smart cities

MDPI

Article

Experimental Verification and Implementation Feasibility Analysis of Remote Smart Meter Error Monitoring System in Smart Cities

Julius Šaltanis [1], Marius Saunoris [1], Robertas Lukočius [1], Vytautas Daunoras [1], Kasparas Zulonas [1], Stefano Rinaldi [2] and Žilvinas Nakutis [1,*]

[1] Electrical and Electronics Engineering Faculty, Kaunas University of Technology, 51368 Kaunas, Lithuania; julius.saltanis@ktu.lt (J.Š.); marius.saunoris@ktu.lt (M.S.); robertas.lukocius@ktu.lt (R.L.); vytautas.daunoras@ktu.lt (V.D.); kasparas.zulonas@ktu.edu (K.Z.)
[2] Department of Information Engineering, University of Brescia, 25123 Brescia, Italy; stefano.rinaldi@unibs.it
* Correspondence: zilvinas.nakutis@ktu.lt

Highlights

What are the main findings?

- Experimental validation proved that the proposed methods achieve a practically reasonable performance of smart meter error estimation.
- Event-driven sampling from smart meters offloads the communication channel.
- Only moderate 1 s synchronization accuracy between smart meters is sufficient for the methods.

What are the implications of the main findings?

- Proposed methods are suitable for Class 1 smart meter accuracy non-compliance detection.
- A WiFi/MQTT communication channel is suitable for collecting readings from smart meters for their remote error surveillance needs.

Abstract

Smart energy meters are widely deployed in modern distribution networks, extending their role beyond revenue billing to real-time monitoring and data-driven smart city applications. However, conventional legal metrology frameworks rely on periodic recalibration and are not intended for the detection of accuracy drift or unexpected malfunctions between scheduled inspections. In scientific publications, various techniques for remote smart meters' error surveillance are presented, but experimental verification on real distribution network data remains limited. The objective of this study is to experimentally verify two previously proposed power event-driven methods for remote estimation of active power measurement error in individual consumer meters, using a feeder-level sum meter as a reference instrument. One-second resolution electrical readings were collected from a real low-voltage distribution branch using ESP32-based local adapters communicating via MQTT over Wi-Fi, with SNTP-based clock synchronization for power event correlation. Under optimized detection parameters, the linear regression method achieved 0.20% RMSE and 0.75% maximum absolute error, and the neural network method 0.09% RMSE and 0.31%, confirming suitability for Class 1 m accuracy surveillance. Feasibility analysis of three MQTT-based deployment scenarios demonstrates that binary encoding limits local adapter buffers to 2.8 kB and worst-case daily channel demand to 2000 kB, confirming the practical viability of the proposed architecture.

Academic Editors: George Cristian Lazaroiu, Mariacristina Roscia, Daniel G. Costa and Pierluigi Siano

Received: 7 May 2026
Revised: 5 June 2026
Accepted: 18 June 2026
Published: 20 June 2026

Keywords: smart metering; measurement error; remote estimation; IoT infrastructure; event-driven acquisition; MQTT; wireless communication channel

1. Introduction

The concept of smart cities and smart grids intersect tightly, the latter focusing on the electrical energy subsystem, which is a key enabler for sustainable and modern society [1–3]. At the interface of the electrical energy field and the digital world, a gateway device, namely a smart meter, is installed and performs tasks of metering of electrical quantities, energy consumption, and is linked to the advanced metering infrastructure (AMI). Smart metering is widely recognized as a critical technological enabler of smart cities, providing trustworthy data streams required for sustainable and intelligent urban environments. Its capabilities extend far beyond traditional billing in terms of supporting real-time situational awareness, predictive planning, and automated control across interconnected infrastructures. Scientific literature consistently highlights smart metering as a foundational component of smart grids. For example, in ref. [4], smart meters are presented as a transformative technology enabling two-way communication, improved energy efficiency, and data-driven grid operation. Similarly, in ref. [5], it is emphasized that reliable measurement and communication technologies are essential for resilient, low-carbon smart city energy systems, enabling outage reduction, renewable integration, fair billing, and enhanced consumption control.

Smart meters offer a wide range of benefits across different stakeholders that contribute to improved energy management, efficiency, and environmental sustainability [4]. Society (consumers in smart city), electrical utilities (business, manufacturers), energy users and service providers (business) are all concerned about the quality of readings from smart meters as it influences payments, revenues, and guides correct decisions [6]. The correct and trusted information can guide long-term planning, infrastructure investments, and policy decisions to create more sustainable and resilient urban environments [7]. Emerging AI/ML-driven applications particularly rely on high-quality data acquired from smart meters. Therefore, accurate measurement of electricity consumption and generation is a fundamental functional requirement of smart meters, followed by more functional (real-time measurement, remote monitoring, two-way communication, appliance control and monitoring, and integration with other systems) and non-functional (security, privacy, dependability, and compatibility) expectations. The widely established metrology regulatory system is maintained to ensure measurement traceability. However, a need for further investigation into the long-term performance and reliability of smart meters indicates that the accuracy over extended periods is not well-understood [4]. Conventional legal metrology frameworks ensure periodic verification of utility-owned meters, but they do not address unexpected malfunctioning or accuracy drift occurring between scheduled inspections. Modern smart meters, however, measure a wide range of electrical parameters—voltage, current, power factor, and harmonics—that can be exploited for continuous, automated health monitoring. Early detection of malfunctioning devices strengthens public trust and reduces financial losses and operational risks.

While utility-owned meters are integrated into AMI systems capable of remote diagnostics, a growing number of submeters, power analyzers, and IoT-based monitoring devices are deployed by building managers, industrial operators, cooperatives, and community organizations. These devices often fall outside legal metrology regulation, yet their measurements are used to control energy flows in buildings, street lighting [8], parking cooperatives, and other public systems. This heterogeneous ecosystem underscores the need

for utility-independent, IoT-based accuracy observation frameworks capable of monitoring both regulated and non-regulated devices.

The widespread deployment of AMI has significantly expanded the availability of high-resolution electricity consumption data in distribution networks. This creates new opportunities for addressing the estimation of non-technical losses (NTL) [9]. A comparative analysis distinguishes: (i) supervised classification methods, enabling high-accuracy detection, with many studies reporting F1-scores exceeding 0.95 on benchmark datasets, yet they require extensive labeled data from AMI systems; (ii) unsupervised clustering techniques, offering a practical alternative for grids with partial or no AMI. Remote detection and estimation of smart meter error is often explored assuming the absence of other causes of anomalous readings from smart meters, like electricity thefts, cybersecurity issues, etc. The shift from physical on-site verification toward purely data-driven remote methods is evident in the last decade [10]. Remote estimation of smart meter errors based on measurement data analysis has come into focus precisely because field verification is expensive and does not scale to millions of meters. Two broad technical lineages are competing: statistical/physics-based estimators and deep learning approaches. Recent works move from detecting binary "faulty/not faulty" to continuous drift estimation [11]. Supervised methods can reach high benchmark scores, but they still require extensive labeled data, while unsupervised clustering is a practical fallback for grids with partial meter roll-out. On the prognostics side, daily measurement-error forecasting is being treated as a health-management problem [9,12]. Papers [10,13,14] push the field toward real-time monitoring and health assessment, including deep or hybrid models for direct fault classification and error estimation. Publication [15] targets real-time error detection and tackles class imbalance in a 20,000-sample dataset. Daily meter measurement error forecasting is emphasized, and techniques are considered in [12,14]. Meter error estimation and anomaly detection, fault classification or detection, and prognostic health monitoring are now three partially overlapping subfields. Smart meter anomaly detection is still constrained by data imbalance, privacy, scalability, interpretability, and the need for models that work without a full network model or large labeled AMI datasets. That is the main research gap related to cross-utility, or cross-region validation, so generalization under changing meter types, climates, load patterns, and deployment conditions, which remains unresolved [14,16].

A comprehensive survey of remote smart meter error estimation methods is given in [17]. The published methods for remote smart meter error (or, in a more general sense, meter fault or anomaly readings) estimation or detection can be classified (i) based on energy preservation law [18–23] and (ii) time series processing [24–27]. Energy consumption readings aimed for revenue metering are typically acquired with a period between 15 min and 1 h. These readings, together with readings of the sum meter, can be used to estimate every smart meter in the grid supervised by the sum meter. In the case of energy preservation-based techniques, a sum (check, totaling, and master) smart meter is required at the electrical grid feeder supplying energy to the group of consumers. Moreover, all consumers must be metered with smart meters in order to deliver readings to the processing location with a similar time resolution. The event-driven method we have introduced and developed [17,28] has unique properties enabling to avoid some limitations inherent to the other techniques: (i) it is not sensitive to partial smart metering roll-out or unmonitored energy consumption like in a case of electricity thefts; (ii) it does not require labeled (faulty/non-faulty) data to build a reference prediction model and thus does not suffer from data imbalance; and (iii) it does not require network topology, parameters and model knowledge due to the ability to adapt the prediction model during training phase. The event-driven method was validated using a synthesized dataset and indicated promising results, like sub-1% RMSE of meter error estimation. Field-validated evidence

of the performance of remote meter fault detection remains limited. For instance, recent reviews highlight the conceptual potential of smart metering infrastructures for smart city development but note the lack of empirical verification of diagnostic techniques [2].

