Architecting a big data-driven software architecture for smart street lighting

Mubashir Ali and Patrizia Scandurra
DIGIP
University of Bergamo
Italy
{mubashir.ali,patrizia.scandurra}@unibg.it

Fabio Moretti and Laura Blaso
Smart Cities and Communities (TERIN-SEN-SCC) lab.
ENEA
Rome-Ispra, Italy
{fabio.moretti,laura.blaso}@enea.it

***Abstract*—Big data is one of the enabling technologies of the vision of Industry 4.0. Technological evolution is able to generate an increasing number of data. From the web, to social networks, from mobile devices to sensors, data is conveyed through the most disparate products of technology, whether physical or virtual. However, the transition from big data to smart data providing insights about information and issues that matter is not simple and obvious to achieve. The greater the amount of data and the more heterogeneous they are, the more complex their processing will be. The data, once collected, is processed by complex analytics algorithms and, to do this, considerable storage units and computing power are needed.**

In this paper, we describe our approach and experience at the Italian national agency ENEA in architecting a big data-driven software architecture for public street lighting. Such a software architecture is called ENEA PELL smart city platform (in brief, PELL SCP) and it is intended to collect, represent, control, predict, and possibly optimize the behaviour of public street lighting plants. In particular, we provide an overview of the analytics features that are being developed in collaboration with the University of Bergamo (Italy) to analyze electric energy data as collected by the PELL SCP.

***Index Terms*—Big data-driven software architecture, smart street lighting, smart city platforms, PELL, city analytics**

I. Introduction

Big data is one of the enabling technologies of the vision of Industry 4.0 and of the concept of Smart City. Big data solutions make the collection and availability of very large amounts of data possible; however, they also demand for new paradigms for massive data and analitics to turn big data into smart data providing new insights about issues that matter, that go beyond processing, storing and accessing records rapidly.

Processing and managing energy consumption in public street lighting is a substantial part of the Smart City concept. The project Public Energy Living Lab (PELL)[1] of the Italian national agency ENEA is dedicated to the census and digitization of public lighting infrastructures and the sharing of a national standard of knowledge representation, monitoring and evaluation of the lighting plants.

This paper presents, in a unified manner, the distinctive features of the ENEA PELL smart city platform (in brief, PELL SCP), which is intended to collect, manage, and analyze large scale structured and unstructured data about public street lighting in real-time. This paper describes our approach and experience in architecting and realizing a big data-driven software architecture for the PELL SCP, including analytics services. We illustrate the different types of data analysis that are being explored to obtain PELL smart data from PELL big data. These include descriptive analytics, diagnostic analitics, and predictive analitics. All of these analytics approaches provide a unique perspective that may help city/municipality and/or utility managers in making the most of their big data for a better management of the lighting plants.

This paper is organized as follows. Section II presents the main characteristics of the public street lighting domain, including the PELL initiative and the type of big data. Section III provides an overview of the big data- driven software architecture for the PELL SCP. Section IV reports some implementation details and lesson learned from our experience. Section V reports on some inspiring related works about big-data driven smart city platforms. Finally, Section VI concludes the paper and outlines future research directions.

II. Application domain

This Sections presents the PELL initiative and the type of Big Data collected by the PELL SCP.

A. The PELL project

The PELL project started in 2014 and involves several Italian stakeholders such as public authorities (like the S.p.A CONSIP and Acquirente Unico), utility providers, and energy service companies. The project goal is developing an efficient and effective management model for public lighting. The public street lighting sector is the first one addressed by the PELL project; in fact, lighting system plants are widespread everywhere in the Italian territory, and a rationalization through a capillary mapping could serve as a baseline for core tasks such as maintenance or energy re-qualification. Moreover, public lighting is one of the highest expense items in municipal administrations, thus the potential improvement can lead to consistent economic savings.

The PELL model was adopted in practice by Consip Spa, as part of the public tender Servizio Luce 4 (2016) and GEIP (2019) for the management and energy efficiency of public lighting systems owned by local authorities. Furthermore, it is

[1]https://www.pell.enea.it/enea/

being adopted by the Italian regions Lombardy and Basilicata (May 2022) for the census and digitization of public lighting systems[2].

B. Big Data in the PELL SCP

We here describe the type of data managed by the PELL SCP in terms of the well known *5V* features (*Volume*, *Velocity*, *Variety*, *Value*, and *Veracity*) characterizing Big Data [9].

Volume: It is amount of data being generated is very high and is related to the urban public lighting plants and their energy consumption status.

Velocity: It is the speed at which data are being generated, collected, retrieved, and processed and depends on the type of analyses to be conducted. Some types of analyses may require short term periods (e.g., real time, every minute, hourly), others may require a time window (e.g. weekly, monthly, yearly) for data observation.

Variety: PELL data include *static data* related to specific characteristics of the lighting plants – including the geographic position/coordinates, points of delivery (POD), electrical panels, associated luminary equipment (e.g., lanterns, street armors, floodlights, street furniture, etc.), the type of area (vehicular traffic, pedestrian circulation and cycle circulation), and road type (classified according to UNI EN 11248 [3] and evaluated according to UNI EN 13201-2[4] –, and *dynamic data* that are all the electric measurements provided by utilities and collected at the level of the electrical panel and aggregated at POD level.

Value: Gathered data are useful for diagnostics and benchmarking, such as evaluation of lighting performance with the verification of the luminaires efficiency based on the road context where are located, energy requalification simulations of historical lighting plants and their economic/financial evaluations, visual analysis across multiple spatial and time scales, energy consumption estimation, etc.

Veracity: It refers to the quality and accuracy of data. The management of multiple data sources from several municipalities and different lighting points (e.g. modern LED modules and old incandescence lamp) could lead to lack of reliability in some data sources, moreover, the presence of noise in the network communication are all causes of uncertainty and imprecision in the PELL data.

III. Architecting the PELL SCP

Figure 1 shows (using a free-style notation) the big data-driven software architecture for the PELL SCP. It is based on a typical data pipeline architecture style for big data systems that need to process data in near real-time [9] [14], and relies on Apache open-source big data frameworks as main technology for large-scale data processing and analytics. The PELL SCP is composed of three data layers: i) *Data sources and ingestion layer* (ii) *Data integration and processing layer*, and (iii) *Data presentation layer*.

