machines

MDPI

Article

A Digital Twin Architecture Model Applied with MLOps Techniques to Improve Short-Term Energy Consumption Prediction †

Tiago Yukio Fujii [1], Victor Takashi Hayashi [1,*], Reginaldo Arakaki [1], Wilson Vicente Ruggiero [1], Romeo Bulla, Jr. [1], Fabio Hirotsugu Hayashi [2] and Khalil Ahmad Khalil [1]

[1] Polytechnic School (EPUSP), University of São Paulo, São Paulo 05508-010, Brazil; tiago.fujii@usp.br (T.Y.F.); reginaldo.arakaki@poli.usp.br (R.A.); wilson@larc.usp.br (W.V.R.); romeo@larc.usp.br (R.B.J.); kha@usp.br (K.A.K.)

[2] Dry Educational Laboratories, Federal University of ABC (UFABC), Santo André 09210-580, Brazil; fabio.hayashi@ufabc.edu.br

* Correspondence: victor.hayashi@usp.br

† This paper is an extended version of our paper published in Fujii, T.Y.; Ruggiero, W.V.; do Amaral, H.L.; Hayashi, V.T.; Arakaki, R.; Khalil, K.A. Desafios para Aplicação de MLOps na Previsão do Consumo Energético. In Proceedings of the 2021 14th IEEE International Conference on Industry Applications (INDUSCON), Saint Paul, Brazil, 16–18 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 455–462.

Abstract: Using extensive databases and known algorithms to predict short-term energy consumption comprises most computational solutions based on artificial intelligence today. State-of-the-art approaches validate their prediction models in offline environments that disregard automation, quality monitoring, and retraining challenges present in online scenarios. The existing demand response initiatives lack personalization, thus not engaging consumers. Obtaining specific and valuable recommendations is difficult for most digital platforms due to their solution pattern: extensive database, specialized algorithms, and using profiles with similar aspects. The challenges and present personalization tactics have been researched by adopting a digital twin model. This study creates a different approach by adding structural topology to build a new category of recommendation platform using the digital twin model with real-time data collected by IoT sensors to improve machine learning methods. A residential study case with 31 IoT smart meter and smart plug devices with 19-month data (measurements performed each second) validated Digital Twin MLOps architecture for personalized demand response suggestions based on online short-term energy consumption prediction.

Keywords: MLOps; digital twin; IoT; machine learning; prediction

check for updates

Citation: Fujii, T.Y.; Hayashi, V.T.; Arakaki, R.; Ruggiero, W.V.; Bulla, R., Jr.; Hayashi, F.H.; Khalil, K.A. A Digital Twin Architecture Model Applied with MLOps Techniques to Improve Short-Term Energy Consumption Prediction. *Machines* **2022**, *10*, 23. https://doi.org/10.3390/machines10010023

Academic Editor: Xiang Li

Received: 30 October 2021
Accepted: 24 December 2021
Published: 28 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A smart grid enables bidirectional communication between utilities and consumers, which may be used to optimize energy usage by demand side management (i.e., demand-response). As increasing energy demand and peak of energy consumption are concerns for utilities, the demand side management enables an effective method to reduce costs of electricity, which in turn restrict the need for more investments in transmission and distribution infrastructure [1,2].

One example of demand side management is employing dynamic hourly energy prices to make consuming energy in peak hours more expensive. Even though demand response has the potential to reduce energy costs and foster more sustainable communities, investigating methods of change consumer behavior towards energy consumption management is an ongoing effort [2].

Future energy facilities for residential and industrial sectors should compose a consumption chain where the behavior of real-time energy usage will be enabled by digital

platforms. These aggregated data will allow analyzing consumption, seasonality, costs and planning in terms of generation, and transmission and distribution capacity [3,4]. With this, the scenario of digital data, ready to be processed by algorithms and artificial intelligence platforms, is quite consistent with product innovations and services in this area.

Smart meters adoption for energy consumption monitoring enables analysing usage habits of home appliances. Added to the direct feedback received, user-customized services such as prediction and classification of energy consumption increase their user's energy awareness and help them reduce their electricity bills [5].

Machine learning (ML) techniques for forecasting and classification of energy consumption are broadly used both academically and in the industry [6,7]. However, academic research focuses on static or offline environments, without analyzing the degradation of accuracy over time due to unexpected changes in the behavior of the time series (concept drift) [5], the sensitivity of the configuration manual of hyper parameters, and training times and prediction of the models.

Residential energy consumption has a large dependence on time of year and temperature [6,8], resulting in concept drift that is not analyzed in experiments in static environments. It is possible to use outdoor temperature data and WiFi thermostat data to improve energy consumption prediction [9], and internal building temperatures can be predicted as well [10]. In addition, although the literature presents standardized metrics for measuring the accuracy of models, there is no consensus on the use of such metrics to measure the aptitude of machine learning systems as to its operation in online environments, rendering comparisons between solutions difficult.

The convergence of digital twin and machine learning is said to improve productivity and quality in smart manufacturing scenarios [11]. Physical appliances could adapt to operational changes in real time and forecast events based on historical data by using a digital twin. However, one of the relevant challenges to build and implement digital twins is the question of how to integrate different engineering models and foster cross-domain collaboration.

This paper has addressed the following research question in order to face the challenge of modeling real-time energy consumption data: Are there computational mechanisms that enable specialized insights from customers employing prediction models? This fundamental question generates the other questions listed below:

Research Question 1. *How do we obtain intelligent real-time database containing information from each user instead of using conventional database structures with raw data?*

The first research question demands that not only raw collected data by IoT are stored and managed in the proposed solution, but its metadata must also be included to allow energy consumption forecast customization.

Research Question 2. *How do we configure Machine Learning Prediction Services for each user that would consider the challenges of real world deployment?*

This second research question shows the need for the proposed solution to consider the constraints of a real world deployment: missing data, multiple time granularity, and diverse metrics.

This paper presents a different approach adding structural topology to build a new category of recommendation platform using the digital twin model fed with real-time data collected by IoT Sensors to improve the existing machine learning approach. Residential study cases with 31 IoT smart meter and smart plug devices with data of 19 months (measurements performed each second) were used to validate Digital Twin MLOps architecture for personalized demand response suggestions based on online short-term energy consumption prediction.

Our main contributions are related to closing the gap between machine learning models used for predicting residential energy consumption and real world deployment by

presenting a solution that includes household metadata so that other systems make better use of prediction results. The results contribute to the state of the art with an approach robust to missing data with multiple time granularity.

This article is an extended version of a conference paper [12], which focused solely on MLOps tests. The text is organized as follows: Section 2 presents the related work, corresponding research gaps found in the literature, and the concepts used in our solution. The method is described in Section 3. MLOps and Digital Twin modeling results are described in Section 4, and the results analysis, comparison with related work, known limitations, and development considerations are presented in Section 5. The article is concluded in Section 6 with final thoughts and suggestions for future work.

2. Research Methodology

2.1. Research Context

Personalized recommendations concerning energy saving may be supported by specialized recommender systems. A proposal found in the literature is based on user profiling and micro-moment recommendations with a mobile user interface to foster energy saving behavior change [13,14]. The solution uses appliance-level energy consumption data collected by sensors deployed in the household to recognize micro-moments for timed recommendations. However, one shortcoming of employing user profiling with collaborative filtering is that the recommendations are not fully personalized, as they are aimed at a cluster of users and not at a specific user.

The gamified management platform application found in the literature exemplifies how gamification could be used to foster demand change based on device-level monitoring [15]. The approach was validated with four households within four months, achieving up to 30% peak period consumption. Even though it is based solely on an user dashboard (i.e., passive instead of the active method a chatbot might interact with users), it organized the platform by individual and group tasks, badges, and informative pages regarding benefits, such as CO_2 emission reductions, grid operation, and electricity bill savings.

Another work uses outside temperature prediction and smart home activity recognition models to propose a controller that concurrently considers both energy savings and comfort requirements at the same time [16]. The proposal was evaluated in four apartments, and it could achieve 5.14% Heating, Ventilation, and Air Conditioning (HVAC) energy consumption reduction over the on/off controller, while simultaneously maintaining the comfort level (i.e., maximum indoor temperature difference of 0.06 °F).

A proposal found in the literature used a digital twin to model energy providers and residences [17]. It employed a reinforcement learning algorithm to optimize smart home appliances scheduling to flatten total household energy consumption to avoid peak demands and reduce the energy bill. They used the the digital twin as a sandbox to test the optimization algorithm before enforcing it to physical devices. The solution presented 17.7% energy cost reduction for a real-life dataset.

One example application of the Digital Twin architecture is energy consumption prediction. Appliance level consumption is heterogeneous, requiring time granularity selection due to complex seasonality [18] of different house appliances. Choosing the wrong granularity might induce information loss [19] due to generalization or erroneous assumptions concerning trends and correlations with features [20].

Just as household data can be used to forecast district level consumption [6], appliance data could be used to forecast residential consumption, helping not only consumers but also utility companies. Most experiments are focused on forecasting only the total house consumption, with few studies on how to analyze and optimize appliances' energy consumption. The authors of [21] used major appliances' consumption data to increase entire house consumption forecasting accuracy. Other exogenous variables are also used as input features, such as weather [6], calendar [22], and socioeconomic and building conditions [8].

In [23], the time granularity for consumption forecasting was chosen by using the Mean Average Percentage Error (MAPE), while [8] used the Normalized Root Mean

Squared Error (NRMSE). In both cases, the normalized errors tended to favor low frequencies of granularity (hourly or daily), while resulting in greater errors for high frequencies (minutely). Conversely, non-normalized errors such as Mean Absolute Error (MAE) or Mean Squared Error (MSE) favor high frequencies to the detriment of lower ones. Thus, there is currently a research gap due to the inadequacy of using error metrics to choose an adequate time frequency, as the result might be biased according to the metric chosen.

To the best of the author's knowledge, [24] is the only study to consider real world deployment challenges on consumption forecasting by using hierarchical models when data such as weather forecasts are missing or unavailable during prediction. Despite not being focused on real-world deployment, [21] used the time taken to train and predict using the forecasting model as a metric to evaluate the trade-off between accuracy and computational resources.

One of the difficulties in comparing results between different studies found in the literature is related to the different metrics used, as observed in Table 1, as well as the various datasets considered, which all use different time granularities and experimental periods, and refer to different countries such as Australia [8], Canada [21], Germany [23], Ireland [6], Portugal [24,25], and the United States [23].

Table 1. Related studies regarding residential consumption forecasting.

Paper	# of Houses	Appliance Data	Models Used	Metrics	Time Granularity	Forecast Horizon	Experiment Period
[6]	25	no	SVR, ANN	NRMSE	30 min	1–24 h	18 months
[8]	27	no	Gradient boosting	RMSE	30 min	1– 24 h	35 months
[23]	7	yes	Multiple	MAPE	15–60 min	15–1440 min	9 months
[21]	1	yes	LSTM	MAPE	30 min	30 min	24 months
[24]	20	no	Hierarchical	NMAE, NQS	hourly	24 h	12 months
[25]	93	no	ANN	R2, MAPE, SDE	hourly	1–24 h	17 months

2.2. Short Term Energy Consumption Prediction

Machine learning (ML) techniques for prediction [6] and classification [7] of energy consumption are widely used in both academia and the industry, applying different learning models, such as neural networks [26], support vector machines [8], and gradient boosting [22].

Residential energy consumption forecasting can be used to assist residents in decision making and conscious spending planning [27] and utilities in medium-scale and large-scale prediction and detection of customer consumption anomalies [28]. They can also facilitate energy transactions between prosumers in peer-to-peer (P2P) energy markets [29,30], promoting the efficient use of the power grid. Home Energy Management Systems (HEMS) can use consumption prediction as an input for predictive control models [31], assisting in planning the usage of controllable applications, such as washing machines, air-conditioning systems, and electric vehicles, in order to optimize the use of energy co-generation and financial savings for users in variable energy tariff schemes.