Acquisition of data demanded by the processing algorithms aiming to detect or estimate smart meter errors is of the utmost importance. Using typically over AMR/AMI acquired 15–60 min resolution energy consumption readings dedicated for revenue billing limits the possibilities to precisely predict meter error drifting. Because introducing new functionality and data collection techniques into the AMI is quite restricted, smart meter anomaly detection (meter errors, electricity thefts, cybersecurity attacks, etc.) systems and their implementation within IoT (Edge/Cloud) frameworks were therefore considered in numerous publications [29–32]. The application of a larger set of electrical parameter readings, obtained with higher time resolution via local interfaces from smart meters and then delivered over less regulated IoT frameworks, makes the design and validation of new techniques much easier.

Examining the performance of meter error detection methods using data collected from real grid extends the validation scope by means of training prediction models (regression or neural network) with readings influenced by (i) real distribution lines, (ii) voltage fluctuations (noise), (iii) synchronization error between CM and SM, and (iv) time overlapping power events in the network monitored by the SM. Some of these factors were mitigated by introducing preprocessing and filtering of sampled data and selecting optimal parameters for preprocessing. In particular, voltage fluctuations causing deviation of voltage, as well as current and active power values before and after the power event, were suppressed by choosing the averaging period (or number of samples). Synchronization error influence was mitigated by discarding a sufficient number of samples in the interval around the power event edge. Overlapping power events were filtered by setting the threshold of expected power losses (10%), i.e., assuming that the difference between power consumption change cannot exceed 10% rate. Moreover, the synthesized data previously used to validate the method was generated by solving the state problem of the specifically designed equivalent network [17]. In this research, the real network with intermediate loads between CM and SM was present during the experimental data collection. Therefore, the method validation conditions were essentially different than used in the earlier publications. Also, the results for the analysis of requirements for an IoT MQTT-based infrastructure in terms of channel capacity, as well as requirements for additional meter adapter devices in terms of buffering memory volume, are provided.

2. Methods

2.1. Meter Error Estimation

Two methods of remote meter active power measurement offset error were proposed in [17] and [33], respectively. Their performance was estimated using synthesized data. Both techniques rely on the minimization of a goal function constructed from the Euclidean distance between measured and predicted values of the power event measured at the location of the sum meter:

$$\hat{\boldsymbol{Y}}_{\boldsymbol{PV}} = \min_{\boldsymbol{Y}_{\boldsymbol{PV}}} \sum_{i=1}^{N} \left(d\hat{P}_S(F^{-1}_{e(V,I,P)}(\boldsymbol{X}_{mi}, g_{V,I,P})) - dP_{Smi} \right)^2, \quad (1)$$

where $\hat{\boldsymbol{Y}}_{\boldsymbol{PV}} = \left| \hat{g}_P, \hat{g}_V \right|$ is the estimate of active power and voltage gain errors, $\boldsymbol{X}_{mi}$ is the i-th vector of synchronously measured quantities by the sum and consumer meters, dP_{Smi} is the i-th power event measured by the sum meter, $F^{-1}_{e(V,I,P)}$ is the inverse function of voltage, current, or active power measurement error, $g_{V,I,P} \equiv (g_V, g_I, g_P)$ are the error gain of

voltage, current, and active power, respectively. We assume the offset measurement errors of the current and voltage channels can be described by a multiplicative model:

$$I_m = F_{eI}(I_0, g_I) = I_0 \cdot \left(1 + \frac{g_I}{100\%}\right), \tag{2}$$

$$V_m = F_{eV}(\mathrm{V}_0, g_\mathrm{V}) = V_0 \cdot \left(1 + \frac{g_V}{100\%}\right); \tag{3}$$

where I_0 and V_0 are actual (real) current and voltage RMS values, and g_I and g_V are the current and voltage error gain coefficients expressed as a percentage.

The measured active power due to errors in current and voltage measurements can be expressed as follows:

$$P_m = I_m V_m cos\varphi = I_0 \left(1 + \frac{g_I}{100}\right) V_0 \left(1 + \frac{g_V}{100}\right) cos\varphi, \tag{4}$$

or

$$P_m = F_{eP}(P_0, g_\mathrm{P}) = P_0 \left(1 + \frac{g_P}{100}\right), \tag{5}$$

where

$$P_0 = I_0 V_0 cos\varphi, \tag{6}$$

$$g_P = g_I + g_V + g_I g_V / 100. \tag{7}$$

The i-th measured power consumption change (power event) by the sum meter is

$$dP_{Smi} = P_{Sbmi} - P_{Sami}, \tag{8}$$

where P_{Sbmi} and P_{Sami} are the corresponding power consumptions before and after the power event.

The prediction of the i-th power consumption change at the location of the power meter $d\hat{P}_S(\boldsymbol{X}_{0\mathrm{i}})$ is obtained using the data-driven models described in Sections 2.2 and 2.3, as well as the expected (actual) values of predictor quantities $\boldsymbol{X}_{0\mathrm{i}}$, which are related to measured quantities $\boldsymbol{X}_{m\mathrm{i}}$ according to (2), (3) or (5) in the case of a consumer meter (CM), and are assumed equal to $\boldsymbol{X}_{m\mathrm{i}}$ if they are measured by the sum meter (SM) (reference instrument).

The temperature, as well as other environmental conditions like humidity, influences both current meter error and contributes to aging-caused error. The method discussed in the paper is dedicated to estimating the meter error despite the causing factors relying on a sum meter (reference instrument) and prediction models build it reference conditions, i.e., when the meter under test was expected to comply with metrological error requirements. Therefore, the influence of environmental conditions can impact the method when temperature drift is present in reference conditions and influence both CM and SM data used for prediction model training. When the environmental conditions change during the monitoring phase, the prediction model accuracy may degrade. Currently, environmental conditions are not taken into account in training or testing phases. In the future, temperature and humidity (measured at the meter location or common in the area) could be made input variables for the prediction model. This could mutually enable the prediction model to be sensitive to the distribution line losses' dependence on the temperature. Nevertheless, in the current publication, we believe the daily impact of temperature/humidity is considerably less than the accuracy of meter error estimation achievable by the considered method. Regarding the sum meter aging error, the method is unable to predict and somehow compensate for it. The sum meter is assumed to be a reference device whose measurements are trusted, and its precision is guaranteed by the metrology system in place (laboratory or on-site verification). In case SM error increases, for example, due to aging, the estimation of CM error accuracy will inevitably degrade in terms that SM error drift

will be recognized as CM error increase. Therefore, metrological maintenance of the sum meter is of high importance for the reliability of the presented method.

2.2. Linear Regression-Based Predictor

The power change at the SM dP_S in response to a consumer load change dP_c can be modeled [33]

$$dP_S = dP_w + dP_{nL} + dP_C \tag{9}$$

where dP_w is the change in active power losses in the energy delivery branch between the SM and CM, dP_{nL} is the active power consumption change in network loads (due to voltage change after a consumer power event). The prediction $d\hat{P}_S$ for the use in (1) can be obtained from (9) by substituting dP_C by its measured value dP_{Cm} and predictions of dP_w and dP_{nL}. It was derived in [33] that the predicted power losses change can be described:

$$d\hat{P}_w = F_{dPw}(\boldsymbol{X}_{Rw}) = (V_{Sa} - V_{Ca})^2 - (V_{Sb} - V_{Cb})^2 / R_{eq} \tag{10}$$

where V_{Sb} and V_{Cb} are the voltage readings measured by the SM and CM, respectively; before the power consumption change event, V_{Sa} and V_{Ca} are voltage readings measured by the SM and CM after the power consumption change event; and R_{eq} is the equivalent resistance of the branch wire, which has to be determined in the training phase when SM and CM readings are trusted.

For the prediction of dP_{nL}, the linear regression model is applied following the analysis performed in [33]:

$$d\hat{P}_n = R_{dPn}(\boldsymbol{X}_{dPn}) \tag{11}$$

where predictor vector $\boldsymbol{X}_{dPn} = (V_{Ca}, V_{Cb}, P_{Ca}, P_{Cb}, V_{Ca}{\cdot}V_{Cb}, V_{Ca}{\cdot}P_{Ca}, V_{Ca}{\cdot}P_{Cb}, V_{Cb}{\cdot}P_{Ca}, V_{Cb}{\cdot}P_{Cb})$ is composed of elements selected according to p-value criteria (p-value < 0.05).

Prediction of active power consumption change after the i-th power event at the consumer meter load can be expressed by substituting actual quantities with measured quantities with compensated measurement errors of the CM using (3) and (5):

$$\begin{aligned} d\hat{P}_S\left(F^{-1}_{e(V,I,P)}(\boldsymbol{X}_{Rmi}, g_{V,P})\right) &= F_{dPw}(\boldsymbol{X}_{dPwi}) + R_{dPn}(\boldsymbol{X}_{dPni}) + dP_{Ci} \\ &= F_{dPw}\left(F^{-1}_{eV}(\boldsymbol{X}_{dPwmi})\right) + R_{dPn}\left(F^{-1}_{e(V,P)}(\boldsymbol{X}_{dPnmi})\right) + F^{-1}_{eV}(dP_{Cmi}), \end{aligned} \tag{12}$$

where the vector of quantities measured by both SM and CM is $\boldsymbol{X}_{Rmi} = (\boldsymbol{X}_{dPwmi}, \boldsymbol{X}_{dPnmi})$. Predictor vectors $\boldsymbol{X}_{dPw}$ and $\boldsymbol{X}_{dPn}$ are expressed from measured values of their elements (voltage and power), which are corrected taking into account measurement voltage and active power gain errors in (3) and (5) (for CM measurement only, because SM is assumed a reference instrument):

$$\boldsymbol{X}_{dPw} = \left(V_{Sma},\ V_{Smb}, V_{Cma}/(1 + \frac{g_V}{100}),\ V_{Cmb}/(1 + \frac{g_V}{100})\right), \tag{13}$$

