IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 25, 2021, accepted December 19, 2021, date of publication December 21, 2021,
date of current version December 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3137352

Big Data Processing for Commercial Buildings and Assessing Flexibility in the Context of Citizen Energy Communities

SIMONA VASILICA OPREA[1], ADELA BÂRA[1], VLAD DIACONITA[1], (Member, IEEE), CĂTĂLIN CEAPARU[1], AND ANCA ALEXANDRA DUCMAN[2]

[1]Department of Economic Informatics and Cybernetics, Bucharest University of Economic Studies, 010374 Bucharest, Romania
[2]Department of Agrifood & Environmental Economics, Bucharest University of Economic Studies, 010374 Bucharest, Romania

Corresponding author: Vlad Diaconita (vlad_diac@ieee.org)

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI -UEFISCDI, project number 462PED/28.10.2020, project code PN-III-P2-2.1-PED-2019-1198, within PNCDI III.

ABSTRACT In this paper, we propose a cloud-based big data processing approach to evaluate the flexibility potential of commercial buildings by type and benefits for the owners. The pandemic times changed electricity consumption patterns with a substantial impact on energy markets. Many activities moved from large commercial offices and schools to residential buildings. With machine learning algorithms, the flexibility forecast can be improved to help energy suppliers, grid operators, and traders better calculate the flexibility potential of commercial buildings. With better forecasts, grid operators can identify and mitigate risks, prevent malfunctions, and schedule maintenance works in advance. Using flexibility forecast as input and results from previous studies regarding flexibility coefficient by state and demand response programs, we propose an original method to assess load flexibility of commercial buildings and calculate the benefits for their owner. The exemplification is done with an extensive hourly dataset from the U.S.A. of 14,976 comma-separated values files with a total of 131.18 million records showcasing the electricity and gas consumptions and their breakdown for one year.

INDEX TERMS Electricity, load management, power generation economics, clouds, databases, environmental economics.

I. INTRODUCTION

In July 2019, the European Union (EU) introduced Citizen Energy Communities (CECs), including residential consumers, prosumers, and local entities such as distributed energy resources, storage facilities, and industrial and commercial buildings. Such communities generate large volumes of smart meter data that can be analyzed to extract useful insights related to flexibility potential and assess the benefits and Enabling Technology Costs (ETCs) for Demand Response (DR) programs. Load flexibility helps CECs to handle the fluctuations and high volatility of load, wind speed, and solar radiation. Smart metering data has multiple applications such as billing, cluster identification, tariff setting, load forecast, optimization, market simulation via blockchain, and flexibility assessment especially when data is provided at the appliance or group of appliances level. While some studies discuss the applications of smart meter data [1] and load forecasting based on Big Data [2], many challenges [3] still need to be addressed as the large volume of data does not provide directly useful insights and hints regarding future trends. Thus, the EU policy envisions an energy transition that allows prosumers and CECs to share, trade, aggregate, and sell the electricity surplus [4], [5] and even own and manage the grid. These activities are accompanied by large volumes of data that can offer useful feedback for energy suppliers, grid operators, traders, consumers, prosumers, and aggregators [6]. DR programs target to extract and reward flexibilities and use them to manage the variation output of Renewable Energy Sources (RES) and the load of the Electric Vehicle (EV) charging stations [7]. Flexibilities are defined in terms of type, size (quantity), duration, control technology considering the specific sector [8]. For instance, [9] defined residential, commercial, and industrial

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

sectors enabling technologies for DR. Furthermore, they define DR service types (shift, shed, and shimmy). Thus, commercial buildings data can be grouped types, Independent System Operator (ISO) affiliation, correlated with several DR services, flexibility coefficient by state, and ETCs to assess the flexibility potential and benefits [10]. Furthermore, COVID-19 pandemic times influenced the load pattern, shifting the load from commercial buildings such as large offices and schools to residential consumption [11]–[14], so improved data-driven architecture for buildings data exchange is needed [15]. Big data technologies and IoT are frequently used in consumer-oriented energy optimization and prediction, including the day-ahead forecast for buildings [16] or total energy consumption [17].

This paper stems from the research underlined in [18] that focused simple data analytics performed with large volumes of data. It also takes into account our previous research in terms of data models for flexibility and DR assessment [19]. Comparing with the method proposed in [19], we identify and emphasize the similarities and differences described in section II. Flexibilities and Direct Load Control (DLC) are also studied in [20]. The novelty of the current study consists in:

- data centers (DCs) are usually the primary choice for storing large quantities of data. Some studies considered DCs as computing facilities of interest for exploiting their energy flexibility and have proposed an Energy Marketplace to allow DCs to act as active energy players integrated into the smart grid [21]. But the study mentioned above and other papers on energy flexibility [22], [23] lack comprehensive analyses of data lake solutions as powerful backends for running DR assessment. To address this gap, we compared two data lake architectures (Hadoop based vs AWS), and different approaches within these architectures as far as costs and speed are concerned. The cost-wise and speed-wise comparisons are conducted to identify the pluses and minuses of the two solutions to store and prepare the data (e.g., to perform data reduction) for running the algorithms;
- a different approach to DR assessment, considering not only shift DR service but also the combination shed & shift. Furthermore, when estimating the DR, we updated the analyses by rethinking the implementation of DR programs. The analyses take into account the results of previous studies [9], [10]. It includes the flexibility forecast performed with Machine Learning - LSTM (Long Short-Term Memory) recurrent neural networks and Regression Analysis aiming to determine future consumption values to evaluate the flexible potential better and estimate the efficiency of DR programs.
- flexibility assessment method that relies on a step-by-step approach consisting in a) Dividing appliances into programmable and non-programmable to separate flexible and fixed consumption; b) Calculate total forecasted consumption of programmable appliances at hour h; c) Calculate daily mean consumption using the load profile; d) Extract peak hours for the analyzed interval: one month, year, etc. e) Identify the start and stop peak hour; f) Apply flexibility coefficient to obtain the shiftable consumption; g) Obtain total consumption to shift from peak to off-peak hours; h) Calculate the gain or benefit that can be obtained by shifting and shifting/shedding programmable appliances; i) Compare the gains and choose the DR program.

The large-scale rollout of smart metering systems that takes place in most European countries generates numerous datasets. Therefore, we propose to extract meaningful insights from this data and raise the awareness of the consumers regarding DR programs potential to bring savings and increase a pro-environmental behavior by assessing the flexibilities in terms of quantities and monetary benefits. The input data comes from numerous smart meters in CSV files grouped by state and building type. Our objective is to store it in S3 (AWS) and HDFS, and reduce it with Athena and Sagemaker or Hive, to be further processed in Python obtaining the flexibility forecast. Then, the flexibility potential is assessed by implementing the method proposed in section II. The input data flow and processes proposed in this paper are graphically described in Figure 1.

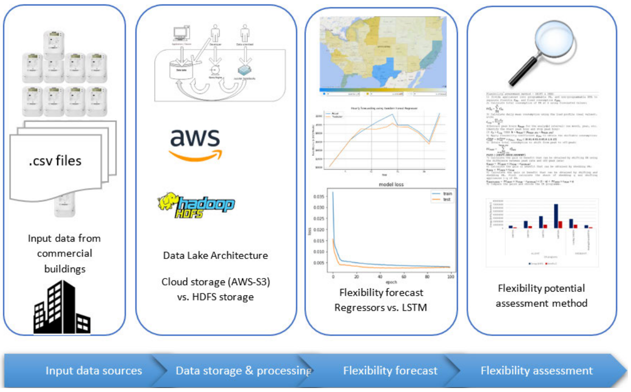

.csv files
Input data from commercial buildings
aws
hadoop HDFS
Data Lake Architecture
Cloud storage (AWS-S3) vs. HDFS storage
model loss
train
test
Flexibility forecast Regressors vs. LSTM
Flexibility potential assessment method
Input data sources
Data storage & processing
Flexibility forecast
Flexibility assessment

FIGURE 1. **Input data flow, analyses, and processes.**

The paper is structured as follows: In Section II, we start with the Big Data processing approach, describing and analyzing the dataset, continuing with comparing the data lake architectures, proposing a flexibility assessment method, with the final subsection comparing two forecast methods; In section III, the simulations using the forecast methods on the dataset are conducted and the results of flexibility method implementation are presented; In section IV, the conclusions are drawn.

II. BIG DATA PROCESSING APPROACH

A. CLOUD VS. HADOOP DISTRIBUTED FILE SYSTEM (HDFS) STORAGE

To store and process the data needed for analyses, we needed a solution that can reliably offer access to various services to multiple types of users, as shown in Figure 2.

Different applications (including IoT devices) can usually interact directly with the data lake by writing semi-structured data. Application developers access the data lake directly or through query engines that offer different views on the data (e.g., by providing SQL functionality). Data scientists prefer Jupyter notebooks [24] to interact with the data through various libraries, using SQL-enabled query engines, data dictionaries, or reading the files directly (e.g., in Pandas DataFrames). Some data scientists prefer the flexibility and standardization of SQL to query their data. Consequently, some schema can be applied (on-read/on-write) to the data, which might lead to the loss of certain nuances of the raw data. Because of this loss of nuances, it is often advisable to work with the initial raw data.

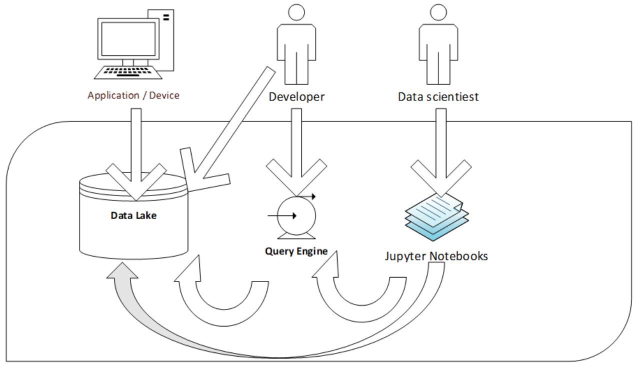


FIGURE 2. The interactions between the actors and the applications.

