energies

MDPI

Article

Edge-Based Short-Term Energy Demand Prediction

Alexios Lekidis and Elpiniki I. Papageorgiou *

Department of Energy Systems, University of Thessaly, Gaiopolis Campus, 41500 Larissa, Greece; alekidis@uth.gr
* Correspondence: elpinikipapageorgiou@uth.gr

Abstract: The electrical grid is gradually transitioning towards being an interconnected area of the smart grid, where embedded devices operate in an autonomous manner without any human intervention. An important element for this transition is the energy demand prediction, since the needs for energy have substantially increased due to the introduction of new and heavy consumption sources, such as electric vehicles. Accurate energy demand prediction, especially for short-term durations (i.e., minutes to hours), allows grid operators to produce the substantial amount needed to satisfy the demand–response equilibrium and avoid peak electricity load conditions that may also lead to blackouts in densely populated areas. However, to achieve such an accuracy level, machine learning (ML) models require extensive training with historical measurements, which is usually resource intensive (e.g., in memory and processing power). Hence, deriving accurate predictions for short-term energy demands is challenging due to the absence of external factors such as environmental data from different regions and seasons and categorical values such as bank/bridging holidays in the ML model. Additionally, existing work focuses on ML model execution on Cloud platforms, which usually does not satisfy the real-time requirements of grid operators for short-term energy demand predictions. To address these challenges, this article presents a new method that considers environmental factors and categorical values to build an energy profile for each consumer in a multi-access edge computing (MEC) framework. The method is also based on the Temporal Fusion Transformer (TFT) ML model, which allows it to learn the temporal dependencies of the gathered historical measurements and predict energy demands with satisfying accuracy. The method is applied to a home energy management system testbed containing photovoltaic systems, smart meters, sensors and actuators for detecting environmental factors (i.e., temperature, humidity and radiation) as well as energy storage systems as an additional energy supply source. The MEC framework is deployed in data concentrator devices where the TFT ML model is executed with low resource requirements, ensuring additional security as the data do not leave the location where they are produced.

Keywords: energy forecasting; multi-access edge computing; Temporal Fusion Transformer; home energy management system

Citation: Lekidis, A.; Papageorgiou, E.I. Edge-Based Short-Term Energy Demand Prediction. *Energies* **2023**, *16*, 5435. https://doi.org/10.3390/en16145435

Academic Editor: Lorand Szabo

Received: 2 June 2023
Revised: 5 July 2023
Accepted: 13 July 2023
Published: 17 July 2023

1. Introduction

Internet of Things (IoT) technologies allow low-resource embedded devices to use sensors, actuators and network interfaces for exchanging data and making decisions autonomously based on embedded intelligence, hence avoiding any human intervention. Recent developments in the Industrial Internet of Things [1] aim to enable automation [2] at a large scale by combining processes, devices and supporting technologies with advanced information and communication technology. The devices can vary from small wireless devices to large systems, communicating locally among themselves or over networks to cloud servers.

IoT technologies are applied in different domains, such as connected vehicles, healthcare and the electrical grid. Their application in the electrical grid aims to interconnect the loosely connected grid segments, i.e., the power production, distribution, transmission and consumption in households. The interconnection aims to increase intelligence and gradually form a highly integrated and autonomous smart grid [3]. The smart grid advances the traditional electrical grid by introducing IoT-based technologies, such as: (1) advanced metering infrastructure, which is based on smart meters, (2) distribution automation using sensors and remote switches to respond to power outages and optimize load balancing, (3) renewable energy source integration, such as solar and wind, (4) energy storage mechanisms to store excess energy during periods of low demand and release it during peak demand, (5) demand response scenarios to optimize energy consumption and reduce peak loads and finally, (6) the integration of electric vehicles and charging infrastructure resources.

All the technologies that are forming the smart grid, though, have a common denominator: the reduction in peak grid demands, which may lead to blackouts in entire neighborhoods and areas. To achieve such a reduction, the electrical grid operators, especially in the production and distribution segments, should be aware of all the historical consumption data from each area as well as the consumer behavioral patterns and characteristics. Then, based on the environmental factors, i.e., weather conditions, algorithms should be developed to forecast consumption in different periods. Specifically, electric energy forecasting can be performed for short-term, middle-term or long-term periods. These periods differ in terms of the planning potential, including the grid resources, that is offered for the electrical grid operators, i.e., short-term periods predict minutes to hours, medium-term periods usually predict days up to a couple of months and finally, the long term predicts periods larger than 3 months and up to one or several years.

Nevertheless, the main challenge for achieving accuracy in energy demand forecasting lies in the variation in meteorological conditions and the energy habits of consumers. To this end, smart meter data combined with an even more diverse set of data inputs from environmental sensors, including temperature, humidity and radiation measurements, can provide an unprecedented holistic view to grid operators and simultaneously enable better operational decisions, especially for areas with high energy demands, such as commercial/industrial buildings or locations with electric vehicle chargers. The challenge is augmented when other technologies of the smart grid are considered, such as solar photovoltaic (PV) and wind turbine renewable energy systems combined with energy storage systems [4]. These electricity production sources have a highly variable output that complicates grid management in order to achieve the required overall grid stability.

Several methods have been proposed for addressing this challenge, including statistical models such as the autoregressive integrated moving average (ARIMA) [5], ML models such as support vector machines [6], and ensemble methods [7], which combine these with environmental factor (e.g., weather) models. The accuracy of these models depends on the quality and the availability of gathered data, as well as the specific environmental conditions and consumer habits of the area to which these models are applied. Hence, the models should be calibrated and executed close to the area where the data are gathered. Additionally, forecasting models are currently trained and executed in centralized data centers, which may increase the processing time and the real-time availability of the forecasts for short-term periods. The recent adoption of multi-access edge computing (MEC) technologies [8] allows us to perform data analysis and computations locally at the edge level using dedicated embedded devices with processing, memory and storage layers. Furthermore, existing frameworks are being developed to support technologies such as the EdgeX Foundry [9].

In this article, a new prediction method is proposed to aid in the resolution of the challenges of short-term energy demand prediction. The method is based on data collected from different sources as well as their aggregation and forecasting analysis using a prediction model based on the Temporal Fusion Transformer (TFT) [10] and deployed in an edge device using an MEC architecture. The method is illustrated in a home energy management system (HEMS) testbed where different sensors for temperature, humidity and radiation, along with other smart grid components such as smart meters, a PV system and a battery storage system, are deployed. The HEMS testbed is then used for short-term load and generation forecasting, and the data being used for future prediction are smart meter and sensor measurements with a 10 min frequency interval, representing the result of energy consumption data gathering in the testbed throughout an entire year. Concretely, this article has the following contributions:

- A novel MEC-based framework for the automated collection of all the necessary data, including historical electricity consumption, temperature and humidity of each household, is used for training the prediction model.
- A prediction model for short-term energy demand prediction using as a basis the TFT architecture and ML mechanisms and deployed in an edge device, which ensures security as the data do not leave the location where they are produced.
- Validation of the proposed method by deploying the introduced framework and the prediction model in a HEMS testbed.

The rest of the article is organized as follows. Section 2 provides an overview of the energy forecasting methodology, including the forecasting categories and the forecasting focus, and analyzes related work in the forecasting field. Then, Section 3 presents the forecasting method by focusing on the developed framework as well as the prediction model with TFT. The method is afterwards validated in a HEMS testbed in Section 4, and a prediction of the electricity demand is derived through the conducted experiments. The benefits of the method along with its limitations are then compared in Section 5 against similar work for short-term energy demand forecasting. Finally, Section 6 provides conclusions and some perspectives for future work.

2. Background

2.1. Energy Forecasting Overview

Energy demand forecasting is an extremely important technology for electrical grid operators, including power plant production personnel, distribution system operators (DSOs) and further electricity market participants. Satisfying the demand–response equilibrium is vital for avoiding electricity demand peaks that would require significant time and effort to analyze and adequate electricity flexibility from the DSO's side to avoid consequences such as blackouts. Furthermore, in the current electrical grid state, existing battery storage systems can only sustain energy for a limited timeframe and also cannot cope with heavier load appliances, such as air conditioning systems, washing machines or electric vehicles. Hence, the electricity demand has to be satisfied in real time for different types of customers, including: (1) low-voltage households, (2) medium-voltage distribution areas and (3) high-voltage commercial and industrial business areas (e.g., manufacturing systems providers, hospitals and banks). For all consumer types, there are four main categories of forecasts based on the time period that is predicted ahead:

- Very short-range (minutes to 1 h) forecasts are required in monitoring a system (load or generation) to detect anomalies. The forecasting model is assumed to represent the normal operation of the system, and large deviations from the predicted value can be considered anomalies.
- Short-range forecasts (1 to 6 h) are useful in cases where the load is expected to fluctuate and additional (distributed) generation capacity must be brought in in real time. Additionally, a short-term load forecast (STLF) is very useful for energy retailers and distribution system operators.
- Medium-range forecasts (a few days up to 1–2 months ahead) are useful for demand-side management, where energy consumers are asked to modify their loads when demand peaks or generation excesses are expected. Furthermore, with the advent of energy auctions, predicting loads and generation capacities with a medium-term load forecast (MTLF) helps actors make decisions to minimize their financial risk. This is mainly useful for renewable sources that are highly variable.
- Long-range forecasts (longer than 3 months and up to years) are useful for grid resource planning. Specifically, a long-term load forecast (LTLF) is generally used

for planning and investment probability analysis, determining upcoming sites or acquiring fuel sources for production plants.

Based on the business operation of electricity markets on a daily basis, the most valuable category of electricity load forecasting is currently STLFs [11]. Figure 1 depicts a reference smart grid architecture based on [12], focusing on the exchanged data as well as illustrating where an STLF is positioned and applied in the architecture.

