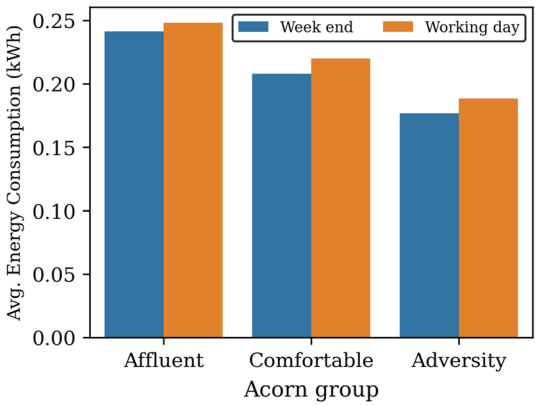

Fig. 13 Average energy for weekends and working days based on acorn category

Table 6 Evaluating metrics for different tested models

Model	For only one block (Block # 0)					For all houses in the LCL project				
	Tuning	*R2*	*MAE*	*MSE*	*RMSE*	Tuning	*R2*	*MAE*	*MSE*	*RMSE*
Plain	NA	0.944	1.721	6.112	2.472	NA	0.963	70.232	9894.756	99.472
Lasso	λ=0.001	0.943	1.778	6.253	2.501	λ= 0.0001	0.961	71.695	10,362.92	101.798
Ridge	λ=0.001	0.943	1.769	6.202	2.490	λ=1e^{-05}	0.965	68.073	9238.572	96.117
Elastic net	λ=0.001 (L2 penalty)	0.943	1.769	6.202	2.490	λ=1e^{-05} (L2 penalty)	0.965	68.073	9238.573	96.117
DTR	Max depth = 5	0.91	2.004	10.338	3.215	Max depth = 5	0.922	107.58	23,989.95	154.88

- The $R2$ metric is higher for the MPR model than the DTR model.
- The performance of regularized and non-regularized models is almost the same. It is because the features were selected wisely before fitting the models. Also, the number of samples used for training the model was very high, which enabled the model to capture the interrelations among the features and avoid overfitting. ML practitioners do not always consider regularizing models when the model performance is the same on testing and training sets. In our case, the $R2$ metric of the cross-validation set is 0.95 and 0.94 for the testing set.
- The Elastic net model does not indicate more advance than the Ridge and Lasso models. The best hyperparameters for the Elastic net model were: $\alpha = 0$, $\lambda = 0.001$. It means the mixing parameter was 0 and the Elastic net is the same as the Ridge model.

On the other hand, the achieved results for the regression models run for all the houses (the right section of Table 6) ensure that the $R2$ metric is better as forecasting load improves when the number of households increases. Nonetheless, other metrics (*i.e.*, *MAE*,*MSE*, and *RMSE*) were higher than the regression model for only one housing block because the target variable (*i.e.,* energy sum) for all houses model have much higher values than for the model of only one block. As mentioned before in Table 2, the target variables for other published studies in the literature differ from