[2]https://geodati.gov.it/geoportale/notizie/403-specifiche-pell-ip-online-la-versione-2-0/

[3]http://store.uni.com/catalogo/uni-11248-2016

[4]http://store.uni.com/catalogo/uni-en-13201-2-2016

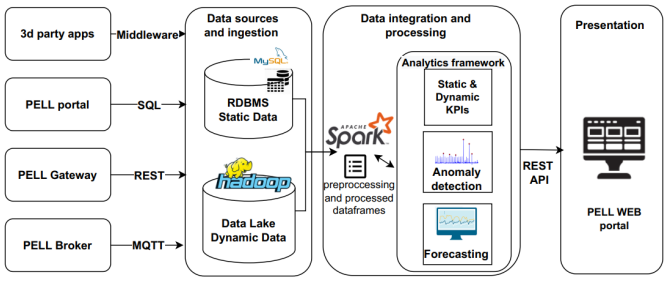


Fig. 1: PELL SCP with analytics features.

In the following, we zoom in on each layer.

A. Data sources and ingestion layer

Data ingestion is carried out through heterogeneous data sources (3d party apps, PELL Portal, PELL Gateway, and PELL Broker). Two different persistence systems are used depending on the type of data. A relational database is used for static data characterizing the lighting plants in terms of its topology, the registry municipality, technical characteristics of the entities involved and other context info such as lighting control policies and annual expenses. An unstructured persistence system (a data lake based on the Apache Hadoop ecosystem) is used instead for dynamic data, i.e. all the electrical measurements provided by utilities and acquired from the smart meters installed into the electrical panels (EP), points of delivery (POD), and lighting spots. This dynamic data is retrieved through a MQTT broker[5] and represented in the JSON (JavaScript Object Notation) format according to a specific ontology-based open data format of ENEA, called *Urban Dataset* (UD) [6], for exchanging data in an interoperable way so avoiding closed proprietary data representations of different vertical platforms of utility companies and municipalities.

The UML sequence diagram shown in Figure 2 details the interactions among the PELL broker and other components involved in transforming and persisting data from an utility solution. In particular:

- PELL Broker is the component that provides all the communication channels through appropriate dedicated topics;
- PELL Bridge provides the interaction of the PELL broker with the Smart City Platform and the data lake using the UrbanDataset Gateway for the former and the HDFS socket communication for the latter;
- MQTT Gateway provides REST API for handling the authentication and permission access, and is used by PELL Broker whenever a new request is performed;
- UD Gateway enables the interaction between PELL Bridge and the SCP through REST API serving the authentication and the sending phases;

[5]MQTT stands for Message Queuing Telemetry Transport. It is a lightweight publish/subscribe messaging protocol.

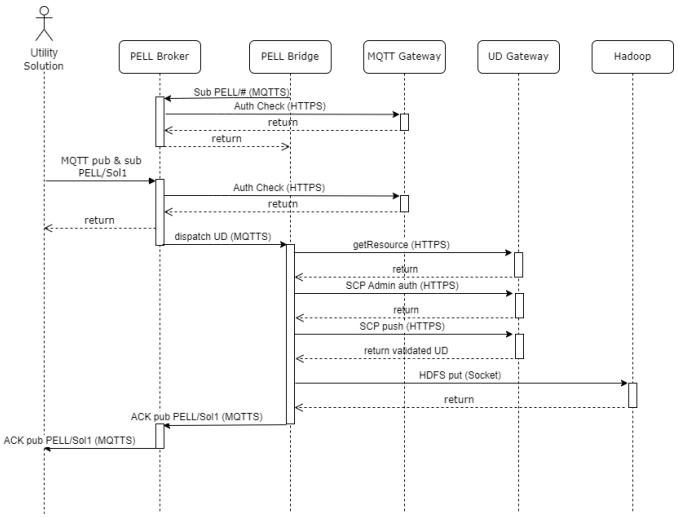


Fig. 2: PELL broker

- Hadoop is the principal component of the big data platform and provides virtualized filesystem where the data ingested is stored.

At the beginning of the process, the PELL Bridge connects in subscribe mode to all the topics with keywords *PELL*, PELL Broker checks the permissions through the MQTT Gateway and the process is concluded. When a utility solution wants to establish a connection it connects to a specific topic of the PELL channel in subscribe (e.g. *PELL/Solution1*). After the permission is checked, solution sends the data package trough the MQTT publish protocol. In order to dispath the package to both SCP and Hadoop, PELL Bridge requests the required resourceID from SCP and validates the format and the content of the package. Once the package is validated, it is sent to SCP trough the Http REST API, and towards Hadoop trough HDFS Socket protocol. Packages sent to Hadoop are stored in their original format (JSON) in the first place, but after the data processing described in subsection III-B the data format becomes Delta Lake [1], an open source software that extends Parquet data files with a file-based transaction log for ACID transactions and scalable metadata handling.

B. Data integration and processing layer

The data integration phase is the process where static and dynamic data are merged together in order to carry out different types of analytics to gain more insights and drive decision making. These include *descriptive analytics*, which by means of KPIs about energy consumption tells us what has already happened through proper measures of the plant performance; *diagnostic analytics* such as anomaly detection; and *predictive analytics*, which through ML models forecast energy consumption in the future. We rely on Apache Spark as main processing framework to perform large-scale data transformations and analyses via dataframes. Spark's underlying execution engine is used to schedule and dispatch analytical tasks and to coordinate the overall input and output operations.

In the following we provide an overview of each type of data analytics we support and put them in context.

a) KPI calculation framework: This framework considers three different types of KPIs according to the context they refer to. *Static* KPIs refer entirely to static data information, e.g. geometric KPI indicates whether the installed power in the lighting points referring to a specific area of the street is compliant to italian Green Public Procurement (GPP) criteria, taking into account the class assigned to the street according to UNI EN 11248. *Dynamic* KPIs refer entirely to dynamic data information, e.g. number of outliers in power consumption per hour. *Hybrid* KPIs refer to both static and dynamic data, e.g. maximum daily consumption deviation between measured energy absorbed by smart meters and the theoretical one according to the operating conditions declared in the lighting plant technical information.

Within the PELL SCP we implemented different types of KPIs. More details on the formulation and implementation of these KPIs are available in [3]. Table I reports an extract of a dataframe of measures for the KPI *Daily Energy Consumption Deviation* (DECD) grouping data by point of delivery (*PODID*) and electrical panel (*ElectricalPanelID*). DECD is an indicator that compares the measured daily consumption and the theoretical maximum consumption of the system operating at maximum power. Such comparison expresses how much the real consumption differs from the expected one, indicating possible malfunctions or abuses.