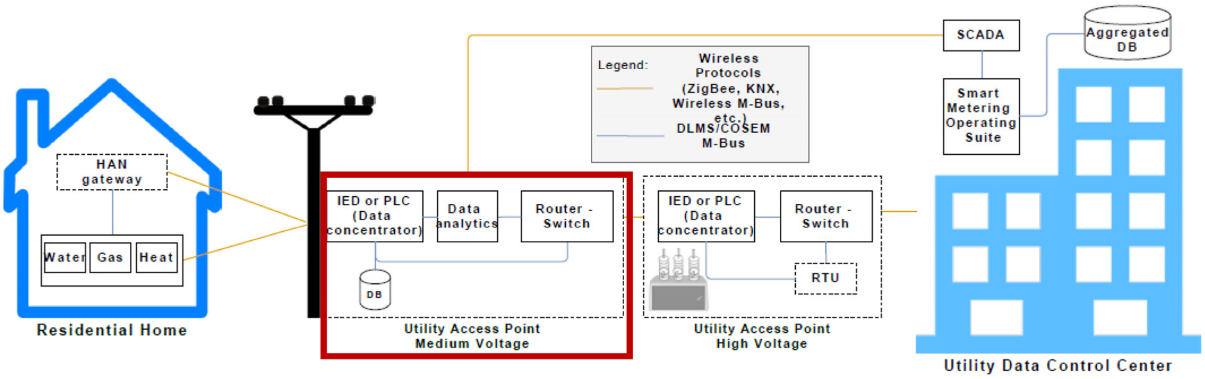

SCADA
Aggregated DB
Legend:
Wireless Protocols (ZigBee, KNX, Wireless M-Bus, etc.)
DLMS/COSEM M-Bus
Smart Metering Operating Suite
HAN gateway
Water
Gas
Heat
Residential Home
IED or PLC (Data concentrator)
Data analytics
Router - Switch
DB
Utility Access Point Medium Voltage
IED or PLC (Data concentrator)
Router - Switch
RTU
Utility Access Point High Voltage
Utility Data Control Center

Figure 1. Smart grid reference architecture (based on [12]) focusing on short-term energy forecasting (red highlighted box).

Specifically, STLFs are performed in areas near distribution transformers using components that are called data concentrators [13]. The concentrators gather energy consumption data from smart meters (i.e., current, voltage, active and reactive power). Smart meters are installed in each household within the neighborhood areas that are served by the distribution transformers. Concentrators have an embedded architecture with resource constraints on processing power and storage memory. Storage memory is used as a dedicated database for the pre-processing and analysis of the smart meter data. After an initial analysis and STLF model training and execution for the prediction, the data, along with the predictions, are forwarded to the utility data control center (Figure 1). In this control center, utilities have adequate resources to allow the training and execution of MTLFs and LTLFs, which require a considerably larger amount of historical data and hence heavier processing power and memory resources. Moreover, in all forecasting time periods, satisfactory prediction accuracy is achieved when the data are coupled with environmental data related to temperature, radiation and humidity.

Even with the availability of such data, though, STLF is quite challenging due to: (1) the variations in terms of the environmental conditions in the area where it is applied and (2) the electricity usage profile of each consumer household. Hence, the electricity usage is different for each working day of the week as well as during the weekend. Moreover, it also varies during the day, i.e., some consumers may usually schedule the use of heavy-load appliances such as electric vehicle chargers and washing machines during the night (electricity load and price are lower), but on occasions when they need these devices, they may also use them during the day. Furthermore, seasonal impacts also have a strong influence on electricity demand. As an example, during extremely hot or cold conditions, the heating or air conditioning are heavily used. Additionally, when household residents are absent during bank/bridging holidays, energy consumption is much lower.

Finally, to measure the energy demand prediction performance in all forecasting time periods, there are different metrics that are generally used for ML-based time series forecasting [14]. Of these metrics for energy demand forecasting, the selection is based on three criteria: (1) overall magnitude of the errors in the forecast (e.g., large errors have to be considered and minimized), (2) weight of outliers should be minimized and (3) easy interpretation of the metric results by the reader for the difference between actual and forecasted values. Based on these criteria, two basic metrics are selected for measuring the STLF prediction accuracy, which are presented in the following equations. Additionally, in these equations, y is the observed variable and $\hat{y}$ the corresponding predicted variable over a set of samples of size N:

(1) The *mean absolute error (MAE)* = $\frac{\sum_1^N |y-\hat{y}|}{N}$, which is the average magnitude of the forecast errors and is measured as the average of the absolute error values. Furthermore, the MAE illustrates the inaccuracy that is expected from the forecast on average [15]. Specifically, the lower the MAE value, the better the model. Hence, a model with MAE = 0 indicates that the forecast is error-free.

(2) The *root mean squared error (RMSE)* = $\sqrt{\frac{\sum_1^N (y-\hat{y})^2}{N}}$, which is the square root of the mean squared error (MSE), allowing us to measure the square root of the average of the squared differences between the actual and predicted values [15]. RMSE is always non-negative, and a value of 0 would indicate a perfect fit to the data. In general, the lower the RMSE, the better the fit of the model.

(3) The *mean absolute percentage error (MAPE)* = $\frac{1}{N}\sum_1^N \frac{|y-\hat{y}|}{y}$, which allows us to measure the average percentage difference between the actual and predicted values in a time series model [16]. Apart from considering this difference, it further divides it by the actual value, allowing us to compare forecasts across different time series as well as on different scales. Usually, the result of the division and the average computed value are provided as a percentage; hence, they are multiplied by 100%.

The following section provides background work in the energy demand forecasting field.

2.2. Related Work

The energy demand forecasting field has been extensively investigated with different ML-based methods in the literature over the last years. The related work in this field is summarized in Table 1 and includes the domain to which the forecasting is applied, the method used to perform the forecasting and the data used to train the models for forecasting.

Table 1. Summary of related work.

Paper	Domain	Method	Dataset
S. S. Rangapuram et al. [17]	Time series forecasting	Deep neural networks	Historical power consumption and traffic samples
A. Alaa et al. [18]	Time series forecasting	Deep neural networks	Healthcare samples
S. Makridakis et al. [19]	Time series forecasting	Deep neural networks	Various including finance and demographic
C. Fan et al. [20]	Time series forecasting	Recurrent neural networks with attention-based methods	Various including healthcare
S. Li et al. [21]	Time series forecasting	Recurrent neural networks with Transformer-based models	Historical power consumption and traffic samples
R. Wen et al. [22]	Time series forecasting	Ensemble methods	Historical power consumption and electricity price
J. Koutnik et al. [23]	Time series forecasting	Ensemble methods	Audio samples
B. Bosco et al. [24]	LTLF demand forecasting	Regression models	Historical power consumption and electricity price
J. Janczura et al. [25]	LTLF demand forecasting	Regression models	Historical power consumption and electricity price

Table 1. *Cont.*

Paper	Domain	Method	Dataset
S. Trueck et al. [26]	LTLF demand forecasting	Regression models	Historical power consumption and electricity price
C. Sigauke et al. [27]	MTLF demand forecasting	Univariate methods	Historical power consumption and electricity price
F. Lisi et al. [28]	LTLF demand forecasting	Regression models	Historical power consumption and electricity price
J. Taylor et al. [29]	MTLF demand forecasting	Exponential smoothing	Historical power consumption
J. Taylor et al. [30]	STLF demand forecasting	Exponential smoothing	Historical power consumption
A. Clements et al. [31]	MTLF demand forecasting	ARIMA	Historical power consumption
M. Ismail et al. [32]	STLF demand forecasting	ARIMA	Historical power consumption
J. Hinman et al. [33]	STLF demand forecasting	ARIMA	Historical power consumption, weather conditions
Y. Feng et al. [34]	MTLF demand forecasting	ARIMA	Historical power consumption, weather conditions
R. M. A. Weron et al. [35]	LTLF demand forecasting	ARIMA	Historical power consumption, weather conditions
E. Yukseltan et al. [36]	Energy demand forecasting	Parametric regression	Historical power consumption
Y. Yang et al. [37]	STLF demand forecasting	Ensemble methods	Historical power consumption
Albahli, S. et al. [38]	LTLF demand forecasting	Extreme Gradient Boosting (XGBoost)	Historical power consumption
Li, D et al. [39]	STLF demand forecasting	Temporal Fusion Transformer	Historical power consumption

The works that are presented in this table are also detailed in the following part. Specifically, deep neural networks (DNNs) exhibit considerably improved performance and accuracy in time series forecasting in comparison with traditional time series models [17–19]. Moreover, recurrent neural networks (RNNs) are enhanced with attention-based methods [20] and transformer-based models [21] to improve the prediction based on a deeper analysis of historical data over different time periods. Further improvements have been introduced recently in time series forecasting models based on ensemble methods combining the existing models with techniques that provide performance improvements and faster training times [22,23]. However, all these models do not consider the different types of external factors such as environmental conditions that have a significant impact on forecasting accuracy.

By focusing on the energy demand forecasting challenge, significant literature work has been performed for all the categories of forecasts using a variety of models [24–28]. Specifically, the authors in [29,30] have proposed exponential smoothing techniques that capture dependencies in energy demand. STLF and MTLF demand forecasting models are based on time series of historical data for energy consumption, where ARIMA models are applicable. Different variations of ARIMA have been proposed in the literature for STLFs, including the Double Seasonal ARIMA (DSARIMA) model, the Double Holt (Winters (D-HW) model) and multiple equations time series (MET) [31,32]. Forecasting accuracy and performance are very crucial criteria to ensure the effectiveness of these models; hence, existing work focuses on external factors such as environmental conditions, which are included in the ARIMA models to improve these criteria [33–35]. Apart from ARIMA, though, regression models can also be used, such as the parametric regression model proposed in [36] to forecast the energy in the Turkish market. Finally, ensemble methods are very promising for energy demand forecasting in terms of accuracy and performance, such as in [37], which introduces a combination of SARIMA and the back propagation neural network (BPNN) model.

Finally, related to the dataset availability for training ML-based models, there are several electric energy datasets in the public domain. Unfortunately, most deal with long-term, large aggregates of power demand or generation. For demand modeling, the available datasets contain no other exogenous information (e.g., weather, user behavior, household occupancy, etc.), therefore limiting the possible accuracy of the models.