Standard databases usually employ a schema-on-write approach. That means that the schema of the data is defined before the data is stored (i.e., CREATE TABLE before INSERT rows). This is more rigid but works well for dense and structured data and where constraints or indexes are necessary. Big Data oriented solutions enable a more flexible schema-on-read approach where semi-structured or unstructured data is stored as it is and the schema is applied (i.e., the data is parsed) later when the data is being consumed. Sometimes, after the schema is applied and the data analyzed, the resulting output is loaded into a schema-enabled database for future use. By combining all these types of data to store, govern and process at scale, we consider a data lake [25]. Many times, such lakes are supplied with data from ingesting real-time flows. Data lakes are being used in many domains, including smart grids [26], and can provide data scientists a plethora of inputs to train better models. Of course, as always, having access to lots of data doesn't tackle by itself the bad or inaccurate data issues [27].

Our dataset is, as previously discussed, is made of multiple CSV files. Useful data comes from the payload of the file, but also from the name of the file and of the folder. The file and folder names contain useful information such as country, state, city, location type, and date (the payload contains the day, month, and hour and the filename, the year). Columnar storage formats such as Parquet [28] or ORC (Optimized Row Column) can greatly decrease query time and costs [29], especially for aggregations (AVG, COUNT, SUM), as some cloud services such as Amazon Athena or Spectrum charge by the queried data size. Therefore, if the CSVs are scanned multiple times, it makes sense to convert them into Parquet.

In this section, we identified and compared two data lake solutions, as shown in Table 1 to see how they work as a backbone to handle the needs of our dataset aggregations and summarizations.

TABLE 1. Data lake architectures.

No	Primary data storage	Schema on read solution	Schema on write solution (real-time processing)	Query Engine
1	HDFS	Hive	HBase	Apache Phoenix
2	S3	Athena + Glue	Redshift or Aurora	Spectrum

The first architecture is comprised of open source which can be used free of charge. HDFS is the Distributed File System of Hadoop that can reliably store files on a cluster of commodity computers (write-once-read-many access). Every file block is stored on more than one node (usually on 3). If one copy is compromised, one of the copies is replicated someplace else. Hive is a data warehouse solution that can apply a schema on HDFS stored flat files. The Hive metadata (e.g., the table definitions) can be stored in the Derby embedded store, a local data store (e.g., MySQL, MS SQL, Oracle), or on a remote metastore (i.e., the metastore is on a distinct JVM). When an embedded store is used concurrent access is not possible because only one Hive session can be opened at once. This problem is mitigated by using the local data store or the remote one. To construct a table based on a folder containing files from our dataset we use a DDL (Data Definition Language) statement:

CREATE EXTERNAL TABLE IF NOT EXISTS csv_dataset (Date_time string, Electricity double, Fans double, Cooling double, HeatingElectricity double, InteriorLights double, InteriorEquipmentElectricity double, Gas double, HeatingGas double, InteriorEquipmentGas double, WaterHeater double)

ROW FORMAT DELIMITED
FIELDS TERMINATED BY ‘,’
LINES TERMINATED BY ‘\n′
STORED AS TEXTFILE
LOCATION ‘/user/root/energy_csv’
tblproperties (“skip.header.line.count” = “1”);

If we need more flexibility, we can define a single column (e.g., Record string) instead of eleven columns and do the parsing through the SELECT statement. Based on this table we evaluate several relevant SQL expressions on a subset of 1.46 GB of the full dataset and a subset almost double in size, 2.9GB (Table 2). The two subsets were chosen to evaluate the increase in query time as the data size increases. The information from the filename is extracted using the

TABLE 2. Full table scans and data reduction for a subset of the full dataset (HDFS + Hive).

<table>
<tr><th>Statement</th><th>1.46 GB Run Time (sec)</th><th>2.9 GB (+98%) Run Time (sec)</th></tr>
<tr><td>CREATE VIEW v_csv_dataset as
SELECT *,
SPLIT(INPUT__FILE__NAME,'/')[6]||'_'||split(INPUT__FILE__NAME,'/')[7] ID,
REGEXP_EXTRACT(SPLIT(INPUT__FILE__NAME,'/')[6],'(?<=_).+?(?=_)',0) state,
regexp_extract(split(split(INPUT__FILE__NAME,'/')[6],'_')[2],'^.[a-zA-Z0-9-]*',0) location,
REGEXP_EXTRACT(split(INPUT__FILE__NAME,'/')[7],'^.[a-zA-Z0-9-]*',0) building
FROM csv_dataset;
SELECT * FROM v_csv_dataset;
(HiveQL1)</td><td>192</td><td>270</td></tr>
<tr><td>CREATE TABLE energy_orc(date_time string, electricity double, fans double, cooling double, heatingelectricity double,
interiorlights double, interiorequipmentelectricity double,
gas   double, heatinggas double, interiorequipmentgas double, waterheater double, id string,
state string, location string, building string, year int) STORED AS ORC;
INSERT INTO TABLE energy_orc select *,
split(INPUT__FILE__NAME,'/')[6]||'_'||split(INPUT__FILE__NAME,'/')[7] ID,
regexp_extract(split(INPUT__FILE__NAME,'/')[6],'(?<=_).+?(?=_)',0) state,
regexp_extract(split(split(INPUT__FILE__NAME,'/')[6],'_')[2],'^.[a-zA-Z0-9-]*',0) location,
regexp_extract(split(INPUT__FILE__NAME,'/')[7],'^.[a-zA-Z0-9-]*',0) building,
substr(regexp_extract(split(INPUT__FILE__NAME,'/')[7],'^.[a-zA-Z0-9-]*',0),-4) year
from csv_dataset;
(HiveQL2)</td><td>200</td><td>255</td></tr>
<tr><td>SELECT * FROM energy_orc;
(HiveQL3)</td><td>14.1</td><td>15.74</td></tr>
<tr><td>SELECT state, location,cast(sum(electricity) as integer) electricity,cast(sum(fans) as integer) fans FROM v_csv_dataset
GROUP BY state,location;
(HiveQL4)</td><td>107</td><td>130</td></tr>
<tr><td>SELECT state,location,cast(sum(electricity) as integer) electricity,cast(sum(fans) as integer) fans FROM v_csv_dataset
GROUP BY state,location LIMIT 10;
(HiveQL5)</td><td>81</td><td>110</td></tr>
<tr><td>SELECT state,location,cast(sum(electricity) as integer) electricity,cast(sum(fans) as integer) fans FROM energy_orc
GROUP BY state,location;
(HiveQL6)</td><td>7.3</td><td>8.53</td></tr>
<tr><td>SELECT state,location,cast(sum(electricity) as integer) electricity,cast(sum(fans) as integer) fans FROM energy_orc
GROUP BY state,location LIMIT 10;
(HiveQL7)</td><td>5.8</td><td>6.9</td></tr>
</table>

INPUT__FILE__NAME pseudo column and regex expressions as shown in HiveQL1 from Table 2. Hive offers ORC (Optimized Row Columnar), a file format somewhat similar to Parquet. In HiveQL2, we converted from the external table, built on the CSV files, to an ORC table. After the conversion to ORC, the data size went down from 1.46 GB to 0.385 GB and from 2.9GB to 0.772 GB.

A full data scan on the CSV data took 192 seconds (HiveQL1) while a similar one on the ORC table took about 14 seconds (HiveQL3). The aggregating functions also performed considerably better on the ORC table compared to the CSV one (HiveQL4 vs. HiveQL6, HiveQL5 vs. HiveQL7).

Even though converting to ORC will make the queries run faster, still, even with an engine like Tez which is much faster than legacy MapReduce, they run rather slow for real-time requirements where such queries usually need to run in under 1 second. To address this problem, the data needed for real-time queries can be exported to HBase, a NoSQL database that stores data into HDFS and manages to provide random access functionalities. Even though HBase has its own language, SQL capabilities can be added using Apache Phoenix. DML (Data Manipulation Language) statements can be used on HBase tables or even for joining SQL tables with Hive tables.

The second stack from Table 1 is comprised of Amazon Web Services cloud solutions. Even though similar solutions exist from other Cloud providers, AWS tends to be the standard for heavy lifting machine learning engineering. For comparison, an Azure Studio-centered solution from Microsoft is more suited for business users who want to train a model in a drag-and-drop manner without using much, if any, code. Cloud solutions are far from free but benefit from the advantages of running on a mature platform offering various services. Using the cloud for data science projects enables easy transitioning from testing and prototyping to production on top of data durability guarantees. It can also help accelerate the training phase (easy access to GPUs, to multiple machines for horizontal scaling) and handle almost unlimited concurrent connections. S3 is the object storage service from AWS. While you can’t run an operating system or a database management system on S3 (you would need EBS – Elastic Block Storage), it is the standard solution for hosting a data lake. It offers multiple tiers each with its advantages, disadvantages, and pricing. Storing 1TB will roughly cost $23/month on S3 classic, $12.5 on S3 Infrequently Accessed, $10 on S3 IA One Zone, $4 on Glacier, and $0.99 on Glacier Deep Archive (long retrieval time, up to 12h). There are also costs related to data access (for PUT, COPY, POST, LIST, GET, SELECT – $0.005/1000 requests), monitoring (for Intelligent Tier), data transfer, transfer acceleration, or cross-region replication. With Amazon Athena, we run SQL queries directly on the S3 data and uses AWS Glue Data Catalog to store the metadata (table definitions). It is interesting to note that Amazon Athena uses Apache Hive DDL to define tables.[1] So, it doesn't require any data movement, like EXTERNAL TABLES in Hive. For Athena, the LOCATION changes from an HDFS folder to an S3 folder in a bucket. The payment is made by the amount of scanned data per query.