$$\begin{aligned} \boldsymbol{X}_{dPn} = \big(&V_{Cma}/(1 + \tfrac{g_V}{100}),\ V_{Cmb}/(1 + \tfrac{g_V}{100}),\ P_{Cma}/(1 + \tfrac{g_P}{100}),\ P_{Cmb} \\ &/(1 + \tfrac{g_P}{100})\big), \end{aligned} \tag{14}$$

$$dP_C = \frac{dP_{Cm}}{(1 + \frac{g_P}{100})}. \tag{15}$$

2.3. Artificial Neural Network-Based Predictor

The input vector (predictor vector) $\boldsymbol{X}_{\boldsymbol{NN}} = (V_{Sb}, I_{Sb}, V_{Cb}, dV_C, dI_C)$ was selected in [17] for the prediction of dP_S using a regression neural network:

$$d\hat{P}_S = N_{Ps}(\boldsymbol{X}_{\boldsymbol{NN}}) = N_{Ps}\left(F^{-1}_{e(V,I,P)}(\boldsymbol{X}_{NNmi}, g_{V,I})\right), \tag{16}$$

where V_{Sb} and I_{Sb} are the corresponding voltage and current readings measured by the sum meter before the power consumption change event; V_{Cb} is the voltage measured by the consumer meter; $dV_C = V_{Cbm} - V_{Cam}$ and $dI_C = I_{Cbm} - I_{Cam}$ are correspondingly voltage and current changes after power events measured by the consumer meter; and N_{Ps} denotes an artificial neural network consisting of three layers, each having 25, 31, and 31 neurons with ReLU activation function.

The predictor vector is composed of measured electrical quantities. In the case of CM measurements, they are compensated by taking into account the CM current and voltage gain errors according to (2) and (3):

$$\boldsymbol{X}_{NN} = \left(V_{Smb},\ I_{Smb},\ \frac{V_{Cb}}{\left(1 + \frac{g_V}{100}\right)},\ \frac{dV_C}{\left(1 + \frac{g_V}{100}\right)},\ \frac{dI_C}{\left(1 + \frac{g_I}{100}\right)} \right), \tag{17}$$

To reduce the influence of randomness of training NNs, the ensemble of $N_e = 50$ neural networks was built, and the prediction $d\hat{P}_S$ was assigned the median of outputs of all neural network instances.

2.4. Experimental Setup

2.4.1. Acquisition of Sum and Consumer Meter Readings

To experimentally validate the proposed CM active power measurement error estimation approach, a data acquisition system was implemented to collect readings from the consumer meter and the sum meter. The purpose of this system was to collect measurement data from both meters in a form suitable for later comparison and joint processing, even though the meters were installed in different locations and used different data formats. For each meter, an ESP32-based controller ESP32-C6-DevKitC-1 manufactured by Espressif Systems Co., Ltd. (Shanghai, China), was used to read the local communication interface, process the received data, and send it to the server for storage and analysis. Since the consumer meter used DSMR over P1 and the sum meter used DLMS over RS485, protocol-specific parsing was required before the data could be converted into a common format.

The acquisition system itself consisted of four main parts: the electricity meters, the ESP32-based controllers, the existing Wi-Fi network, and a computer acting as the central server (see Figure 1). Each controller received data from its meter, extracted the required values, and sent them to the server over Wi-Fi using MQTT. The server ran the MQTT broker, handled communication with the controllers, processed incoming messages, and stored the measurements in a time-series database.

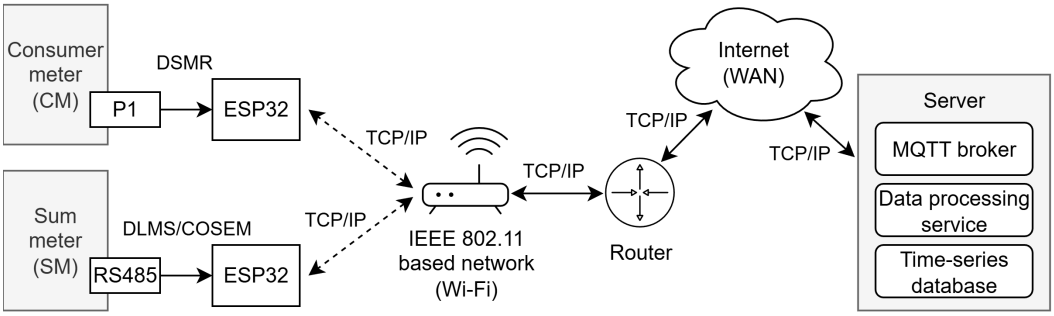

Figure 1. Architecture of the smart meter data acquisition system. Readings from the consumer meter and the sum meter are acquired by ESP32-based controllers and transmitted using MQTT over a Wi-Fi network to a central server for data processing and storage.Solid lines with arrows correspond to wired communication link, and dashed correspond to wireless link.

Both meters provided new readings once per second, but their internal clocks were not synchronized. In a non-invasive setup, direct synchronization of the meters is difficult

and, in many cases, not possible without changes by the manufacturer or the electricity network operator. Maintaining long-term time alignment is also a practical challenge. For this reason, time synchronization was implemented in the ESP32 controllers using the SNTP protocol (Simple Network Time Protocol). This provided sufficient accuracy for aligning meter data with 1 s resolution. When the first byte of a message was received from the meter, the controller assigned a local timestamp, which was later sent together with the processed data to the server. This allowed the two data streams to be aligned and corresponding power events to be identified.

The approach was non-invasive, since there were no modifications to the meters. Only the local communication interfaces already available on the meters, namely P1 and RS485, were used. This kept the setup simple and made it suitable for experimental validation of the proposed methods.

2.4.2. Test Site

The experiments were conducted in a university office building equipped with typical office infrastructure, including personal computers, large display screens, electrical kettles, etc. The SM and the CM were both connected to the same single-phase branch on the same floor of the building. The wiring length between the SM and the CM connection points was approximately 40 m. Electrical installation in the building was implemented with copper wires with a cross-sectional area of 2.5 mm^2.

The loads connected between the SM and the CM were manually operated, unsynchronized with switchable loads toggling. Consequently, there may have been rare moments when other loads and switchable loads' switch moments coincided. Such a situation reflects the real-life load changing in a short moment (overlapping power events), though probabilistically, this should not happen often. The power demand of the uncontrolled loads reached values of up to several kilowatts.

2.4.3. Accelerated Generation of Power Events

The number of power events caused by consumer daily activities is personal behavior- and weekday-dependent. A preliminary expectation is that a sufficient number of power events can be acquired within several months. Therefore, the collection of experimental data for method verification needs was accelerated by implementing specific switchable loads. Selected typical electrical loads (see Table 1) were toggled according to a preprogrammed sequence every 15 s using relays controlled by the controller unit based on the Arduino module (Figure 2). The same power event was repeated five times to acquire SM and CM readings representing cases with different voltage values due to natural voltage fluctuations in the LV grid. Combining the six available loads, four different power events (0.2, 0.6, 0.85 and 1.45) kW, and three constant power consumption levels (0, 1.85 and 2.05) kW before the power event were generated. Reactive powers were recorded during the experiment, but were relatively very low (for Qp of CM, the highest recorded value was 12 VAR when the heater with fan was turned on for constant power, and for Qn of SM, the highest recorded value was 3 VAR). The whole experiment's active power consumption pattern can be seen in Figure 3.

The SM was installed at the floor electrical panel, and the test bed was plugged into the wall socket wired to the electrical branch supplied through the SM.

Consequently, the primary objective of the present study is to rigorously validate and stress-test these methods in an operational setting—namely, a representative university office building. This setting intrinsically introduces realistic operational uncertainties, ambient noise, and grid instabilities. For statistical reliability of estimates, multiple subsets were randomly sampled from the full experimental dataset, with each subset containing an

identical number of observations. Comparative evaluations were then conducted using both methods across these subsets, enabling a systematic assessment of a broad range of load variations and profiles. From our perspective, this empirical strategy represents the most stringent framework currently available for method verification. Future research will extend this experimental paradigm to encompass strongly non-linear load profiles in order to further probe the performance limits of the proposed methods.

Table 1. Switchable loads and their parameters.

Load ID	Load Type	Nominal Active Power
P_{C1}	Block of 3 incandescent lamps	0.25 kW
P_{C2}	Heater with fan No. 1	2 kW
P_{C3}	Heater No. 2	1.5 kW
P_{C4}	Block of 4 incandescent lamps	0.25 kW
P_{C5}	Heater No. 3	0.4 kW
P_{C6}	Incandescent lamp	0.2 kW

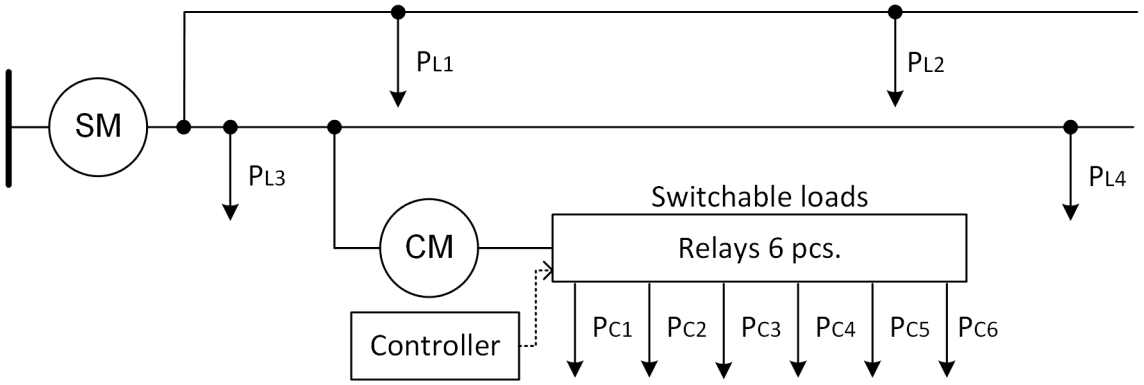

Figure 2. Diagram test network including power events generating switchable loads. A sum meter measures the power consumption of the distribution network, including connected loads P_{L1}–P_{L4}. Solid lines denote power distribution lines, dashed lines denote digital control lines.