Unlike medium-scale and large-scale energy consumption, individual hourly consumption is more volatile, with daily consumption peaks occurring at different times. Due to this characteristic, traditional metrics for measuring forecasts, such as the mean absolute error (MAE), end up measuring only point-to-point accuracy and do not analyze temporal or shape errors.

Figure 1 shows an example of a constant forecast (F1), which does not introduce any significant value to its user and has a smaller point-to-point error than a forecast with behavior closer to the real one but displaced in time (F3). While the F1 forecast has an MAE of 0.82, the F3 forecast has an MAE of 0.99.

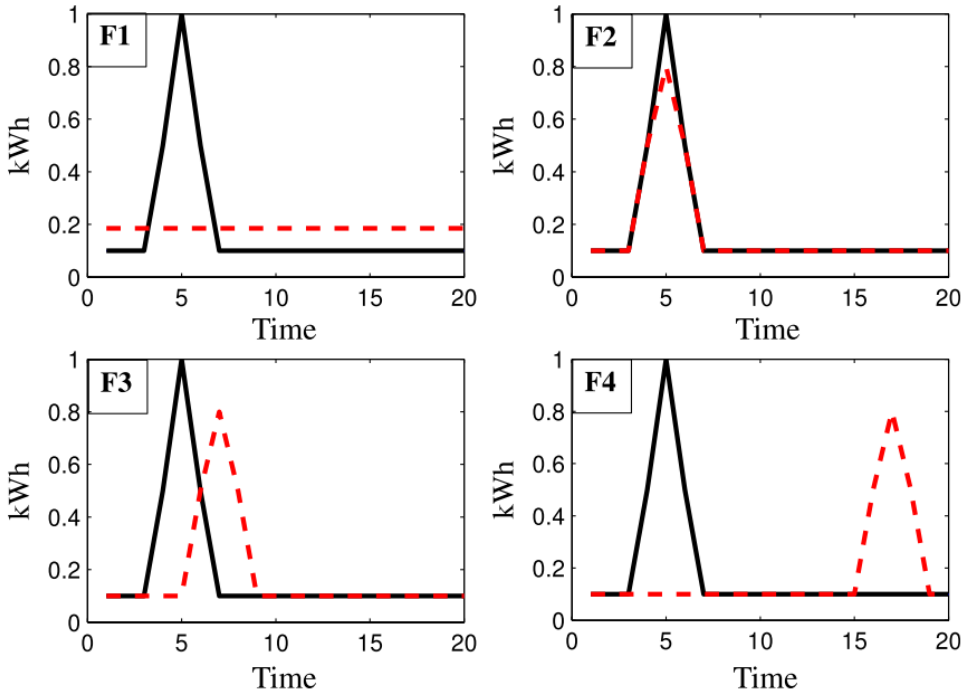


Figure 1. Four different predictions F1, F2, F3, and F4 (dotted lines) compared to the actual value (solid lines). Source: Reprinted with permission from [32].

For a satisfactory analysis of prediction models, it is necessary to use metrics that consider shape and temporal errors such as Dynamic Time Warping, Move-Split-Merge [33], DILATE [34], or the adjusted error [32].

Good forecasts are not measured only by their accuracy. Not only can different metrics can produce different results, but it is also important to consider other types of goodness, such as correspondence to human specialists judgment (consistency), similarity between forecast and previous observations (quality), and insight generation to their users (value) [35].

Additionally, the main features used to improve the accuracy of consumption forecasts in the short term (next hours or next days horizon) include weather data, such as temperature, precipitation, or wind speed, and calendar data, such as time of day, day of the week, or occurrence of holidays [6,8,24].

In order to create value for residential consumers, it is important to capture the multiple seasonalities and trends in their energy consumption. Energy Consumption has complex seasonality [18], with hourly, daily, weekly, and yearly components. In order to better analyze them, the Auto Correlation Function (ACF) plot can be used to compare similarity between time series and its lagged versions.

Most experiments performed, however, are performed in offline environments, not providing due importance to the treatment of erroneous or incomplete data in addition to the degradation of accuracy over time [24]. The use of MLOps has been deployed to address these challenges in other applications of machine learning [36,37].

2.3. MLops

In order to integrate the stages of software development and operations of information technology systems, DevOps culture uses test automation, monitoring and integration, and infrastructure management as code, among other techniques, thus allowing continuous delivery and deployment of the system [38].

The application of DevOps culture in Machine Larning (ML) systems, known as MLOps [39], seeks to adapt DevOps techniques to the area, distinguish itself from practices used in traditional software systems due to its dependence on data quality through correct extraction and processing its exploratory nature during development by testing different

configurations, model architectures, and feature generation and its error monitoring derived not only from erroneous system programming but also caused by obsolete or biased models and training data.

Thus, testing systems before introducing them into production environments and monitoring their performance is considered good practice in the development and operation of software systems. However, due to their predictive nature, such practices are difficult to define and implement in ML systems [40].

Google Research uses 28 metrics to measure the readiness of ML systems in production [36]. These metrics involve tests related to 4 categories, which are input data, the model used, the infrastructure, and system monitoring. Each category has seven tests, such as ensuring privacy control for data, tuning hyper parameters, testing integration throughout the pipeline, and monitoring code dependencies.

In addition to these metrics, another good practice in ML projects is the separation of its steps into pipelines [37] to facilitate the integration of the different steps, the scalability of the system, and the reproducibility of the results.

One of the differentials of online systems is the need of continually training their models to avoid concept drift. In [41], a strategy is defined for simulating and evaluating the effects of periodic retraining in time series, finding the seasonality of the input data and updating the model at each seasonal cycle by using training and validation data that reflect the most recent cycle.

2.4. Digital Twin

One of the most critical aspects of creating a higher engagement level of human user and digital service interaction involves advanced personalization techniques. In this context, real-time data obtained from IoT devices (Technical IoT) and from humans (Human IoT) could be combined to represent digital users in both dimensions: structural/static and dynamic/behavior. The digital twin-based model might bring more engagement elements by offering helpful information with request–response interactions [42]. Instead of the Human IoT concept presented in the literature [43], which aims to develop IoT solutions focused on usability guidelines, the Human IoT is used to refer to cooperation between humans and machines, considering that IoT may enable machine–machine [44] and machine–human cooperation.

Digital modeling of a naval building, an oceanic petroleum platform, civil construction, and health care are examples of digital twin techniques for improving operational efficiency. Real-time data collected from IoT devices are mapped directly to the corresponding element digitally created in these cases.

With the digital twin model, each part of a physical structure is linked to precise data, and each behavior is recognized and registered to help in such operational procedures. Moreover, applying prediction models allow efficiency in terms of cost reduction or risk mitigation in some use cases [45–47].

How could all this be performed in the smart-home demand-side management scenario? Energy-consuming profiles can be collected and analyzed in a real-time fashion and specifically to each customer. A digital twin model organizes structural and behavioral data, which means precision and prediction information. This prediction and meta-data information may orient customers with customized suggestions to help people reduce their energy bill.

One of the alternatives to model a smart home digital twin that could be useful to our approach is by using ontologies. These semantics-related knowledge representations are understandable by humans and readable by machines. As found in the literature, it is possible to use ontologies to model a smart-home digital twin [48,49]. For example, a digital twin based on the Web of Things (W3C) description [50] is compatible with JSON format and supports SPARQL queries [48]. Other authors designed modular and independent ontologies with the Protégé editor tool to model a home automation system digital twin with the environment, equipment, resources, and their possible relations [49].

3. Digital Twin Mlops Method

3.1. Digital Twin Architecture Requirements

This project considered some requirements to prioritize efforts related to scientific and industrial lines of using a digital twin (DT) model, as described below:

1. How may it improve human–computer interaction (HCI) by applying personalization techniques considering the customer as an energy consumer, home environment, home places, appliances, and specific and distributed IoT devices to measure power consumption?
2. Considering that HCI may use natural language to implement natural language interactions, it is crucial to consider the memory aspect to create continuous and evolutive engagement levels. In this context, investigating how using a digital twin supports natural language interactions is a must.
3. How does it integrate digital twin and machine learning models to map seasonal behavior of energy consumption and to execute prediction functions and to help with energy awareness personalized suggestions?

The project requirements can be met with conventional engineering mechanisms to build human–computer Interaction (HCI). However, in this research project, the decision was to apply Digital Twin technology to facilitate integration with other emergent technologies, including IoT and Machine Learning tools, to obtain a more effective HCI. Table 2 shows some characteristics to compare digital twin to a conventional implementation. It also summarizes conclusions regarding architecture decisions in both implementation alternatives: conventional and enhanced by digital twin.

In conventional modeling, the database structure is centralized, and register fields are sufficient for adding such attributes to static data and events collected in real-time integration. The personalization configuration to HCI, Machine Learning for Seasonal and Prediction, and Natural Language Processing uses a set of user profile parameters. Note that personalization considers a set of similar profiles to deal with the trade off between volume and performance. It is an impracticable process individually for each user-customer: low performance with a substantial impact on usability. It is a crucial highlight—the centralized database is enormous in volume and not prepared for individual access and processing—that the processing balance is performed by grouping users with the same profiles to process a set of registers. For customized interactions, seasonal profile analysis, and energy-consumption prediction, the same rationale is valid. Therefore, personalization is limited to similar profiles parameterization.

The digital twin applied to this research project is different in crucial aspects, and the results are more effective in terms of personalization in general with a positive impact on all requirements listed. The first difference appears in modeling. Each user and his/her home, places, appliances, and IoT devices correspond to a digital twin that is different from conventional implementation, whereby the centralized database includes registers for both static and temporal series of events.

Software objects with data and functions organize and implement each user energy-consuming database; that is, software objects are connected with abstraction: User-customer is connected to home; it connects to place-spaces of home; it connects to each family member; and it connects to each appliance and to each IoT device. The organization forms a structural ontology that supports all personalized natural language interactions and all prediction functions in each user database.

Table 2. Architectural aspects comparison using the Digital Twin.

Aspect	Conventional	Using Digital Twin
Database entities (energy consumption data)	Energy-consumption registers Database entities: 1. House; 2. User (energy consumer); 3. Family (people and energy consumer); 4. Convenience (places of the house); 5. Electrical appliances; 6. IoT devices.	Digital space implementing real world elements as software objects from Digital Twin mechanisms includes the following: 1. A digital object of the house; 2. A digital object of user-consumer; 3. A digital object of each family member (energy consumers); 4. A digital object of each convenience (each one with energy consumer devices); 5. A digital object of electrical appliances; 6. A digital object of each IoT device.
Data titular and controller (According to the Brazilian's LGPD law)	All data stored as registers of a centralized database, and operated by the digital platform. Each data titular can go along only with his/her registers processing.	Each digital twin stores the corresponding data collected. According to law requirements, the user owns data and acts as titular and controller, in cooperation with the platform that acts as a data operator.
Real-time data (IoT)	Registers on database (events): 7. Timestamp; 8. Measurements; 9. IoT devices; 10. Relationship (House, User-consumer, Convenience).	Digital software objects register their own collected events: Convenience(4), Electrical appliance(5), and IoT device (6): 7. TimeStamp (4) (5) (6); 8. Measurements (4) (5) (6).
Natural language for information obtained from the user	The centralized database registers provides information in the right column table.	Each digital software objects register data provided by the user, connecting directly to digital twin implementation (house, user-consumer, convenience, electrical appliance, and IoT device)
Machine Learning (e.g., Chatbot)	Historical data include talking with limited and centralized memory. The conversational interaction is almost repetitive and focused on a set of users profiles.	With the digital twin, all interaction and memorization connect the correct user. In this implementation, the more historical data, the more maturity accomplished.
Machine Learning (Prediction)	Machine Learning implemented for: 11. Learning and showing seasonal information, using events data; 12. Predicting energy-consuming data, using events data. All data parametrization refers to sets of the same profile user.	Machine Learning parameters mapping directly for each digital twin software objects; that is, all data and all objects relations of real-world elements (house, user, family, conveniences, appliances, and IoT): 9. Learn and show seasonal data with precision and helpful information; 10. Predict energy-consumption data with precision according to all parameters related to his software objects.
Data organization	Huge centralized database, where sets of user-profile foundation to process intelligent services.	A federation of databases. Each database corresponds to one group of user-consumer implementing digital twin of real-world energy consumers.