TABLE I: Example of dataframe measures for the KPI DECD

dateFMT	PODID	ElectricPanelID	DECD
2020-09-25	IT001E04172906	QEID2	1.02
2020-09-25	UVAX	UVAXPANELID	1.05
2020-09-26	IT001E04172906	QEID1	1.33
2020-09-26	IT001E04172906	QEID2	1.02
2020-09-26	UVAX	UVAXPANELID	1.1
2020-09-26	IT000000000ID2	IT000000000ID2	0.86
2020-09-27	UVAX	UVAXPANELID	1.09
2020-09-28	UVAX	UVAXPANELID	1.06
2020-09-29	UVAX	UVAXPANELID	1.06
2020-09-30	UVAX	UVAXPANELID	1.06
2020-10-01	UVAX	UVAXPANELID	0.98
2020-10-02	UVAX	UVAXPANELID	0.97
2020-10-03	UVAX	UVAXPANELID	1.04
2020-10-04	UVAX	UVAXPANELID	1.06

b) Anomaly detection: We have been developing an anomaly detection framework for the PELL street lighting data. By anomaly in this context, we mean abnormal or unusual behaviour or activity pattern that does not conform to the expected or normal energy consumption. These anomalies can be due to errors in the instruments, human errors, malicious activities or miscalculation of missing values, sampling errors, etc. Our primary goal was to detect anomalies for time-series electricity consumption data of the PELL SCP by using unsupervised clustering techniques, as no labelled and training data are available.

The proposed anomaly detection framework is a sequence of three main modules: *data preprocessing*, *clustering model*

formulation, and *model evaluation*. The first module is responsible for preparing the data for further processing and analysis. The second module is responsible for selecting features and for building the clustering model. The last module carries out the anomaly detection for the PELL electricity consumption data. Figure 3 shows a snippet example of the PELL street lighting dataframe obtained after the pre-processing steps. It contains the time stamps indicating the start and the end of the time interval, respectively, the identifier of the point of delivery, the identifier of the electric panel, and amount of energy consumed in the specified time interval (*ActiveEnergy*).

start_time	end_time	PODID	ElectricalPanelID	ActiveEnergy
2020-10-19 00:00:00	2020-10-19 00:15:00	IT012345678901	IT012345678901	0.1739
2020-10-19 00:15:00	2020-10-19 00:30:00	IT012345678901	IT012345678901	0.1737
2020-10-19 00:30:00	2020-10-19 00:45:00	IT012345678901	IT012345678901	0.1733
2020-10-19 00:45:00	2020-10-19 01:00:00	IT012345678901	IT012345678901	0.1735
...	...	...	...	...

Fig. 3: Dataframe after preprocessing

Because of the characteristics of the PELL street lighting model, we considered *point anomalies* and *collective anomalies* [8]. A point anomaly is the most common form of anomaly and it is when a data point significantly deviates from the rest of the data. In the context of electricity consumption, a point anomaly could be, for example, a faulty instrument or meter reporting an hour's consumption measure that is higher than usual. A collective anomaly is when a collection or sequence of data points deviates from rest of the data. An example of collective anomaly is a power outage causing the energy consumption to drop and remain unavailable for several consecutive hours. Figure 4 shows examples of energy consumption series for the PELL data without and with point and collective anomalies, respectively. These anomalies are introduced by following the domain-specific scenarios, and we then utilized unsupervised learning-based techniques to detect these synthetic anomalies.

We have explored different clustering methods, and in particular one prominent one is the well-known unsupervised clustering algorithm Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [13]. DBSCAN requires two input parameters: the minimum number of points (the threshold *MinPts*) clustered together for a region to be considered dense, and the neighborhood radius *eps* (ϵ), a distance measure used to locate the points in the neighborhood of any point. Figure 5 shows the anomaly distribution detected by DBSCAN by using *eps=0.35* and *MinPts=10* (chosen with appropriate heuristics) on a data time series collected by smart meters of electrical panels located at the Smart District of ENEA in Casaccia (Rome). We have used electric consumption measures which were collected from July 2020 to December 2020 with the time granularity of 15 minutes.

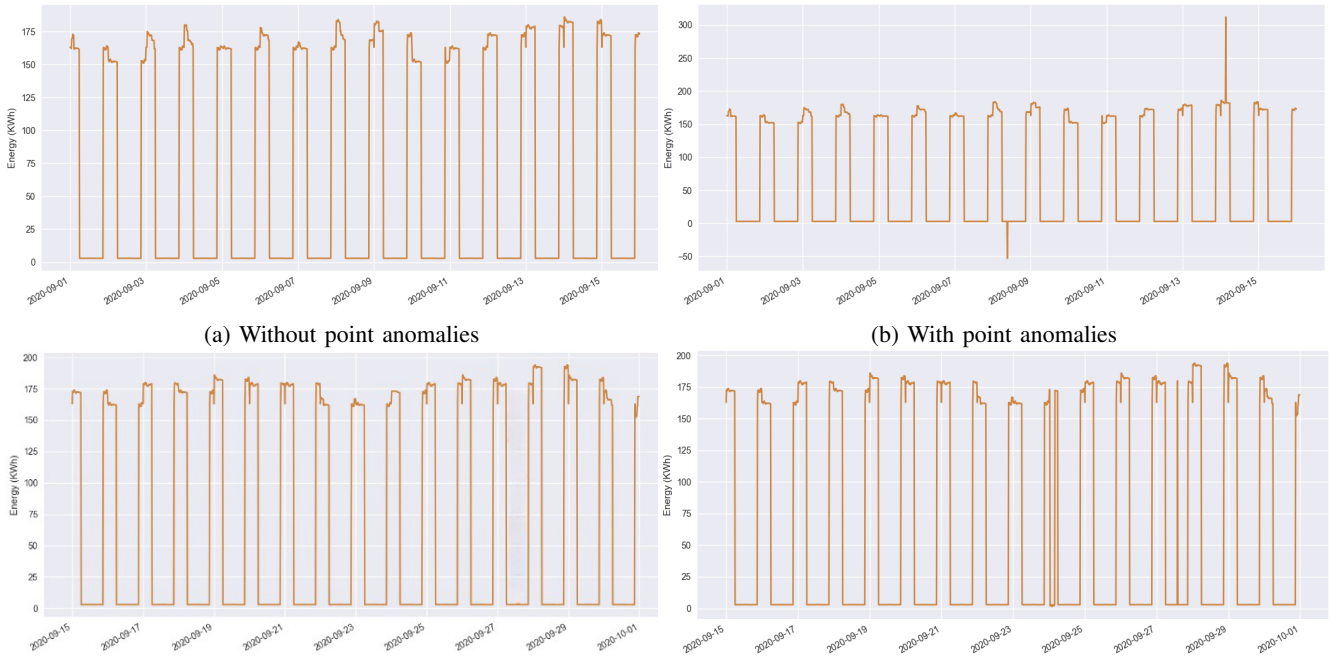
Another data set with known synthetic anomalies helped us to evaluate the accuracy of the clustering algorithms with different parameter settings. As an example, Figure 6 plots the accuracy results of DBSCAN on the chosen dataset in term of the well-known metrics precision, recall, and F1-score [27], using different values of the parameter eps.

c) Energy consumption forecasting: We have developed an energy consumption forecasting module as part of the analytical framework of the PELL SCP. The primary objective of this module is to forecast energy consumption by using ML forecasting models. The proposed forecasting workflow (see Figure 7) consists of three main stages: *data preprocessing*, *data splitting*, and *forecasting model building and evaluation*.