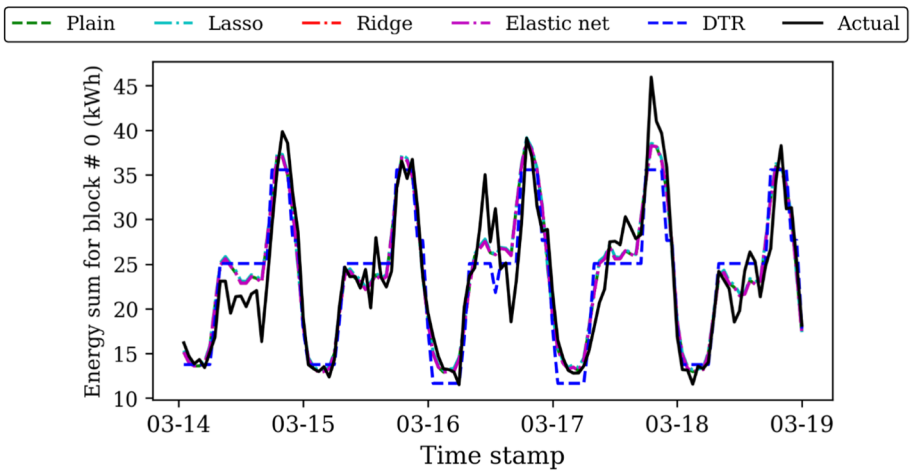

Fig. 14 Actual and predicted load profiles for the sum of energy of Block # 0

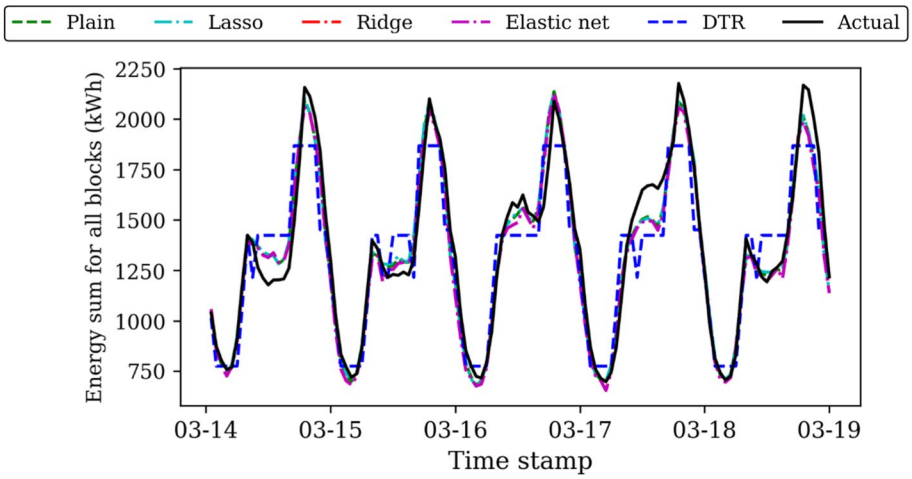

Fig. 15 Actual and predicted load profiles for the sum of energy of all blocks

the target variable applied in this work. Hence, comparing the performances of ML models used in this work and the works mentioned in Table 2 is not applicable.

During the interval from 14th March to 19th March 2013, the actual and predicted load profiles for forecasting the sum of energy of settlement Block # 0 are revealed in Fig. 14, while the sum of energy of all blocks is displayed in Fig. 15. As shown in the figures, DTR is less flexible than other models. The actual load profile for Block # 0 is more dispersed than the load profile for all the households. The noticed notches on the 16th and 17th of March are because a CM event was issued on those days. Also, the SF (supply following) event was applied on 14th March with a high tariff between 5:00 a.m. and 7:30 a.m. Through other days (*i.e.*, 15th and 18th March), the prediction was mostly precise.

DTR had low performance regarding its accuracy as the maximum depth of the tree was not allowed to increase above 5 to limit the computational efforts and have better insights. Tuning the depth of the tree by cross-validation should give more accurate