IoT devices collect each event, and each energy-consumption datum corresponds to one user-consumer software object. Conversational interactions may be supported by the memory associated with this user-consumer software object. This implementation uses a NoSQL implementation tool and organizes one database for each energy-consumption user as detailed, with a digital twin implemented as a set of software objects. With this architectural decision, personalization is superior to the conventional approach.

A machine learning parametrization procedure is superior when compared to the conventional approach because all data applied are corrected to their own control: user-customer structure data (home, family, convenience, appliances, IoT devices) and time series. With this, seasonal modeling is accurate and valuable according to the data provided with respect to energy-consumption prediction, considering that the parameters show more precision than the conventional implementation.

Digital Twin adoption brings research challenges, especially for industrial applications. One relevant aspect is data organization. As described in Table 2, the database federation is the foundation of data organization, whereby each user-consumer is the owner—there is no centralized and colossal database.

In this context, other opportunities arise about data usage: the European GDPR [51] (General Data Protection Regulation) and Brazil's LGPD [52] (Lei Geral de Proteção de Dados) are laws regarding data privacy. This database federation creates the condition to enable user empowerment as a data principal and controller. In this case, the platform acts as a data operator service, providing user autonomy and coverage.

3.2. Smart Home Testbed

The smart home testbed was based on data collection architecture of energy consumption presented in [3], implemented in the early 2020 in four Brazilian households.

Specifically, the digital twin Proof of Concept for this work is built upon a household with four inhabitants and 31 energy consumption time series collected with smart meter and smart plug IoT devices.

The smart meters have a data collection system tolerant to connection failures, ensuring the integrity of data during network outages through the connection with an intermediary for data temporary storage [53].

The hourly consumption and internal temperature data of the residence are sent to a remote database, which are used by the proposed solution to forecast energy consumption and train machine learning models.

Currently, the database has information on the period from January 2020 to December 2021, with a gap from January 2021 to April 2021 due to modifications made to the smart meters used, resulting in a total of 19 months of data. Figure 2 shows load profiles of minute granularity in a weekday from different appliances monitored, such as television, refrigerator, computer, air conditioner, and living room light bulbs.

3.3. Mlops

Each stage of the pipeline has multiple steps, as shown in Figure 3. In the offline environment, tests were performed for prototyping models and experimenting new functionalities.

The online pipeline, on the other hand, albeit similar to the offline environment, has differences regarding the degree of automation, runtime constraints, and error handling. In the first stage of the pipeline, which only occurs in online environments, the automated search for data is performed, either in internal databases or through external interfaces, requiring the correct handling of exceptions due to unavailability or transfer errors. The next pre-processing step includes feature cleaning and engineering, in addition to the treatment of anomalies and missing values performed manually in offline environments.

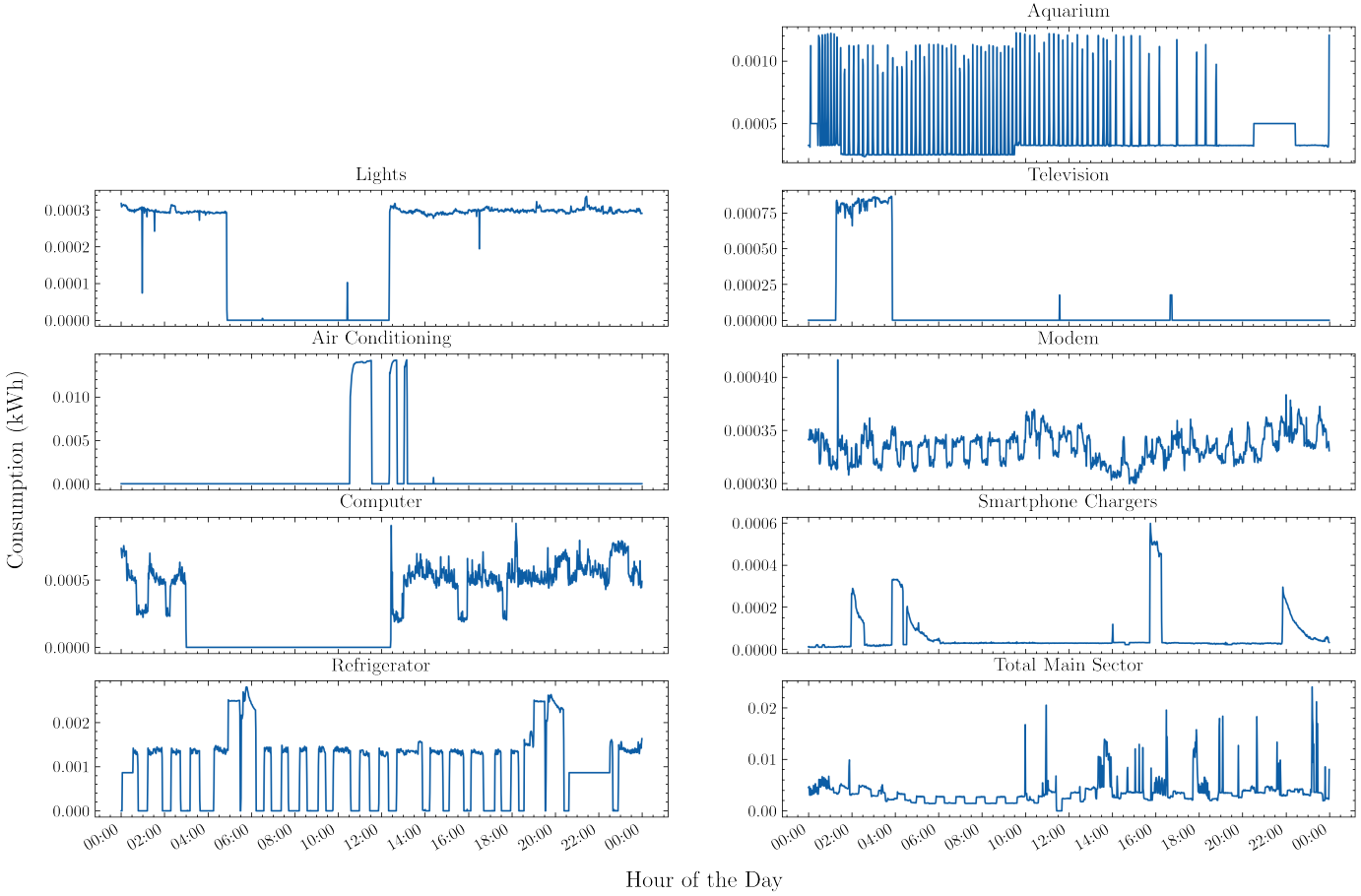


Figure 2. Daily load profile of different appliances during a weekday.

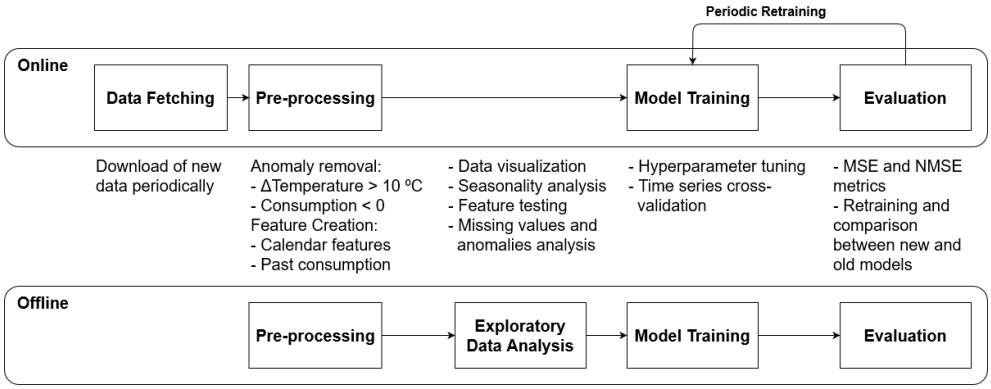


Figure 3. Online and offline pipelines for machine learning projects.

In offline environments, exploratory data analysis is then performed, in which data familiarization, anomaly detection, and distribution and correlation analysis between features occur in order to iteratively refine the previous pre-processing step. In the next step, the model is built by defining its hyper parameters, either manually or automatically through grid search, and trained according to available data.

Finally, in the model evaluation step, its accuracy is measured, and the hyper parameters that optimize the defined metric are selected. Thus, it is important to analyze and to choose which metrics will be the most appropriate and relevant to the problem.

3.3.1. Data Loading

Input data are received from cloud storage service, which stores the total energy consumption and by sectors, as well as the internal temperature of the residence, with a sampling frequency of one hour. Every hour, files are searched for in the cloud, and if the files not present in the local files are found, they are downloaded to the local directory.

3.3.2. Pre-Processing

During pre-processing, raw data are checked for missing hours and anomalies. To be considered an anomalous value, the consumption must be less than zero, and temperature must vary by more than 10ºC from the previous value.

There were occurrences of temperature anomalies in which variations in relation to the previous hour exceeded 20 ºC, as well as instants with missing temperature and consumption readings. In these cases, it was assigned as a reading error, and these values were discarded.

For defining features, three past hourly consumptions were added, referring to 25, 24, and 23 h ago in relation to the instant to be predicted, in addition to calendar-related attributes, such as the time of day, day of month, and month of the year to be predicted. The choice of these features for the final prediction model was made in the exploratory data analysis step.

However, new features can be added by modifying input files, such as adding the internal temperature of the residence, or also generated by modifying the source code of the pre-processing step, such as adding the first derivative of hourly energy consumption. In this manner, the other steps of the pipeline do not need further modifications.

3.3.3. Exploratory Data Analysis

In [3], energy consumption prediction models were developed by using Extreme Gradient Boosting (XGBoost), long short-term memory neural networks (LSTM), and support vector machines (SVM) architectures. The results showed that the XGBoost architecture obtained better accuracy in most of the monitored households, and this architecture was chosen for this study.

XGBoost is an open source ML library for regression and classification models using decision tree ensembles [54]. Its implementation allows training models in a parallelized and distributed fashion. The models also accept the existence of missing values in the input data in both training and prediction stages.

In order to analyze the gain of introducing new features, a base reference model was deployed, using only the last 24-hour consumption and calendar data: time, day of the week, day of the month, month, and year. This reference model was compared with other models with additional features in addition to those used in the reference model, as shown in Table 3. Cross-validation was used for each household, obtaining the mean squared error (MSE) and the adjusted error with a 2-hour window and norm 4 [32] from all households. Table 3 also shows the percentage reductions of MSE and the adjusted error relative to the reference model.

Table 3. Analysis of the addition of features on model accuracy.

Model	MSE	Adjusted Error
Reference Model	0.0479 (0%)	0.2535 (0%)
Reference + 1st Derivative	0.0476 (−0.62%)	0.2529 (−0.24%)
Reference + Indoor Temperature	0.0502 (+4.80%)	0.2568 (+1.30%)
Reference + Consumption from 25 and 23 h ago	0.0415 (−13.36%)	0.2043 (−19.40%)

The addition of the residence internal temperature as a feature of the model eventually reduced its accuracy, while the use of the first derivative of the energy consumption had little significant gains. Due to these results, these features were not considered in the final model.

Low weekly correlation was observed for all residences, with no great variation between weekdays and weekends, as shown in Figure 4 for one of the residences. Note that the consumption data refers to the year 2020, and this low variation may be related to the quarantine period due to the COVID-19 pandemic. This effect confirms what is presented by [55], in which consumption during weekends was higher than in weekdays for the residential sector in 2018 and 2019 but had similar consumptions for weekdays and weekends in 2020.