The first module is responsible for preparing the data for further processing and analysis; feature selection is also the responsibility of the first module. In the second module data is divided into training set, validation set and testing set. In the last module, PELL energy consumption forecasting is performed. As an example of empirical setting, for a 3-months PELL data set of the Smart District of ENEA in Casaccia (Rome) (from September 2020 to November 2020), we analyzed JSON files containing 96 values (4 values per hour for 24 hours per day) for a total of 8736 data points of three months consumption. The consumption is measured with the time granularity of 15 minutes. The training set contains 45 days data, the validation set contains 15 days, and testing is performed on one-month data.

We have been exploring different forecasting models, as implemented in the Scikit-learn open source machine learning library: linear regression (LR), support vector regression (SVR), decision tree (DT), random forest (RF), k-nearest neighbours (KNN), and in particular one prominent one is the well-known forecasting model multilayer perceptron (MLP) [18]. MLP consists of two phases: 1) the training phase of the model in which weights of the connections between neurons are optimized by employing various algorithms, for example, backpropagation combined with stochastic gradient descent (SGD) [5] to minimize a loss function, and 2) the testing phase in which the trained MLP model is applied on the unseen data to fulfill the desire objective.

To evaluate the forecasting performance of the MLP model, we consider two different versions of the model. The only difference is which features are used to train the model. The first version contains the historic consumption (HC) with autoregressive AR(1) value, and the second version of the model also contains time information (hour of the day). In order to determine the optimal parameter settings for the MLP model, we employed grid search. Table II presents the parameter settings used for the MLP model in the empirical evaluation for the 3-months PELL data set of Casaccia plant, while Table III reports the forecasting accuracy results of the two MLP models in terms of forecast errors usually considered as accuracy metrics in time series forecasting, namely root mean square errors (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Correlation coefficient (R). Figure 8 and Figure 9 shows the comparison of actual and predicted energy consumption values of the model for both versions.



(a) Without point anomalies

(b) With point anomalies

(c) Without collective anomalies

(d) Width collective anomalies

Fig. 4: Examples of anomalies in the PELL data series

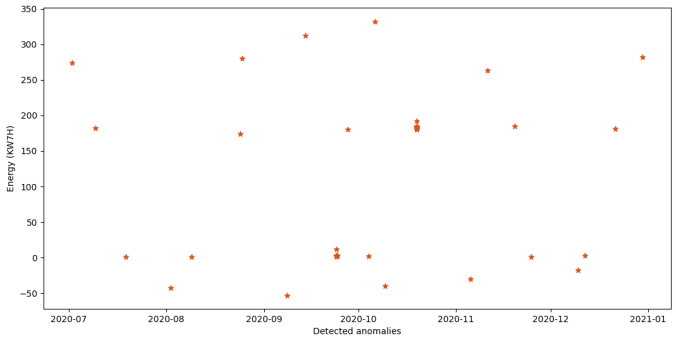


Fig. 5: Example of distribution of anomalies correctly detected by DBSCAN

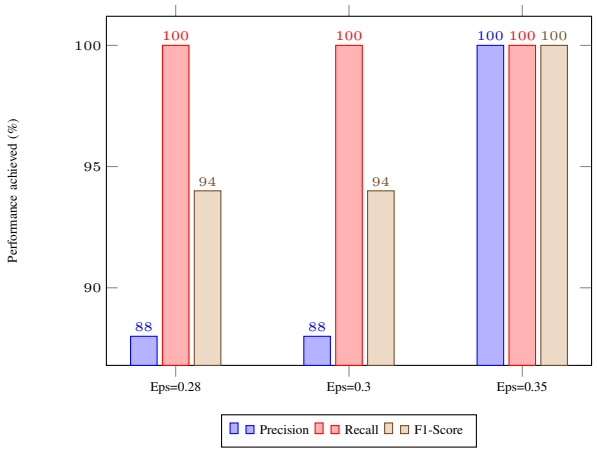


Fig. 6: DBSCAN accuracy with different *eps* values

TABLE II: Example of parameter tuning of the MLP model.

Parameters	Used value
Hidden Layers	2
Activation Function	Relu
Optimizer	Adam
Batch size	16
Epochs	150

TABLE III: MLP forecasting errors.

Model	Input Features	RMSE	MAE	MAPE	R
MLP	HC	0.0121	0.0051	0.3729	0.9889
MLP	HC & time (hour)	0.0127	0.0043	0.3696	0.9856

C. Data presentation layer

This is the final layer of the PELL SCP. It refers to represent knowledge, which is usually exposed as API services and delivered through intuitively web dashboards or via CLI commands. By tracking multiple data sources, this software layer allows monitoring and analysis of KPIs and analytical queries' results.

Figure 10 shows an example of dashboard provided to the user to visualize time series of electric consumption, navigate lighting plant elements on the map, visualize custom widget representing static or dynamic KPIs and so on. Since every user has a corresponding role and permission, the dashboard is configured such that data is accessed and/or aggregated according to corresponding permissions.

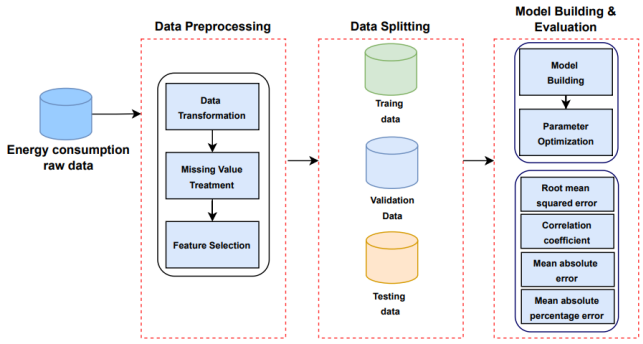


Fig. 7: Stages of the PELL energy consumption forecasting

IV. Discussion and lesson Learned

In this section, we summarize a couple of aspects in our experience with the PELL SCP. In addition to the study of the peculiarities of the public street lighting domain and the PELL data model, the development and evaluation of the algorithms adopted by the analytics framework required us a learning curve for gaining a wide spectrum of multi-disciplinary skills. In particular, we focus here on the technical challenges in learning and using big data technologies and data analysis libraries.