Redshift is the main AWS OLAP solution that easily integrates with S3 and Glue Data Catalog whereas Aurora is the AWS-built OLTP solution for the cloud. Both use SQL natively. Still, Redshift Spectrum can spread queries across multiple stores such as Redshift and S3 similar to how Phoenix can query across HDFS, Hive, and HBase. AWS offers Sagemaker, a Jupyter environment that makes available the libraries and the infrastructure to make use of the data lake. For example, to run a 5 note Spark cluster from a Sagemaker Notebook (e.g., to assess data quality) we use:

```
p = PySparkProcessor(base_job_name = 'summarize_data',
  role = role,
  framework_version = '2.4',
  instance_count = 5,
  instance_type = 'ml.m5.xlarge',
  max_runtime_in_seconds = 500).
```

To evaluate the AWS solution, we loaded the CSV files in S3, including the folders. Then, we defined the SCHEMA on top of those files (observation: the SQL statements can be run from Sagemaker or Athena query editor):

```
CREATE EXTERNAL TABLE IF NOT EXISTS
energy.energy_csv(
Date_time string, Electricity double, Fans double,
Cooling double, HeatingElectricity double, InteriorLights double,
InteriorEquipmentElectricity double, Gas double, HeatingGas double,
InteriorEquipmentGas double, WaterHeater double)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' LOCATION 's3://sagemaker-eu-west-2-537954404244/energy'
TBLPROPERTIES ('skip.header.line.count' = '1').
```

The files can now be queried using standard SQL. To extract information from the file and folder names, we use split_part and regexp_extract on $PATH (similar to INPUT__FILE__NAME from HiveQL1), part of a view based on a query. The information can be retrieved by querying the view as shown in SQL1 from Table 3. As in HDFS, no pre-processing of the files is needed (e.g., concatenating files, adding columns, etc.) and when new files are added to the S3 folder, the view will see the new information.

Whereas Hive offers easy conversion between TEXTFILE tables and ORC, AWS offers, among others, easy conversion from Redshift, Aurora, or Athena to Parquet. This can be done using SQL statements as shown in SQL2 from Table 3. SQL2 moves and compresses the data from one S3 folder to another, creating subfolders for each distinct value of the partitioned_by clause (e.g., if there is data for 40 states, 40 subfolders will be created in /energy_parquet). As shown in SQL3, we can do a full query of energy_parquet and get the same result as in SQL1. This improves query time and lowers costs by reducing the data scanned. One problem here is that as CSV files are added or updated so has to be the parquet table (e.g., by using INSERT+SELECT) as it doesn't automatically see the new data.

The run times from the tables are for comparison reasons (speed increments when adding more data or switching from one format to another). They can easily vary by up to 10% depending on many factors, including the general AWS load. We notice that the SQL implementation of Athena is similar to the one from Hive (there are some differences in the functions and the accepted regex expressions). By studying Table 3, it is interesting to notice that the conversion from CSV to PARQUET was done in less than 15 seconds and the size of the dataset was reduced by a factor of 3.5 (due to compression). If new CSV files are added to the S3, the Parquet table can be easily updated by using a Lambda function [30] and a CloudWatch Events rule.

The results are similar to the ones when converting from CSV to ORC from Table 2 in respect to size reductions and improved query time. Even though the full scan from SQL3 took almost the same amount of time as SQL1, the amount of data scanned was much less, so the costs will be more than three times smaller (on Athena you pay by the amount of scanned data per query). The results suggest that it makes sense to convert to Parquet if you have to do more than one full table query on the CSV data. To further our research, we also conducted aggregation queries on the view constructed directly on CSV data and the parquet table (SQL4-SQL7).

We can observe that when using the CSV-based view, all the data gets scanned for the group by while when using the Parquet table, a lot less data is scanned thus lower costs. Using a limit clause to get only a subset of the rows has little impact on query time or the amount of scanned data for both cases. If we add to the SQL4-SQL7 queries a sum function on a different column (e.g., sum(cooling)), the amount of scanned data remains the same for the first 2 queries and increases by 55% for the latter two.

In this section, we have shown that the two architectures described in Table 1 are similar in many ways. The conversion from CSV to a columnar format brings important speed

[1] https://aws.amazon.com/athena/faqs/

TABLE 3. Full table scans for two subsets of the full dataset, one of 1.46 GB and another of 2.9 GB (S3+Athena).

<table>
<tr><th rowspan="2">Statement</th><th colspan="3">1.46 GB</th><th colspan="3">2.9 GB</th></tr>
<tr><th>Run Time (sec)</th><th>Data scanned</th><th>Cost/ 30 Queries ($)</th><th>Run Time (sec)</th><th>Data scanned</th><th>Cost/ 30 Queries ($)</th></tr>
<tr><td>CREATE VIEW energy_csv_v1 as
SELECT *,
split_part("$PATH",'/',5)||'_'||split_part("$PATH",'/',6) ID,
regexp_extract
(split_part("$PATH",'/',5),'(?<=_).+?(?=_)') state,
regexp_extract
(split_part("$PATH",'/',5),'(?<=_)([^_^.]*)(?=\.)') location,
regexp_extract
(split_part("$PATH",'/',6),'^.[a-zA-Z0-9-]*(?=_)') building
FROM energy_csv;
select * from energy_csv_v1;
(SQL1)</td><td>99</td><td>1.46 GB</td><td>0.21</td><td>183.56</td><td>2.9 GB</td><td>0.42</td></tr>
<tr><td>CREATE TABLE IF NOT EXISTS energy_parquet
WITH (format = 'PARQUET',
external_location = 's3://sagemaker-eu-west-2-537954404244/energy_parquet',
partitioned_by = ARRAY['state'])
AS SELECT *,
split_part("$PATH",'/',5)||'_'||
split_part("$PATH",'/',6) ID,
regexp_extract
(split_part("$PATH",'/',5),'(?<=_)([^_^.]*)(?=\.)') location,
regexp_extract
(split_part("$PATH",'/',6),'^.[a-zA-Z0-9-]*(?=_)') building,
SUBSTR
(regexp_extract(split_part("$PATH",'/',6),'^.[a-zA-Z0-9-]*(?=_)'),-4) year,
regexp_extract
(split_part("$PATH",'/',5),'(?<=_).+?(?=_)') state
FROM energy_csv;
(SQL2)</td><td>9.22</td><td>1.46 GB</td><td>0.21</td><td>14.91</td><td>2.9 GB</td><td>0.42</td></tr>
<tr><td>SELECT * FROM energy_parquet;
(SQL3)</td><td>91</td><td>410 MB</td><td>0.06</td><td>179.04</td><td>808.69</td><td>0.12</td></tr>
<tr><td>SELECT state, location,cast(sum(electricity) as integer) electricity,cast(sum(fans) as integer) fans FROM energy_csv_v1 GROUP BY state,location;
(SQL4)</td><td>1.75</td><td>1.46 GB</td><td>0.21</td><td>2.87</td><td>2.9 GB</td><td>0.42</td></tr>
<tr><td>SELECT state,location,cast(sum(electricity) as integer) electricity,cast(sum(fans) as integer) fans FROM energy_csv_v1 GROUP BY state,location LIMIT 10;
(SQL5)</td><td>1.57</td><td>1.46 GB</td><td>0.21</td><td>2.59</td><td>2.9 GB</td><td>0.42</td></tr>
<tr><td>SELECT state,location,cast(sum(electricity) as integer) electricity,cast(sum(fans) as integer) fans FROM energy_parquet GROUP BY state,location;
(SQL6)</td><td>1.19</td><td>122.31 MB</td><td>0.02</td><td>1.41</td><td>243.7 MB</td><td>0.04</td></tr>
<tr><td>SELECT state,location,cast(sum(electricity) as integer) electricity,cast(sum(fans) as integer) fans FROM energy_parquet GROUP BY state,location LIMIT 10;
(SQL7)</td><td>1.17</td><td>122.31 MB</td><td>0.02</td><td>1.95</td><td>243.7 MB</td><td>0.04</td></tr>
</table>

benefits, but the comparable queries will run considerably faster on AWS (see HiveQL6 vs. SQL6 and HiveQL7 vs. SQL7). We chose the AWS solution because of the speed benefits and better integration with Sagemaker for building the models discussed in the next section. On the other hand, the AWS solution incurs monthly costs which are reasonable for our data preparation needs, but will get higher as the volumes increase (e.g., running 30 queries/day in Athena, each query scanning 500 GB, the monthly cost would be $2299, not including the tier dependent S3 costs).

B. LOAD FLEXIBILITY FORECAST AND ASSESSMENT METHOD

The flexible energy consumption forecasting can play a very important role at a global level for DR program event creators, such as energy suppliers, grid and system operators, aggregators [31], [32]. The flexibility potential can be improved by performing the forecast for programmable appliances which allows suppliers and grid operators to make the right decisions at the right time. Thus, the prediction of future consumption allows a better assessment of the flexibility and efficiency of the DR programs.

There are many methods to predict consumption[33]–[35] and most of them depend on historical data. Historical data series can be used to create prediction patterns with the aim of predicting future consumption. Machine learning - Long Short-Term Memory (LSTM) is a type of recurrent neural network used in deep learning that can be used to forecast consumption. Another method by which future flexible consumption can be forecasted is regression analysis [36]. Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable and one or more independent variables. Both methods are suitable for time series forecasting. The linear regression algorithm attempts to minimize the sum of the squares of the differences between the observed value and the predicted one, whereas recurrent neural networks hold a hidden layer that acts as a memory function that considers previous time steps while calculating what the next value in the sequence should be. Unlike regression that deals especially with linear dependencies, recurrent neural networks can also cope with nonlinearities. Also, regarding the dataset, recurrent neural network, being a method of deep learning, requires a large set of data to train the model compared to the regression model that needs at least a little bit more data than the number of its parameters. In Table 4, the most important steps for building the forecast models are briefly described.

One flexibility assessment method is envisioned starting from the results of previous studies that identify the flexibility coefficient by state [37] and propose and fully describe the DR enabling costs and DR programs such as SHIFT, SHED, and SHIMMY to control appliances and flatten the load curve [9]. The most important aspect of this method is to update the load profile at least monthly because the load can suffer from seasonal changes and unexpected phenomena such as the pandemic leading to significant changes in consumption patterns from offices to residents.