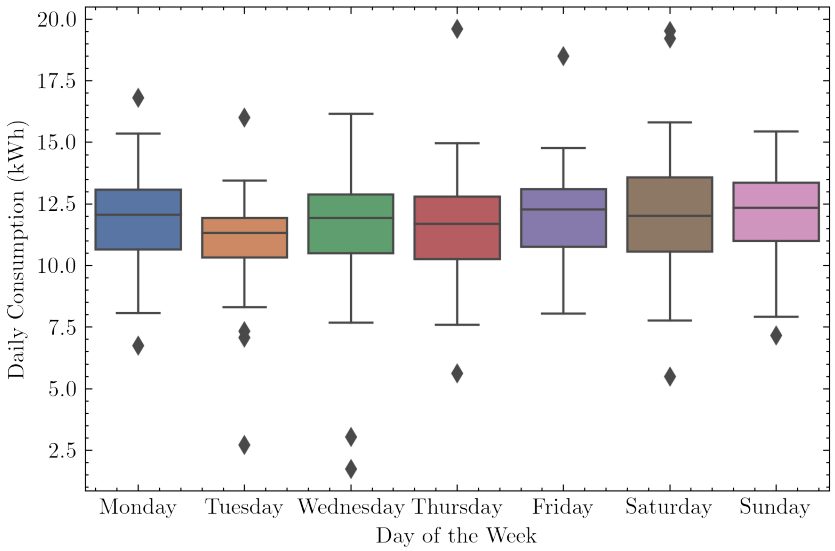


Figure 4. Box plot of the total consumption per day of the week for one of the households.

Energy consumption was higher during winter, as observed in Figure 5, due to the increased usage of air conditioning. A greater variance of consumption can be observed during the summer, although its median is similar to other seasons. This could be explained by greater air conditioning usage, as well as holidays and inhabitants absences.

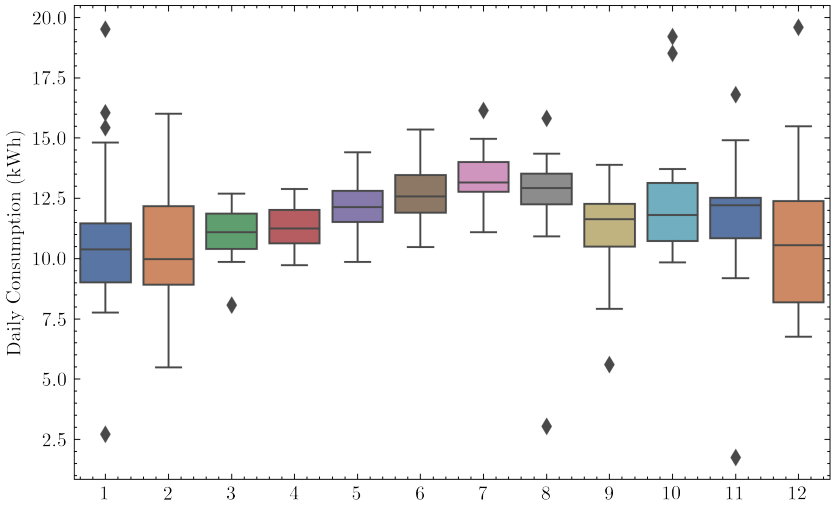


Figure 5. Box plot of the daily consumption for each month for one of the households.

Figure 6 shows the autocorrelation function of sampled energy consumption with hourly frequency for one of the monitored households. In the plot, a larger value on the

ordinate axis indicates high correlation between the time series and the series lagged in time by k units, with k represented by the abscissa axis. One can observe autocorrelation peaks for 24-h lags, evidencing daily seasonality.

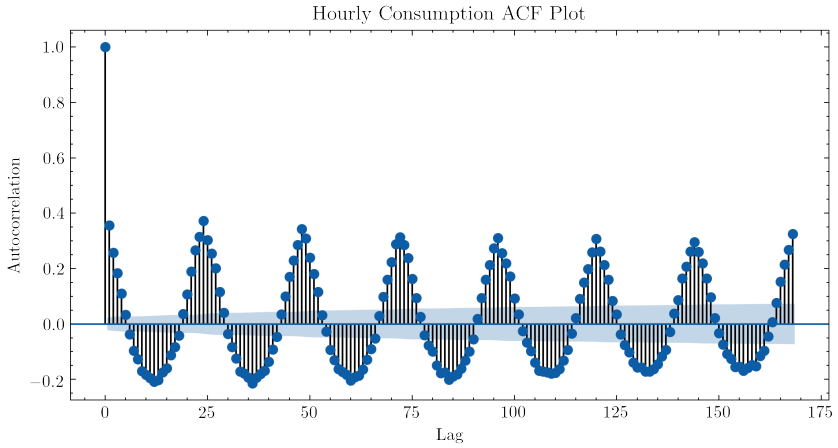


Figure 6. Autocorrelation function of the hourly energy consumption (95% significance band).

Figure 7 shows the first decision tree of the model, which observed the relevance of the time of day in model prediction, while Figure 8 shows the importance of the features for one of the households calculated as the number of times each feature appeared in XGBoost's decision trees. The possibility of obtaining information related to the internal structure of the model is important as it allows debugging the operation and investigating performance drops or instability.

The models are also evaluated with extreme or even invalid inputs, assessing their robustness. The inputs tested are as follows: consumption equal to zero, negative, infinite, and with missing values.

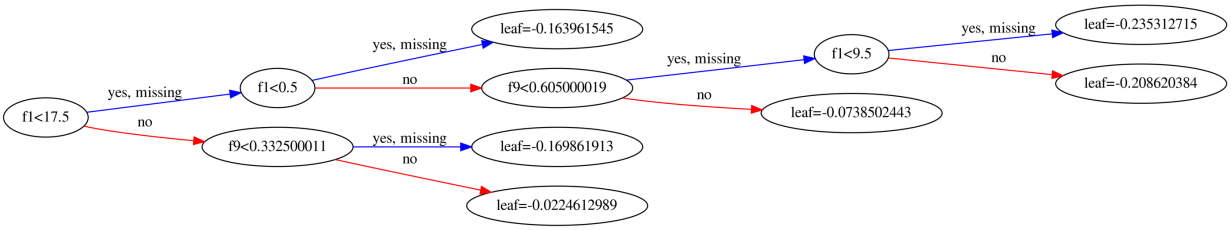


Figure 7. First decision tree for XGBoost prediction model.

3.3.4. Model Training and Prediction

The model uses the XGBoost library to predict the hourly consumption of the next 24 h, and it is trained with consumption features of the last 23, 24, and 25 h, the time of day, day of the week, day of the month, day of the year, and month.

In order to perform XGBoost hyper parameter tuning, a grid search is performed with cross validation with partitions of four subsets for each household, varying tree size, learning rate, and objective function to be minimized. After training the models for each combination of hyper parameters, the one with the smallest mean square error is chosen. The random seed used by XGBoost is fixed automatically, ensuring the reproducibility of results. Both data and code are versioned via Git and DVC version control systems.

Figure 9 shows an example of energy consumption prediction for one of the households performed during the month of July, and it is possible to observe the daily seasonality of energy consumption.

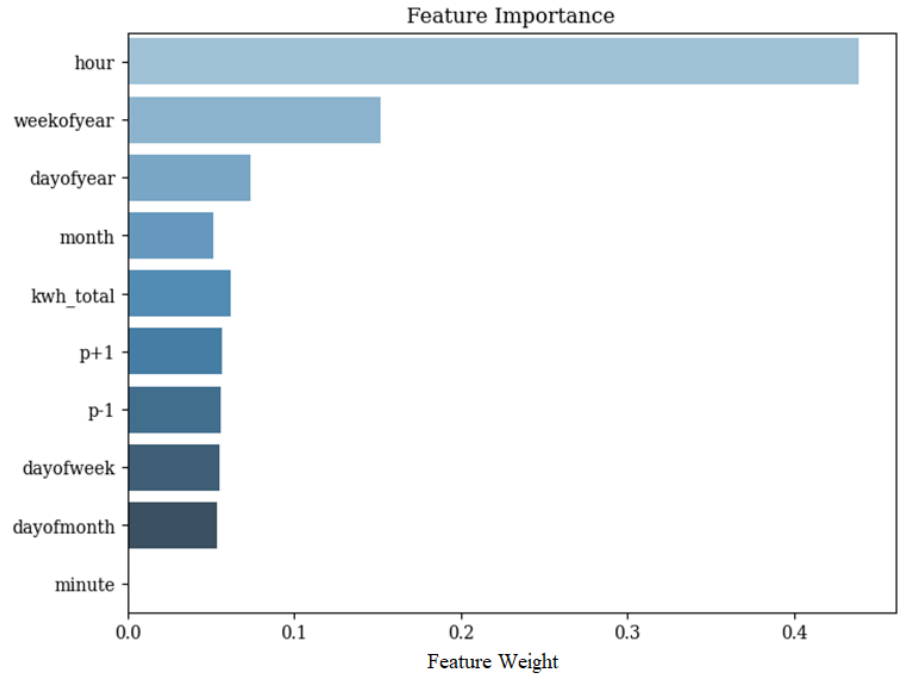


Figure 8. Importance of features for the prediction model of one of the project residences.

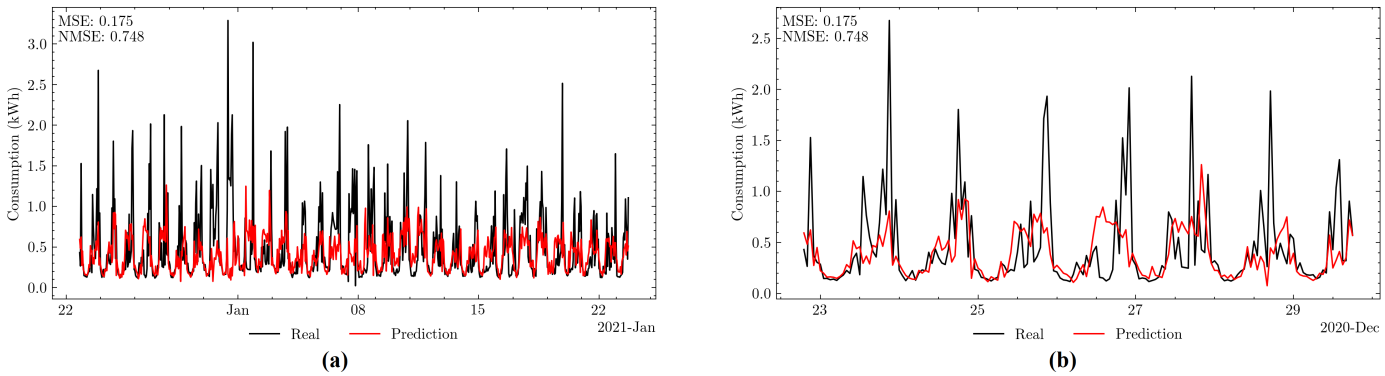


Figure 9. (**a**) Prediction (red) and actual value (black) of energy consumption for the months of December 2020 and January 2021. (**b**) Zoom in on the first week of the test data.

3.3.5. Inference

The system in online environment was deployed as a Flask application on Apache2 server hosted on Amazon Elastic Compute Cloud (EC2), performing retraining periodically every 24 h and permitting the reception of calls in REST API format for the consumption forecast of households monitored.

The API can be used by other systems to query users' consumption forecast. Figure 10 shows an example of an application, whereby a website was developed in Dash platform [56] to perform consumption forecasts in user-customizable time periods.

The calls made to the API and training time are monitored and saved in log files. When an anomalous value is encountered, as defined in Section 3.3.2 (negative consumption or temperature variation greater than 10 °C), an alert is added to the log files.