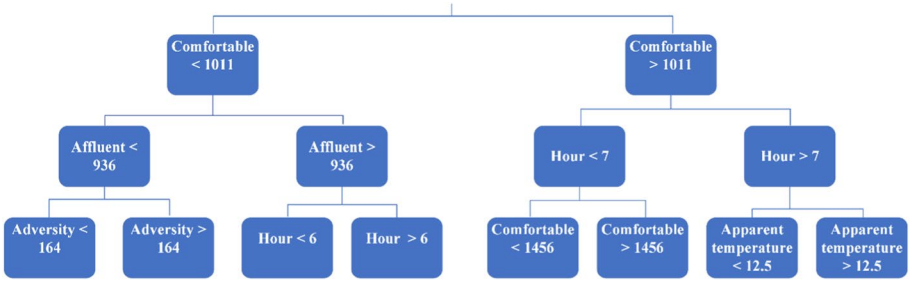

Fig. 16 DTR for load prediction of all houses

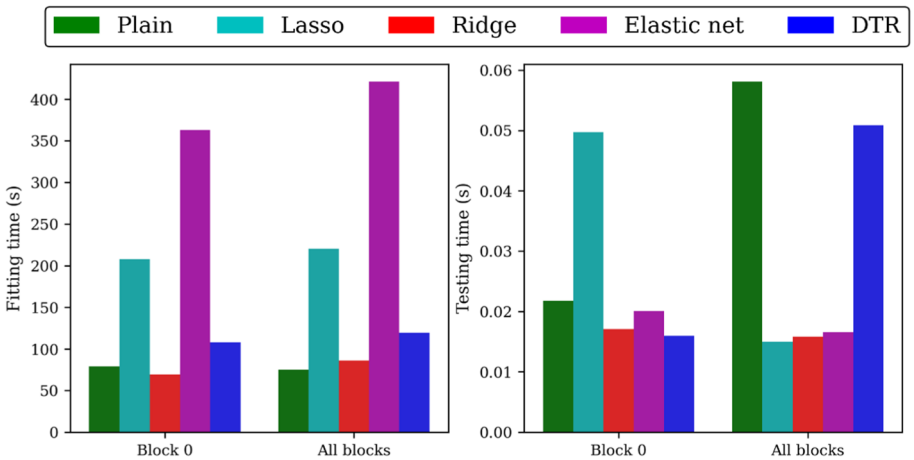

Fig. 17 Fitting and testing times for all the models run for only Block # 0 and all blocks

results. For example, $R2$ metric would reach 0.982 if the tree depth becomes 12 for the model predicting the sum of load consumption of all houses. To get more insights, the complete DTR can be plotted by the D3.js library. As the original tree is so large and complex, the top nodes of the tree for LF of all houses are shown in Fig. 16. The plotted tree provides insights into how the model estimated the target and which are the most important features.

Computational efforts for different tested models

In BD applications, computational efforts are considered because these efforts increase exponentially as data size increases. The fitting and testing times of the models run on Spark MLlib are presented in Fig. 17. The exact time values differ according to the platform capability, but the order of the models is still the same. As displayed, Elastic net model has the highest fitting times as the hyperparameter tuning tries all the possible combination values of regularization parameter λ and mixing parameter α. Other regularization models test only the possible values of regularization parameter λ. However, testing times for regularized models were the lowest. The testing time parameter is more important than fitting time as the model is fitted once but used many times for predictions. The fitting times were higher for Lasso and Elastic net models than for the Ridge

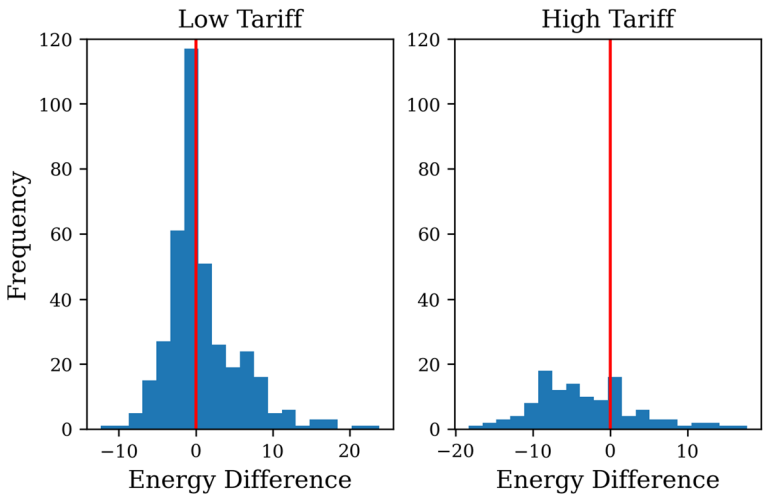

Fig. 18 A histogram of energy difference for low and high tariffs

model. Thus, regularized models are preferred as they deliver faster results than plain non-regularized and DTR models.