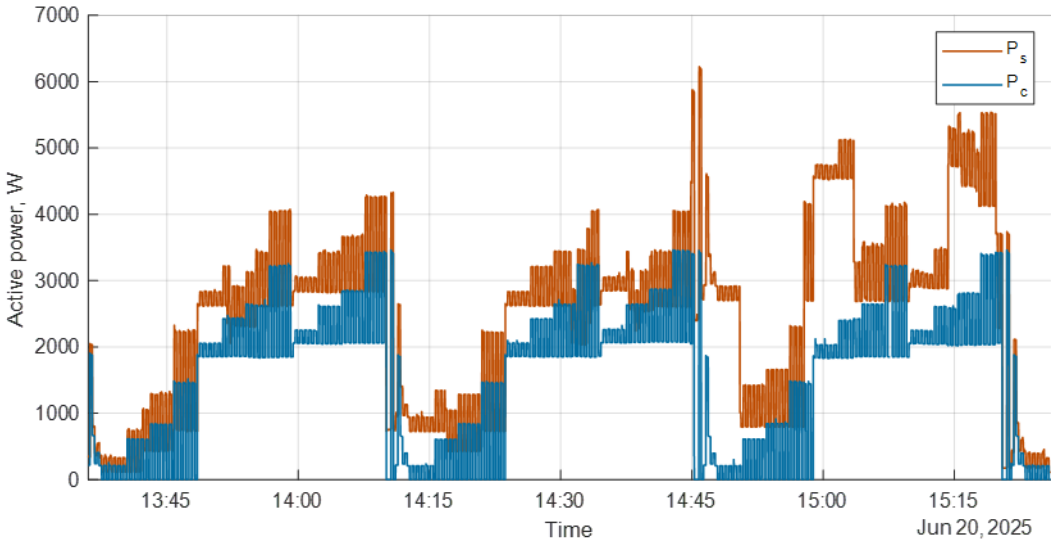

Figure 3. Readings of active power of the sum meter Ps and client meter Pc (full dataset).

3. Results

3.1. Collected Dataset

Both meters send data every 1 s, but they are not synchronized with each other. To synchronize the data being sent, the SNTP timestamp is added to each message. Along with the SNTP timestamp, all readings transmitted via the P1 interface are stored in the database for each meter separately (Table 2). Collected measured values for each meter, and their resolution is provided in Table 2.

Table 2. Measured values and their parameters.

Value	Resolution and Unit	Notation	Example
NTP timestamp	1 ms	t	2025-06-20 13:36:00.490
Timestamp of the meter clock	1 s		250620133600W
Total imported energy	1 Wh	E_i	141966
Total exported energy	1 Wh	E_e	3
Instantaneous imported active power	1 W	P	218
Instantaneous exported active power	1 W	P_e	0
Instantaneous imported reactive power	1 VAR	Q_p	0
Instantaneous exported reactive power	1 VAR	Q_n	0
Instantaneous voltage	0.1 V	V	229.7
Instantaneous current	1 A	I_m	0
Equipment identifier (meter)			3034393839353540
Reporter identifier (WiFi transmitter)			404CCAFFFE4E

Due to the low resolution of the instantaneous current reported by the meters, it was not used in the calculations. The current value used in the calculations was estimated from active, reactive power, and voltage measured values:

$$I = \frac{\sqrt{P^2 + (Q_p + Q_n)^2}}{V} \tag{18}$$

where P, V and Q_p, Q_n are measured values.

The difference in measured active power between SM Ps and CM Pc is due to the power losses and consumed by other loads, P_{Li}, connected to the same distribution branch (see Figure 3). This is very noticeable in the last part of the dataset, starting from time 14:45. However, the size of the power event in the P_s mostly replicates the power event in P_C that was generated by the testbed. The power events caused by the activities of other loads and overlapping with testbed loads switching have to be discarded from the prediction models' training and testing datasets. The criteria for discarding are unexpectedly high differences between dP_s and dP_c, which cannot be justified by power delivery losses and CM error.

3.2. Acquired Meter Readings Preprocessing and Power Event Detection

The detection of power events from readings of SM and CM stored in database is performed in four stages: (i) detecting power events separately in SM and CM active power records following the below specified criteria, (ii) matching SM and CM measured power events according to timestamps, (iii) locating samples of power measured by SM and CM, which are not related to the power event edge, (iv) calculating mean values of samples before and after power event for all quantities (voltage, current, and active power).

In the first stage (i), a power event is detected when the following criteria are met (in detail, power event detection is described in [28]):

(1) Active power consumption changes $dP_C > dP_{min}$ and $dP_S > dP_{min}$;

(2) Duration of power consumption change edge Δt_c (rise time or fall time) is not more than three sampling periods;
(3) Standard deviation of power consumption samples in the range of N_b samples before the power event edge $s_b(P_C) < s_{pMax}$;
(4) Standard deviation of power consumption samples in the range of N_a samples after the power event edge $s_a(P_C) < s_{pMax}$.

In the second stage (ii), the same power events in both SM and CM are located by comparing timestamps. Since moments of transmitting P1 packets (containing electrical quantities readings) from meters are not synchronized, the power event is detected when the time difference in the first power event edge samples between SM and CM is less than one second.

In the third stage (iii), power change edge durations of the same power event measured by SM and CM are compared. If the durations of these edges are not equal, then the shorter edge is extended to avoid further using samples of SM or CM related to the transition moment.

In the final fourth stage (iv), the mean power consumption before and after power event P_{Sbmi} and P_{Sami} are calculated from the number of samples N_b and N_a, respectively. Average voltage and current before and after the power event are calculated accordingly.

Six power events sampled by SM and CM can be seen (three increasing power consumption and three decreasing power consumption) in Figure 4. Three power consumption changes are not detected as power events (edges 2, 3, and 4) because they do not meet power event detection criteria: edge duration of SM longer than three samples (can be due to some network loads overlapping switching).

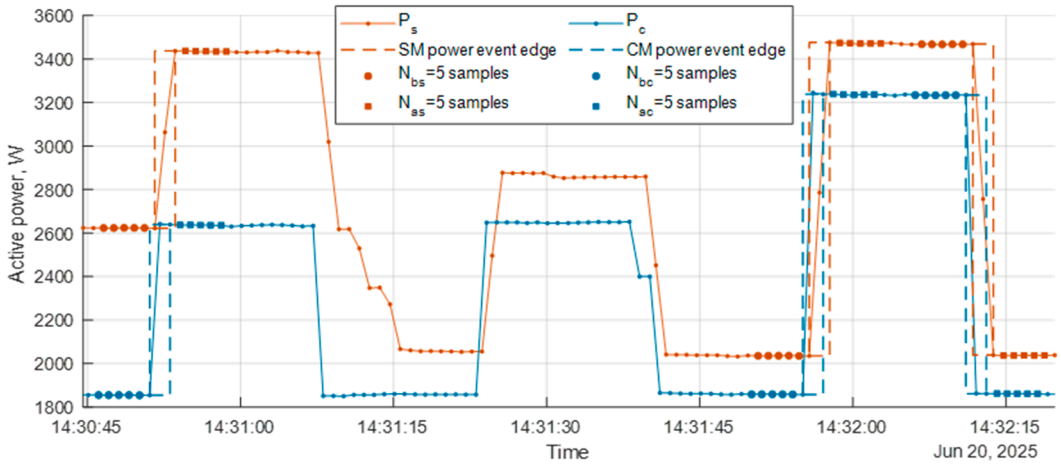

Figure 4. Sum meter Ps and client meter Pc active power measurement readings with accepted and rejected power events.

The criteria (3) and (4) of stage (i) of power event detection applied to the readings of SM prevent the detection of overlapping events of more than one consumer. Power events failing to comply with the condition $(|dP_S - dP_C|/dP_S)\cdot 100\% > L_{nMax}$ are rejected too. L_{nMax} = 10% was chosen, instructing the rejection of power events whose power change at SM and CM cannot be justified by network losses and most probably appear as a result of overlapping consumer power events. The detected power event in the SM and the monitored CM is considered suitable when there are no power changes due to other consumers, i.e., the power consumption P_{Ln} (see Figure 2) of all other consumers, which are constant in the interval surrounding the event edge. When more CMs are connected to a single SM on the same branch, the event detection method would be applied separately to each monitored CM and SM pair. If the power events from several CMs coincide or if

they overlap and take place during the time interval $[t - \Delta t_s \cdot N_b, t + \Delta t_s \cdot N_a]$, they will be rejected by applying detection rules to the SM active power consumption samples. An obvious influence of overlapping power events at more than one CM is seen in Figure 4 (the second falling edge), with power fluctuations in SM power after the power event edge. Therefore, this power event will be rejected and not included in the training or testing datasets used for CM error estimation. The higher the number of CMs observed by the same SM, the larger the probability that power events of different consumers will overlap. As the number of rejected power events increases, the time for collecting the demanded number of power events data will increase, too.