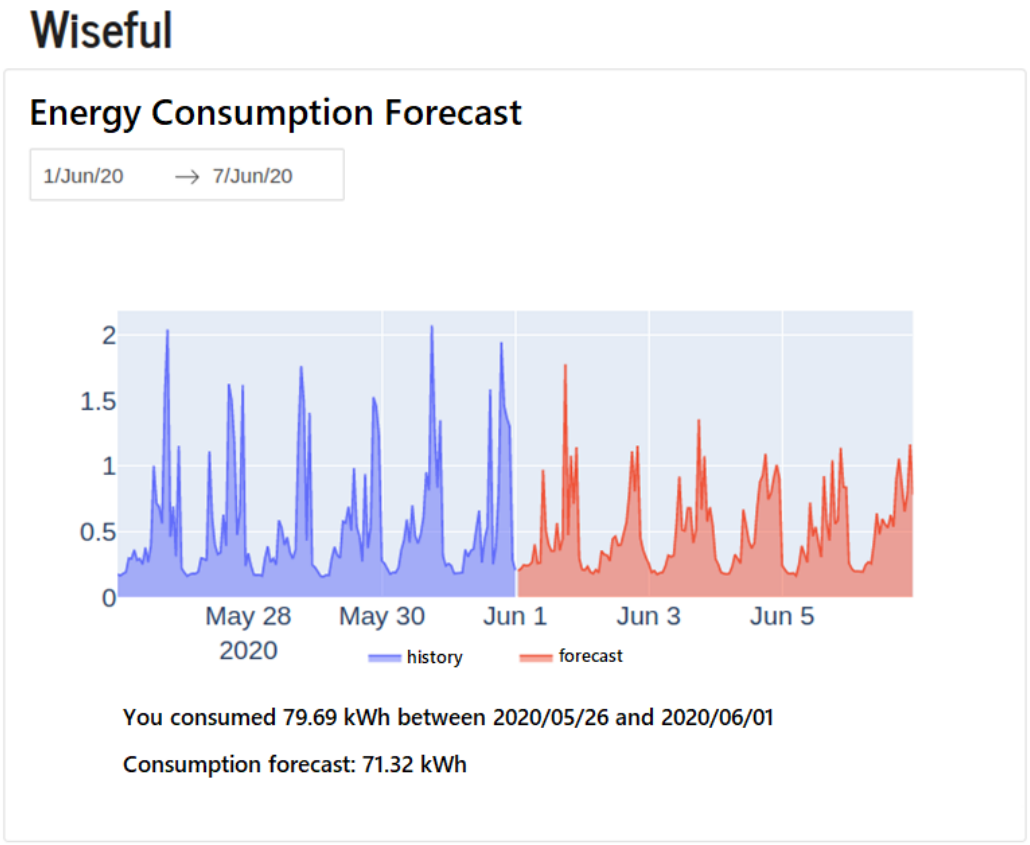


Figure 10. Website for visualizing consumption forecasts.

3.3.6. Evaluation

In order to analyze the effects of time granularity, different house appliances were evaluated using multiple time granularities. The prediction result for each combination of appliance and granularity is compared by using both error metrics and ACF for seasonality analysis.

Accuracy evaluation in a static environment is performed using the method proposed in [41]. In this method, multiple models are trained, each based on training data from different instants, to reflect the arrival of new data in an online environment.

Daily training seasonality is considered, with data partitioning 80% for training and 20% for testing. The hyper parameters are set by means of grid search. The adjusted error is used to compare the updated model with the previous one, and the one with the lowest error is used.

4. Results

4.1. Mlops Tests

Table 4 shows the tests performed by the system automatically following the metrics defined in [36] and whether they were performed autonomously (A), manually (M), not performed (-), or are not applicable (N/A).

Data Tests 4 and 5 are not applicable to the project in the current status as there is no personal data collection that allows identifying them for privacy concerns of users. Model 2 test does not apply because they are not currently monitored online metrics. Test Infrastructure 6 is not applicable due to the insufficient number of users to launch new versions (rollouts), nor is Monitoring Test 3 because there is no difference between offline and online training data.

Table 4. Tests related to data.

Category	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Data	A	M	M	N/A	N/A	A	M
Model	A	N/A	A	A	M	M	-
Infrastructure	A	-	-	A	M	N/A	-
Monitoring	-	A	N/A	A	M	A	-

Due to the relatively low complexity of the pipeline and low retraining cost, no tests regarding integration (Infrastructure 3 test) and rollback (test Infrastructure 7) were performed. Since the XGBoost library already performs a large series of unit tests to ensure correct code execution for training and predicting models, the verification of the model specification was considered as outside the scope of the project (Infrastructure 2).

As there is a low number of users at the moment, the project has not yet addressed issues of social inclusion of the system (test Model 7). When new users are invited to participate, representativeness of the Brazilian population will be important so as not to bias the system.

So far, there have been no changes in the structure of the input data; thus, monitoring changes (Monitoring test 1) are not currently performed, although in future steps if new features obtained from external sources, such as the weather forecast, are entered, this test will be of greater importance.

Since the model forecasts consumption for the next 24 h, real-time monitoring of the quality of forecasts made (Monitoring test 7) was not performed, as its accuracy can only be measured 24 h after the forecast.

Exploratory data analysis proved to be extremely important, satisfying several tests (Data 2, Model 5, Infrastructure 5, and Monitoring 5), which, despite performed manually, can be reused in the future for additions to the pipeline running automatically.

4.2. Use of Digital Twin Data to Improve Forecasting Accuracy

As mentioned in Section 2, using error metrics to choose the most adequate time frequency for prediction model training might generate biased results depending on which metric is chosen.

In order to analyze if these results, from entire residence consumption, also applied to appliance level consumption, forecasting models were trained for nine different appliances—lights, air conditioning, computer, refrigerator, aquarium, television, modem, smartphone chargers, and total main sector.

Each appliance was trained with data from five different time granularities—1 min, 15 min, 1 h, 6 h, and 1 day. Thus, a total of 45 models were trained. The forecasts were evaluated by using MSE and NMSE metrics. Figure 11 shows the mean results for each metric. It can be observed that better results are achieved for higher time series frequencies when normalized metrics are used, while better results for lower frequencies are achieved with non-normalized metrics, confirming what was observed in [8,23] with total residence consumption data.

A possible solution for time granularity selection for appliance level forecasting is using Digital Twin house metadata to categorize appliances by ACF plots and analyzing their prevailing seasonalities. This solution allows the scalability and customization of forecasts according to specific digital twin models and improved quality and value for the user, as defined by [35].

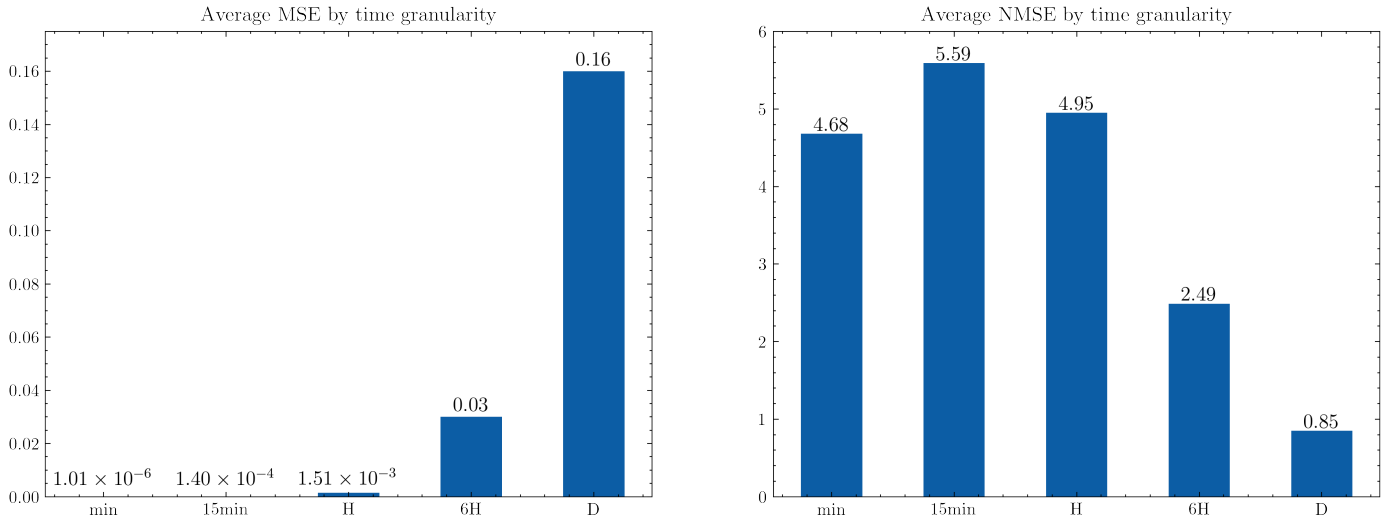


Figure 11. Average MSE and NMSE for 1 min, 15 min, 1 h, 6 h, and 1 day time period granularities.

Figure 12 shows different seasonalities for light energy consumption. Figure 13 shows their respective forecasts in different frequencies. From these results, choosing an adequate frequency is important for improving consistency, quality, and value, as defined by [35], as well as for avoiding information loss [19].

The minute and hourly frequency predictions show little temporal and shape dissimilarity when compared with daily data. There is daily seasonality present in the data, as observed in the ACF plot, which can be used to select the most adequate frequency for forecasting. Thus,in order to assist and automate this decision, the appliance classes retrieved from the digital twin model can be used in conjunction with ACF plots to select time frequencies to optimize information value for users by consumption forecasts.

4.3. Digital Twin Ontology

Figure 14 presents the smart home digital twin. It comprises persons, home, facility, room classes, and subclasses. The instances are related to the household used in the proof of concept. The four individuals live in Household ABC, which is an instance of Home. The Home class has the Room subclass, which is related to Household ABC instance. There are four household energy consumption sectors, all related to Household ABC. Each person may have a relation of private or shared room, and a device has a relation installed in some room. All these relations are illustrated in Figure 15. Additionally, each device has a data property describing its MQTT Topic, which is the Publish-Subscriber protocol used in smart home implementation.

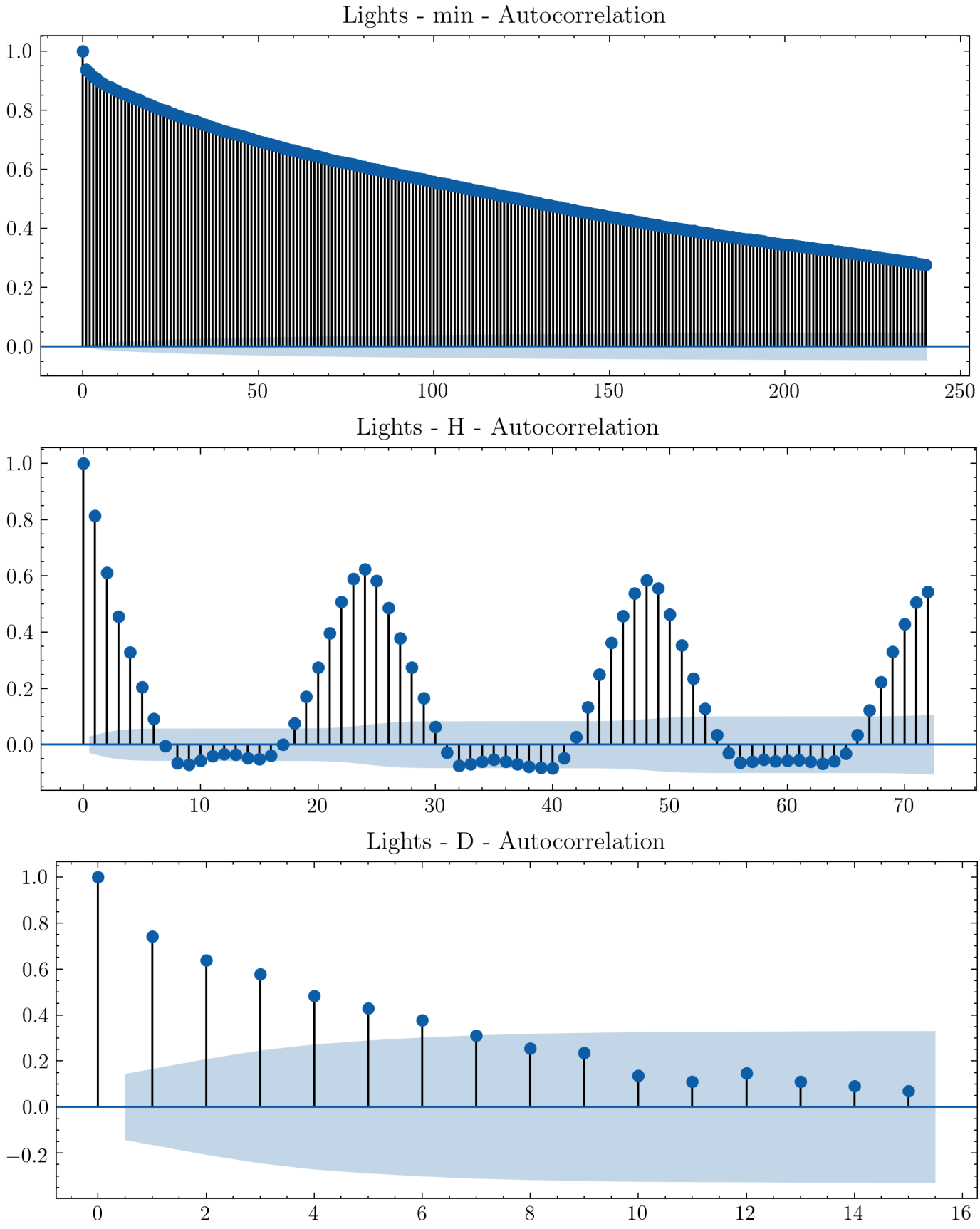


Figure 12. Lights ACF plot for minute, hourly, and daily frequency data.