As revealed, the prediction results are accurate and interpretable even though we applied standalone models and avoided using compound models (like fuzzy and clustering models) to enhance the model's accuracy. Using simplified models is crucial in the BD context to reduce computational power. Reducing the number of features fed to the ML models made the models scalable and faster.

Results of customer response assessment

Summarized information about CM events conducted in the LCL project is offered in [23]. The CM events on settlement blocks # 2, 4, 6, 8, 46, and 48 are evaluated. These events were applied to households enrolled in the DToU program, and then the assessment was done for these houses only. For these designated blocks, the total number of households was 300, while only 96 were enrolled in the DToU program. The evaluation was achieved according to "Assessing Customer Responsiveness to DToU Tariffs" section, and the results are presented in this section. The predictions were made using the Ridge model based on the recommendation in "Computational efforts for different tested models" section that regularized models have better performance regarding prediction computational effort.

Through all CM events, the number of occurrences of low tariffs was 383 times and 120 times for high tariffs. Figure 18 displays the histogram for energy consumption differences between actual and prediction values. It is noticed that customers were more interactive with high tariffs as the histogram is left-skewed. Conversely, customer response was worse for low tariffs.

Figure 19 displays the actual and predicted energy consumptions during six different examined CM events, as examples. The events were chosen to cover different months throughout the year. The selected events are P4_3D_0 in March, P1_3D_0 in May, P9_2D_0 in January, P4_2D_0 in February, P9_2D_1 in March, and lastly P1_2D_0 in November, as discussed in [23]. From the graphs of Fig. 19 (a to f), it is noticed that:

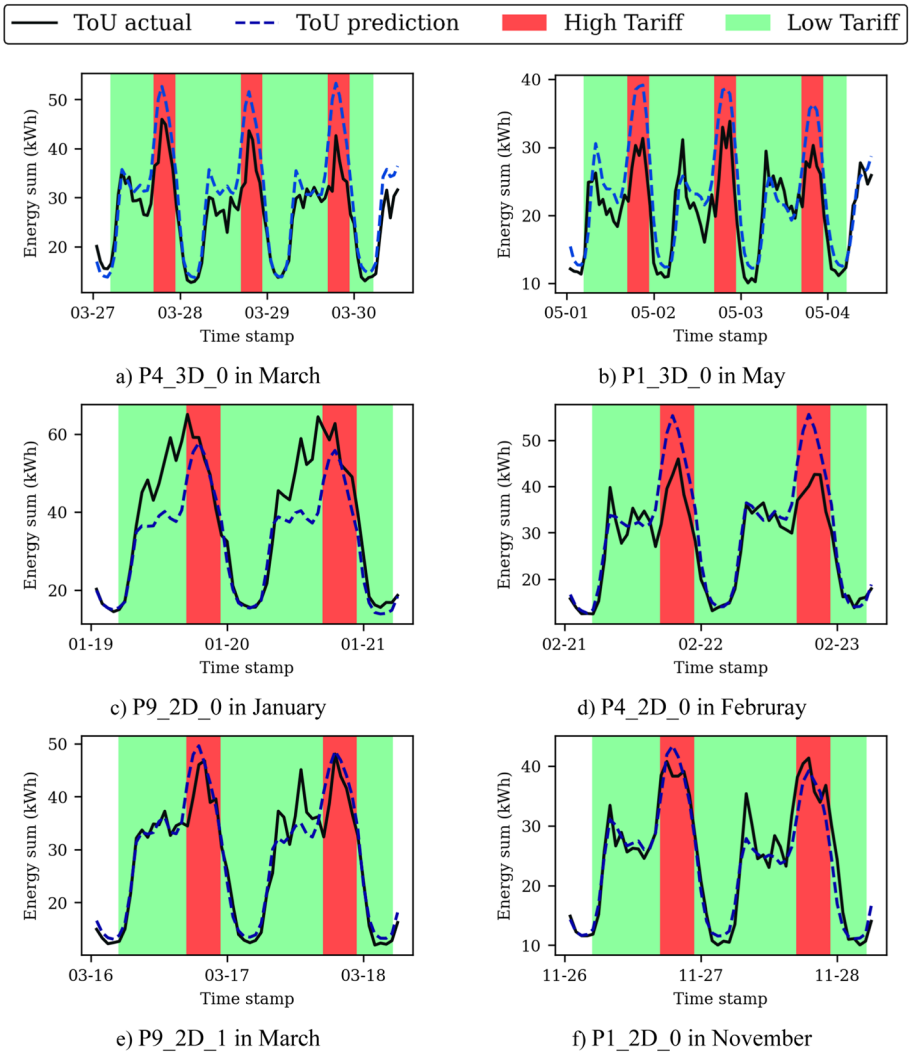

Fig. 19 Six examples of CM events for blocks # 2, 4, 6, 8, 46, and 48

- High tariffs were applied in peak hours from 17:00 to 23:00 in the six examined events. Accordingly, a considerable reduction in actual consumption is achieved in the three events P4_3D_0, P1_3D_0, and P4_2D_0 (as revealed in Fig. 19a, b, d). On the contrary, no decrease in energy consumption is attained in the other three examined cases P9_2D_0, P9_2D_1, and P1_2D_0 (as revealed in Fig. 19c, e, f).
- During low tariffs, and as illustrated in Fig. 19c (for P9_2D_0), there was a good response from customers on the 19th and 20th of January, which were Saturday and Sunday. However, the correlation between the good response to low tariffs during weekends should be investigated more by applying more CM events at weekends. The customer responses on 2nd May, 3rd May (Fig. 19b), 17th March (Fig. 19e), and 27th November (Fig. 19f) were also good but less compared with the responses on 19th and 20th of January. Finally, it was found that the customers did not respond to low tariffs applied in the last night hours (i.*e*., after 00:00) in any of the examined events.

Limitations and challenges

One main limitation of the applied ML models in this study is that their forecasting accuracy increases as the number of houses increases. Thus, these models are not suitable to forecast the energy consumption for only one house or a small group of houses. On the other hand, another limitation arises when the data size increases beyond the limit of the cluster RAM as the used cluster should be scaled up. Besides, obtaining a huge dataset to be used as a case study is a key challenge because of the data privacy issues.

Regarding the available LCL project data, obtaining detailed information about the time of applying DToU events, the RES capacity, and grid capability limits was not possible. Such data was essential to make a complete analysis of the proposed framework and to enrich the study results.

Conclusions

The developing usage of smart meters and information technologies in SG creates a large amount of data. This study highlights the significance of deploying BD tools to support the operation of SG. Apache Spark, as an integrated data analytics engine, is used to perform descriptive and predictive analytics on 167 million records of smart meters data from the LCL project in addition to exogenous data (like weather and demographic data). Spark SQL descriptive analytics offer new insights that help DNO recognize consumption behavior (such as consumption patterns and trends). Descriptive analytics displayed how residential energy consumption relies highly on weather and demographic conditions.

Spark MLlib was used to train MPR and DTR models for the LF task because these models were proven in the literature to be precise and self-interpretable. As a result of reducing the number of features based on the exploratory analysis (*i.e.,* four selected features for one block model and six features for the model of all blocks), ML models become more scalable, more computationally efficient, and less prone to overfitting. The $R2$ metric of fitted models exceeded 0.96 to predict the sum of energy consumption for all houses. So, this study recommends the regularized models, especially the Ridge model, as it has the least fitting and testing times concerning all tested cases.

Assessing the customer response to CM events highlighted the weak response to the DToU program, especially the low tariffs. Customer response to SF events is supposed to be weaker than the reaction to CM events which are designed to accomplish better customer response.