3.3. Performance Assessment Methodology

To assess the performance of CM error estimation methods, the acquired experimental dataset was split into training and testing datasets. The total number of samples (electrical quantities corresponding to each power event) in every dataset was dependent on power event detection parameters (s_{pMax}, t_m, dP_{min}) and maximum reasonable power loss parameter L_{nMax}, which was used to reject power events overlapping with other loads in the grid activities. Some typical values of full dataset samples after power event detection are shown in Table 3. In this chapter, we explore the influence of all these parameters, aiming to select their optimal values maximizing the performance of meter error estimation.

Table 3. Size of dataset of power events (L_{nMax} = 10%).

dP_{min}, W	s_{pMax} = 10 W		s_{pMax} = 20 W		s_{pMax} = 30 W	
	t_m = 4 s	t_m = 10 s	t_m = 4 s	t_m = 10 s	t_m = 4 s	t_m = 10 s
50	254	210	334	310	353	325
250	181	146	243	223	258	233

The methodology shown in the flowchart (Figure 5) was applied to evaluate statistical parameters of smart meter power and voltage gain error estimation (separately based on linear regression and neural network models):

$$e\left(g_{(V,P)}\right) = \hat{g}_{(V,P)} - g_{0(V,P)}. \tag{19}$$

Power events exhibiting the difference between dP_S and dPc exceeding L_{nMax} = 10% were excluded because this level of active power losses in the distribution network is little probable and the difference might be caused by the overlapping power events taking place in the loads monitored by SM but not by CM.

Two performance characterizing criteria, namely RMSE and MaxAE (maximum absolute error) of $e\left(g_{(V,P)}\right)$ were considered in the research. The RMSE was obtained from r = 300 instances of $e\left(g_{(V,P)}\right)$ (as shown in Figure 5) and represents the integral performance of the methods investigated. The confidence interval of the RMSE estimate is specified using the chi-squared distribution $[\sqrt{r \cdot RMSE^2/\chi^2_{1-\alpha/2,r}}, \sqrt{r \cdot RMSE^2/\chi^2_{\alpha/2,r}}]$ for the probability between 5% and 95% (r = 300).

Alongside RMSE, the largest estimation error from r = 300 attempts to estimate meter power/voltage error (MaxAE) as a metric of the worst case is reported. Fluctuations of $e\left(g_{(V,P)}\right)$ are possibly dependent on the training and test datasets, and, therefore, worst-case estimation is important in order not to overestimate the performance of methods reported.

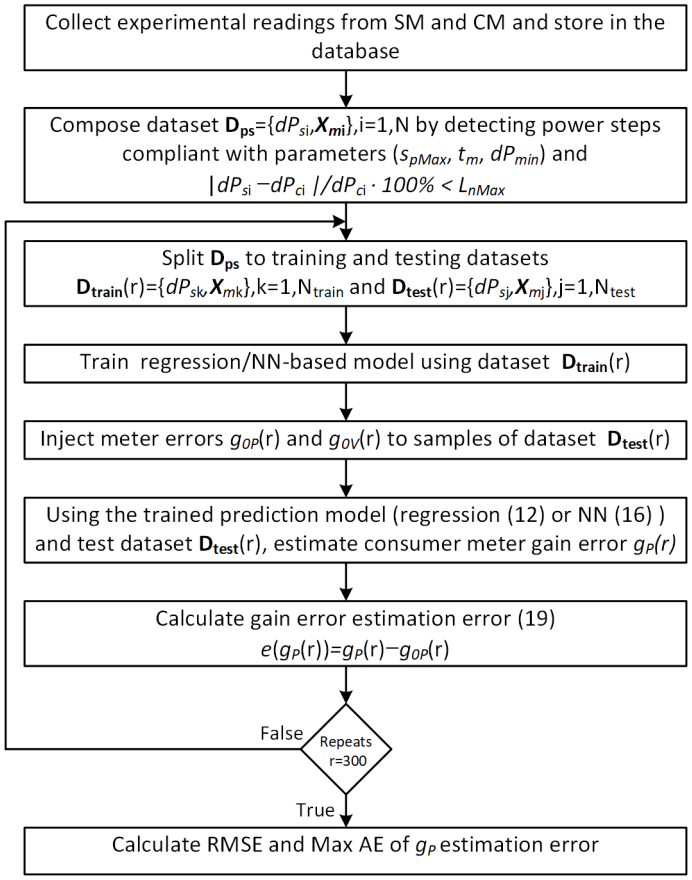

Figure 5. Flowchart of method verification methodology.

3.4. Performance Assessment Results

In Figure 6, RMSE and MaxAE dependence on the averaging interval t_m for both methods are shown. It can be seen that the NN-based method clearly outperforms the linear regression-based method and achieves the lowest RMSE when s_{pMax} = 30 W and t_m = 4 s. Considering that the aim of smart meter error estimation is to detect if accuracy class compliance is not violated (Class 1 (1% error)), meters are most often used for residential consumers' revenue metering), the target precision of the smart meter power measurement error estimation should not exceed 1%. In such a case, the RMSE of both techniques features a sufficient accuracy. However, considering the MaxAE criteria, only the NN-based technique ensures the sub-1% power gain error estimation.

The combination of parameters t_m and s_{pMax} controls how many power events are included in the training and test datasets (see also Table 3). Longer averaging interval t_m determines that electrical grid voltage fluctuations may influence average voltages of SM and CM before and after the power consumption event moment. Reducing s_{pMax} can reject power events due to noisier power consumption before and after the power event. Therefore, if noisier power events are not rejected, then a larger number of power events are included in the training and test datasets. It is seen from Figure 6 that the NN-based method benefits from a larger number of power events in training and test datasets, though they are noisier. On the opposite, the regression-based method exhibits better RMSE and MaxAE in the case of more thoroughly cleaned power events (s_{pMax} allowance threshold is lower).

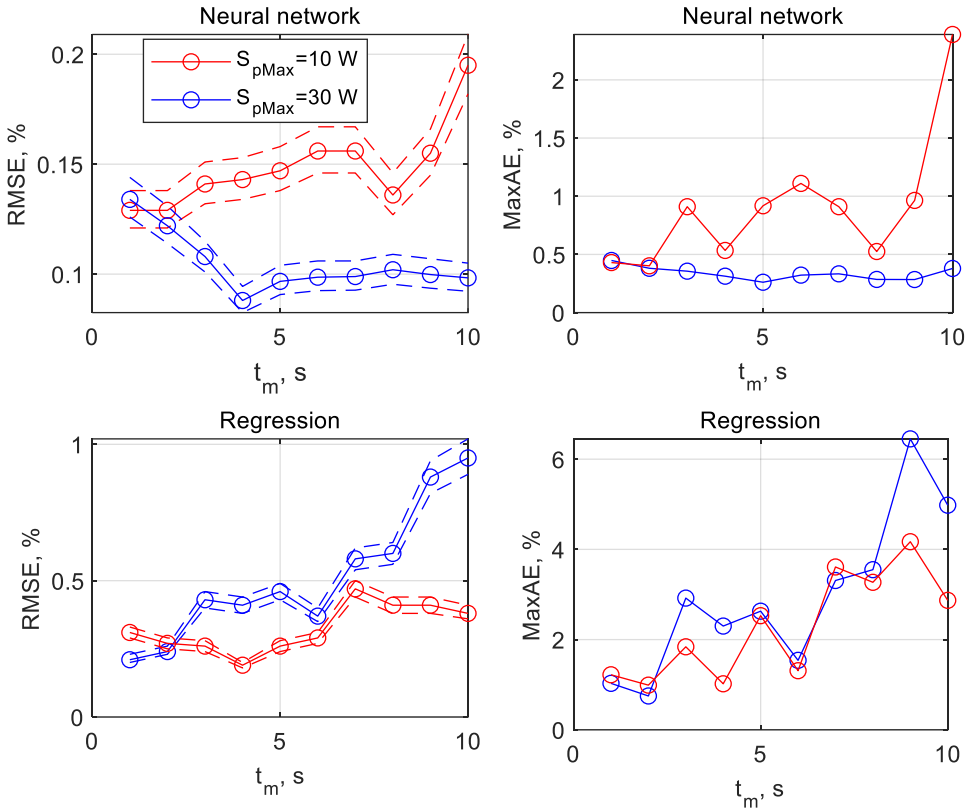

Figure 6. Power measurement gain error $e\left(g_{(P)}\right)$ RMSE and MaxAE vs. averaging interval t_m at different s_{pMax} values. Dashed lines denote 5 % to 95 % confidence limits of the RMSE estimate.

To further investigate the influence of training and test datasets on RMSE and MaxAE, Table 4 was produced. Different volumes of training and test datasets N_{Train}/N_{Test} were examined, as well as the influence of partial overlapping of samples between training and test datasets. Individual electricity consumers, as well as specific groups of consumers, exhibit characteristic consumption patterns that depend on the day of the week, time of day, and other influencing factors. Furthermore, an energy consumer typically operates a limited set of household equipment (electrical loads). This provides a basis for assuming that the same consumer devices are switched on and off at similar times under analogous load conditions and comparable power system operating states. Consequently, analogous power events are expected during both training and test (CM error monitoring) periods. Therefore, the partially overlapping training and test datasets (O mark in Table 4) were obtained by allowing the same power event samples to be randomly included in both training and test datasets. Therefore, the partially overlapping training and test datasets (O mark in Table 4) were obtained by allowing the same power event samples to be randomly included in both training and test datasets. In fully nonoverlapping (N mark in Table 4) type of datasets, none of the experimental power event samples were admitted to both training and test datasets.