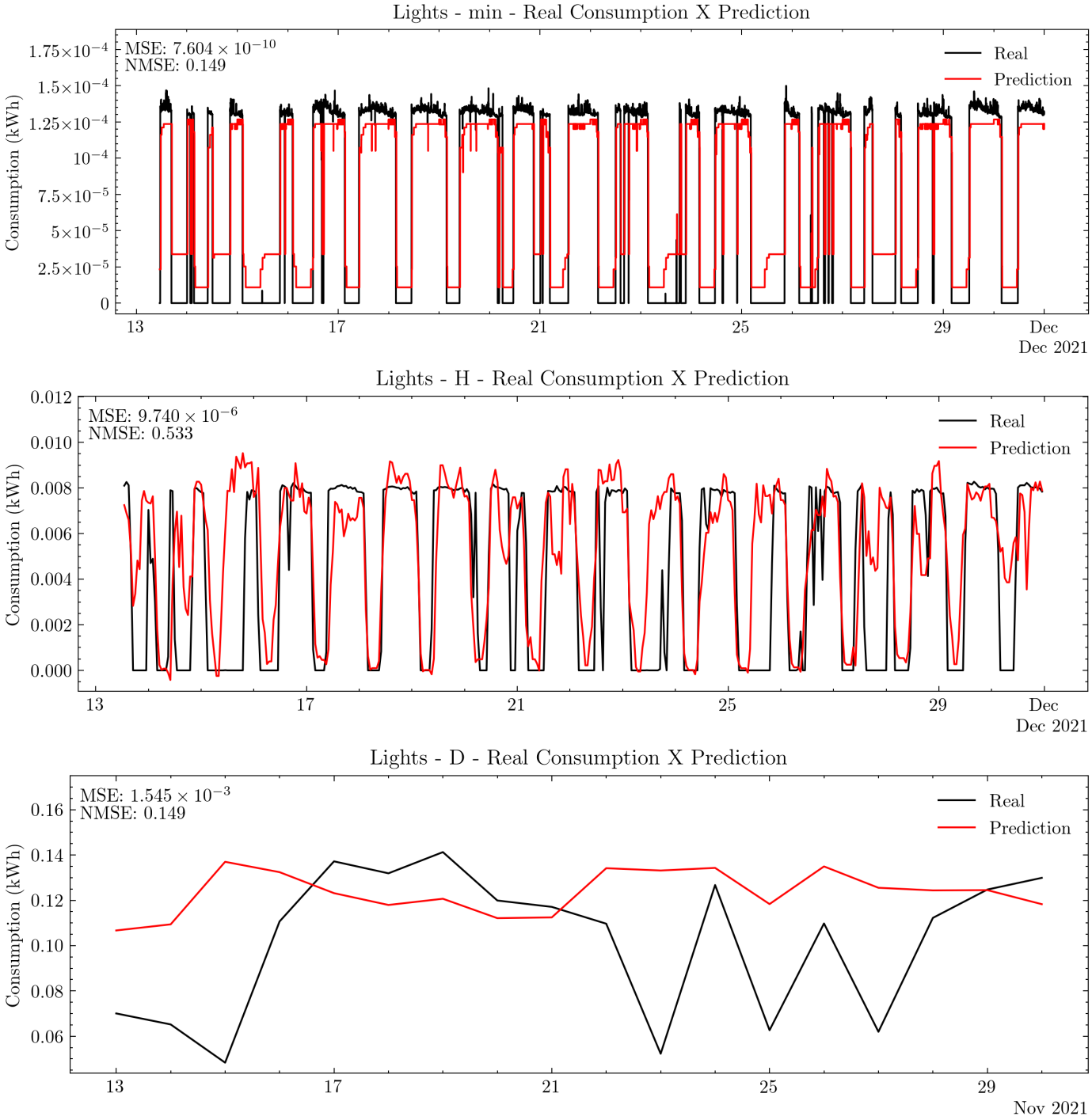


Figure 13. Lights forecast for minute, hourly, and daily frequency data.

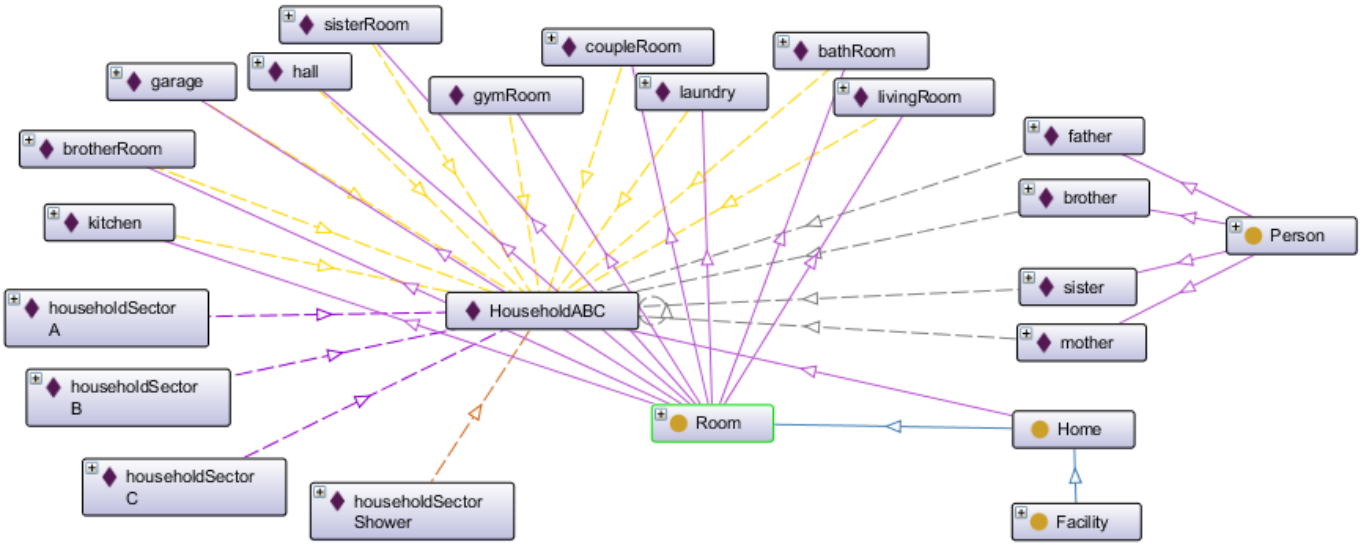


Figure 14. Household ABC digital twin ontology.

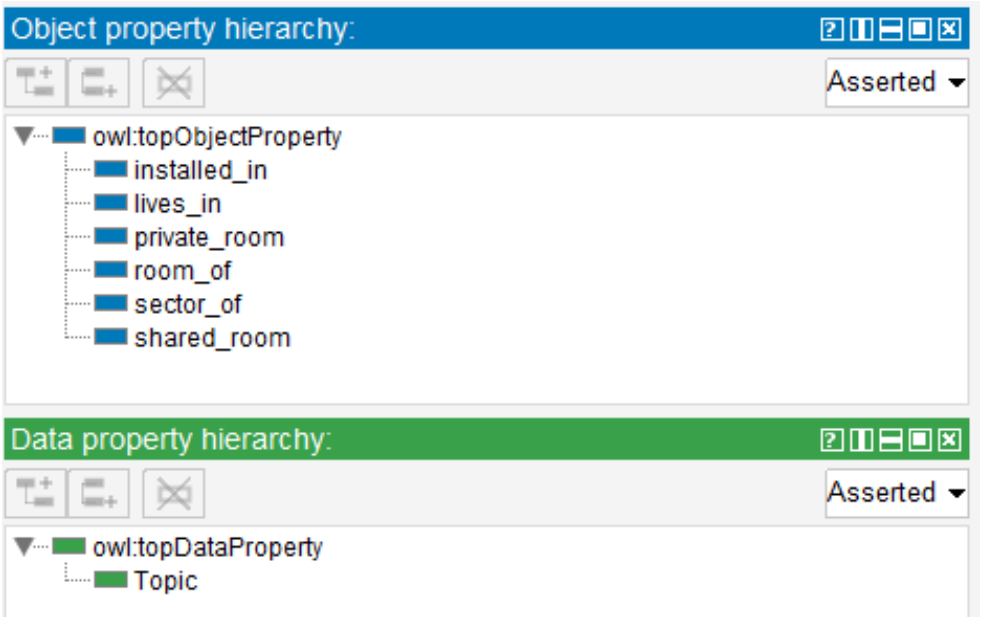


Figure 15. Digital twin smart home ontology relations and data property.

The brother person perspective is depicted in Figure 16. One may observe that the brother has a private room relation with his room. Brother room is a room of Household ABC, and home office and light bulb devices are installed in this room. A conversational agent may use this knowledge to recognize the speaker as the brother, and process the command "turn my light off" to infer that it must switch off the LightBrotherRoom and not another light bulb present in another room, thus saving a conversation iteration for increased usability. The automation command may be issued to the smart home backend based on the MQTT Topic of the LightBrotherRoom device.