From a IT operations engineer and system administrator, it is required to know how to set up the runtime environment, i.e. how to install the Apache Hadoop and Spark frameworks for big data architectures in clustered mode. Each framework contains an extensive ecosystem of open-source technologies that prepare, process, manage and analyze big data. The seamless connection of all these software solutions through a sequence of installation and configuration steps on a target machine requires a certain amount of time and practical experience. Maintenance is also an important factor, as technology advances and new software versions are released, a careful check on the compatibility issue is required in order to avoid the jeopardize of the environment.

From a software engineer/system architect point of view, a challenge is how to design a big data pipeline architecture and include all frameworks necessary to incorporate the data injection, preparation and processing activities, distinguishing among requirements and roles that these frameworks represent. Both Apache Hadoop and Spark allow building a data analysis infrastructure with a limited budget since they are open source. According to the state of the art and practice in big data computation, Hadoop and Spark are the most prominent tools enabling a scalable and higly performing pipeline. In some contexts and scenarios, we have seen Hadoop and Spark both used to solve vast and intricate data problems with some roles' overlapping. Instead, in our architecture solution, as explained in Section III, we employ them in two different stages with different roles. In general, Hadoop is most effective for scenarios that involve batch processing with tasks that exploit disk data storage, while Spark allows for faster in-memory processing. Some other emerging database technologies can be considered in the future upgrades of the platform as valid alternatives, especially when the data is focused mainly on time series, as in our case. For example, TimescaleDB is a Postgres-based DBMS particularly performing on time series as well as MongoDB that starting from version 6 enables an high performing querying on time series semi-structured data.

From a data analyst point of view, python programming and dataframe processing skills are required. Moreover, the built-in support of libraries for data processing offered by Hadoop and Spark frameworks could be not enough for more complex analysis. So, external Python libraries for data analytics, such as pandas and scikit-learn, could be necessary (as in our case) to learn and apply in a seamless manner. The main focus should be put on applying the stack of libraries of Spark for large scale data processing, and Python ML library (scikit-learn) for the implementation of clustering and forecasting algorithms. We used PySpark to transform the data set from the JSON file to Spark DataFrame for big data processing. After data processing we transformed Spark DataFrame to the pandas DataFrame because MLib Apache Spark's ML library does not support the implementation of advanced clustering algorithms like DBSCAN and OPTICS. After this conversion, we exploited scikit-learn for the implementation of clustering algorithms.

Finally, to build maintainable, exposable and scalable data services, knowledge of principles for developing small (micro-)services and REST controllers in Python is highly suggested.

V. Related work

Big data analytics is being actively used in the development of smart city software solutions. Here, we first discuss in Section V-A, the most notable studies from the literature that inspired us as smart city software architectures adopting big data analytics. In Section V-B, we discuss the existing works for anomaly detection using clustering techniques. Then, in Section V-C, we present some works in the literature related to energy consumption forecasting using machine learning.

A. Smart city software architectures adopting big data analytics

Azzam et al. [4] proposed the architecture of the *CitySPIN* project for the development of smart services. The platform of the *CitySPIN* is assisted with methods and techniques, which are based on Semantic WEB and Linked Data technologies for the acquisition and integration of heterogeneous data of different formats (structured, unstructured, and semi-structured), including open data and social data. The *CitySPIN* project is based on a three-layered architecture: 1) back-end layer, which is responsible for data collection, pre-processing and data integration, 2) service layer, which provides the services of analysis by applying queries and prediction model. The prediction model is based on machine learning algorithms to facilitate the prediction by using historical data that can assist in decision making, and 3) front end layer or presentation layer, which facilitate the users to interact with the system and perform different kinds of analysis of their need.

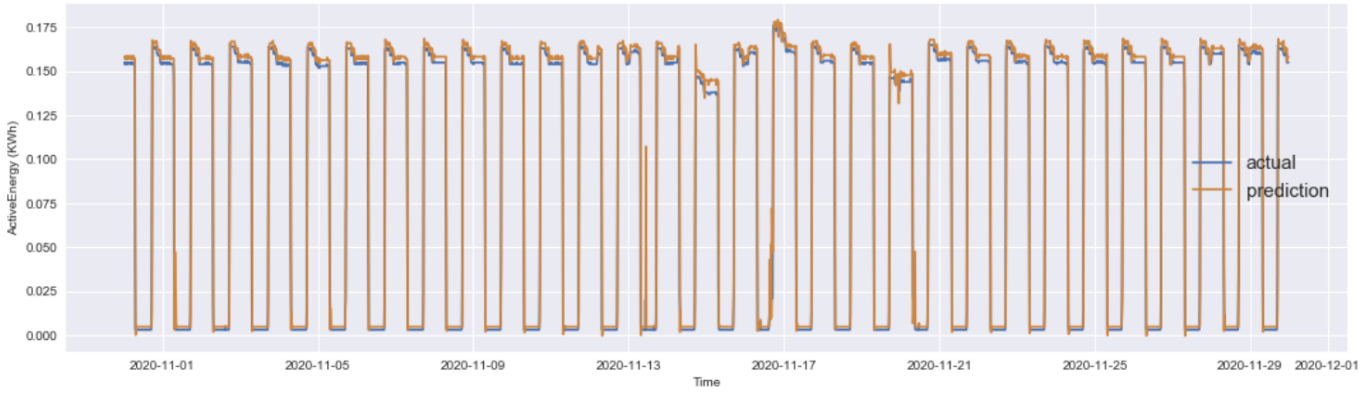


Fig. 8: Actual and predicted energy consumption forecasting of MLP with HC feature.

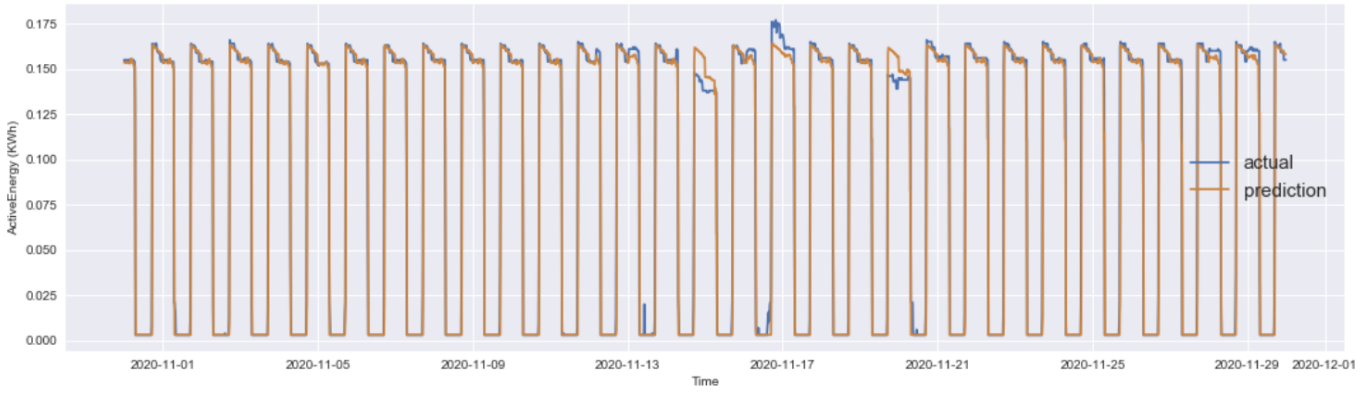


Fig. 9: Actual and predicted energy consumption forecasting of MLP with HC and time (hour of the day) feature.