The following can be summarized from results presented in Table 3: (i) overlapping of training and test datasets improves RMSE and MaxAE for both Regression and NN based methods, (ii) rejection of active power events with less than dP_{min} = 250 W did not improve RMSE nor MaxAE for neither of methods, (iii) though NN-based method outperformes regression-based one RMSE of both methods stays well below 1% estimation error, (iv) considering MaxAE metrics, only some combinations of parameters t_m and s_{pMax} enabled only NN-based method to achieve a sub-1% estimation error $e(g_P)$.

Table 4. Performance criteria (RMSE/MaxAE) of smart meter error assessment methods (t_m = 4 s).

Method	Case	dP_{min}, W	N_{Train}/N_{Test} Samples, %	s_{pMax} = 10 W		s_{pMax} = 30 W	
				RMSE, %	MaxAE, %	RMSE, %	MaxAE, %
Regression R_{Ps}	1.	50	40/70, O	0.28	1.80	0.52	2.13
	2.	50	50/70, O	0.19	1.02	0.41	2.30
	3.	250	40/70, O	0.31	2.14	0.51	3.07
	4.	250	50/70, O	0.20	0.75	0.45	2.41
	5.	50	40/50, N	0.37	2.13	0.64	3.14
	6.	50	50/50, N	0.32	1.36	0.54	2.66
NN N_{Ps}	7.	50	40/70, O	0.20	0.92	0.12	0.41
	8.	50	50/70, O	0.14	0.53	0.09	0.31
	9.	250	40/70, O	0.38	1.79	0.21	1.06
	10.	250	50/70, O	0.22	1.08	0.13	0.54
	11.	50	40/50, N	0.28	1.29	0.18	0.50
	12.	50	50/50, N	0.22	0.75	0.16	0.51

O—overlapping, N—nonoverlapping training and test datasets.

To reveal the statistical distribution of CM voltage, current, and active power gain error estimation, standard boxplots in Figures 7 and 8 present their median, IQR (inter quartile range) (25–75%), 1.5-fold IQR (25–75%), and outliers beyond for the estimation errors of $e(g_V)$, $e(g_I)$ and $e(g_P)$. Every boxplot was generated from r = 300 (see Figure 5) CM error estimates. It is evident that in the case of some randomly selected training datasets, less successful error estimates may appear, and they are denoted as outliers in Figures 7 and 8. During the CM error estimation in real conditions, these cases cannot be distinguished and rejected, which means that the performance of methods should not be overstated by focusing on only RMSE metrics.

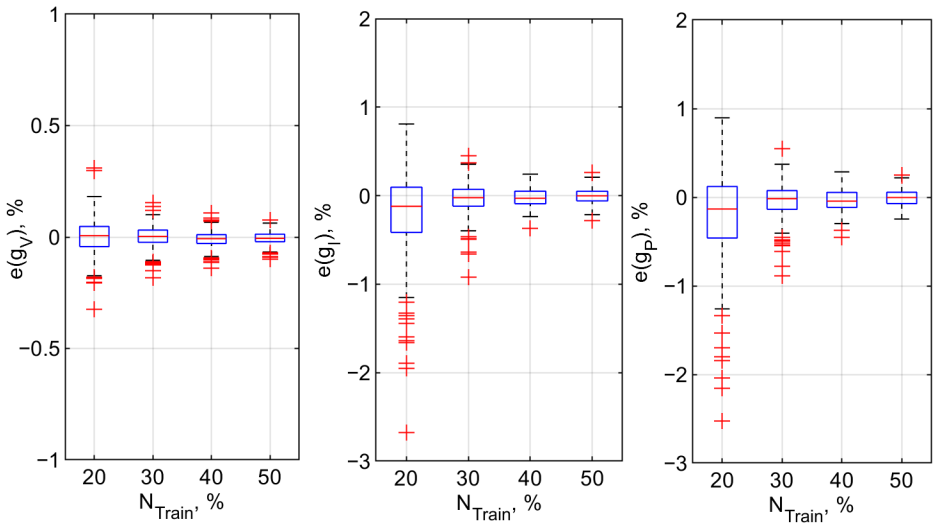

Figure 7. Voltage, current and active power gain error estimation error of NN-based method (16) box plots N_{test} = 70% testing samples overlapping with training samples, s_{pMax} = 30 W, t_m = 4 s, dP_{min} = 50 W).

Considering that the volume of the initial experimental dataset for parameters t_m = 4 s and s_{pMax} = 30 W was 353 power events (Table 3), the absolute number of training power

events (N_{Train} = 50%) is 176, which is realistic to collect over a period of one month (approximately 6 power events exceeding dP_{min} = 50 W in residential apartments).

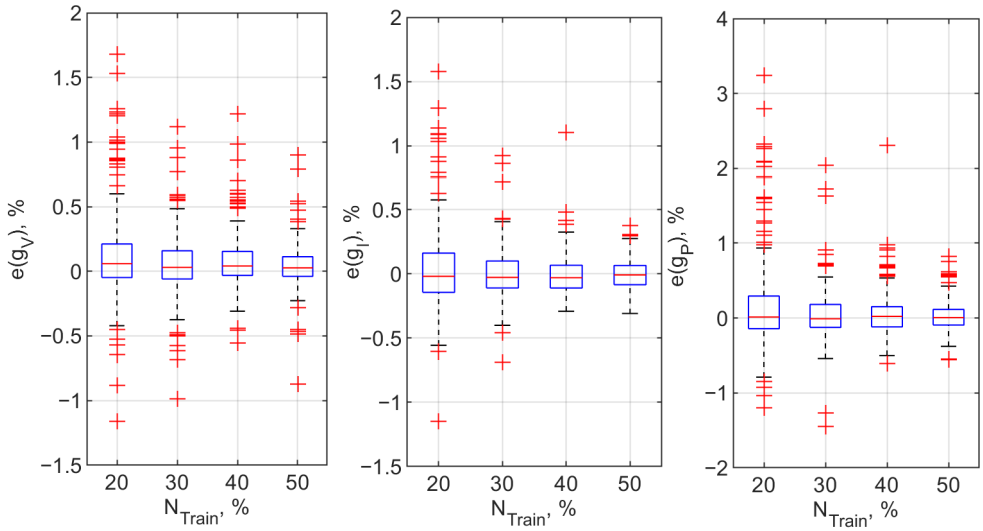

Figure 8. Voltage, current, and active power gain error estimation error of linear regression (12) based method box plots (N_{test} = 70% testing samples overlapping with training samples, s_{pMax} = 10 W, t_{ms} = 4 s, dP_{min} = 50 W).

The scatterplot diagrams of injected (actual) vs. estimated power measurement gain error are shown in Figure 9 and indicate that the estimation error is not dependent on the size of the actual meter power measurement gain error.

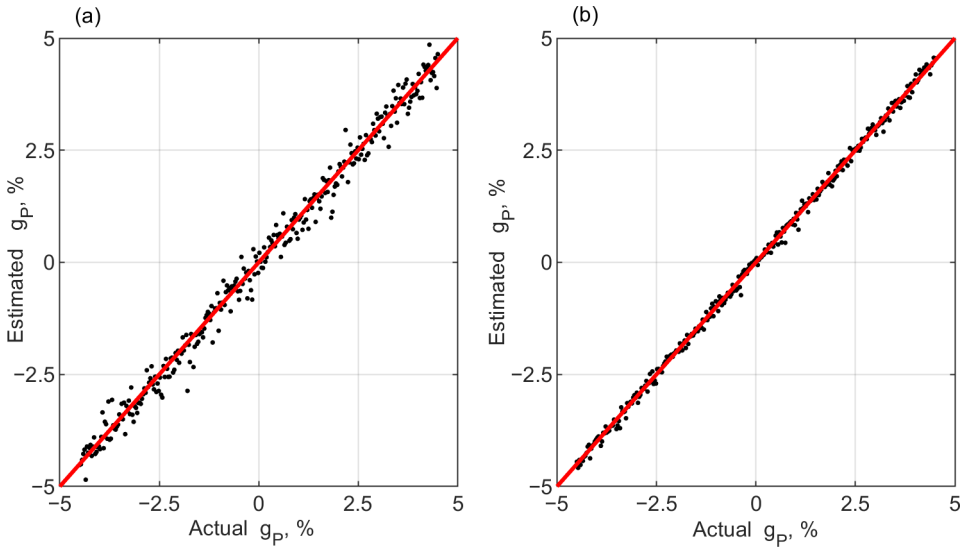

Figure 9. Active power measurement gain error estimation scatter plot (N_{train} = 50%, N_{test} = 70%, overlapping with training samples, S_{pMax} = 30 W, t_{ms} = 4 s, dP_{min} = 50 W): (**a**) using linear regression-based method; (**b**) NN-based method.

The NN-based method described in [17] was reused in this research. The number of NN ensembles (n = 50) was suggested to reduce the random nature of training by averaging n = 50 predictions from separately trained NNs. This means that n = 50 NN training is performed on the same training dataset, and n = 50 NN models have to be stored in the memory of a processor. Also, n = 50 NN will be used to predict dP_S according to (16). In Figure 10, the influence of NN ensembles on RMSE and MaxAE is shown. It can be seen that a smaller ensemble degrades method performance. However, considering the MaxAE

metrics, the sub-1% CM power measurement gain error estimation can be achieved even if the NN ensemble is reduced to n = 10, retrained, and applied in the test-period NNs.