The main steps of the proposed method are presented in Table 5. First, the appliances are classified into programmable (PA) and non-programmable (NPA) to separate the flexible consumption from fixed one. Then, the hourly total consumption of forecasted PA is summed up. Using a valid load profile, the average consumption is calculated to identify the start $h_{PEAK-STA}$ and stop peak hours $h_{PEAK-STO}$. By comparing the consumption of each hour with the average, one or more peak intervals will be extracted. Then, the flexibility coefficient (gamma flex) will be applied to the total hourly consumption of the forecasted PA indicating the consumption that is available for DR programs or the reliable flexibility potential that can be summed up to get the total. By multiplying the peak and off-peak rate difference with the total, we will obtain the gain from shifting PA. The same approach is carried out for the SHED DR program, but usually not all PAs are appropriate for shedding, thus a combination of SHED and SHIFT is proposed instead. In the case of shedding, the gain is obtained by multiplying the total with the peak rate as that consumption is shifted but not canceled. The proposed method is implemented with the consumption data of commercial buildings from the U.S. in the next section Simulation and Results.

Comparing with the method proposed in [19], we identify the similarities and differences described in Table 6:

Thus, the current method is structured in 9 steps, providing more precision, and the results are more accurate as it starts from the proposed flexibility forecast using one of the regressors or LSTM.

III. SIMULATION AND RESULTS

A. ANALYSIS OF THE DATASETS AND EXTRACTING INSIGHTS FROM BIG DATA

As discussed in the previous section, the files have been stored in S3, the object storage service from Amazon Web Services (AWS), and the data preparation, model training, and deployment have been done mainly in Amazon Sagemaker (a Jupyter environment). We used the AWS Data Wrangler library to seamlessly connect to the S3 stored data files and read them into Pandas DataFrames. To analyze the data and extract valuable insights, we used Athena to query data and to build a table-to-S3 dictionary in Amazon Glue. For the current study, the preliminary analysis data coming in.csv format. The dataset contains 14,976 *.csv* files with a total of 131.18 million records.[2] The files contain both electricity and gas consumption and their corresponding breakdown, but as our study is related to flexibility potential in terms of electricity consumption, we will focus on electricity. The programmable load consists of *fans*, *cooling*, and *heating appliances*, whereas the non-programmable load consists of *interior lighting* and *equipment*. Comparing programmable (flexible) and non-programmable loads in Table 7, we notice that at some intervals the flexibility is zero (Minimum = 0) and the total consumption of programmable appliances is smaller than interior lights and equipment meaning that there is more room for flexibility in case lighting and interior equipment become more flexible in the future. This aspect could be considered for buildings depending on their type.

The dataset we are studying consists of 16 types of buildings: *Full-Service Restaurant, Hospital, Large Hotel, Large Office, Medium Office, Midrise Apartment, Outpatient*

[2] https://openei.org/datasets/dataset/commercial-and-residential-hourly-load-profiles-for-all-tmy3-locations-in-the-united-states

TABLE 4. Forecasting model.

Step	**LSTM Model**	**Regression**
Data Preparation	Transforming the raw data: ✓ Data Cleaning ✓ Feature Selection ✓ Feature Engineering ✓ Dimensionality Reduction	The same steps are performed.
Setting up models	The dataset is divided into 2: 1 set for training and 1 for testing. The models will be developed using the training dataset and will make predictions on the test dataset.	To demonstrate the *predict_model()* function on unseen data, a sample of records will be withheld from the original dataset for predictions.
Data transformation	✓ Time Series to Supervised Learning To transform the time series into supervised learning, a function can be defined to use the observation from the last time step (t-1) as input and the observation at the current time step (t) as output. ✓ Time Series to Scale It consists in transforming data from time series to scale using the *MinMaxScaler*. Process Reversal-to bring the forecasts on the differentiated series to their original scale	Data will be used without being transformed.
Model development	For the LSTM, the model must be specified: ✓ a loss function and an optimization algorithm; ✓ the number of neurons in the first visible and from the exit layers; ✓ the training period and the size of the batch.	The following functions are used: ✓ *setup()* function initializes the environment and creates the transformation pipeline to prepare the data for modeling and deployment. ✓ *compare_models()* function trains all models in the model library and scores them using k-fold cross-validation to evaluate measurements (more than 20 models are trained and evaluated using cross-validation). ✓ *predict_model()* function is used to predict the model using an unseen dataset.

Healthcare, Primary School, Quick Service Restaurant, Secondary School, Small Hotel, Small Office, Stand-Alone Retail, Strip Mall, Supermarket, Warehouse. Thus, more sensitive buildings in terms of load variation such as hospitals and outpatient healthcare could be excluded or treated separately.

Furthermore, in Table 8, we analyze the statistic indicators for peak and off-peak hours since their particularities are significant for DR programs. However, peak hours could be split into mild-peak and critical-peak hours depending on the season, schedules, and working shifts, aiming to analyze load flexibility potential. Thus, we calculate the statistics indicators splitting the appliances into programmable and non-programmable ones operating at peak and off-peak hours.

Therefore, we target to apply DR programs to programmable appliances that operate at peak, whereas non-programmable appliances will remain unchanged.

Using the reduced data approach proposed in section II.A, we can extract valuable insights regarding commercial building operation and flexibility potential. Considering the state and ISO affiliation (the nine regional Independent System Operators that control the load in the U.S. are depicted in Figure 3(a) as CAISO, ERCOT, ISO-NE, MISO, NORTHWEST, NYISO, SOUTHEAST, SOUTHWEST, and SPP) and DR [37], we calculate the average flexibility potential in percentage for our dataset of commercial buildings at the ISO level (as in Figure 3(b)). ISO affiliation is added to the initial dataset grouping the states to corresponding ISOs since it is important for control areas to know the flexibility potential and envision strategies to use it.

There is a quite significant difference between ISO-NE, NORTHWEST with around 2%, and MISO with over 13%. Thus, the flexibility of buildings is not uniformly distributed among ISOs. In Figure 4, the breakdown of electricity consumption is showed for each ISO. The appliances that can provide flexibility are heating, cooling, and fans, whereas interior equipment and lights are considered less flexible.

The electricity consumption by appliance type is provided in Figure 5. Cooling systems and fans are the most numerous programmable appliances. They represent 76% of the total programmable appliances.

The electricity load curves for the 16 types of buildings are provided in Figure 6. These curves are interesting as they

TABLE 5. Flexibility assessment method.

Flexibility assessment method – SHIFT & SHED

1) Divide appliances into programmable PA and non-programmable NPA to separate flexible $\boldsymbol{C_{PA}}$ and fixed consumption $\boldsymbol{C_{NPA}}$.

2) Calculate total consumption of PA at hour *h* using forecasted values:

$$TC_{PA}^{h} = \sum_{i=1}^{m} C_{PA}^{i}$$

3) Calculate daily mean consumption using the load profile (real values from the monthly updated profile), n=24:

$$C_{avg} = \frac{\sum_{i=1}^{n} C_h}{n}$$

4) Extract peak hours $\boldsymbol{h_{PEAK}}$ for the analyzed interval: one month, year, etc.

Identify the start peak hour $\boldsymbol{h_{PEAK-STA}}$ and stop peak hour $\boldsymbol{h_{PEAK-STO}}$:

IF $C_h > C_{avg}$ THEN $h = h_{PEAK} \in \{h_{PEAK-STA} \div h_{PEAK-STO}\}$

5) Apply flexibility coefficient $\boldsymbol{\gamma_{flex}}$ to obtain the shiftable consumption:

$$C_{FLEX}^{h_{PEAK}} = TC_{PA}^{h_{PEAK}} \times \gamma_{flex},\ \gamma_{flex} \in \{0.01, 0.03, 0.05, 0.1, 0.15\}$$

6) Obtain total consumption to shift from peak to off-peak:

$$TC_{FLEX} = \sum_{h=h_{PEAK-STA}}^{h_{PEAK-STO}} C_{FLEX}^{h}$$

$$FLEX \in \{SHIFT, SHED, SHIMMY\}$$

7) Calculate the gain or benefit that can be obtained by shifting PA using the difference between peak rate and off-peak rates:

$$G_{SHIFT} = TC_{SHIFT} \times (r_{PEAK} - r_{OFFPEAK})$$

8) Calculate the gain or benefit that can be obtained by shedding PA:

$$G_{SHED} = TC_{SHED} \times r_{PEAK}$$

8) Calculate the total gain or benefit that can be obtained by shifting and shedding PA. First, calculate the share of shedding *q* and shifting appliances *1-q* of PA:

$$G_{SHIFT\&SHED} = TC_{SHIFT} \times (r_{PEAK} - r_{OFFPEAK}) \times (1 - q) + TC_{SHED} \times r_{PEAK} \times q$$

9) Compare the gains and choose the DR program (i.e. ALLSHIFT, SHIFT & SHED, etc.).

reveal the buildings with the highest consumption at peak hours and in general. For instance, hospitals, large offices, secondary schools, and large hotels are buildings with the highest consumption. However, some complementarity could be identified in terms of consumption hours among buildings with the highest consumption. Large hotels have two peaks: late evening and morning peaks, whereas the others reflect more activity during 6 and 18.

However, gas load curves are much different (as in Figure 7) and they can be considered in additional analyses to reflect the possible transfer to release the grid stress from electricity to gas and vice-versa.

Load profiles are drawn for total electricity and gas consumption of all buildings as in Figure 8 and breakdown by appliance type as in Figures 9 and 10.

Most of the heating is done by gas, probably because the gas price was more convenient. Therefore, it is reasonable that heating by electricity is small. Furthermore, the load profile for electrical appliances is very relevant for strategy makers as some rules have to be taken into account. For instance, cooling even it is the highest programmable load, cannot be shifted entirely to the night hours. Thus, a mix of strategies in terms of DR programs should be considered.

Moreover, very relevant is Figure 11, showing the potential of each building type. It seems that hospitals have the highest potential, with the highest flexibility, but it is sensitive in terms of patients' comfort and safety so the DR program should be adjusted accordingly considering the building type.

Thus, large offices and secondary schools could be more targeted in terms of DR programs penetration as they will allow scholars and employees shifts of even working from home that will somehow transfer the consumption from commercial to residential buildings, especially during pandemic times.