In [21], a framework called *CityPulse* is presented for the development of smart city services by enabling the integration of heterogeneous data streams, interoperability, (near-) real-time data analytics, and applications development in a scalable way. The CityPulse framework is composed of a powerful data analytics module, which is empowered to perform intelligent data aggregation, quality assessment, event detection, contextual filtering, and decision support. All the components of the *CityPulse* have been developed as reusable entities and application development is facilitated by open APIs.

Pedro et al. [17] proposed a project called *CityAction* to facilitate the city managers to take actions/decisions on the bases of real-time city data. The main objective of the project is to support the design and development of an integrated platform that has the ability to combine city data coming from different sources with heterogeneous devices and perform intelligent data analysis. The architecture of the *CityAction* is based on four independent layers: 1) Device layer, in which IoT sensors, actuators and communication gateways correspond to different vertical systems, 2) M2M Connectivity layer, which is responsible for the devices interconnection to the internet, 3) Middleware layer, this layer has the responsibility to integrate several blocks like data broker, monetization, data management and analytics, vertical management M2M management, and API management, 4) Application layer that also has an ability to incorporate the open data to enrich the application portfolio. Mohamed et al. [10] came up with another approach to transform big data into a smart data. In this study, they introduced a system called *CityPro*. The architecture of the *CityPro* is discussed for surveillance system. In the architecture of the *CityPro*, a federated star-schema is used in the storage repository and repository only store the summarized data instead of huge amount of data.

In another study [12], an approach is discussed for the development of next-generation big data applications. They proposed *CAPIM* (Context-Aware Platform using Integrated Mobile services), a platform designed to automate the process of collecting and aggregating the context information on a large scale. An intelligent transportation system is developed by using *CAPIM*, which helps the user and city managers to understand the traffic problems of their city. In another interesting study, Paula et al. [26] proposed a simple and scalable *hut* architecture to extract the valuable historical insights and actionable knowledge from IoT data streams. The developed *hut* architecture support both historical as well as real-time data analysis. It was applied on two real smart city contexts, namely transportation and energy management. The implementation of the *hut* architecture is based on open source solutions and can be replaced or customized according to the need. In another work [23], a system, called *CrowdNav*,

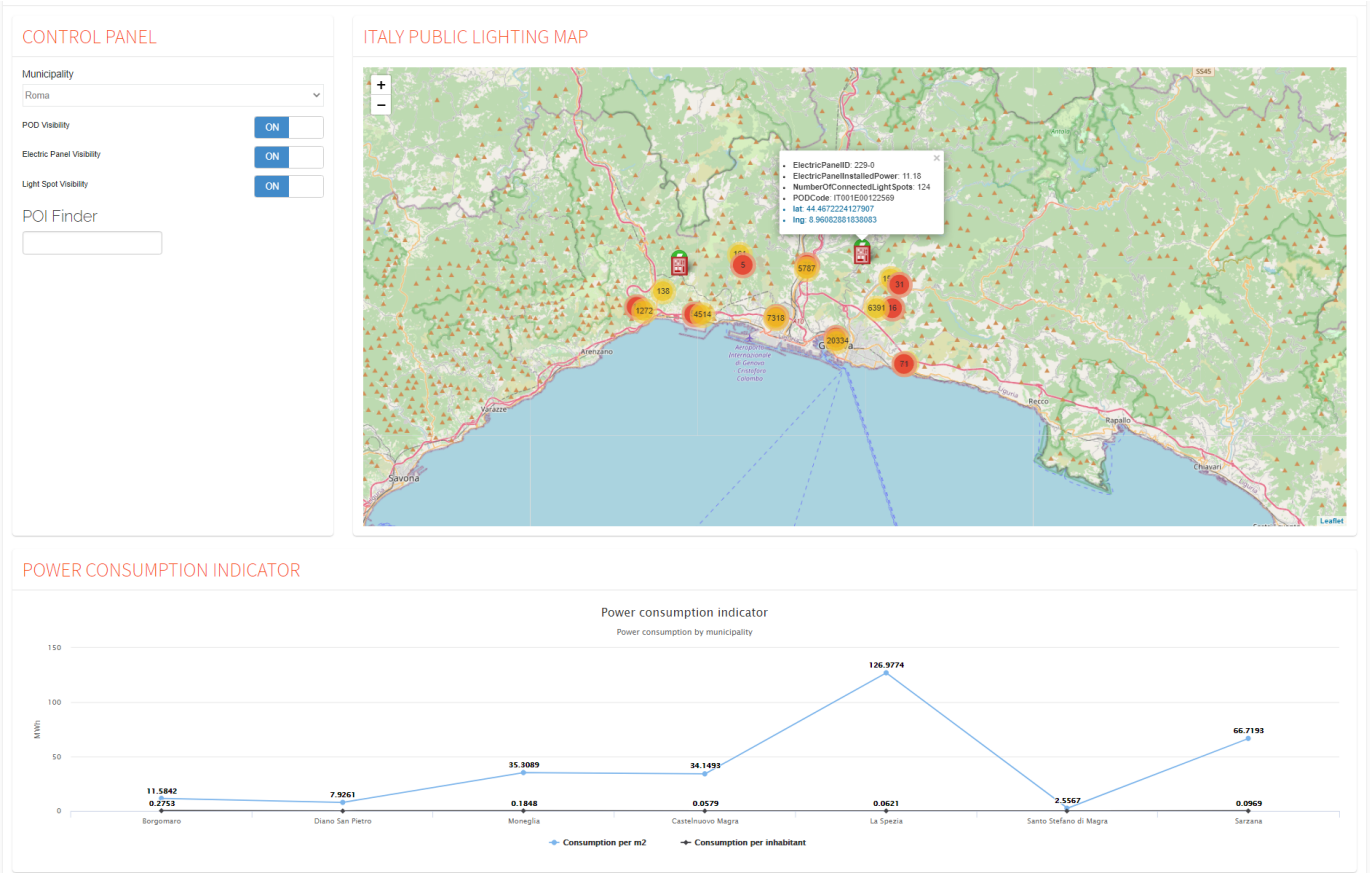


Fig. 10: PELL dashboard for power consumption

is proposed to enable self-adaptation in a complex large scale software-intensive distributed system by using big data analytics.