3. Proposed Method for Energy Demand Forecasting

The main challenge that is faced in forecasting energy demand lies in the fact that forecasting a future value is usually not enough. Specifically, actual values depend on many environmental parameters that can be seasonal or stochastic. Since decisions on the electrical grid operation have very high economic value and may impact a large percentage of the population, we need to limit our risk. This can be accomplished by knowing the prediction interval of the forecast. This interval allows quantification of the accuracy of the prediction, including probabilistic upper and lower bounds for estimating the future energy demand value as an output of the edge-based prediction model. Even though the forecasting models that are also provided in the related work of Section 2.2 may have an interval that is close to the future value, an STLF requires extremely high accuracy to be an effective tool for grid operators. Hence, the challenges that are present in the development of a forecasting model are:

- The energy demand is formed based on a multi-variate time series since, apart from the historical energy consumption measurements, it depends on a variety of evolving external factors (i.e., environmental values) such as temperature, humidity, radiation, etc.
- Categorical values should also be considered in the model, such as bank/bridging holiday periods that influence the historical energy consumption.
- The prediction accuracy that the model should reach has to be high, but its specific level must be carefully selected. In particular, even though the accuracy is increased by incorporating external factors, it might not be adequate for an STLF utility operation tool, and hence the model might need to be retrained and executed with the latest energy consumption measurements. However, very frequent model execution may lead to exhaustion of available resources (e.g., memory and processing power), especially at the data concentrator level.

Considering these challenges, we have developed a framework that can be deployed at the data concentrator level and whose architecture is presented in Section 3.1. The framework relies on an ML-based prediction model, which is detailed in Section 3.2.

3.1. Framework Architecture

The framework that was developed for providing a potential solution to the challenge of energy demand forecasting is based on a MEC-based architecture along with the underlying technologies [8]. The reason for choosing such technologies is that the data concentrators have resource constraints at the processing, storage and network data exchange levels. MEC technologies provide a software architecture where the data gathered from embedded devices (i.e., sensors/actuators) are processed and then actual verdicts are reached. Such verdicts subsequently lead to autonomous actions, such as informing energy utility operators about potential loads in certain areas and performing corrective actions to avoid them. Moreover, by using the MEC architecture, data storage can be performed at the data concentrator level, as only the necessary measurements for training the prediction model (Section 3.2) are kept, thus avoiding memory or cache overflow issues. Overall, the framework allows STLF applications to be built and deployed in data concentrators by maintaining a real time and critical operations.

Furthermore, the presence of an edge platform ensures less communication latency since the edge resources are closer to the embedded devices and the data are only exchanged with them. Additionally, performance improvements are observed due to the reduced processing actions for the embedded IoT devices. In this case, the processing is handled by edge resources, and the processing result is returned to the embedded devices. Hence, in this work, the developed framework is based on an edge platform, where the forecasting is performed using a prediction model (Section 3.2). The architecture of the developed framework is based on three main layers (depicted in Figure 2):

(1) *Device layer*: This layer consists of smart meters and embedded devices that are employed in order to gather data such as temperature and humidity, which aid in training the prediction model. Specific data that are collected also include: (a) inside and outside surface temperatures as well as indoor air temperature; (b) recorded schedules of occupants, equipment and lighting; (c) recorded weather profiles and household heating and cooling demands.

(2) *Service layer*: This layer is used to provide auxiliary services for the framework, including the storage of the collected data. Specifically, the historical data from the smart meters and the embedded devices of the device layer are stored in dedicated storage databases for later processing, which enables the training of the prediction model based on the occupants' behavior. MQ Telemetry Transport (MQTT) topics are used to subscribe in these databases and are implemented using the Mosquitto client and server mechanisms [40]. Additionally, this layer is used to provide metadata about the framework related to its capabilities (CPU, storage, networking, etc.) as well as the framework registries and configurations.

(3) *Application layer*: This layer includes the prediction model, which uses dedicated subscription topics to consume the data that are available in the service layer and then initiate the training phase for energy demand predictions.

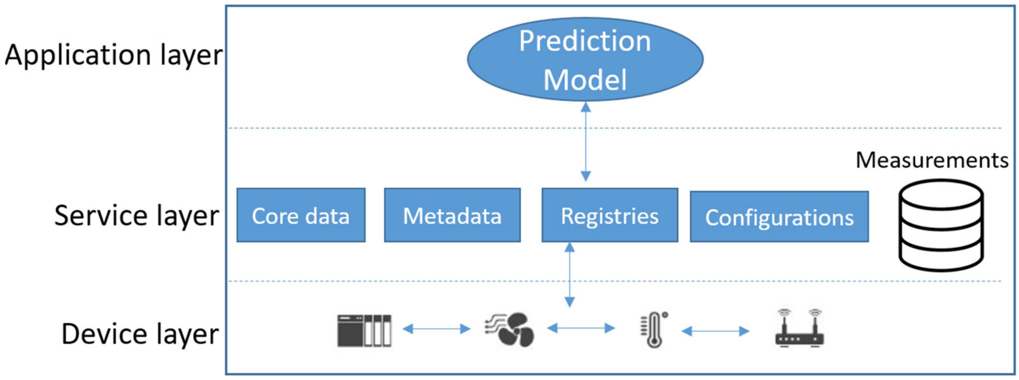


Figure 2. Proposed method architecture.

Overall, the layers of the architecture are implemented using EdgeX Foundry [9], an open-source framework from the Linux Foundation that enables the development of interoperable edge computing solutions for the Internet of Things (IoT) ecosystem. Furthermore, it provides a standardized platform for managing and orchestrating edge devices, data and applications, allowing for seamless integration and interoperability across different hardware and software components.

3.2. Prediction Model

The prediction model that is proposed in this article is based on the Temporal Fusion Transformer (TFT), which is a prominent ML-based solution developed by Google for energy demand forecasting [10]. The TFT is a deep learning model designed for time series forecasting. It combines two powerful architectures: (1) Transformer and (2) Temporal Fusion Decoder (TFD). Figure 3 shows the high-level architecture of the Temporal Fusion Transformer (TFT), as presented in [10].

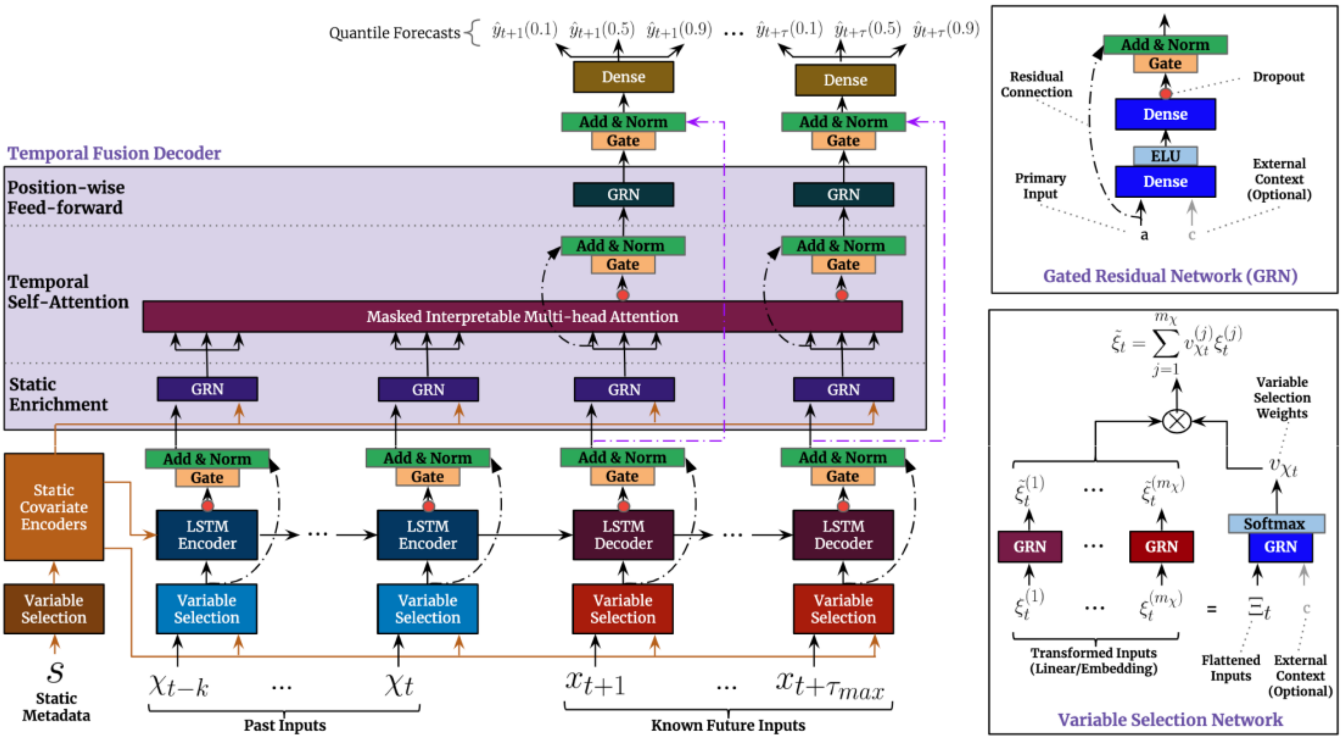

Quantile Forecasts
$\hat{y}_{t+1}(0.1)$ $\hat{y}_{t+1}(0.5)$ $\hat{y}_{t+1}(0.9)$ … $\hat{y}_{t+\tau}(0.1)$ $\hat{y}_{t+\tau}(0.5)$ $\hat{y}_{t+\tau}(0.9)$
Dense
Add & Norm
Gate
GRN
Temporal Fusion Decoder
Position-wise Feed-forward
Temporal Self-Attention
Masked Interpretable Multi-head Attention
Static Enrichment
Static Covariate Encoders
LSTM Encoder
LSTM Decoder
Variable Selection
s
Static Metadata
χ_{t-k} … χ_t
x_{t+1} … $x_{t+\tau_{max}}$
Past Inputs
Known Future Inputs
Residual Connection
Dropout
Dense
ELU
Primary Input
External Context (Optional)
a
c
Gated Residual Network (GRN)
$\tilde{\xi}_t = \sum_{j=1}^{m_\chi} v_{\chi_t}^{(j)} \tilde{\xi}_t^{(j)}$
Variable Selection Weights
v_{χ_t}
$\tilde{\xi}_t^{(1)}$ … $\tilde{\xi}_t^{(m_\chi)}$
Softmax
GRN
$\xi_t^{(1)}$ … $\xi_t^{(m_\chi)}$ = Ξ_t
c
Transformed Inputs (Linear/Embedding)
Flattened Inputs
External Context (Optional)
Variable Selection Network

Figure 3. The TFT prediction model architecture (based on [10]).