B. ANALYSIS OF THE DATASETS AND EXTRACTING INSIGHTS FROM BIG DATA

Using the reduced dataset according to section II.A, we intend to perform the flexibility forecast that will be the input data for the proposed flexibility assessment method to evaluate the flexibility potential and the efficiency of DR programs. From the analysis of the data, we discovered that states such as Texas and California have the highest consumption values on the flexible component (fans, cooling, and heating) and states such as Delaware and Alaska, the lowest ones (as in Figure 12).

For forecast exemplification and analysis, we used the consumption data recorded in Texas, as it is one of the most representative state, aggregated at the hourly level, as follows:

TABLE 6. Similarities and differences between the current method and the method proposed in [19].

No.	Differences	Similarities
1	In the current method, we consider the flexibility forecast as input, not the historical data as in [19] which provides more accurate results;	Both compute flexibility potential and savings;
2	Peak and off-peak hours are differently selected, in the current method the selection considers the average consumption and it has to be updated periodically, whereas in [19] the peak and off-peak hours are selected following the load curve without a precise rule. The current approach provides more accurate and realistic results since the load curve is seasonally changing or it suffers from changes;	Both take into account the same appliances as programmable PA and non-programmable NPA;
3	The current method calculates the energy for SHIFT and SHED considering the share of the appliances in each category, whereas in [19] the consumption of appliances by SHIFT and SHED is calculated separately. It does not impact the output significantly.	Both consider two DR programs: 1) all shift and 2) the combination between SHIFT and SHED;

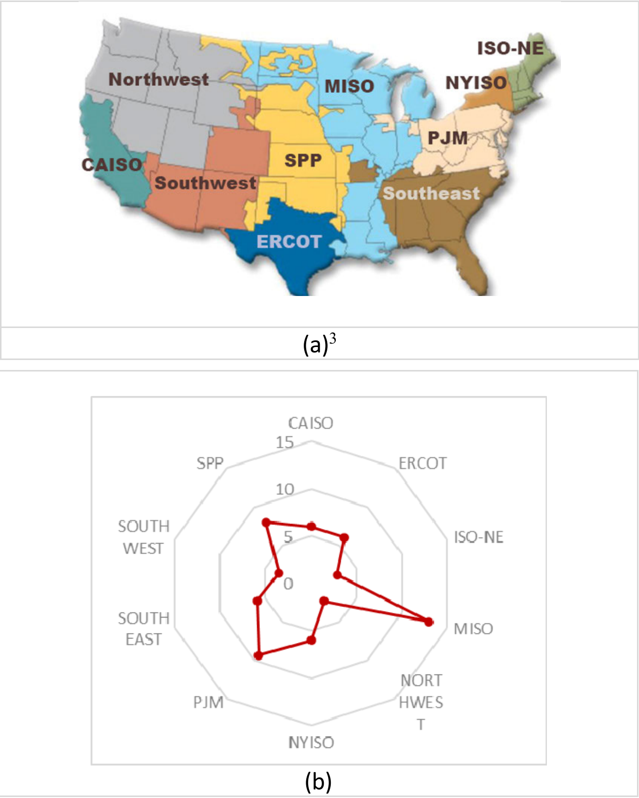
TABLE 7. Statistic indicators for programmable and non-programmable appliances.

Statistic indicators	*PA [kW]*	*NPA [kW]*
Mean	23,220,457.59	29,295,097.36
Standard Error	2,183,909.43	2,230,055.88
Median	4,808,748.42	12,497,281.58
Mode	0	2,019,138.49
Standard Deviation	42,795,710.04	43,699,992.08
Sample Variance	1.83147E+15	1.90969E+15
Kurtosis	5.018	7.439
Skewness	2.432	2.685
Range	171,935,895.40	222,772,877.40
Minimum	0	716,730.68
Maximum	171,935,895.40	223,489,608.10
Sum	8,916,655,716.00	11,249,317,388.00

- combined with weather data from NOAA (National Center for Environmental Information/ https://www.ncdc.noaa.gov/crn/qcdatasets.html) for regression analysis using PyCaret;

TABLE 8. Statistic indicators for programmable and non-programmable appliances at peak and off-peak hours.

Stat. Ind.	***Peak PA [kW]***	***Peak NPA [kW]***	***Off-peak PA [kW]***	***Off-peak NPA [kW]***
Mean	25,234,391	29,210,824	22,549,146	29,323,188
Std. Error	4,454,835	3,821,704	2,508,076	2,690,228
Med.	6,998,947	12,782,877	4,172,706	12,386,520
Std. Dev.	43,648,298	37,444,904	42,563,475	45,654,688
Smp Var.	1.90517E15	1.40212E15	1.81165E15	2.08435E15
Kr.	4.758	4.786	5.231	7.615
Skew.	2.345	2.177	2.477	2.755
Range	171,478,972	188,532,236	171,935,895	222,772,877
Min.	123,271	816,074	0	716,730
Max.	171,602,244	189,348,311	171,935,895	223,489,608
Sum	2,422,501,577	2,804,239,170	6,494,154,139	8,445,078,218
Count	96	96	288	288



(a)[3]

(b)

FIGURE 3. Flexibility potential at ISO level.

- without enhancing the data (no adding information about the weather) for the LSTM model.

The weather values taken from NOAA are values at the hourly level and contain information such as air temperature,

[3] https://www.ferc.gov/electric-power-markets

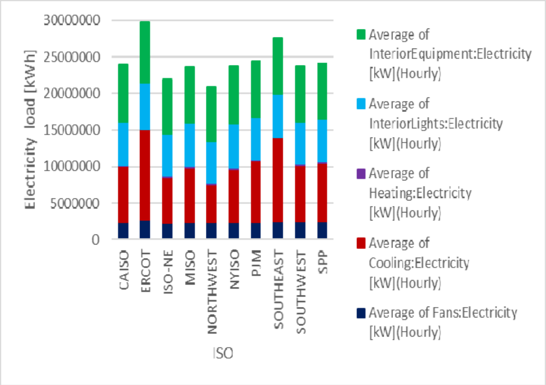


FIGURE 4. Electricity consumption on average by appliance type and ISO.

precipitation, global solar radiation, relative humidity, surface infrared temperature, soil moisture, and soil temperature. They are forecasting variables used as input data in the regression model.

As proof of concept, we aim to predict future consumption on one of the flexible components namely ''Fans: Electricity [kW](Hourly)''. Of course, the models can be applied to any component in the dataset.

For regression analysis (using different regressors), we follow several steps:

1) **Data Preparation:** The first step in performing the analysis was to prepare the data. From the initial dataset of 131 million records, 8.5 million related to consumption in the state of Texas were extracted using aggregation queries as in section II.A. The new dataset obtained was then aggregated at the hourly level to be combined with the weather dataset. Because weather data is provided for specific time slots (hours: 2, 6, 8, 11, 13, 14, 15, 17, 19, 21, 23), the combination of the two, the consumption records and the weather time slots, resulted in a final set of 7,270 records.
2) **Setting up the model: Unseen Data.** From the obtained dataset, a 10% sample was retained from the initial dataset to be used for prediction.
3) **Model implementation and comparison:** After the environment was initialized, all the regression models available in the library were run and compared. Out of them, we chose to further analyze and plot the results of Random Forest Regressor, Linear Regression, and AdaBoost Regressor to determine the accuracy of different regressors. The forecast results with 14 regressors are presented in Table 9.

A model was created for regressors that were used to predict consumption for 24 hours. From the dataset, the day of July, 15th, a summer day in which flexible consumption usually registers significant increases, is selected and to that day we apply the above-mentioned models. The results are shown in Figures 13-15.

After checking the results, we've noticed that for Random Forest Regressor and Linear Regression, the forecasted consumption represents 96% of the actual consumption (in total, prediction is lower with 4% than actual consumption). In contrast, with AdaBoost Regressor, prediction is higher with 9% than actual consumption. For the LSTM model, we use the data with the initial variables, without weather data, representing the consumption in Texas aggregated at date level. The number of observations in the dataset is 8,395. Unlike the regression simulation, where we considered variables related to the weather conditions, for this model we intend to determine the future consumption based on the previous consumptions recorded for the other components (flexible or not).

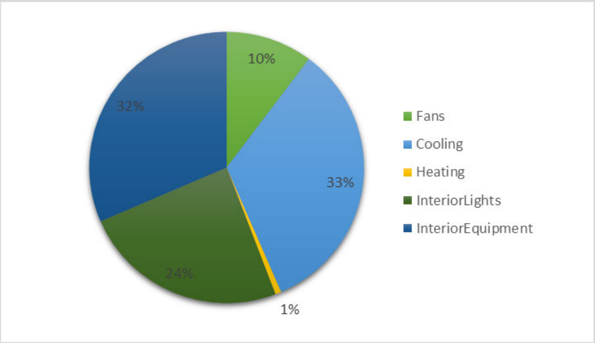


FIGURE 5. Electricity consumption on average by electricity appliance type.

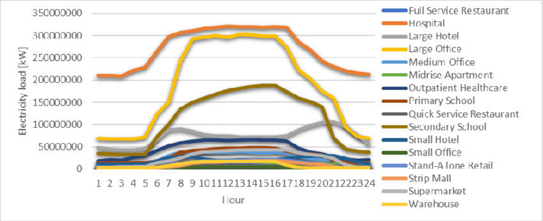


FIGURE 6. Electricity load curves for 16 building types.

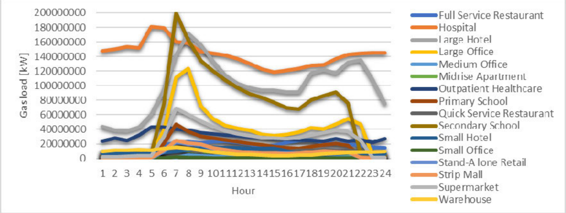


FIGURE 7. Gas load curves for 16 building types.

The difference between the two models consists in dividing the dataset into two: a training and a test set, followed by the transformation of the data from time series to supervised learning, respectively into scales for the LSTM model. To build the model, the following parameters are used: train data - 6000; loss function - ''mean_squared_error''; optimization algorithm - ADAM; neurons in the first visible layer - 100; epochs - 100; batch size - 70. As in the case of regression, we intended to forecast the consumption for 24 hours. The results are presented in Figures 16 and 17.