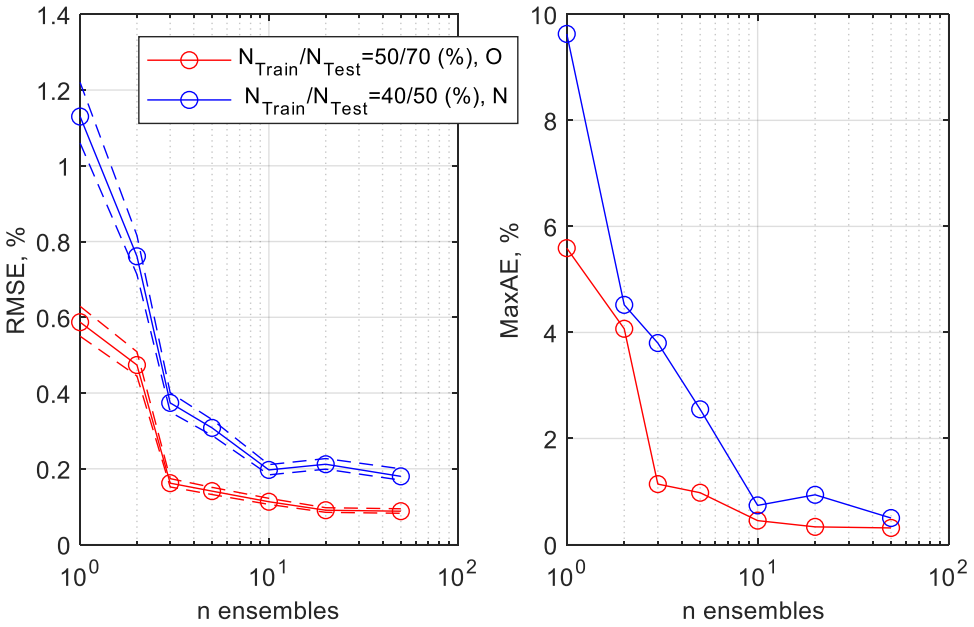

Figure 10. RMSE and MaxAE vs. NN ensemble size. Dashed lines denote 5 % to 95 % confidence limits of the RMSE estimate.

We consider that the linear regression-based technique might be executed on embedded systems in IoT edge infrastructure, whereas the NN-based method is a cloud server application, especially in the training stage.

4. Implementation Feasibility Analysis

4.1. Requirements for the Implementation of Proposed Methods

The implementation of the technique involves the collection of data from CM and SM (see the sequence diagram in Figure 11), assisted by an Aggregator unit. An Aggregator unit could be located on the Edge or Cloud, and is server-dependent. The final destination of all collected data for the training of prediction models (12) and (16) and later CM error estimation (1) is a Central server connected to the WAN and maintaining a database for meter readings storage.

The proposed technique does not require real-time CM-sampled reading delivery to the Aggregator and Central processing server. Moreover, the data would be collected for one month (we expect typically 10 power events per meter per day), and an error estimation procedure could then be performed using this data. This does not require that the detected power event data be delivered to the central processing unit.

The data payload X_{cEi} (Figure 11) from CM to Aggregator (according to Sections 2.2 and 2.3) is n = 20 samples of k_{CM} = 2 electrical quantities and one timestamp per one power event. In binary format, each sample timestamp is a 64-bit UTC Unix epoch. The data payload from SM to Aggregator X_{sEi} (Figure 11) is n = 20 samples of k_{SM} = 3 electrical quantities and one timestamp per one power event. If transmitted in text-format frames (like JSON), the amount of payload is higher correspondingly.

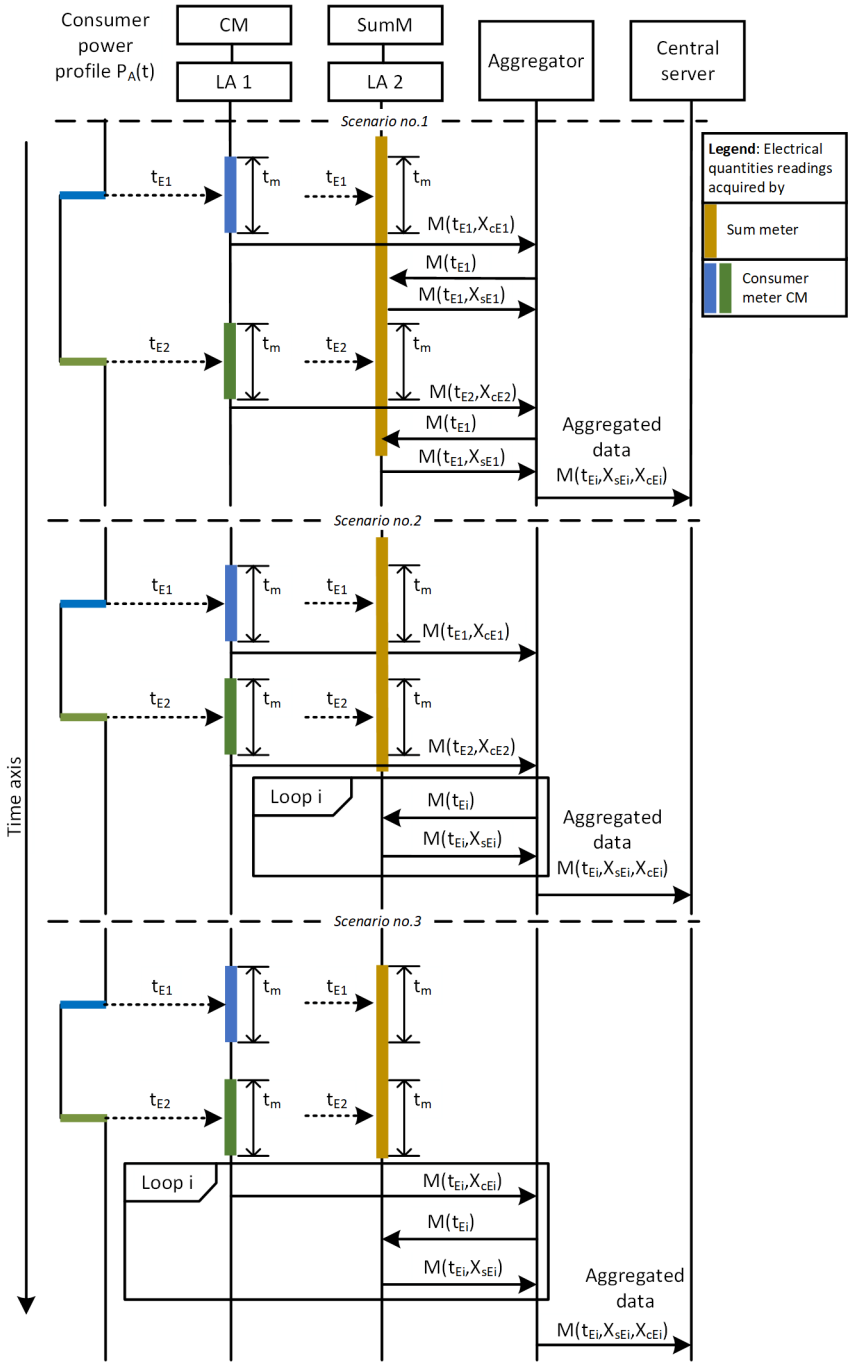

Figure 11. Scenarios of the application-level sequence diagram of power event data collection.

The proposed acquisition architecture requires only moderate relative time synchronization between the consumer meter and the sum meter, as the event-correlation procedure allows synchronization accuracy of approximately 1 s. Event timestamps are assigned locally at the meter-side adapter upon data reception from the meter, prior to MQTT transmission. Consequently, any communication latency associated with an increasing number of CMs does not influence the accuracy of the clock used to generate the timestamp, which is assigned to the detected power event. As a result, timestamp-based correlation of CM- and SM-detected power events, which is applied to locate the same consumer power event in both CM and SM data streams, is not affected by the latency in the communication channel. The timestamp accuracy depends on clock synchronization, which each adapter maintains independently via the NTP protocol. NTP protocol compensates for the delays in the uplink and downlink channels between the NTP server and the remote unit (LA of

each meter in our application). Under this assumption, Network Time Protocol version 4 (NTPv4) provides the most feasible baseline solution when the meters or associated local adapters (LAs) communicate via an IP-enabled communication infrastructure. In resource-constrained embedded nodes, the SNTP may be utilized as a more lightweight implementation, as it is specifically designed for scenarios where a complete NTP implementation is unnecessary. Furthermore, the NTPv4 protocol can achieve significantly superior performance compared to the 1 s requirement of the considered application, especially in local network configurations. Therefore, NTP/SNTP-based synchronization is adequate to ensure the necessary timestamp consistency for event association.

From an implementation point of view, an efficient solution is to distribute the UTC time from a site-local reference node, such as the gateway or edge server, to all LAs involved in the acquisition process. This approach minimizes the sensitivity of clock synchronization to fluctuating wide-area network latencies and facilitates the administration of a unified time reference throughout the monitored facility. Furthermore, when synchronization messages traverse untrusted IP networks, authenticated NTP is preferred. In this scenario, Network Time Security (NTS) offers cryptographic measures for client-server NTP communications and should be considered whenever the integrity and authenticity of timing data are critical.

4.2. Scenarios for Data Acquisition

We consider several application-level scenarios for CM readings collection as shown in Figure 11. The sequence diagram is adapted to describe a system consisting of one CM and an SM. In a real situation, the number of consumer meters N_C monitored by a single SM in the electrical grid section can be higher.

Every meter is equipped with a local adapter (LA) which reads measurements over the local interface, for example, P1 [34], and is able to communicate with the Aggregator unit over LAN, for instance, a WiFi network. A typical LA implementation could be an ESP32 microcontroller or a similar embedded system-based low-cost electronic unit.