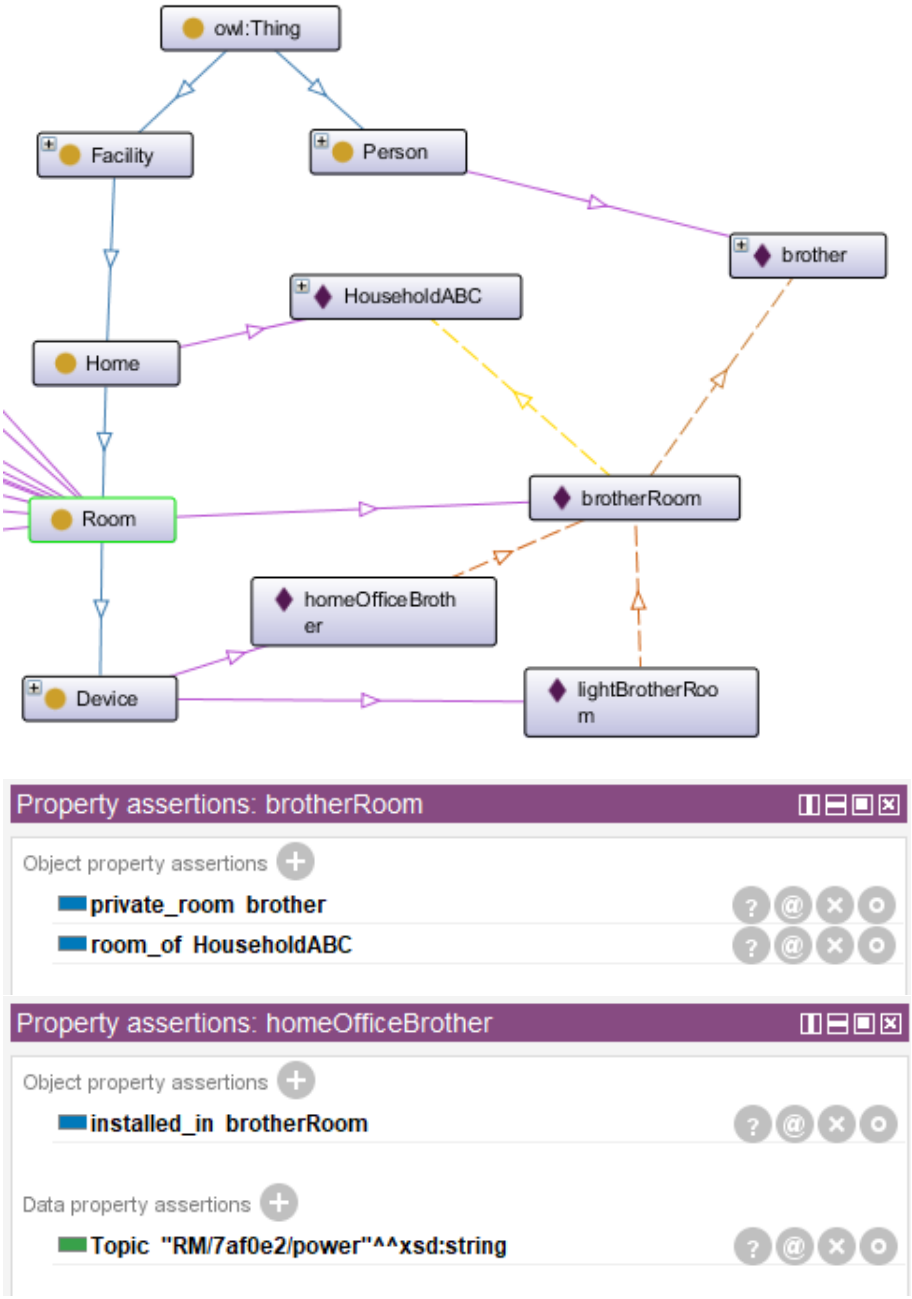


Figure 16. Brother perspective in digital twin smart home ontology.

The kitchen perspective shown in Figure 17 may be useful for a smart home automation and energy management system that must know all the devices installed in the kitchen. Based on smart plugs with device-level monitoring, household-level monitoring may be performed based on smart home digital twin ontology.

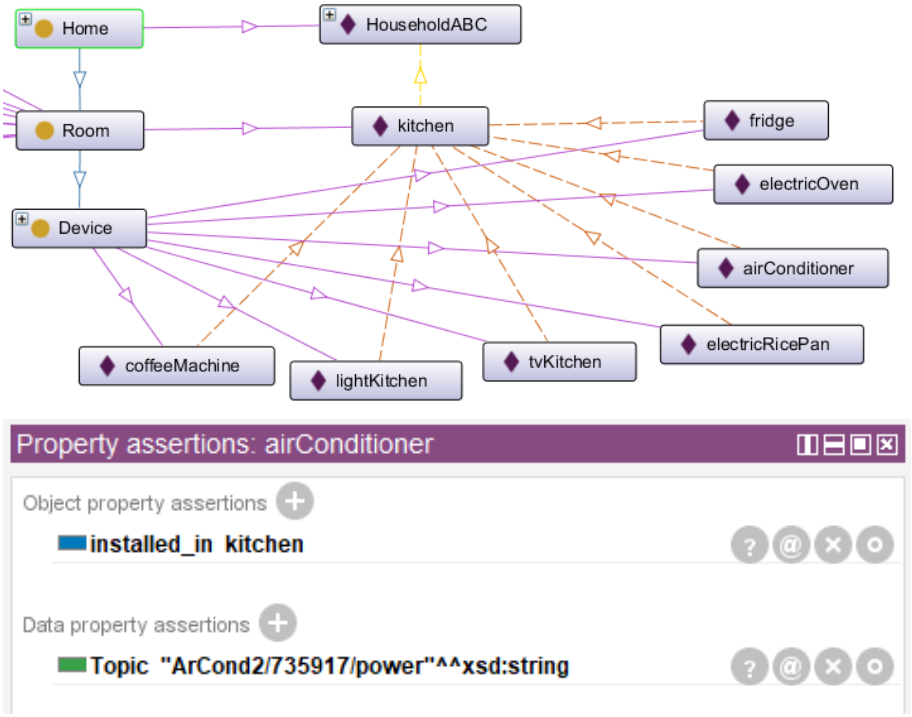


Figure 17. Kitchen perspective in digital twin smart home ontology.

5. Discussion

5.1. Comparison with Related Work

As shown in Table 5, most works used offline experiments, disregarding MLOps challenges to energy consumption forecasting. While [24] used hierarchical models to increase its resilience to missing data, his focus was mainly on data and model related challenges and did not address infrastructure and monitoring issues. A wider range of time granularities and forecasting horizons were used in comparison with related studies on residential energy consumption, and the period available for experiments is similar to other studies.

Table 5. Residential consumption forecasting papers comparison.

Paper	# of Houses	Appliance Data	Models Used	Metrics	Time Granularity	Forecast Horizon	Experiment Period
[6]	25	no	SVR, ANN	NRMSE	30 min	1–24 h	18 months
[8]	27	no	Gradient boosting	RMSE	30 min	1–24 h	35 months
[23]	7	yes	Multiple	MAPE	15–60 min	15–1440 min	9 months
[21]	1	yes	LSTM	MAPE	30 min	30 min	24 months
[24]	20	no	Hierarchical	NMAE, NQS	hourly	24 h	12 months
[25]	93	no	ANN	R2, MAPE, SDE	hourly	1–24 h	17 months
Wiseful	4	yes	Gradient Boosting	MSE, NMSE	1–1440 min	15–1440 min	19 months

One of the main difficulties in quantitatively comparing our results with related work is related to the different datasets and metrics used, as discussed in Section 2. Even if all studies used the same metrics, unless a universal dataset is used, comparing them quantitatively is unfeasible. The main focus of our study was to analyze the development approaches used to tackle MLOps challenges, as well as to optimize appliance level load forecasting.

Considering appliance consumption forecasting, it was possible to generate accurate predictions by choosing custom time frequencies for each appliance class using digital twin appliance metadata and ACF analysis for seasonality, as opposed to the approaches in [8,23] in which they used accuracy metrics.

5.2. Known Limitations and Future Work

The proposed solution supports more granular personalized recommendations than the approach based on collaborative filtering, as each household and its users are modeled in a specialized ontology [13,14]. However, a future study direction is to test, with real users, how smart home ontology supports engaging conversations and making personalized suggestions.

Human IoT is a concept explored in this project and added as an essential aspect of the platform. Distinct from the original objective of using IoT to collect data in real-time from the physical world [57], another key element is allowing the user to complement digital twin information regarding house physical dimensions; family people; energy monthly cost; electrical appliances with details, such as vendor, age, and technologies; and other pieces of information that, if combined with curate attitude, might produce valuable information. In this study, ontology was constructed with a manual method, but automating this process with user inputs in natural language may be a promising future research direction.

One opportunity is to integrate the gamification elements and other motivational factors to extend the gamified management platform proposed in [15] with a conversational interface based on a smart home digital twin ontology. Another opportunity is to use our smart home digital twin to investigate MLOps aspects when deploying reinforcement learning models as the ones presented in [17], in addition to the prediction models presented herein and found in the literature [16].

Finally, one of the most relevant challenges is to secure the digital twin [11], which is considered out of the scope of this article.

5.3. Development Considerations

One of the main fears at the beginning of the project is related to the large number of functional changes arising from the start of the project, and the assumption that the tests implemented at this stage would quickly become obsolete. However, this fear proved unfounded, since the simple definition of the tests not only verified the correct execution of the code but also guided the process of development, following Test-Driven Development (TDD) [58].

Data versioning proved to be important in experimentation during exploratory data analysis, ensuring the reproducibility of experiments performed in previous versions. During the development process, it was necessary to balance delivering results and running tests so that the definition of priorities was extremely important in the course of the project. The choice of priorities was calculated according to the probability of related problems to occur, considering the impact of these problems on the system.

Another consideration for running the tests was the modularization of pipeline steps. By accurately defining its expected features, inputs, and outputs, it makes it easier to change the source code and experiment with new settings and it is easier to observe how specific changes impact the results.

6. Conclusions

Research Question 1 was properly addressed by the digital twin household ontology model that includes topological and behavioral aspects of the residence and relevant metadata that may be used with the energy consumption forecasts by other systems (e.g., smart home with energy management system with solar photovoltaic panels and batteries).

Research Question 2 was also addressed in MLOps experiments that show how the proposed solution is robust to missing data and supports multiple time granularity of 1 to 1440 min and MSE and NMSE metrics.

A smart home digital twin integrated with MLOps is proposed to effectively predict energy consumption at a device level. Our approach may be useful for tackling the challenges of deploying machine learning prediction models in online environments, considering the specific scenario of energy consumption forecast. Household metadata is modeled in an ontology to support facilitated integration of real-time monitoring and prediction information with new interfaces, such as personalized conversational agents and dashboards.

The approach was validated by using a residential study case with 31 IoT smart meter and smart plug devices with 19-month data (measurements performed each second). Our results show that choosing custom time frequencies for each appliance class and ACF analysis allowed generating accurate predictions, actively tackling MLOps challenges in the energy forecast scenario.

Author Contributions: Conceptualization, T.Y.F. and R.A.; methodology, R.A.; software, T.Y.F., V.T.H., R.B.J., F.H.H. and K.A.K.; validation, R.A. and W.V.R.; investigation, T.Y.F., R.A. and V.T.H.; resources, V.T.H. and F.H.H.; data curation, T.Y.F.; writing—original draft preparation, T.Y.F., V.T.H. and R.A.; writing—review and editing, V.T.H.; supervision, R.A.; project administration, V.T.H.; funding acquisition, V.T.H. and R.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by FAPESP under grant number 20/05763-6 and PPGEE ("Programa de Pós Gradução em Engenharia Elétrica") from the Polytechnic School of the University of São Paulo.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Iqbal, S.; Sarfraz, M.; Ayyub, M.; Tariq, M.; Chakrabortty, R.K.; Ryan, M.J.; Alamri, B. A Comprehensive Review on Residential Demand Side Management Strategies in Smart Grid Environment. *Sustainability* **2021**, *13*, 7170. [CrossRef]
2. Cruz, C.; Palomar, E.; Bravo, I.; Aleixandre, M. Behavioural patterns in aggregated demand response developments for communities targeting renewables. *Sustain. Cities Soc.* **2021**, *72*, 103001. [CrossRef]
3. Hayashi, V.T.; Arakaki, R.; Fujii, T.Y.; Khalil, K.A.; Hayashi, F.H. B2B B2C Architecture for Smart Meters using IoT and Machine Learning: A Brazilian Case Study. In Proceedings of the 2020 International Conference on Smart Grids and Energy Systems (SGES), Perth, Australia, 23–26 November 2020. [CrossRef]
4. Hayashi, V.; Fujii, T.; Arakaki, R.; Amaral, H.; Souza, A. Boa Energia: Base de Dados Pública de Consumo Residencial com Qualidade de Dados. In *Anais de XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*; Sociedade Brasileira de Telecomunicações: Rio de Janeiro, Brazil, 2020.
5. Carrie Armel, K.; Gupta, A.; Shrimali, G.; Albert, A. Is disaggregation the holy grail of energy efficiency? The case of electricity. *Energy Policy* **2013**, *52*, 213–234. [CrossRef]
6. Humeau, S.; Wijaya, T.K.; Vasirani, M.; Aberer, K. Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households. In Proceedings of the 2013 Sustainable Internet and ICT for Sustainability (SustainIT), Palermo, Italy, 30–31 October 2013.
7. Martins, P.B.d.M.; Pinto, R.G.D.; Bittencourt, S.P. Load Disaggregation of Industrial Machinery Power Consumption Monitoring Using Factorial Hidden Markov Models. In Proceedings of the International Workshop on Non-Intrusive Load Monitoring (NILM), Austin, TX, USA, 7–8 March 2018; p. 6.
8. Lusis, P.; Khalilpour, K.R.; Andrew, L.; Liebman, A. Short-term residential load forecasting: Impact of calendar effects and forecast granularity. *Appl. Energy* **2017**, *205*, 654–669. [CrossRef]
9. Alanezi, A.; P Hallinan, K.; Elhashmi, R. Using Smart-WiFi Thermostat Data to Improve Prediction of Residential Energy Consumption and Estimation of Savings. *Energies* **2021**, *14*, 187. [CrossRef]
10. Villa, S.; Sassanelli, C. The Data-Driven Multi-Step Approach for Dynamic Estimation of Buildings' Interior Temperature. *Energies* **2020**, *13*, 6654. [CrossRef]
11. Kaur, M.J.; Mishra, V.P.; Maheshwari, P. The convergence of digital twin, IoT, and machine learning: Transforming data into action. In *Digital Twin Technologies and Smart Cities*; Springer: Berlin, Germany, 2020; pp. 3–17.