It can be observed from Figure 17 that the train and test performances are pretty close, and we can infer that the

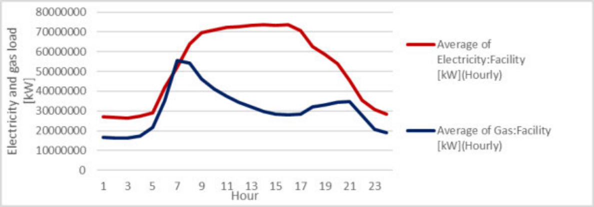


FIGURE 8. Electricity and gas load profile in total.

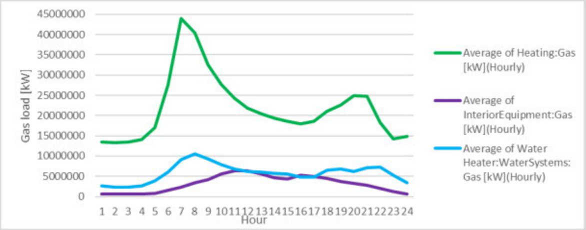


FIGURE 9. Gas load profile breakdown by appliance type.

model converges quite quickly. Starting from this observation, we can say that LSTM is a better choice for our model.

If we compare with the results depicted in Table 10, it can be noticed that LSTM performs better, considering that RMSE is 1,234.40. While performing the analysis, after several iterations, we discovered that the larger the training set, the better the result obtained. For LSTM models using increasingly larger datasets ensures a better training of the model. Thus, we can infer that this model is more suitable for solving time series problems which don't require additional calculation efforts, considering that the initial dataset (without weather variables) is used. Of course, the computing power available when using large sets of data must be considered.

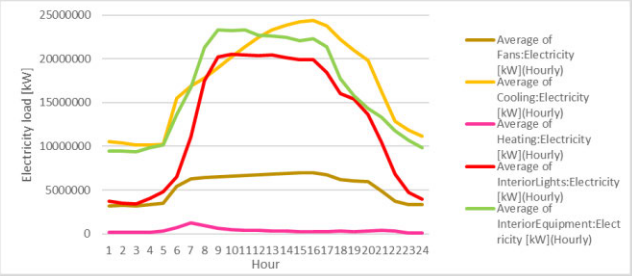


FIGURE 10. Electricity load profile breakdown by appliance type.

C. DR PROGRAMS AND FLEXIBILITY ASSESSMENT IMPLEMENTATION

The current much higher variety of energy sources, compared with the more predictable large power plants and global load leads to the necessity of more back-ups from the generation and load side. The generation reserve consists of rapid generating units (gas and hydro) that can rapidly change the output or even consume at night (pump-hydro power plants). The Demand Side Management (DSM) includes the DR concept that targets the load flexibility to balance generation and consumption. This concept is progressing a lot due to the advancement of Information and Communications Technology (ICT) and increasing awareness and motivation from demand. As mentioned, the flexible loads in our dataset are heating, cooling, and fans that could be engaged in at least one of the DR services: shift, shed, or shimmy with specific enabling technology. Shift DR services consist in rearranging the loads and involve shifting the operation of flexible loads (programmable appliances) from high-rate to low-rate hours (using a Time-of-Use (ToU) tariff), with the benefits of the price difference (peak rate could be 0.38 Euro cents/kWh and off-peak rate 0.09 Euro cents/kWh). Shed DR services rely on the capacity of some appliances to temporally reduce the load at peak intervals. For instance, the consumption of cooling, fans, or heating systems will be reduced for short intervals without disturbing the consumers' comfort for the residential sector or commercial activities. Even if the reduction is just for short intervals such as 5 or 10 up to 15 minutes, the system will benefit from multiple loads simultaneously reduced that will lower the consumption curve. The perception of DR services is very important because if the consumers perceive the DR program as negative, altering their commercial activities, they will not participate in DR. Shimmy DR services are more complex, remotely controllable, and usually imply enabling technologies that allow the appliance to follow a precise dispatch signal increasing and decreasing (more often) the load. They also vary in terms

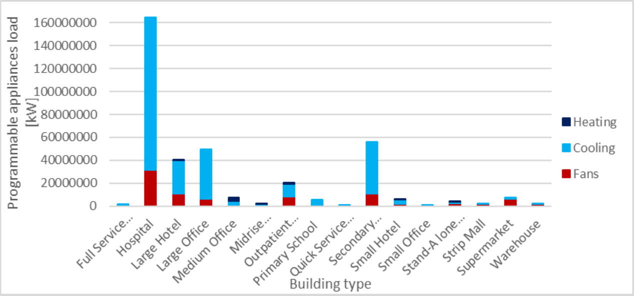


FIGURE 11. Programmable appliances breakdown by building type.

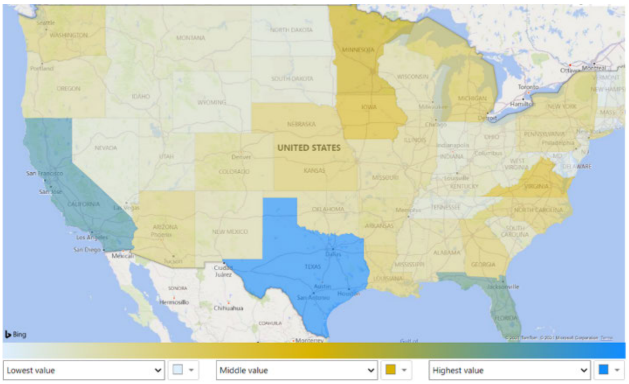


FIGURE 12. Flexible consumption (fans, cooling, and heating) per state.

TABLE 9. Model comparison result.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
rf	Random Forest Regressor	754.0837	2437332.75	1548.9295	0.9342	0.0839	0.0403	0.086
lightgbm	Light Gradient Boosting Machine	846.5858	2441569.36	1551.2981	0.9341	0.0852	0.0468	0.096
et	Extra Trees Regressor	682.8883	2522755.49	1584.4508	0.9319	0.0847	0.0358	0.08
gbr	Gradient Boosting Regressor	1279.2859	3487373.75	1858.8847	0.906	0.1049	0.0722	0.04
dt	Decision Tree Regressor	852.1163	3830193.65	1934.3649	0.8965	0.1042	0.0446	0.011
llar	Lasso Least Angle Regression	2130.0449	7098403.11	2660.3356	0.8086	0.1937	0.1288	0.008
lasso	Lasso Regression	2102.8321	7107392.3	2661.6943	0.8084	0.2023	0.128	0.368
ridge	Ridge Regression	2106.2143	7106934.55	2661.7367	0.8084	0.201	0.1282	0.363
br	Bayesian Ridge	2106.3101	7107273.89	2661.7995	0.8084	0.201	0.1282	0.009
lr	Linear Regression	2103.2053	7113251.45	2662.8367	0.8082	0.2031	0.1281	0.447
omp	Orthogonal Matching Pursuit	2726.8532	10851969.2	3290.9694	0.7072	0.2037	0.1542	0.008
ada	AdaBoost Regressor	3015.8491	12295438.2	3503.2298	0.6687	0.1842	0.1625	0.038
knn	K Neighbors Regressor	4035.8092	25466760.6	5041.2061	0.3133	0.2631	0.2245	0.024
en	Elastic Net	4334.0464	25907620	5088.8324	0.3009	0.2971	0.2626	0.364

of time horizon implementation; shift DR service is applied on long and medium-term, shed on medium and short-term, shimmy on short and ultra-short-term. However, the DR services implementation depends on the building type, physical infrastructure, or the existence of the enabling technologies that involve considerable costs. According to [37], [38], for controlling Heat Ventilation Air Conditioning (HVAC) with shed & shift, the ETC or cost is around 242$, with shed only is 169$, whereas with shimmy is considerable: 2376$. The cost includes control technology, communication, and hardware. There are different costs for lighting, refrigerators warehouse, and water heaters, but they are not included in our dataset. Thus, when estimating the benefits from DR services, we have to subtract the ETCs. As the DR capability estimation is usually below 10%, we assumed these percentages from programmable appliances consumption as flexible with the results of 4 scenarios multiplied by the DR services for which the ETCs are available. However, it is not possible to simulate shimmy DR service as it implies fine up/down tunning depending on the system real-time balancing requirements that can be paid at a fixed price as in Florida Power & Light[4] or variable as in auctions. Therefore, we analyze shed & shift DR service, shedding cooling systems at peak and partially shifting the heating and fans systems that totalize 24% of the programmable appliances as in Figure 18.

For simulation, we start identifying programable appliances or the flexibility potential of the commercial buildings, peak (6-21) and off-peak hours, and possible DR programs or their efficient combination that can be implemented for our dataset. Reasonable DR capability percentage – flexibility coefficients (between 1 and 10%) of the programmable appliances are considered according to previous studies. The results of the DR programs simulation are provided in Table 11 and Figure 19 using the flexibility forecast results described in the previous section.

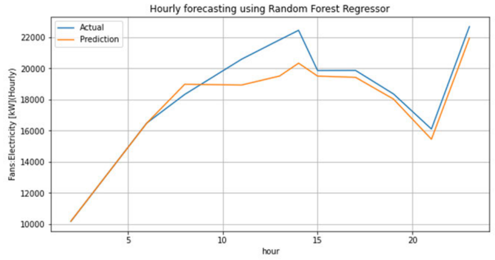


FIGURE 13. Random forest regressor.

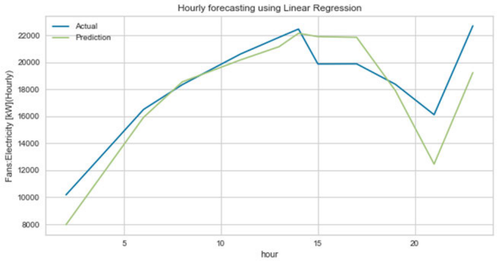


FIGURE 14. Linear regression.