Initially, the Transformer architecture can capture long-range dependencies in sequential data. It consists of an encoder–decoder structure with self-attention mechanisms, which allow the model to focus on different parts of the input sequence when making predictions. Then, the Temporal Fusion Decoder (TFD) is specifically designed for time series forecasting. Hence, it incorporates several modules to handle the temporal nature of the data, including autoregressive encoding, temporal convolutional layers and gating mechanisms.

The TFD model can capture both short-term and long-term temporal dependencies in the data, enabling accurate predictions. The TFT uses the Transformer to encode the temporal features of the time series data and generate context vectors, which are then fed into the TFD for forecasting. The TFD takes the context vectors as input and applies temporal convolutions and gating mechanisms to generate the final predictions. Therefore, by combining the Transformer and TFD architectures, the TFT prediction model can capture both global and local temporal dependencies in the time series data, making it well-suited for accurate forecasting tasks. Additionally, to ensure accuracy in forecasting the energy demand, the model that is employed for the TFD is based on long short-term memory network (LSTM) layers [41]. LSTM is chosen as the technique due to its capability of learning long-term dependencies.

The TFD uses a series of layers for learning the temporal relationships of the input data, including energy consumption measurements and environmental conditions from temperature, humidity and radiation sensors/actuators. More specifically, utilizing local contexts in time series data can enhance the identification of significant points, including anomalies, change points and cyclical patterns, by considering their surrounding values. By constructing features that incorporate pattern information along with individual data points, attention-based architectures can achieve better performance improvements.

For instance, in [42], a sole convolutional layer is employed to enhance locality by extracting local patterns using a consistent filter across the entire time span. Nevertheless, this methodology may not be suitable when dealing with cases where the observed inputs differ in terms of the number of past and future inputs, which is common for energy consumption measurements as each household exhibits a different consumption pattern and this may also vary over time. To overcome this challenge, a sequence-to-sequence model is utilized that inherently accommodates these variations by feeding data into both the encoder and decoder components. Subsequently, a collection of uniform temporal features is produced, which are then fed as inputs into the TFD. To ensure comparability with widely employed sequence-to-sequence baselines, our method is based on the use of an LSTM encoder–decoder, although alternative models could also be considered. This approach also serves as a substitute for conventional positional encoding, offering a suitable inductive bias to preserve the temporal order of the input measurements.

Following the linear transformation of the output obtained from the TFD, the generation of quantile forecasts takes place. These forecasts are designed as prediction intervals that encompass future time intervals in addition to point forecasts. To achieve this, multiple percentiles (such as the 10th, 50th and 90th) are simultaneously predicted at each time step.

4. Demand Forecasting in a Home Energy Management System

A home energy management system (HEMS) was set up for testing the proposed method and the accuracy of the TFT prediction model presented in Section 3. The HEMS system includes solar plants (photovoltaics), temperature and humidity sensors, as well as energy storage batteries (lithium iron phosphate - LFP technology) with a battery management system (BMS) as well as direct current/alternate current (DC/AC) inverters for energy storage. The batteries were initially manufactured by the Sunlight Group (https://www.the-sunlight-group.com/en/global/), a Renewable Energy Semiconductor Manufacturing company industrial with factories in Athens and Xanthi, Greece.

A hybrid home resident scenario consisting of a real resident home as well as multiple emulated ones is used to demonstrate a complete smart grid energy flow. Specifically, the flow starts with energy production and continues with its allocation to the distribution substations until it is finally consumed by scenario residents. Moreover, the BMS system ensures that the energy equilibrium between demand and response is always sustained and that the energy from batteries is not drained. Additionally, apart from storing the energy, the batteries also allow the residents to use the energy in real time, for example, when sunlight is present, and the photovoltaics are charging them. Moreover, the HEMS testbed uses smart meters for measuring energy consumption and derives profiles for each resident. On the production side, the HEMS system is fed loads from the electrical grid as well as the home PV system. Furthermore, controllers are also used, such as programmable logic controllers and a supervisory control and data acquisition (SCADA) system, for diagnostics and fault detection. The architectural overview of the HEMS testbed is illustrated in Figure 4.

The EdgeX Foundry framework for the collection of sensor data from the temperature and humidity sensors, as well as the prediction model for the HEMS testbed, is deployed in the command and control (C&C) workstation (Figure 4). The HEMS system allows monitoring of the building behavior (receiving information and statistics) for the consumed/produced energy. Our target is to receive detailed knowledge about energy consumption and production profiles and the smart control of the household via the proposed method. Additionally, the deployment of the EdgeX Foundry framework enables optimal scheduling over the appliances of the HEMS system. The dashboard interface of the EdgeX Foundry framework is also depicted in Figure 5, which is setup to receive the measurements from the temperature and humidity sensors of the HEMS testbed.

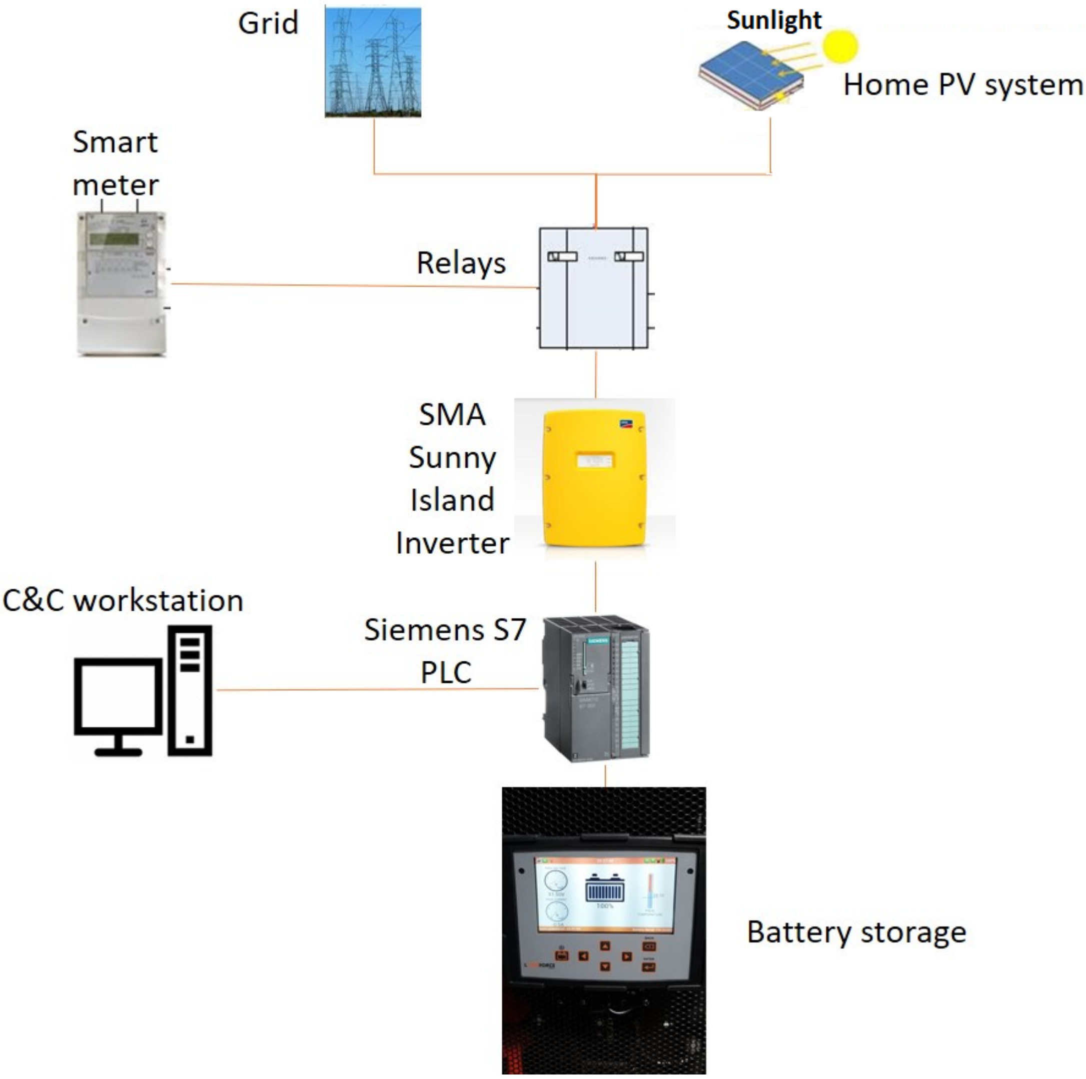
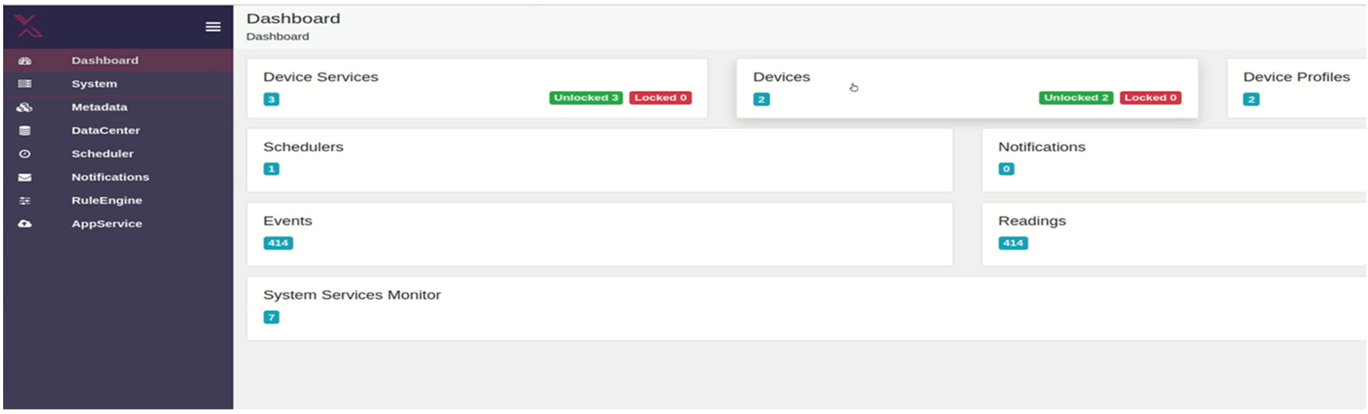


Figure 4. Home energy management system testbed.