12. Fujii, T.Y.; Ruggiero, W.V.; do Amaral, H.L.; Hayashi, V.T.; Arakaki, R.; Khalil, K.A. Desafios para Aplicação de MLOps na Previsão do Consumo Energético. In Proceedings of the 2021 14th IEEE International Conference on Industry Applications (INDUSCON), São Paulo, Brazil , 15–18 August 2021; pp. 455–462.
13. Alsalemi, A.; Sardianos, C.; Bensaali, F.; Varlamis, I.; Amira, A.; Dimitrakopoulos, G. The role of micro-moments: A survey of habitual behavior change and recommender systems for energy saving. *IEEE Syst. J.* **2019**, *13*, 3376–3387. [CrossRef]
14. Alsalemi, A.; Himeur, Y.; Bensaali, F.; Amira, A.; Sardianos, C.; Varlamis, I.; Dimitrakopoulos, G. Achieving domestic energy efficiency using micro-moments and intelligent recommendations. *IEEE Access* **2020**, *8*, 15047–15055. [CrossRef]
15. Zehir, M.A.; Ortac, K.B.; Gul, H.; Batman, A.; Aydin, Z.; Portela, J.C.; Soares, F.J.; Bagriyanik, M.; Kucuk, U.; Ozdemir, A. Development and field demonstration of a gamified residential demand management platform compatible with smart meters and building automation systems. *Energies* **2019**, *12*, 913. [CrossRef]
16. Zhang, Y.; Srivastava, A.K.; Cook, D. Machine learning algorithm for activity-aware demand response considering energy savings and comfort requirements. *IET Smart Grid* **2020**, *3*, 730–737. [CrossRef]
17. Fathy, Y.; Jaber, M.; Nadeem, Z. Digital Twin-Driven Decision Making and Planning for Energy Consumption. *J. Sens. Actuator Netw.* **2021**, *10*, 37. [CrossRef]
18. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*; OTexts: Melbourne, Australia, 2021.
19. Rossana, R.J.; Seater, J.J. Temporal Aggregation and Economic Time Series. *J. Bus. Econ. Stat.* **1995**, *13*, 441–445.
20. Sprenger, J.; Weinberger, N. Simpson's Paradox. In *The Stanford Encyclopedia of Philosophy*, 2021th ed.; Zalta, E.N., Ed.; Metaphysics Research Lab, Stanford University: Palo Alto, CA, USA, 2021.
21. Kong, W.; Dong, Z.Y.; Hill, D.J.; Luo, F.; Xu, Y. Short-term residential load forecasting based on resident behaviour learning. *IEEE Trans. Power Syst.* **2018**, *33*, 2017–2018. [CrossRef]
22. Ben Taieb, S.; Hyndman, R.J. A gradient boosting approach to the Kaggle load forecasting competition. *Int. J. Forecast.* **2014**, *30*, 382–394. [CrossRef]
23. Veit, A.; Goebel, C.; Tidke, R.; Doblander, C.; Jacobsen, H.A. Household electricity demand forecasting—Benchmarking state-of-the-art methods. In Proceedings of the e-Energy 2014—Proceedings of the 5th ACM International Conference on Future Energy Systems, Cambridge, UK, 11–13 June 2014; pp. 233–234.
24. Gerossier, A.; Girard, R.; Bocquet, A.; Kariniotakis, G. Robust day-ahead forecasting of household electricity demand and operational challenges. *Energies* **2018**, *11*, 3503. [CrossRef]
25. The daily and hourly energy consumption and load forecasting using artificial neural network method: A case study using a set of 93 households in Portugal. *Energy Procedia* **2014**, *62*, 220–229. [CrossRef]
26. Kong, W.; Dong, Z.Y.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. *IEEE Trans. Smart Grid* **2019**, *10*, 841–851. [CrossRef]
27. Serrenho, T., Bertoldi, P. *Smart Home and Appliances: State of the Art*; Technical Report; Publications Office of the European Union: Luxembourg, 2019.
28. Amaral, H.L.; Maginador, J.A.; Ayres, R.M.; De Souza, A.N.; Gastaldello, D.S. Integration of consumption forecasting in smart meters and smart home management systems. In Proceedings of the SBSE 2018—7th Brazilian Electrical Systems Symposium, Niteroi, Brazil, 12–16 May 2018; pp. 1–6.
29. Wang, Y.; Chen, Q.; Hong, T.; Kang, C. Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. *IEEE Trans. Smart Grid* **2019**, *10*, 3125–3148. [CrossRef]
30. Tushar, W.; Saha, T.K.; Yuen, C.; Liddell, P.; Bean, R.; Poor, H.V. Peer-to-Peer Energy Trading With Sustainable User Participation: A Game Theoretic Approach. *IEEE Access* **2018**, *6*, 62932–62943. [CrossRef]
31. Pratt, A.; Krishnamurthy, D.; Ruth, M.; Wu, H.; Lunacek, M.; Vaynshenk, P. Transactive Home Energy Management Systems: The Impact of Their Proliferation on the Electric Grid. *IEEE Electrif. Mag.* **2016**, *4*, 8–14. [CrossRef]
32. Haben, S.; Ward, J.; Vukadinovic Greetham, D.; Singleton, C.; Grindrod, P. A new error measure for forecasts of household-level, high resolution electrical energy consumption. *Int. J. Forecast.* **2014**, *30*, 246–256. [CrossRef]
33. Stefan, A.; Athitsos, V.; Das, G. The move-split-merge metric for time series. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 1425–1438. [CrossRef]
34. Guen, V.L.; Thome, N. Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models. *arXiv* **2019**, arXiv:1909.09020.
35. Murphy, A.H. What Is a Good Forecast? An Essay on the Nature of Goodness in Weather Forecasting. *Weather Forecast.* **1993**, *8*, 281–293. [CrossRef]
36. Breck, E.; Cai, S.; Nielsen, E.; Salib, M.; Sculley, D. The ML test score: A rubric for ML production readiness and technical debt reduction. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; Volume 47, pp. 1123–1132.
37. Sugimura, P.; Hartl, F. Building a reproducible machine learning pipeline. *arXiv* **2018**, arXiv:1810.04570.
38. Senapathi, M.; Buchan, J.; Osman, H. DevOps Capabilities, Practices, and Challenges. In Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, Christchurch, New Zealand, 28–29 June 2018; ACM: New York, NY, USA, 2018; pp. 57–67.
39. Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.F.; Dennison, D. Hidden technical debt in machine learning systems. *Adv. Neural Inf. Process. Syst.* **2015**, *2015*, 2503–2511.

40. Google Cloud. *MLOps: Continuous Delivery and Automation Pipelines in Machine Learning*; Google LLC: Mountain View, CA, USA, 2020.
41. Guajardo, J.A.; Weber, R.; Miranda, J. A model updating strategy for predicting time series with seasonal patterns. *Appl. Soft Comput. J.* **2010**, *10*, 276–283. [CrossRef]
42. Kent, L.; Snider, C.; Hicks, B. Early stage digital-physical twinning to engage citizens with city planning and design. In Proceedings of the 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 23–27 March 2019; pp. 1014–1015.
43. Koreshoff, T.L.; Leong, T.W.; Robertson, T. Approaching a human-centred internet of things. In Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, Adelaide, Australia, 19–25 November 2013; pp. 363–366.
44. Chen, S.; Xu, H.; Liu, D.; Hu, B.; Wang, H. A vision of IoT: Applications, challenges, and opportunities with china perspective. *IEEE Internet Things J.* **2014**, *1*, 349–359. [CrossRef]
45. Eyre, J.; Freeman, C. Immersive Applications of Industrial Digital Twins. *Ind. Track EuroVR* **2018**, 11–20. Available online: https://publications.vtt.fi/pdf/technology/2018/T339.pdf (accessed on 29 October 2021).
46. Bezborodova, O.; Bodin, O.; Gerasimov, A.; Kramm, M.; Rahmatullov, R.; Ubiennykh, A. *«Digital Twin» Technology in Medical Information Systems*; Journal of Physics: Conference Series; IOP Publishing: Bristol, UK, 2020; Volume 1515, p. 052022.
47. Raes, L.; Michiels, P.; Adolphi, T.; Tampere, C.; Dalianis, T.; Mcaleer, S.; Kogut, P. DUET: A Framework for Building Secure and Trusted Digital Twins of Smart Cities. *IEEE Internet Comput.* **2021**. [CrossRef]
48. Kuller, M.; Kohlmorgen, F.; Karaoğlan, N.; Niemeyer, M.; Kunold, I.; Wöhrle, H. Conceptual design of a digital twin based on semantic web technologies in the smart home context. In Proceedings of the 2020 IEEE 3rd International Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE), Budapest, Hungary, 18–19 November 2020; pp. 000167–000172.
49. Maryasin, O. Home Automation System Ontology for Digital Building Twin. In Proceedings of the 2019 XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP), Samara, Russia, 3–6 September 2019; pp. 70–74.
50. Raggett, D. The web of things: Challenges and opportunities. *Computer* **2015**, *48*, 26–32. [CrossRef]
51. Goddard, M. The EU General Data Protection Regulation (GDPR): European regulation that has a global impact. *Int. J. Mark. Res.* **2017**, *59*, 703–705. [CrossRef]
52. Pinheiro, P.P. *Proteção de Dados Pessoais: Comentários à Lei n. 13.709/2018-LGPD*; Saraiva Educação SA: São Paulo, Brazil, 2020.
53. Arakaki, R.; Hayashi, V.T.; Ruggiero, W.V. Available and Fault Tolerant IoT System: Applying Quality Engineering Method. In Proceedings of the 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Istanbul, Turkey, 12–13 June 2020; pp. 1–6.
54. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
55. Burleyson, C.D.; Rahman, A.; Rice, J.S.; Smith, A.D.; Voisin, N. Multiscale effects masked the impact of the COVID-19 pandemic on electricity demand in the United States. *Appl. Energy* **2021**, *304*, 117711. [CrossRef]
56. Plotly Technologies Inc. *Dash*; Plotly Technologies Inc.: Montreal, QC, Canada, 2021.
57. Ashton, K. That 'internet of things' thing. *RFID J.* **2009**, *22*, 97–114.
58. Beck, K. *Test-Driven Development: By Example*; Addison-Wesley Professional: Upper Saddle River, NJ, USA, 2003.