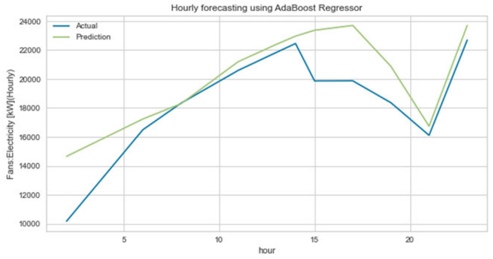


FIGURE 15. AdaBoost regressor.

Benefits are calculated considering the price difference. Thus, we consider that the peak rate is 0.38$ and the off-peak rate is 0.09$. When shedding the colling systems, the entire shed energy is saved.

[4] https://www.fpl.com/

TABLE 10. LSTM data sample.

Electricity Facility [kW]	Fans [kW]	Cooling [kW]	Heating [kW]	Interior Lights [kW]
116764.20	10699.06	36984.23	1027.66	11793.80
115953.16	10353.21	37143.77	966.76	11531.67
115519.70	10455.80	37675.26	1149.13	10339.19
114935.08	10443.04	37221.31	1105.49	10339.19
117844.01	10590.74	37367.62	1287.28	10998.35
120769.41	10993.92	36994.24	1244.30	11422.56
141279.77	12680.70	39508.67	1407.14	14648.98
142873.73	12722.66	39977.20	1283.09	15087.56
152355.91	14088.27	42189.98	1187.09	16897.91

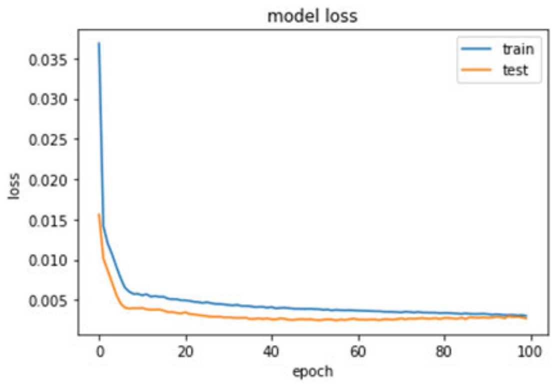


FIGURE 16. LSTM model loss (mean_squared_error over epochs).

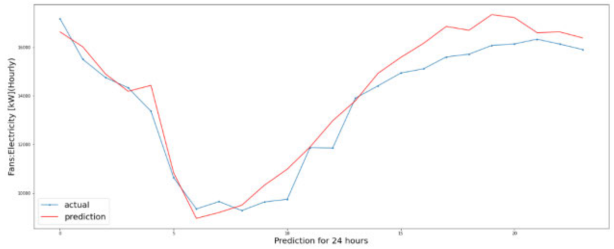


FIGURE 17. LSTM prediction for 24 hours.

Comparing the two main DR programs that rely on 5% of the programmable appliances, the combination SHED & SHIFT is better than ALLSHIFT by 19%. Also, the advantage of the combination SHED & SHIFT is given by the different time horizons of implementation. For ALLSHIFT, the shifted energy is proportional to the benefits, the more energy is shifted the benefits are higher as they depend on the difference of peak and off-peak rates. But shedding also implies load reduction without rescheduling the appliances that represent not only money savings, but energy savings with social implications in terms of CO_2, deforestation reduction, standard coal avoidance, etc.

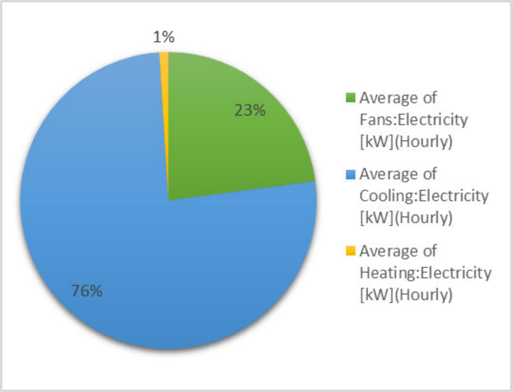


FIGURE 18. Share of the programmable appliances.

TABLE 11. Results of the DR programs simulation – per year.

DR program	DR services	Energy [kWh]	Benefits $
ALL SHIFT	SHIFT1%	70,657,689.4	20,490,729.9
	SHIFT3%	211,973,068.3	61,472,189.8
	SHIFT5%	353,288,447.1	102,453,649.7
	SHIFT10%	706,576,894.2	204,907,299.3
SHED& SHIFT	Cooling SHED5%	268,499,219.8	102,029,703.5
	Heating & Fans SHIFT5%	84,789,227.3	24,588,875.9

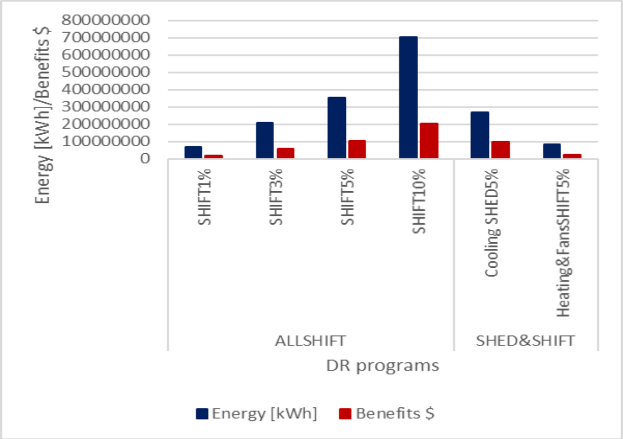


FIGURE 19. Results of the DR programs simulation per year.

IV. CONCLUSION

Starting from a large consumption dataset of commercial buildings of different types, we tested two approaches to store and reduce the data. We chose the AWS approach over the Hadoop one, because of lower processing time and a more mature and ready-to-use environment centered around SageMaker and S3.

After choosing the data lake architecture for big data, we applied two machine learning methods to forecast the flexibility: Long Short-Term Memory (LSTM) and regression analysis using the PyCaret library. Both methods provided promising results since roughly both estimations were pretty close to the actual consumption values. However, it can be concluded that the LSTM model is more suitable for the time series analysis as it provides a better RMSE score than the regressors. An additional insight that stems from the analysis is that for the LSTM model, a larger dataset is required to train the model and thus obtain a high accuracy forecast. Compared to LSTM, the regression model also offers good results on aggregated datasets, so it does not depend on having big datasets available to be able to return relevant forecasts.

With the reduced dataset and a reliable forecast, we proposed a flexibility assessment method to identify the flexibility potential of commercial buildings and the efficiency of several DR programs. The merit of our method is twofold: first, it uses the results from previous studies regarding the flexibility coefficients and DR programs; second, it is a novel contribution as it provides an approach to evaluate the flexibility that can be used by the grid operators to balance the systems and suppliers to enhance their market acquiring strategies. The results are better when combining the DR services SHIFT & SHED. As future work, we consider enhancing the triplet: big data - forecast - flexibility assessment approach. Furthermore, we aim to improve the forecasting models featured in this study to obtain an even more accurate forecast. To do this, the scope will be broadened, and we plan to use an extended period. The benefit of this extension is that it will allow the models to train better and, as a consequence, we will have increased accuracy. Additional enhancements are to be investigated, such as trying to find the best parameters for the models, as well as identifying and selecting the most relevant variables for the newly created context.

REFERENCES

[1] G. Dileep, "A survey on smart grid technologies and applications," *Renew. Energy*, vol. 146, pp. 2589–2625, Feb. 2020, doi: 10.1016/j.renene.2019.08.092.

[2] B. Deng, Y. Wen, and P. Yuan, "Hybrid short-term load forecasting using the Hadoop MapReduce framework," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Aug. 2020, pp. 1–5, doi: 10.1109/PESGM41954.2020.9282094.

[3] D. Syed, A. Zainab, A. Ghrayeb, S. S. Refaat, H. Abu-Rub, and O. Bouhali, "Smart grid big data analytics: Survey of technologies, techniques, and applications," *IEEE Access*, vol. 9, pp. 59564–59585, 2021, doi: 10.1109/ACCESS.2020.3041178.

[4] F. Ceglia, P. Esposito, E. Marrasso, and M. Sasso, "From smart energy community to smart energy municipalities: Literature review, agendas and pathways," *J. Cleaner Prod.*, vol. 254, May 2020, Art. no. 120118, doi: 10.1016/j.jclepro.2020.120118.

[5] G. Dóci, E. Vasileiadou, and A. C. Petersen, "Exploring the transition potential of renewable energy communities," *Futures*, vol. 66, pp. 85–95, Feb. 2015, doi: 10.1016/j.futures.2015.01.002.

[6] J. Lowitzsch, C. E. Hoicka, and F. J. van Tulder, "Renewable energy communities under the 2019 European clean energy package—Governance model for the energy clusters of the future?" *Renew. Sustain. Energy Rev.*, vol. 122, Apr. 2020, Art. no. 109489, doi: 10.1016/j.rser.2019.109489.

[7] E. M. Gui and I. MacGill, "Typology of future clean energy communities: An exploratory structure, opportunities, and challenges," *Energy Res. Social Sci.*, vol. 35, pp. 94–107, Jan. 2018, doi: 10.1016/j.erss.2017.10.019.

[8] S. Homaei and M. Hamdy, "Quantification of energy flexibility and survivability of all-electric buildings with cost-effective battery size: Methodology and indexes," *Energies*, vol. 14, no. 10, p. 2787, May 2021, doi: 10.3390/en14102787.

[9] J. Potter and P. Cappers, "Demand response advanced controls framework and assessment of enabling technology costs," Dept. Energy Office Energy Efficiency Renew. Energy, Lawrence Berkeley Nat. Lab., Berkeley, CA, USA, Tech. Rep. LBNL-2001044, 2017.

[10] A. Faruqui, D. Harris, and R. Hledik, "Unlocking the 53 billion savings from smart meters in the EU: How increasing the adoption of dynamic tariffs could make or break the EU's smart grid investment," *Energy Policy*, vol. 38, no. 10, pp. 6222–6231, Oct. 2010, doi: 10.1016/j.enpol.2010.06.010.