B. Clustering-based techniques for Anomalies Detection

Celik et al. [7] performed a comparative analysis between DBSCAN and a statistical method to discover the anomalies in temperature data. According to the authors, DBSCAN is robust in discovering anomalies as compared to statistical model, because statistical model can only detect the anomalous data points that are below or above the threshold values but it is not able to detect those anomalies that are less frequent and exist in the threshold range. In [16], three well known clustering algorithms including K-means, DBSCAN, and OPTICS are compared based on the performance measures such as accuracy, outliers formation and cluster size prediction. In the light of authors findings, K-means produced good quality clusters when apply on large scale data, but K-means is sensitive to outliers and also does not perform well with clusters of arbitrary shapes. The highlighted deficiencies of the K-means can be overcome by using DBSCAN, and OPTICS. DBSCAN has an ability to form the clusters of arbitrary shape and size. It also has an ability to determine which data points should be classified as noise point or outliers, in comparison to other clustering models it is very fast algorithm. However, the drawback of DBSCAN algorithm is that when clusters with different densities are located to each other then DBSCAN will not be able to distinguish them. These deficiencies are addressed by OPTICS. It ensures good quality clustering by giving the priority to high-density clusters over low density clusters [15].

Another DBSCAN based anomaly detection model was proposed by Jith et al. [22]. The objective of their work is to detect the anomalies from traffic data set, in which a path is labeled as anomaly if it does not match with the pre-trained model. In another study [19], the authors proposed a K-means clustering based approach to detect the anomalies from traffic data set. In [25], the authors employed K-means algorithm to detect the anomalies from call detail records (CDR) data set. In order to evaluate the effectiveness of the proposed anomaly detection approach, the authors train a neural network model on anomaly and anomaly-free data. During the model training, they observed the effects of anomalous activities and also noted the mean square error of anomaly and anomaly free data. In [28], the authors proposed a K-means clustering algorithm based approach to detect the anomalies in software measurement data. In [29], an unsupervised learning based anomaly detection approach from system's log files is presented by using the OPTICS algorithm.

C. Energy consumption forecasting using ML techniques

Pedro et al. [2] utilized regularized machine learning models i.e. Lasso, Lars, Lasso Lars, Ridge, Elastic Net, and Random Forest (RF), with the intention to forecast Brazilian power power electricity consumption for short and medium terms. They performed predictions for 6 different forecast horizon: 1, 7, 15, 30, 60, and 90 days ahead. They divided the horizons into three groups such as (1) very short-term forecast group (VSTFG) that includes 1 and 7 days, (2) short-term forecast group (STFG) containing 15 and 30 days and (3) medium-term forecast group (MTFG) including 60 and 90 days. Beside the power electricity consumption, the authors also used calendar variables, weather variables, price of electrical energy, and several economic variables. After experimental evaluation, they concluded that machine learning methods, especially RF and Lasso Lars more accurate results for all horizons.

Jihoon et al. [18] conducted a comparative analysis of diverse ANN models for short-term load forecasting (STLF). They constructed different ANN models by tuning two hyperparameters such as number of hidden layers and activation functions. They considered rectified linear unit (ReLU), leaky rectified linear unit (LReLU), parametric rectified linear unit (PReLU), exponential linear unit (ELU), scaled exponential linear unit (SELU) as activation functions, and the number of hidden layers from 1 to 10. In order to compare the prediction performance with two hyperparameters for the STLF model, the authors used electric load data collected from five different types of buildings for 2 years, and two performance metrics such as coefficient of variation of the root mean square error (CVRMSE) and MAPE. They concluded that SELU-based model with five hidden layers produced better average performance than other ANN-based models for shor-term load forecasting. In [24], the authors proposed kCNN-LSTM, a deep learning framework for robust and reliable building energy consumption. kCNN-LSTM uses (i) k-means clustering to perform cluster analysis for trend characterization; (ii) Convolutional Neural Networks (CNN) to extract complex energy related features with non-linear interactions; and (iii) Long Short Term Memory (LSTM) neural networks to handle long-term dependencies and model temporal information in energy consumption data. The effectiveness of the proposed kCNN-LSTM framework was demonstrated by using a real time building energy consumption data acquired from a four-storeyed building in IIT-Bombay, India. The ability to learn the spatio-temporal dependencies in the energy consumption data makes the kCNN-LSTM a suitable deep learning model for energy consumption forecasting.

Pavlicko et al. [20] utilized and compared different forecasting models to predict the maximum hourly electricity consumption per day. They categorized the used models into two groups. The first group is based on the transverse set of Grey models and Nonlinear Grey Bernoulli models. The second group is based on a multi-layer feed-forward back-propagation network. Furthermore, they also proposed a new potential hybrid model by combining these approaches, which is used to forecast the maximum hourly electricity consumption per day. Experimental results show that the hybrid model offered the best results according to the used performance metrics.

Divina et al. [11] performed a comparative analysis to analyze the performance of statistical and ML models in predicting energy consumption in non-residential smart buildings. Electricity energy consumption data was collected from thirteen smart buildings located on a university campus in Spain. The authors demonstrated that highly accurate prediction accuracy can be reached in favour of strategies based on ML approaches and the historical window's optimal size optimization.

From this preliminary state-of-the-art review, we found there are no big data-driven software architectures for public street lighting intended to collect, represent, control, predict, and possibly optimize the behavior of public street lighting plants. To fulfill this gap, we designed and implemented one for the analysis of energy consumption data in the context of public street lighting. We have designed and prototyped different types of analytics services as part of the PELL SCP.

VI. Conclusion and future work

In this paper, we have presented the main building blocks of the PELL SCP. It realizes a big data-driven software architecture for processing and managing energy consumption in public street lighting. We have also provided some insights on how we have been defining and implementing an analytical framework based on KPIs and ML algorithms for measuring the performance of the electrical energy plants, discovering consumption patterns and anomalies, and predicting energy consumptions.

In the future, through predictive models we would like to conceive also a sort of *prescriptive analytics* for suggesting energy efficiency strategies to eventually plan and actuate by the city/utility managers.

Currently the analytical functions are carried out in a batch mode, i.e. the data processing is executed by scheduling periodic tasks. From a software engineer perspective, we also aim in the future at re-engineering the exposition layer of the analytical framework to the presentation layer by re-engineering it in terms of a micro-service architecture layer. In this way, the analytical processing could be served also on-demand by the user by specifying appropriate parameters of the processing request (municipality ID, PODID, observation period, time granularity, etc.).

Security also is an aspect that should be targeted more deeply in future, as the exposure increases, probability of being targeted by malicious attacks increases as well. Enhancing security both from sysadmin and software design point of views is paramount in our opinion.