The results ensure that BD analytics can provide insights into load consumption patterns, trends, and predictions. Predictions are vital for planning, initiating DToU programs, and evaluating customer response to such programs. Also, these predictions can be utilized in a framework that facilitates RES integration. Accordingly, all these measures will make the grid operation more secure, efficient, economical, sustainable, and automated as the framework will generate the tariffs automatically.

Future work may extend to cover more analyses to catch the reason for weak responses to DToU programs and explore other solutions that can control consumption behavior for customers. For attaining a comprehensive grid simulation to

investigate the framework proposed in this study, actual data is required for a particular grid (RES production, grid capability limits, and load consumption). The main challenge of conducting such work is data availability because of the privacy and security issues related to making the data available for public use.

Abbreviations

ANN	:Artificial neural network
API	:Application programming interface
ARIMA	:Auto-regressive integrated moving average
ARMA	:Auto regressive moving average
BD	:Big data
CM	:Constraint management
DNO	:Distribution network operator
DR	:Demand response
DToU	:Dynamic time of use
DTR	:Decision tree regression
ETL	:Extract transform load
GCP	:Google cloud platform
HDFS	:Hadoop distributed file system
LCL	:Low carbon London
LCLid	:Identification number of a house in the Low Carbon London project
LF	:Load forecasting
LR	:Linear regression
ML	:Machine learning
MPR	:Multivariate polynomial regression
NARX	:Nonlinear autoregressive exogenous model
RDD	:Resilient distributed dataset
RES	:Renewable energy sources
SF	:Supply following
SG	:Smart grid

Acknowledgements
Not applicable.

Author contributions
HAE performed the study, developed the models, performed the analysis, and drafted the manuscript. MHA worked with HAE to develop the article framework and interpret the results. DKI is a leader of the research, conducted the research, and approved the results. All authors read and approved the final manuscript.

Funding
Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Availability of data and materials
The dataset used and analyzed during the current study for the Low Carbon London Project is open for reuse on the website of UK Power Networks [24].

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 26 August 2023 Accepted: 7 April 2024
Published online: 28 April 2024

References

1. Guerrero-Prado JS, Alfonso-Morales W, Caicedo-Bravo E, Zayas-Pérez B, Espinosa-Reza A. The power of big data and data analytics for AMI data: a case study. Sensors. 2020;11:3289.
2. Zhou K, Fu C, Yang S. Big data driven smart energy management: From big data to big insights. Renew Sustain Energy Rev. 2016;56:215–25.