Dashboard
Dashboard

Device Services 3 Unlocked 3 Locked 0

Devices 2 Unlocked 2 Locked 0

Device Profiles 2

Schedulers 1

Notifications 0

Events 414

Readings 414

System Services Monitor 7

Figure 5. EdgeX Foundry dashboard for the HEMS testbed.

The MEC framework was deployed in a data concentrator device in the HEMS testbed, which included a 1.66 GHz dual core processor, 3 GB of RAM and 256 GB of SSD storage memory.

4.1. Dataset Preparation

This section describes how the input data for training the prediction model were first gathered and then explains the pre-processing step that was followed in order to select the parameters employed for training.

4.1.1. Input Data Collection

The input data gathered for training the prediction model are based on the energy supply ID of each household over a period of the last four years. This selection was made (1) to demonstrate the method's scalability in terms of data storage within an edge device (i.e., a data concentrator) and (2) to have sufficient measurements for training the TFT model, which also allows better accuracy in the predictions. Moreover, they include all the latest measurements from the HEMS testbed by using the MQTT subscription topics of EdgeX Foundry. The gathered data are divided into three categories:

(1) *Sensor data*, gathered from the sensors (i.e., temperature, humidity and radiation);
(2) *Smart meter measurements*, from the smart meters (e.g., active/reactive power);
(3) *Categorical values*, which are related to geographical data from each household and significantly impact the consumer energy profile. Such data are gathered based on manual consumer questionnaire answers upon deploying the proposed framework in their household.

The reactive power of the second category is computed based on active power by including reactive elements like inductive or capacitive loads in the household. It is measured in volt-ampere reactive (VAR) and calculated based on the reactive elements present in the household. The power factor indicates the ratio of real power (active power) when compared with the total power (apparent power) in an electrical system. It ranges between 0 and 1, where a power factor of 1 signifies a purely resistive load and a power factor less than 1 indicates the presence of reactive elements. Concretely, it is calculated by the following equation:

$$\text{Reactive Power (RP)} = \text{S } x \sqrt{1 - \text{PF}^2)}$$

where S indicates the apparent power (measured in volt-amperes), which is equal to the active power (P) divided by the power factor (PF). Specifically:

$$\text{Apparent Power (S)} = \frac{\text{P}}{\text{PF}}$$

The average power factor for many household appliances and devices is around 0.9 to 1.0. Specifically, power factors close to 1.0 indicate that the load is primarily resistive, meaning it consumes mostly active power and has minimal reactive power.

Before proceeding to data preprocessing, an exploratory data analysis (EDA) step was followed. During the EDA, data are analyzed and visualized to understand patterns, explore correlations and dependencies between the variables and summarize the main characteristics of the dataset. Then, the feature selection process takes place, where a subset of the features that are the most informative and have a significant contribution to the STLF prediction model is selected. This reduces the dimensionality of the dataset and improves the model's performance by focusing on the most important features.

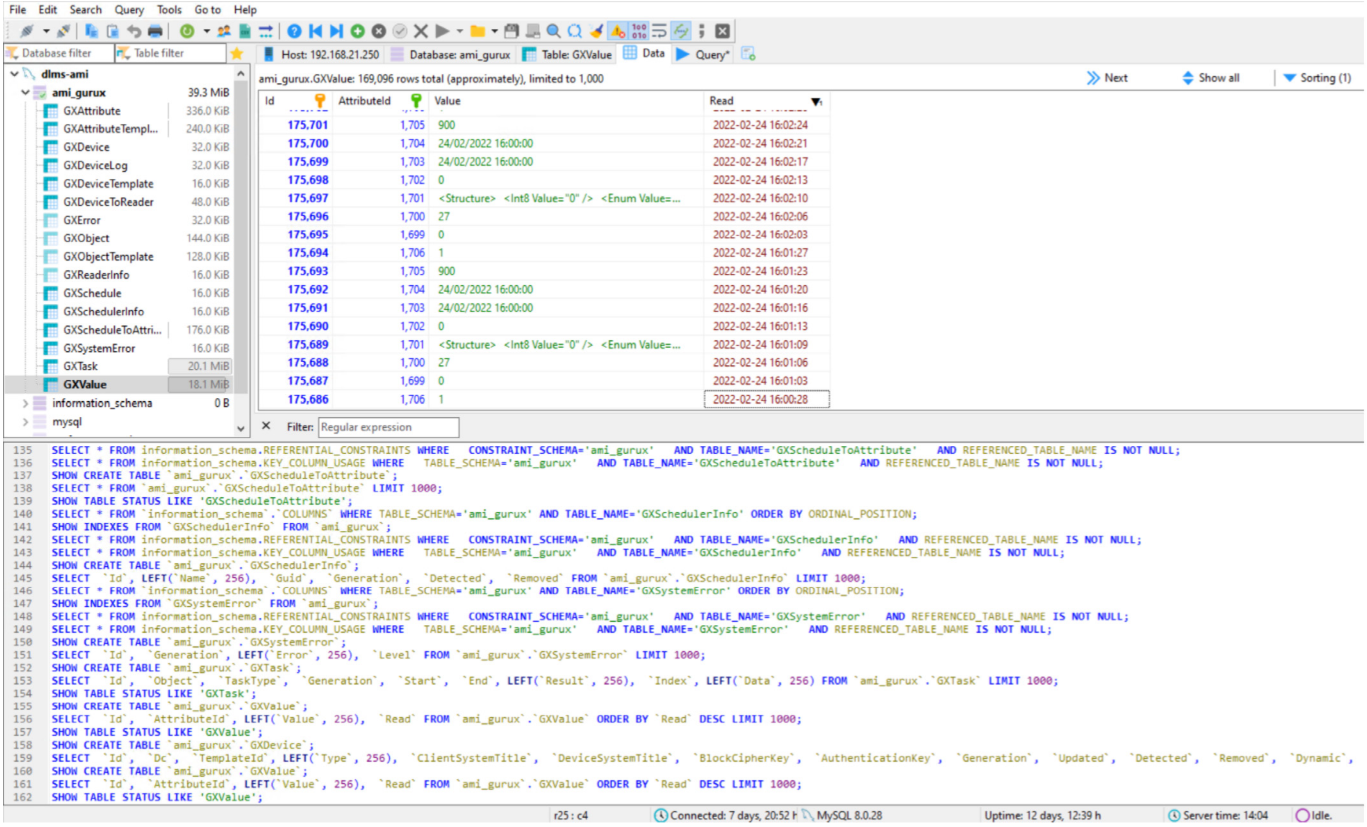
Upon finishing the EDA analysis, the input parameters for the prediction model are selected. The parameters that constitute the STLF dataset, along with their sample values, are provided in Table 2. The parameters are divided into the aforementioned categories and the common reference for them is the energy supply ID, which is unique for each household and provided as an anonymized example value in the table for security and data privacy issues.

Table 2. Dataset input parameters overview.

Parameter	Category	Description	Example
Energy supply ID	Generic	Unique energy supply ID (anonymized)	4qxiUAKo2BHN8p/RXfiXag==
Timestamp	Sensor data	Date and time of measurement	3
Indoor temperature	Sensor data	Indoor household temperature	22 degrees Celsius
Outdoor temperature	Sensor data	Outdoor air dry-bulb temperature	32 degrees Celsius
Relative indoor humidity	Sensor data	The humidity percentage inside the household	30%
Relative outdoor humidity	Sensor data	The humidity percentage in the external environment	40%
Radiation	Sensor data	Global horizontal radiation	800 watts per square meter (W/m^2)
Active power	Smart meter measurements	The real energy consumption from the household in kilowatts (kW)	5910 kW (1600 watts (morning) + 1250 watts (afternoon) + 2700 watts (evening) + 360 watts (night))
Reactive power	Smart meter measurements	The unused power in the lines that is consumed by the household in kilowatts (kW)	2858.51 VAR
Voltage	Smart meter measurements	Average voltage (volts)	230 V
Current intensity	Smart meter measurements	Average current intensity (amperes)	25,695 A
Electrical appliances	Categorical values	List of basic electrical appliances in the household	Electric hobs Oven Refrigerator Washing machine TV
Primary heating system	Categorical values	The type of the primary heating appliance the household uses	Air condition
House year built	Categorical values	The year the house was built	1975
Floor number	Categorical values	The house floor	2nd floor
Heating water appliance	Categorical values	The appliance used to heat water	Electric water heater
Household type	Categorical values	The type of the building	Detached house

For the above table, the reactive power was calculated based on an average active power consumption of a household (5910 kW). First, the apparent power (S) is computed as 6566.67 VA using a 0.9 power factor. Then, using the above formula, the reactive power is computed as 2858.51 VAR.