Acknowledgment

ENEA's activities were carried out within the Project 1.7 *Technologies for the efficient penetration of the electric vector into end uses* within the PTR 22-24 *Research on Electrical Systems*.

M. Ali acknowledges the support of the PhD scholarship in Engineering and applied sciences co-financed for round XXXV by the Lombardy Region (Italy) under the collaboration agreement between the Lombardy Region and the Italian National Agency ENEA for New Technologies, Energy and Sustainable Economic Development.

REFERENCES

[1] Delta Lake. https://delta.io

[2] Albuquerque, P.C., Cajueiro, D.O., Rossi, M.D.: Machine learning models for forecasting power electricity consumption using a high dimensional dataset. Expert Systems with Applications **187**, 115917 (2022)

[3] Ali, M., Scandurra, P., Moretti, F., Blaso, L., Leccisi, M., Leccese, F.: From big data to smart data-centric software architectures for city analytics: the case of the pell smart city platform. In: 2021 IEEE International Conference on Smart Data Services (SMDS). pp. 95–104. IEEE (2021)

[4] Azzam, A., Aryan, P.R., Cecconi, A., Di Ciccio, C., Ekaputra, F.J., Fernández, J., Karampatakis, S., Kiesling, E., Musil, A., Sabou, M., et al.: The cityspin platform: A cpss environment for city-wide infrastructures (2019)

[5] Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. Advances in neural information processing systems **20** (2007)

[6] Brutti, A., Sabbata, P.D., Frascella, A., Gessa, N., Ianniello, R., Novelli, C., Pizzuti, S., Ponti, G.: Smart city platform specification: A modular approach to achieve interoperability in smart cities. In: Cicirelli, F., Guerrieri, A., Mastroianni, C., Spezzano, G., Vinci, A. (eds.) The Internet of Things for Smart Urban Ecosystems, pp. 25–50. Springer (2019). https://doi.org/10.1007/978-3-319-96550-5_2, https://doi.org/10.1007/978-3-319-96550-5_2

[7] Çelik, M., Dadaşer-Çelik, F., Dokuz, A.Ş.: Anomaly detection in temperature data using dbscan algorithm. In: 2011 international symposium on innovations in intelligent systems and applications. pp. 91–95. IEEE (2011)

[8] Chandola, V., Banerjee, A., Kumar, V.: Survey of anomaly detection. ACM Computing Survey (CSUR) **41**(3), 1–72 (2009)

[9] Chen, C.P., Zhang, C.Y.: Data-intensive applications, challenges, techniques and technologies: A survey on big data. Information sciences **275**, 314–347 (2014)

[10] Dbouk, M., Hakim, M., Sbeity, I.: Citypro: From big-data to intelligent-data; a smart approach. In: BDCSIntell. pp. 100–106 (2018)

[11] Divina, F., Garcia Torres, M., Goméz Vela, F.A., Vazquez Noguera, J.L.: A comparative study of time series forecasting methods for short term electric energy consumption prediction in smart buildings. Energies **12**(10), 1934 (2019)

[12] Dobre, C., Xhafa, F.: Intelligent services for big data science. Future generation computer systems **37**, 267–281 (2014)

[13] Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96, pp. 226–231 (1996)

[14] Habibzadeh, H., Kaptan, C., Soyata, T., Kantarci, B., Boukerche, A.: Smart city system design: A comprehensive study of the application and data planes. ACM Computing Surveys (CSUR) **52**(2), 1–38 (2019)

[15] Hurst, W., Montañez, C.A.C., Shone, N.: Time-pattern profiling from smart meter data to detect outliers in energy consumption. IoT **1**(1), 92–108 (2020)

[16] Kanagala, H.K., Krishnaiah, V.J.R.: A comparative study of k-means, dbscan and optics. In: 2016 International Conference on Computer Communication and Informatics (ICCCI). pp. 1–6. IEEE (2016)

[17] Martins, P., Albuquerque, D., Wanzeller, C., Caldeira, F., Tomé, P., Sá, F.: Cityaction a smart-city platform architecture. In: Future of Information and Communication Conference. pp. 217–236. Springer (2019)

[18] Moon, J., Park, S., Rho, S., Hwang, E.: A comparative analysis of artificial neural network architectures for building energy consumption forecasting. International Journal of Distributed Sensor Networks **15**(9), 1550147719877616 (2019)

[19] Münz, G., Li, S., Carle, G.: Traffic anomaly detection using k-means clustering. In: GI/ITG Workshop MMBnet. pp. 13–14 (2007)

[20] Pavlicko, M., Vojteková, M., Blažeková, O.: Forecasting of electrical energy consumption in slovakia. Mathematics **10**(4), 577 (2022)

[21] Puiu, D., Barnaghi, P., Toenjes, R., Kümper, D., Ali, M.I., Mileo, A., Parreira, J.X., Fischer, M., Kolozali, S., Farajidavar, N., et al.: Citypulse: Large scale data analytics framework for smart cities. IEEE Access **4**, 1086–1108 (2016)

[22] Ranjith, R., Athanesious, J.J., Vaidehi, V.: Anomaly detection using dbscan clustering technique for traffic video surveillance. In: 2015 Seventh International Conference on Advanced Computing (ICoAC). pp. 1–6. IEEE (2015)

[23] Schmid, S., Gerostathopoulos, I., Prehofer, C., Bures, T.: Self-adaptation based on big data analytics: a model problem and tool. In: 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). pp. 102–108. IEEE (2017)

[24] Somu, N., MR, G.R., Ramamritham, K.: A deep learning framework for building energy consumption forecast. Renewable and Sustainable Energy Reviews **137**, 110591 (2021)

[25] Sultan, K., Ali, H., Zhang, Z.: Call detail records driven anomaly detection and traffic prediction in mobile cellular networks. IEEE Access **6**, 41728–41737 (2018)

[26] Ta-Shma, P., Akbar, A., Gerson-Golan, G., Hadash, G., Carrez, F., Moessner, K.: An ingestion and analytics architecture for iot applied to smart city use cases. IEEE Internet of Things Journal **5**(2), 765–774 (2017)

[27] Tatbul, N., Lee, T.J., Zdonik, S., Alam, M., Gottschlich, J.: Precision and recall for time series. arXiv preprint arXiv:1803.03639 (2018)

[28] Yoon, K.A., Kwon, O.S., Bae, D.H.: An approach to outlier detection of software measurement data using the k-means clustering method. In: First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007). pp. 443–445. IEEE (2007)

[29] Zeufack, V., Kim, D., Seo, D., Lee, A.: An unsupervised anomaly detection framework for detecting anomalies in real time through network system's log files analysis. High-Confidence Computing **1**(2), 100030 (2021)