[11] A. T. Z. R. Sausen, M. Campos, P. S. Sausen, M. O. Binelo, M. F. B. Binelo, J. M. L. V. Da Silva, and M. D. Santos, "Classification of the social distance during the COVID-19 pandemic from electricity consumption using artificial intelligence," *Int. J. Energy Res.*, vol. 45, no. 6, pp. 8837–8847, May 2021, doi: 10.1002/er.6418.

[12] M. Carvalho, D. B. de Mello Delgado, K. M. Lima, M. C. Cancela, C. A. dos Siqueira, and D. L. B. De Souza, "Effects of the COVID-19 pandemic on the Brazilian electricity consumption patterns," *Int. J. Energy Res.*, vol. 45, no. 2, pp. 3358–3364, Feb. 2021, doi: 10.1002/er.5877.

[13] I. Santiago, A. Moreno-Munoz, P. Quintero-Jiménez, F. Garcia-Torres, and M. J. Gonzalez-Redondo, "Electricity demand during pandemic times: The case of the COVID-19 in Spain," *Energy Policy*, vol. 148, Jan. 2021, Art. no. 111964, doi: 10.1016/j.enpol.2020.111964.

[14] S. Halbrügge, P. Schott, M. Weibelzahl, H. U. Buhl, G. Fridgen, and M. Schöpf, "How did the German and other European electricity systems react to the COVID-19 pandemic?" *Appl. Energy*, vol. 285, Mar. 2021, Art. no. 116370, doi: 10.1016/j.apenergy.2020.116370.

[15] V. Marinakis, "Big data for energy management and energy-efficient buildings," *Energies*, vol. 13, no. 7, p. 1555, Mar. 2020, doi: 10.3390/en13071555.

[16] X. J. Luo, L. O. Oyedele, A. O. Ajayi, C. G. Monyei, O. O. Akinade, and L. A. Akanbi, "Development of an IoT-based big data platform for day-ahead prediction of building heating and cooling demands," *Adv. Eng. Informat.*, vol. 41, Aug. 2019, Art. no. 100926, doi: 10.1016/j.aei.2019.100926.

[17] X. J. Luo and L. O. Oyedele, "Forecasting building energy consumption: Adaptive long-short term memory neural networks driven by genetic algorithm," *Adv. Eng. Informat.*, vol. 50, Oct. 2021, Art. no. 101357, doi: 10.1016/j.aei.2021.101357.

[18] S.-V. Oprea, A. Bâra, C. Ceaparu, A. Ducman, V. Diaconiţa, and G. D. Ene, "Insights with big data analysis for commercial buildings flexibility in the context of smart cities," in *Proc. 10th Int. Conf. Smart Cities Green ICT Syst. (SMARTGREENS)*, Apr. 2021, pp. 118–124, doi: 10.5220/0010409801180124.

[19] S.-V. Oprea, A. Bâra, R. C. Marales, and M.-S. Florescu, "Data model for residential and commercial buildings. Load flexibility assessment in smart cities," *Sustainability*, vol. 13, no. 4, p. 1736, Feb. 2021, doi: 10.3390/su13041736.

[20] S.-V. Oprea and A. Bâra, "Edge and fog computing using IoT for direct load optimization and control with flexibility services for citizen energy communities," *Knowl.-Based Syst.*, vol. 228, Sep. 2021, Art. no. 107293, doi: 10.1016/j.knosys.2021.107293.

[21] T. Cioara, I. Anghel, I. Salomie, M. Antal, C. Pop, M. Bertoncini, D. Arnone, and F. Pop, "Exploiting data centres energy flexibility in smart cities: Business scenarios," *Inf. Sci.*, vol. 476, pp. 392–412, Feb. 2019, doi: 10.1016/j.ins.2018.07.010.

[22] Z. Foroozandeh, S. Ramos, J. Soares, Z. Vale, and M. Dias, "Single contract power optimization: A novel business model for smart buildings using intelligent energy management," *Int. J. Electr. Power Energy Syst.*, vol. 135, Feb. 2022, Art. no. 107534, doi: 10.1016/j.ijepes.2021.107534.

[23] Y. Zhou and S. Cao, "Quantification of energy flexibility of residential net-zero-energy buildings involved with dynamic operations of hybrid energy storages and diversified energy conversion strategies," *Sustain. Energy, Grids Netw.*, vol. 21, Mar. 2020, Art. no. 100304, doi: 10.1016/j.segan.2020.100304.

[24] J. M. Perkel, "Why Jupyter is data scientists' computational notebook of choice," *Nature*, vol. 563, no. 7729, pp. 145–146, Nov. 2018, doi: 10.1038/d41586-018-07196-1.

[25] H. Fang, "Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem," in *Proc. IEEE Int. Conf. Cyber Technol. Autom., Control, Intell. Syst. (CYBER)*, Jun. 2015, pp. 820–824, doi: 10.1109/CYBER.2015.7288049.

[26] A. A. Munshi and Y. A.-R.-I. Mohamed, "Data lake Lambda architecture for smart grids big data analytics," *IEEE Access*, vol. 6, pp. 40463–40471, 2018, doi: 10.1109/ACCESS.2018.2858256.

[27] R. Mahanti, *Data Quality: Dimensions, Measurement, Strategy, Management, and Governance*. Birmingham, AL, USA: Quality Press, 2019.

[28] D. Vohra, "Apache parquet," in *Practical Hadoop Ecosystem*. Berkeley, CA, USA: Apress, 2016, pp. 325–335, doi: 10.1007/978-1-4842-2199-0_8.

[29] T. Ivanov and M. Pergolesi, "The impact of columnar file formats on SQL-on-Hadoop engine performance: A study on ORC and Parquet," *Concurrency Comput., Pract. Exp.*, vol. 32, no. 5, Mar. 2020, Art. no. e5523, doi: 10.1002/cpe.5523.

[30] A. Mishra, "AWS Lambda," in *Machine Learning in the AWS Cloud*. Hoboken, NJ, USA: Wiley, 2019, pp. 237–255.

[31] C. Behm, L. Nolting, and A. Praktiknjo, "How to model European electricity load profiles using artificial neural networks," *Appl. Energy*, vol. 277, Nov. 2020, Art. no. 115564, doi: 10.1016/j.apenergy.2020.115564.

[32] J. Zhang, Y.-M. Wei, D. Li, Z. Tan, and J. Zhou, "Short term electricity load forecasting using a hybrid model," *Energy*, vol. 158, pp. 774–781, Sep. 2018, doi: 10.1016/j.energy.2018.06.012.

[33] A. Yang, W. Li, and X. Yang, "Short-term electricity load forecasting based on feature selection and least squares support vector machines," *Knowl.-Based Syst.*, vol. 163, pp. 159–173, Jan. 2019, doi: 10.1016/j.knosys.2018.08.027.

[34] I. K. Nti, M. Teimeh, O. Nyarko-Boateng, and A. F. Adekoya, "Electricity load forecasting: A systematic review," *J. Electr. Syst. Inf. Technol.*, vol. 7, no. 1, pp. 1–19, Dec. 2020, doi: 10.1186/s43067-020-00021-8.

[35] S.-V. Oprea and A. Bara, "Machine learning algorithms for short-term load forecast in residential buildings using smart meters, sensors and big data solutions," *IEEE Access*, vol. 7, pp. 177874–177889, 2019, doi: 10.1109/ACCESS.2019.2958383.

[36] U. Gain and V. Hotti, "Low-code AutoML-augmented data pipeline—A review and experiments," *J. Phys., Conf. Ser.*, vol. 1828, no. 1, Feb. 2021, Art. no. 012015, doi: 10.1088/1742-6596/1828/1/012015.

[37] R. Hledik, A. Faruqui, T. Lee, and J. Higham, "The national potential for load flexibility: Value and market potential through 2030," Brattle Group, Boston, MA, USA, 2019. [Online]. Available: https://brattlefiles.blob.core.windows.net/files/16639_national_potential_for_load_flexibility_-_final.pdf

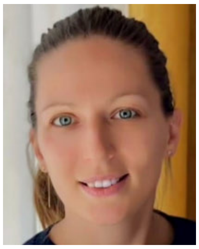
[38] R. Hledik, A. Faruqui, T. Lee, and J. Higham, "Brattle study: Cost-effective load flexibility can reduce costs by more than $15 billion annually," Brattle Group, Boston, MA, USA, Jun. 2019. [Online]. Available: https://www.brattle.com/insights-events/publications/brattle-study-cost-effective-load-flexibility-can-reduce-costs-by-more-than-15-billion-annually/

SIMONA VASILICA OPREA received the M.Sc. degree through the Infrastructure Management Program from Yokohama National University, Japan, in 2007, the Ph.D. degree in power system engineering from Bucharest Polytechnic University, in 2009, and the Ph.D. degree in economic informatics from the Bucharest University of Economic Studies, in 2017. She is currently a Lecturer. She teaches databases, database management systems, and software packages with the Faculty of Economic Cybernetics, Statistics, and Informatics, Bucharest University of Economic Studies.

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

VLAD DIACONITA (Member, IEEE) received the Ph.D. degree in Statistics and Cybernetics from the Bucharest University of Economic Studies. He was a Postdoctoral Fellow of the Research Group on Agent-Based, Social and Interdisciplinary Applications, Complutense University of Madrid, under the coordination of Professor J. Pavon. He is currently a Professor and a Ph.D. Supervisor with the Bucharest University of Economic Studies. He has authored over 50 papers in international journals and conferences. His research interests include databases, big data, system integration, decision support, cloud computing, and renewable energies. He is also an Associate Editor of the IEEE Access journal. He also served as a reviewer for numerous journals and conferences.

CĂTĂLIN CEAPARU is currently pursuing the Ph.D. degree with the Economic Informatics Doctoral School, Bucharest University of Economic Studies. He works under the supervision of Ph.D. Professor Adela Bâra, with "IT Solutions for the Analysis of Large Volumes of Data" as the chosen scientific research theme.



ANCA ALEXANDRA DUCMAN received the bachelor's degree in foreign languages and the master's degree business communication. She is currently pursuing the Ph.D. degree in economics with the Bucharest Academy of Economic Studies. She is working in critical economic fields. As general interests she is a Convinced Environmentalist, advocating for sustainable projects that aim to use and preserve natural resources, reduce pollution, and its effects.