Given that the data were collected from the sensors and the smart meters every minute, there were 5,256,000 records per year for each household, as well as 7 more records including six categorical values and one generic value for the energy supply ID. Each record was occupied by 50 bytes of storage, thus resulting in a total of 250 MB of data storage for each household. Moreover, as the datasets were collected for the last four years, a total size of 1 GB was required for storage memory by the data concentrator device where the proposed MEC framework was deployed. Such a size was adequate for storing and later processing the data from the HEMS testbed. The datasets were then placed in a PostgreSQL database of the storage layer, a fragment of which is presented in Figure 6.



dlms-ami	
ami_gurux	39.3 MiB
GXAttribute	336.0 KiB
GXAttributeTempl...	240.0 KiB
GXDevice	32.0 KiB
GXDeviceLog	32.0 KiB
GXDeviceTemplate	16.0 KiB
GXDeviceToReader	48.0 KiB
GXError	32.0 KiB
GXObject	144.0 KiB
GXObjectTemplate	128.0 KiB
GXReaderInfo	16.0 KiB
GXSchedule	16.0 KiB
GXSchedulerInfo	16.0 KiB
GXScheduleToAttri...	176.0 KiB
GXSystemError	16.0 KiB
GXTask	20.1 MiB
GXValue	18.1 MiB
information_schema	0 B
mysql	

ami_gurux.GXValue: 169,096 rows total (approximately), limited to 1,000

Id	AttributeId	Value	Read
175,701	1,705	900	2022-02-24 16:02:24
175,700	1,704	24/02/2022 16:00:00	2022-02-24 16:02:21
175,699	1,703	24/02/2022 16:00:00	2022-02-24 16:02:17
175,698	1,702	0	2022-02-24 16:02:13
175,697	1,701	<Structure> <Int8 Value="0" /> <Enum Value=...	2022-02-24 16:02:10
175,696	1,700	27	2022-02-24 16:02:06
175,695	1,699	0	2022-02-24 16:02:03
175,694	1,706	1	2022-02-24 16:01:27
175,693	1,705	900	2022-02-24 16:01:23
175,692	1,704	24/02/2022 16:00:00	2022-02-24 16:01:20
175,691	1,703	24/02/2022 16:00:00	2022-02-24 16:01:16
175,690	1,702	0	2022-02-24 16:01:13
175,689	1,701	<Structure> <Int8 Value="0" /> <Enum Value=...	2022-02-24 16:01:09
175,688	1,700	27	2022-02-24 16:01:06
175,687	1,699	0	2022-02-24 16:01:03
175,686	1,706	1	2022-02-24 16:00:28

```
SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE   CONSTRAINT_SCHEMA='ami_gurux'   AND TABLE_NAME='GXScheduleToAttribute'   AND REFERENCED_TABLE_NAME IS NOT NULL;
SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE   TABLE_SCHEMA='ami_gurux'   AND TABLE_NAME='GXScheduleToAttribute'   AND REFERENCED_TABLE_NAME IS NOT NULL;
SHOW CREATE TABLE `ami_gurux`.`GXScheduleToAttribute`;
SELECT * FROM `ami_gurux`.`GXScheduleToAttribute` LIMIT 1000;
SHOW TABLE STATUS LIKE 'GXScheduleToAttribute';
SELECT * FROM `information_schema`.`COLUMNS` WHERE TABLE_SCHEMA='ami_gurux' AND TABLE_NAME='GXSchedulerInfo' ORDER BY ORDINAL_POSITION;
SHOW INDEXES FROM `GXSchedulerInfo` FROM `ami_gurux`;
SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE   CONSTRAINT_SCHEMA='ami_gurux'   AND TABLE_NAME='GXSchedulerInfo'   AND REFERENCED_TABLE_NAME IS NOT NULL;
SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE   TABLE_SCHEMA='ami_gurux'   AND TABLE_NAME='GXSchedulerInfo'   AND REFERENCED_TABLE_NAME IS NOT NULL;
SHOW CREATE TABLE `ami_gurux`.`GXSchedulerInfo`;
SELECT  `Id`, LEFT(`Name`, 256),  `Guid`,  `Generation`,  `Detected`,  `Removed` FROM `ami_gurux`.`GXSchedulerInfo` LIMIT 1000;
SELECT * FROM `information_schema`.`COLUMNS` WHERE TABLE_SCHEMA='ami_gurux' AND TABLE_NAME='GXSystemError' ORDER BY ORDINAL_POSITION;
SHOW INDEXES FROM `GXSystemError` FROM `ami_gurux`;
SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE   CONSTRAINT_SCHEMA='ami_gurux'   AND TABLE_NAME='GXSystemError'   AND REFERENCED_TABLE_NAME IS NOT NULL;
SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE   TABLE_SCHEMA='ami_gurux'   AND TABLE_NAME='GXSystemError'   AND REFERENCED_TABLE_NAME IS NOT NULL;
SHOW CREATE TABLE `ami_gurux`.`GXSystemError`;
SELECT  `Id`,  `Generation`, LEFT(`Error`, 256),  `Level` FROM `ami_gurux`.`GXSystemError` LIMIT 1000;
SHOW CREATE TABLE `ami_gurux`.`GXTask`;
SELECT  `Id`,  `Object`,  `TaskType`,  `Generation`,  `Start`,  `End`, LEFT(`Result`, 256),  `Index`, LEFT(`Data`, 256) FROM `ami_gurux`.`GXTask` LIMIT 1000;
SHOW TABLE STATUS LIKE 'GXTask';
SHOW CREATE TABLE `ami_gurux`.`GXValue`;
SELECT  `Id`,  `AttributeId`, LEFT(`Value`, 256),  `Read` FROM `ami_gurux`.`GXValue` ORDER BY `Read` DESC LIMIT 1000;
SHOW TABLE STATUS LIKE 'GXValue';
SHOW CREATE TABLE `ami_gurux`.`GXDevice`;
SELECT  `Id`,  `Dc`,  `TemplateId`, LEFT(`Type`, 256),  `ClientSystemTitle`,  `DeviceSystemTitle`,  `BlockCipherKey`,  `AuthenticationKey`,  `Generation`,  `Updated`,  `Detected`,  `Removed`,  `Dynamic`,
SHOW CREATE TABLE `ami_gurux`.`GXValue`;
SELECT  `Id`,  `AttributeId`, LEFT(`Value`, 256),  `Read` FROM `ami_gurux`.`GXValue` ORDER BY `Read` DESC LIMIT 1000;
SHOW TABLE STATUS LIKE 'GXValue';
```

r25 : c4 | Connected: 7 days, 20:52 h | MySQL 8.0.28 | Uptime: 12 days, 12:39 h | Server time: 14:04 | Idle.

Figure 6. Storage layer database entries for the proposed framework (visualized through the GuruX AMI library) [42].

The measurements in this figure are also taken from the GuruX AMI library [42], which is also deployed in the HEMS testbed for storing measurements and scheduling data and events from the smart meters. The forecasted values from the prediction model can also be presented in the AMI library, as they are linked to energy measurements, i.e., the forecasted active and reactive power for the consumption of the HEMS testbed.

As a final remark and to facilitate the reader's understanding, the selection of the dataset timeframe that was made for the HEMS testbed, i.e., four years, was indicative of sufficient training measurements. The framework has the ability to stop data collection and start retraining based on user input, which means that it can be performed in periods less than a year, i.e., on a monthly or even daily data collection basis.

4.1.2. Data Preprocessing

Upon data gathering from the sensors/actuators and the smart meters, necessary preprocessing is applied, which is a vital step in data analysis. Entries with duplicate or incomplete values are removed as they may give an incorrect interpretation of the overall statistics. Outliers and inconsistent data points are removed as well. Data preprocessing involves steps to transform or encode the data so that it can be easily manipulated by a machine. This ensures that the framework and the underlying model are accurate and precise in their predictions, as the features of the data are easily interpreted.

Then, we performed a correlation analysis to identify if the parameters of the dataset are highly correlated with each other, in which case a multivariate method would be used as the features are likely to be affected by the same underlying patterns and outliers.

Following the correlation analysis, the four-year household dataset is split into a three-year training set and a one-year test set and down-sampled into 10 min increments to match the smart meter dataset. Hence, it was split into 80% training and 20% testing. Splitting was performed to prevent overfitting as well as to evaluate the prediction model's accuracy. Then, we trained our proposed model on the data and validated it on the testing data.

Moreover, the dataset was provided to the TFT prediction model using the parameters of Table 1 as input variables and the forecasted active and reactive power for the consumption of the HEMS testbed as output variables. The model was deployed within the EdgeX Foundry framework, and relevant MQTT topic listening processes ran constantly to receive new data from the sensors and the smart meters. Whenever such data are received, the framework initiates a new procedure to store them in the database of the storage layer and form a new dataset, which will later be used as an up-to-date version for training the TFT model. Afterwards, the TFT can initiate predictions based on the new dataset.

4.2. Experiments

The prediction model is assumed to be executed at midnight every day and produces a forecast for the next 24 h (144 different 10-min intervals). Throughout the experiments, we ran the model with 432 time-steps, 287 past samples, 1 current sample and 144 future samples. These correspond to 2 days of 10 min intervals for past and current observations and 1 day of 10 min intervals for prediction. The prediction model uses a stacked architecture with three stacked layers, while the attention layer has four heads. Due to the large memory requirement for explainability processing and the use of the edge computing-based framework, the model was not able to rank parameters in terms of their contribution to the predictions. However, they are present and have been considered by the prediction model. Furthermore, apart from a data concentrator device, we also deployed the framework on a Cloud platform that is also present in the infrastructure of the HEMS testbed.

For the considered testbed, two types of forecasts are produced based on the conducted experiments:

(1) Active and reactive power forecasts in 10 min intervals for a whole day. The model is executed at midnight every day, producing a forecast for the next day in 10 min intervals. This forecast would be useful to plan intra-day actions like storage allocation, power source management for grid balancing, etc.

(2) Daily energy production forecasts. These would be useful for energy auctions and grid balancing.

Figure 7 illustrates the predictions for 10 min power consumption as well as energy consumption for every day of the last year in the dataset produced. Furthermore, the 50% quantile prediction is considered the actual prediction, while the 10% and 90% quantiles give the margin of possible power variation.