3. Abdelaziz AY, Biswal M, Dewangan F. Load forecasting models in smart grid using smart meter information: a review. Energies. 2023;16(3):1404.
4. Wilcox T, Jin N, Flach P, Thumim J. A big data platform for smart meter data analytics. Comput Ind. 2019;105:250–9.
5. Mokhade A, Fund N, Bokde ND, Shewale A. An overview of demand response in smart grid and optimization techniques for efficient residential appliance scheduling problem. Energies. 2020;13:4266.
6. Bartschat A, Ludwig N, Braun E, Waczowicz S, Renkamp N, Peter N, Düpmeier CD, Mikut R, Hagenmeyer VH, Ordiano JÁG. Concept and benchmark results for Big Data energy forecasting based on Apache Spark. J Big Data. 2018;5:11.
7. Zhou K, Yang S. Understanding household energy consumption behavior: the contribution of energy big data analytics. Renew Sustain Energy Rev. 2016;56:810–9.
8. El Hannani A, Aqqal A, Haidine A, Dahbi A, Daki H. Big data management in smart grid: concepts, requirements, and implementation. J Big Data. 2017;4:13.
9. Syed D, Zainab A, Ghrayeb A, Refaat SS, Abu-Rub H, Bouhali O. Smart grid big data analytics: survey of technologies, techniques, and applications. IEEE Access. 2020;9:59564–85.
10. Martinez-Pabon M, Eveleigh T, Tanju B. Smart meter data analytics for optimal customer selection in demand response programs. Energy Procedia. 2017;107:49–59.
11. Vakili VT, Bahrak B, Ansari MH. Evaluation of big data frameworks for analysis of smart grids. J Big Data. 2019;6:109.
12. Moharm K. State of the art in big data applications in microgrid: a review. Adv Eng Inf. 2019;42:100945.
13. R. J. Bessa, Chapter 10—future trends for big data application in power systems, In: Y. Z. Reza Arghandeh, Eds., Big data application in power systems, 2018, pp. 223–242.
14. Kwac J, Rajagopal R, Demand response targeting using big data analytics. In: IEEE International Conference on Big Data, Silicon Valley, CA, USA, 2013.
15. Vajjala VAH, A novel solution to use Big Data technologies and improve demand response program in aggregated residential houses, In: IEEE Conference on Technologies for Sustainability (SusTech), Phoenix, AZ, 2016.
16. Ghorbanian M, Dolatabadi SH, Siano P. Big data issues in smart grids: a survey. IEEE Syst J. 2019;13:4158–68.
17. Safhi HM, Frikh B, Ouhbi B. Energy load forecasting in big data context, In: 5th International Conference on Renewable Energies for Developing Countries (REDEC), 2020, Ben Guerir &Marrakech, Morocco.
18. Saber AY, Alam AKMR. Short-term load forecasting using multiple linear regression for big data. In: IEEE Symposium Series on Computational Intelligence, Honolulu, Hawaii, USA, pp. 1-6, 2017,
19. Lei J, Jin T, Hao J, Li F. Short-term load forecasting with clustering–regression model in distributed cluster. Clust Comput. 2019;22:10163–73.
20. Mamun AA, Sohel M, Mohammad N, Sunny MSH, Dipta DR. A comprehensive review of the load forecasting techniques using single and hybrid predictive model. IEEE Access. 2020;8:134911–39.
21. Oprea S-V, Bâra A. Machine learning algorithms for short-term load forecast in residential buildings using smart meters, sensors and big data solutions. IEEE Access. 2019;7:177874–89.
22. Syed D, Refaat SS, Abu-Rub H. Performance evaluation of distributed machine learning for load forecasting in smart grids. In: 2020 Cybernetics & Informatics (K&I), Velke Karlovice, Czech Republic, 2020,
23. Schofield J, Carmichael R, Tindemans S, Woolf M, Bilton M, Strbac G. Residential consumer responsiveness to time-varying pricing. London: Imperial College London; 2014.
24. UK Power Networks, Low Carbon London. UK Power Networks, 2015. [Online]. Available: https://libguides.sccsc.edu/findcitationinfo/websites. Accessed Jan 2023.
25. Chen R, Lai CS, Zhong C, Pan K, Ng WW, Li Z. MultiCycleNet: multiple cycles self-boosted neural network for short-term electric household load forecasting. Sustain Cities Soc. 2022;76:103484.
26. A. C. Müller and S. Guido, Introduction to machine learning with Python: a guide for data scientists, O'Reilly Media, Inc., 2016.
27. Al-Jarrah OY, Yoo PD, Muhaidat S, Karagiannidis GK, Taha K. Efficient machine learning for big data: a review. Big Data Res. 2015;2(3):87–93.
28. Friedman JH, Tibshirani R, Hastie T. The elements of statistical learning: data mining, inference, and prediction. New York: Springer; 2009.
29. Apache Spark Foundation, "MLlib - Decision Tree," [Online]. https://spark.apache.org/docs/1.1.0/mllib-decision-tree.html. Accessed Jan 2023.
30. EmilioCarrizosa CMADRM. Mathematical optimization in classification and regression trees. TOP. 2021;29:5–33.
31. Apache Spark Foundation, "Extracting, transforming and selecting features," [Online]. https://spark.apache.org/docs/latest/ml-features. Accessed Jan 2023.
32. Apache Spark Foundation, "ML Tuning: model selection and hyperparameter tuning," [Online]. https://spark.apache.org/docs/latest/ml-tuning.html. Accessed Jan 2023.

Publisher's Note