The predicted power demand of Figure 7 would be useful in a demand–response system since it predicts the max and min power demands as well as the (approximate) time they happen. In this case, consumers could be alerted to a pending rate change.

Additionally, Figure 8 provides insights into calculating the energy demand for daily energy auctions or simply for making decisions to bring in new power sources in order to balance the grid. Each data point in this figure represents the sum of all points in forecasts, as in Figure 7 for every day of the year. The resulting MAE obtained during the execution of the prediction model was 68 kilowatt hours (kWh) and the RMSE was 86 kWh. Furthermore, the MAPE for the prediction difference according to the actual value was 87.88%.

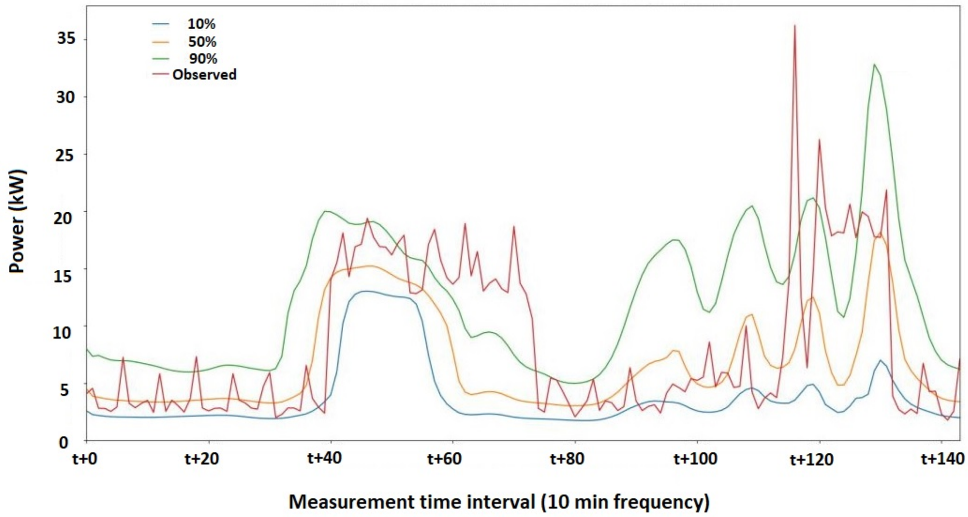


Figure 7. HEMS testbed power consumption (based on 10 min smart meter measurement intervals).

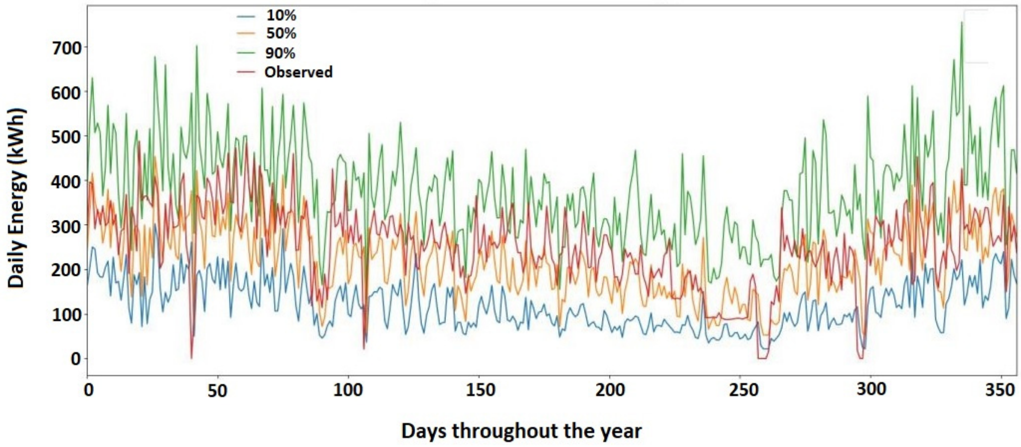


Figure 8. Forecasted energy consumption per day for an entire year.

If we further focus on the last 100 days of the year, we obtain the energy consumption prediction that is illustrated in Figure 9.

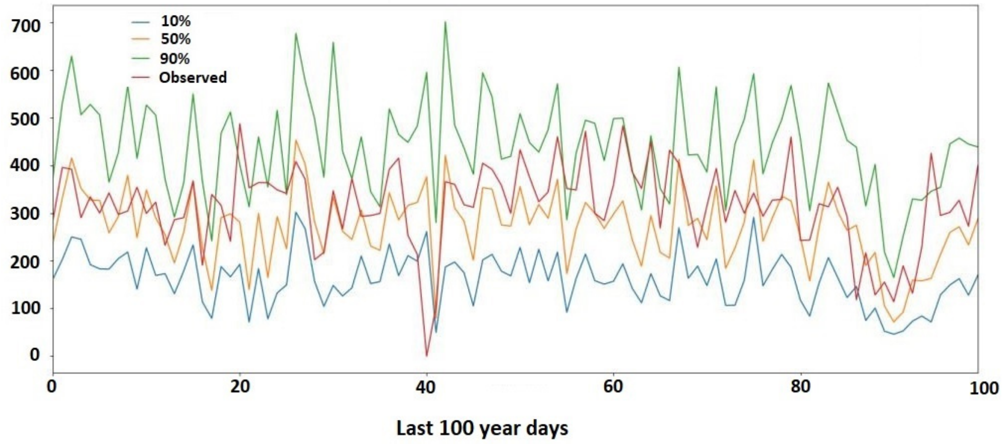


Figure 9. Forecasted energy consumption per day for the last 100 days of the year.

Table 3 illustrates the key performance indicator (KPI) metrics for the proposed framework. The metrics include: (1) average training time for the edge-based prediction model,

(2) average processing power, (3) storage memory for training and prediction at the edge-level, (4) prediction accuracy in comparison with future values for energy demand and (5) average time duration for the availability of the STLF to the utility operators at the distribution level. The results are also compared with the respective values when the framework is deployed directly in the utility data control center of Figure 1.

Table 3. Performance metrics of edge-based prediction framework.

KPI	Edge-Based Deployment	Control Center Deployment
Av. training time	15 min 29 s	15 min 58 s
Av. processing power	1.66 GHz	2 GHz
Av. storage memory	2 GB	2.5 GB
Prediction accuracy	94.1%	81.3%
Av. STLF time	1.3 s	5.7 s

In the above table, it is depicted that both the edge-based and the control center deployments have similar average training times. More specifically, the edge-based deployment performs slightly better since it is closer to the smart meter and sensors, which are devices providing the measurements, whereas due to communication latency issues, the control center deployment requires more time to gather all the necessary training measurements. On the contrary, the control center has an extensive amount of resources, including processing power and storage, allowing it to score significantly higher in the processing power KPI than the edge-based deployment. However, since the data concentrator is only used for STLF applications, there is no impact on the resources of the edge-based device. Additionally, the prediction accuracy for the STLF is significantly higher than the control center deployment since, at the edge level, there is real-time availability for smart meter and environmental measurements that can be leveraged for better training using historical data, which subsequently leads to an increased prediction accuracy. Finally, the edge-based deployment provides the forecast in near real-time (1.3 s) to the utility operators at the distribution level, whereas the time that is needed to provide such a forecast in a control center deployment is significantly higher (5.7 s). This is mainly due to the communication latency for transmitting the forecasts to the distribution substations.

5. Discussion

This section provides the main benefits and limitations of the proposed method when compared to similar work on energy forecasting methods. Initially, STLF prediction is usually performed using Cloud computing platforms [43], which leverage large computing and data storage capabilities in order to train the models using, for example, the Extreme Gradient Boosting (XGBoost) library [38]. Even though these models have satisfying accuracy, communication latency is added for their availability in the utility substations in order to be used by distribution service operators (DSOs) for STLFs. Moreover, certain DSOs employ strong security mechanisms for protecting access to the substation infrastructure or even use virtual private networks (VPNs), which provide an added time duration for encrypting and decrypting the data at the substation level. Furthermore, the presence of edge devices allows scalability for STLF applications and the avoidance of a single point of failure. Specifically, the presence of a utility data control center deployment constitutes an important risk for the electrical network, as a potential failure may lead to a loss of data, processing and management capabilities. The utility data control center deployment may fail due to a potential overload or even as a result of a cyber-attack. With a decentralized MEC architecture, an issue or fault in the utility data control center will have no impact on the STLF application, which is executed at the substation level.

In comparison with the existing work for STLFs in Section 2, and to the best of our knowledge, the proposed method in this article constitutes the first effort towards STLFs using the TFT. Furthermore, existing work in STLFs focuses on applying statistical approaches such as ARIMA [31,32] or regression [33–35] models for the analysis of smart meter measurements, including voltage, current and active/reactive power, which allow the capture of linear and non-linear patterns from temporal data. However, to produce accurate and tailored predictions for each household area, including behavioral habits, environmental variables and categorical values shall be considered in the prediction model. Moreover, environmental variables exhibit seasonal variations and usually do not remain constant over time. Nevertheless, this is not possible with statistical approaches such as ARIMA models since they require that the input parameters remain constant over time and have no variation. To cope with this challenge, this article is based on the use of the TFT as an ML-based prediction model. Apart from its low resource consumption, the TFT aids in identifying all the temporal relationships in the historical data that are fed into the STLF model. Specifically, the TFT incorporates: (1) static covariate encoders as context vectors, (2) gating mechanisms and sample-dependent variable selection for filtering the contribution of unnecessary variables, (3) a sequence-to-sequence layer to allow edge-based processing of the input energy consumption and environmental data and (4) a temporal self-attention decoder to learn long-term dependencies that may be present within the historical values of the gathered data. All these features facilitate the interpretation of data, including the identification of only the necessary values for the prediction as well as the presence of any temporal patterns in the prediction model by incorporating LSTM encoder/decoder layers in the architecture (Figure 3).

Finally, TFT models have been introduced only recently for multi-horizon time series forecasting in the literature [10]. Nevertheless, they have been used for MTLF predictions over the energy load [39], where they have demonstrated substantial improvements in forecasting accuracy as well as the incorporation of uncertainty estimation in time series forecasting. However, the data that are used for training do not include environmental factors and categorical values and are also performed with sufficient resources in terms of memory, processing power and GPU availability.

6. Conclusions

This article presents a novel energy demand prediction method that is based on an edge computing framework and the TFT prediction model for building a profile tailored to the behavioral characteristics of each consumer. The framework is based on the Foundry platform for gathering and analyzing data that are required for producing accurate forecasts, such as smart meter measurements as well as temperature, humidity and radiation data from embedded devices (i.e., sensors and actuators). The proposed method is applied on a HEMS testbed, which includes a PV system, smart meters and a battery storage system as an additional electricity supply unit in case of peak demands.

As a part of our future work, we plan to apply the forecasting method to a large-scale testbed covering the area of an entire neighborhood. This will demonstrate the impact of the method and allow us to demonstrate the seasonal dependence of the predictions. Moreover, it will enable the prediction of electricity load demand peaks at an early stage, i.e., before they cause blackouts. Hence, the avoidance of blackouts will provide high utility service availability, which in turn will increase customer satisfaction and avoid manual actions by operator personnel to restore the electricity network in such conditions. Finally, we plan to apply an optimization model for the optimal scheduling of shiftable loads (electric vehicles, a building energy management system—BEMS, etc.), based on day-ahead forecasting of the power produced by the photovoltaic system and the energy demand of the households or buildings.

Author Contributions: Conceptualization, A.L. and E.I.P.; methodology, A.L.; software, A.L.; validation, A.L. and E.I.P.; resources, A.L. and E.I.P.; data curation, A.L.; writing—original draft preparation, A.L. and E.I.P.; writing—review and editing, A.L. and E.I.P.; visualization, A.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets generated and/or analyzed during the current work are available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gilchrist, A. *Industry 4.0: The Industrial Internet of Things*; Springer: Berlin/Heidelberg, Germany, 2016.
2. Nof, S.Y. (Ed.) *Handbook of Automation*; Springer: Berlin/Heidelberg, Germany, 2009.
3. Xi, F.; Misra, S.; Xue, G.; Yang, D. Smart grid—The new and improved power grid: A survey. *IEEE Commun. Surv. Tutor.* **2011**, *14*, 944–980.
4. Li, X.; Hui, D.; Lai, X. Battery energy storage station (BESS)-based smoothing control of photovoltaic (PV) and wind power generation fluctuations. *IEEE Trans. Sustain. Energy* **2013**, *4*, 464–473. [CrossRef]
5. Lee, Y.-S.; Tong, L.I. Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. *Knowl. Based Syst.* **2011**, *24*, 66–72. [CrossRef]
6. Sapankevych, N.I.; Sankar, R. Time series prediction using support vector machines: A survey. *IEEE Comput. Intell. Mag.* **2009**, *4*, 24–38. [CrossRef]
7. Qiu, X.; Zhang, L.; Ren, Y.; Suganthan, P.N.; Amaratunga, G. Ensemble Deep Learning for Regression and Time Series Forecasting. In Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL), Orlando, FL, USA, 9–12 December 2014; pp. 1–6.
8. ETSI Industry Specification Group (ISG): GR MEC 017 Group Report v1.1.1, Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV Environment, February 2018. Available online: https://www.etsi.org/deliver/etsi_gr/mec/001_099/017/01.01.01_60/gr_mec017v010101p.pdf (accessed on 17 February 2018).
9. Villali, V.; Bijivemula, S.; Narayanan, S.L.; Prathusha, T.M.V.; Sri, M.S.K.; Khan, A. Open-source Solutions for Edge Computing. In Proceedings of the 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 7–9 October 2021; pp. 1185–1193.
10. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]
11. Kyriakides, E.; Polycarpou, M. Short term electric load forecasting: A tutorial. In *Trends in Neural Computation*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 391–418.
12. Ismail, B.; Lekidis, A.; dos Santos, D.R. Security and Privacy in Smart Grids: Challenges, Current Solutions and Future Opportunities. *ICISSP* **2020**, *10*, 733–741.
13. Derviškadić, A.; Romano, P.; Pignati, M.; Paolone, M. Architecture and experimental validation of a low-latency phasor data concentrator. *IEEE Trans. Smart Grid* **2016**, *9*, 2885–2893. [CrossRef]
14. Islam, M.A.; Che, H.S.; Hasanuzzaman, M.; Rahim, N.A. Energy Demand Forecasting. In *Energy for Sustainable Development*; Academic Press: Cambridge, MA, USA, 2020; pp. 105–123.
15. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [CrossRef]
16. Vivas, E.; Allende-Cid, H.; Salas, R. A systematic review of statistical and machine learning methods for electrical power forecasting with reported mape score. *Entropy* **2020**, *22*, 1412. [CrossRef]
17. Rangapuram, S.S.; Seeger, M.W.; Gasthaus, J.; Stella, L.; Wang, Y.; Januschowski, T. Deep state space models for time series forecasting. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 7785–7794.
18. Alaa, A.; van der Schaar, M. Attentive State-Space Modeling of Disease Progression. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 11338–11348.
19. Makridakis, S.; Spiliotis, E.; Assimakopoulos, V. The m4 competition: 100,000 time series and 61 forecasting methods. *Int. J. Forecast.* **2020**, *36*, 54–74. [CrossRef]
20. Fan, C.; Zhang, Y.; Pan, Y.; Li, X.; Zhang, C.; Yuan, R.; Wu, D.; Wang, W.; Pei, J.; Huang, H. Multi-horizon time series forecasting with temporal attention learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2527–2535.
21. Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.X.; Yan, X. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS'19), Vancouver, BC, Canada, 8–14 December 2019; pp. 5243–5253.
22. Wen, R.; Torkkola, K.; Narayanaswamy, B.; Madeka, D. A multi-horizon quantile recurrent forecaster. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Time Series Workshop, Long Beach, CA, USA, 4–9 December 2017; pp. 1–9.
23. Koutnik, J.; Greff, K.; Gomez, F.; Schmidhuber, J. A clockwork RNN. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 2014; pp. 1863–1871.
24. Bosco, B.; Parisio, M.; Pelagatti, L.P. Deregulated wholesale electricity prices in Italy: An empirical analysis. *Int. Adv. Econ. Res.* **2007**, *13*, 415–432. [CrossRef]

25. Janczura, J.; Weron, R. An empirical comparison of alternate regime-switching models for electricity spot prices. *Energy Econ.* **2010**, *32*, 1059–1073. [CrossRef]
26. Trueck, S.; Weron, R.; Wol, R. Outlier treatment and robust approaches for modeling electricity spot prices. In Proceedings of the 56th Session of the International Statistical Institute (ISI), Paris, France, 22–29 September 2007; New Economics Papers (NEP-ENE-2007-09-09). Available online: https://mpra.ub.uni-muenchen.de/4711/1/MPRA_paper_4711.pdf (accessed on 17 February 2018).
27. Sigauke, C.; Chikobvu, D. Prediction of daily peak electricity demand in south africa using volatility forecasting models. *Energy Econ.* **2011**, *33*, 882–888. [CrossRef]
28. Lisi, F.; Nan, F. Component estimation for electricity prices: Procedures and comparisons. *Energy Econ.* **2014**, *44*, 143–159. [CrossRef]
29. Taylor, J.; De Menezes, L.; McSharry, P. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *Int. J. Forecast.* **2006**, *22*, 1–16. [CrossRef]
30. Taylor, J.; McSharry, P. Short-term load forecasting methods: An evaluation based on european data. *IEEE Trans. Power Syst.* **2007**, *22*, 2213–2219. [CrossRef]
31. Clements, A.; Hurn, A.; Li, Z. Forecasting day-ahead electricity load using a multiple equation time series approach. *Eur. J. Oper. Res.* **2016**, *251*, 522–530. [CrossRef]
32. Ismail, M.; Zahran, A.; El-Metaal, E. Forecasting hourly electricity demand in egypt using double seasonal autoregressive integrated moving average model. In Proceedings of the ALLDATA 2015: The First International Conference on Big Data, Small Data, Linked Data and Open Data, Barcelona, Spain, 19–24 April 2015; pp. 42–45.
33. Hinman, J.; Hickey, E. Modeling and forecasting short-term electricity load using regression analysis. *J. Inst. Regul. Policy Stud.* **2009**, 1–51.
34. Feng, Y.; Ryan, S. Day-ahead hourly electricity load modeling by functional regression. *Appl. Energy* **2016**, *170*, 455–465. [CrossRef]
35. Weron, R.M.A. Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models. *Int. J. Forecast.* **2008**, *24*, 744–763. [CrossRef]
36. Yukseltan, E.; Yucekaya, A.; Bilge, A. Forecasting electricity demand for turkey: Modeling periodic variations and demand segregation. *Appl. Energy* **2017**, *193*, 287–296. [CrossRef]
37. Yang, Y.; Wu, J.; Chen, Y.; Li, C. A new strategy for short-term load forecasting. *Abstr. Appl. Anal.* **2013**, *2013*, 208964. [CrossRef]
38. Albahli, S.; Shiraz, M.; Ayub, N. Electricity price forecasting for cloud computing using an enhanced machine learning model. *IEEE Access* **2020**, *8*, 200971–200981. [CrossRef]
39. Li, D.; Tan, Y.; Zhang, Y.; Miao, S.; He, S. Probabilistic forecasting method for mid-term hourly load time series based on an improved temporal fusion transformer model. *Int. J. Electr. Power Energy Syst.* **2023**, *146*, 108743. [CrossRef]
40. Light, R.A. Mosquitto: Server and client implementation of the MQTT protocol. *J. Open Source Softw.* **2017**, *2*, 265. [CrossRef]
41. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef]
42. Lieskovan, T.; Hajny, J.; Cika, P. Smart Grid Security: Survey and challenges. In Proceedings of the 2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Dublin, Ireland, 28–30 October 2019; pp. 1–5.
43. Dillon, T.; Wu, C.; Chang, E. Cloud Computing: Issues and Challenges. In Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, WA, Australia, 20–23 April 2010; pp. 27–33.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.