The proposed method transmits only time-stamped power-step windows and auxiliary identifiers, excluding continuous waveform streams. Thus, the communication service is event-driven, decentralized across numerous meter-side adapters, and tolerant to delivery latencies ranging from seconds to minutes. The method is compatible with various IoT and smart metering protocol stacks [35], depending on the ability of LA to locally buffer short records and maintain a proper level of time synchronization.

On the meter-facing side, the LA separates the proposed data acquisition logic from the native meter interface. This role can be executed by a P1 or similar serial/optical port for customer meters, whereas in utility-oriented implementations, the same function can be realized through DLMS/COSEM or OMS data objects already present in the AMI ecosystem [36]. The protocol selected below the application layer does not affect the semantics of the collected datasets X_{cEi} and X_{sEi}; it simply affects the rate and efficiency of event record transmission to the Aggregator.

The MQTT/RESTful over Wi-Fi option represents the most direct implementation for laboratory validation, pilot installations, and buildings with an existing IP network. It provides sufficient throughput for the immediate transmission of all accepted power-step events, simple bidirectional configuration of the LA, and seamless integration with edge or cloud software. Consequently, it is the optimal selection for Scenario 1 in Figure 11, where the aim is to transmit detected events with minimal latency and negligible local buffering. The options of NB-IoT, wireless M-Bus, and PLC represent solutions that align more closely with current smart-meter infrastructures. DLMS/COSEM and OMS are appealing in these instances due to their established applications in structured meter reading, register access,

and device management. Their primary advantage lies in the feasibility of deployment within utility environments, where coverage, security policies, and fleet management might favor standardized metering stacks over generic IP connectivity. Their disadvantage is increased latency and reduced effective uplink capacity compared to Wi-Fi [37], particularly when numerous devices compete for access or when the channel is in duty cycle. Thus, these options are most compatible with Scenario 2, where the LA initially stores the event locally and subsequently uploads the complete record upon request or at predetermined intervals. Wired serial connections, such as RS485 with Modbus, are relevant when the Aggregator is situated within the same cabinet, substation, or building control room as the meters. This stack is robust, cost-effective, and easily integrable with industrial gateways, while avoiding radio-coverage issues. Conversely, the LoRa or Zigbee protocols represent an open low-power alternative rather than a singular, fixed protocol mapping. These technologies are appropriate only when the payload is compact or when multiple events are aggregated, as their data rate, duty-cycle constraints, and downlink asymmetry make them unsuitable for frequent large uploads. Therefore, they correspond more closely to Scenario 3, which prioritizes communication efficiency over latency.

The key requirement is that the selected stack must support: (i) time-stamped transmission of brief event records, (ii) asynchronous control messages for synchronization and configuration, and (iii) temporary local storage in the LA when the channel is inaccessible or intentionally scheduled. According to these criteria, MQTT over IP is utilized in Section 4.4 as a specific reference mapping; however, the same acquisition logic can be maintained across DLMS/COSEM, OMS, or Modbus-based implementations.

4.3. AMI Integration and Protocol Mapping

The proposed acquisition layer is intended to operate as a parallel retrofit system with respect to existing AMI installations, rather than as a replacement or modification of the utility metering infrastructure. The existing AMI channel continues to support regulated billing, remote configuration, and manufacturer-specific meter management. The additional LA is connected only to already available local or communication interfaces and reads the electrical quantities required by the proposed power-event method. Therefore, the method can be deployed without changing meter firmware, without affecting the certified metrological functions of installed devices, and without requiring a homogeneous fleet of meters from the same manufacturer.

AMI compatibility is achieved by separating the meter-facing protocol from the cloud/edge-facing acquisition protocol. On the meter-facing side, the LA implements the parser required by the installed device, for example, DSMR/P1, DLMS/COSEM over optical, RS485, PLC or cellular links, OMS/wireless M-Bus, or Modbus in cabinet-level installations. The LA converts vendor- or protocol-specific registers into the common event-window structures XcEi and XsEi used by the estimation algorithms. On the northbound side, these normalized event records are exposed to the Aggregator through MQTT topics or through an equivalent IP/edge gateway interface. This decoupling makes the application layer independent of the original meter manufacturer and of the native AMI backhaul.

Three practical integration paths can be distinguished. First, in a customer-premises or building retrofit, the LA reads the local P1, optical, or serial interface and publishes event windows through MQTT over Wi-Fi/Ethernet, leaving the utility AMI untouched. Second, in a utility-operated deployment, the LA function can be embedded in, or connected to, an existing data concentrator that already communicates with meters through DLMS/COSEM, PLC, NB-IoT, or wireless M-Bus. In this case, the concentrator performs the same normalization before forwarding events to the edge or cloud broker. Third, in installations where direct meter access is not allowed, the Aggregator may receive equivalent high-resolution

register data exported by the AMI head-end system, provided that timestamp resolution and event-window buffering satisfy the requirements of Section 4.1. In all cases, authentication, topic-level authorization, and separation between the billing AMI channel and the diagnostic event-acquisition channel are required to avoid interference with revenue metering and utility operation.

Accordingly, MQTT is not imposed as a meter protocol. It is used in this work as a lightweight northbound transport after meter-specific data have been normalized by the LA. This choice was adopted because MQTT natively supports asynchronous publication, retained configuration messages, Last Will status information, QoS-based delivery, and straightforward deployment in cloud and edge infrastructures. Other AMI-compatible protocols can therefore be retained below the LA, while the diagnostic application receives a uniform data model above it.

4.4. Mapping Data Acquisition Scenarios to the MQTT Protocol

This paper uses MQTT over an IP network as the reference implementation because it fits the asynchronous, event-driven nature of the proposed acquisition method very well, but it limits at the same time the communication delay as discussed in [38]. Each meter-side LA acts as an MQTT client that connects to the LAN network. The broker can be hosted at the central server in a cloud deployment. The Aggregator is the logical consumer of raw event messages and the producer of merged records, while the Central server stores the merged datasets and makes them available to the training and error-estimation workflow. This setup separates meter access from back-end processing, removes direct LA-to-LA communication, and allows individual wireless links to be temporarily disconnected without breaking the application logic in Section 4.2.

For each accepted power step event i, the CM-side LA publishes $m_C^i = \{id_C, t_{Ei}, X_{cEi}, seq_{Ci}, q_{Ci}\}$, while the SM-side LA publishes $m_S^i = \{id_S, t_{Ei}, X_{sEi}, seq_{Si}, q_{Si}\}$. t_{Ei} is the event timestamp, X_{cEi} and X_{sEi} are the local measurement windows defined in Section 4.1, seq_* is a monotonically increasing sequence number used for duplicate detection, and q_* is an optional quality field describing local conditions such as missing samples, clock resynchronization, or buffer replay. In the remainder of the paper, JSON and compact binary mapping are considered and compared in the validation phase. After reception, the Aggregator associates CM and SM messages that belong to the same electrical event by combining the site identifier, meter-pair configuration, and timestamp proximity $| t_C - t_S | \leq 1$ s, and plausibility rules on the sign and magnitude of the power step. The merged record is then formed as $m_A^i = \{id_C, id_S, t_{Ei}, X_{cEi}, X_{sEi}, meta_i\}$.

The topic tree is *sc/<site>/cm/<meterId>/event* for CM event windows, *sc/<site>/sm/<meterId>/event* for SM event windows, *sc/<site>/agg/<pairId>/event* for merged records, *sc/<site>/agg/<pairId>/request* for pull commands or retransmission requests, *sc/<site>/<nodeId>/status* for heartbeat and availability information, and *sc/<site>/<nodeId>/cfg* for remote configuration. Event topics are non-retained because they represent transient time-series windows, whereas configuration topics can be retained so that a reconnecting LA immediately receives the current sampling, threshold, and buffering settings. Status topics should use MQTT Last Will and Testament to mark an LA as offline whenever the connection is lost unexpectedly. This is important because Scenarios 2 and 3 rely on local buffering, so the Aggregator must distinguish between normal delayed delivery and a communication fault.

With respect to the MQTT service parameters, QoS 1 is the preferred configuration for event and control messages. It guarantees at-least-once delivery and therefore prevents silent loss of accepted events, while the residual risk of duplicate delivery can be handled deterministically by the Aggregator using (seq$_*$) and (t_{Ei}). QoS 0 is sufficient for frequent heartbeats and non-critical diagnostics, whereas QoS 2 is unnecessary because its stronger

handshake increases protocol overhead without improving the utility of the final merged dataset. Persistent sessions are useful when the underlying channel is intermittent, as they enable broker-side storage of undelivered messages during temporary communication interruptions. In MQTT 5, this functionality is controlled by session expiration timers; in MQTT 3.1.1, the equivalent effect is achieved by deactivating clean sessions. From a deployment perspective, TLS protection, per-device authentication, and topic-based access control are necessary to ensure that an LA can exclusively publish to its designated namespace and subscribe only to the control topics intended for that node.

Scenario 1 in Figure 11 maps to immediate event-driven publishing. As soon as LA1 detects a power step, it sends the full window (X_{cEi}) and publishes it directly to the broker. Upon reception of the power step message from LA1, the Aggregator requests power step data from SM by sending a message containing a power step timestamp t_{E1}. Then the Aggregator performs timestamp-based correlation in near real time and forwards the merged record (m_{Ai}) to the Central server (that hosts the broker server) with no intentional waiting time. This mapping minimizes local memory, simplifies replay logic, and supports online supervision of acquisition quality. Its drawback is that every accepted power step generates a distinct MQTT transaction on both the CM and the SM sides, so the number of broker operations grows linearly with the total event count. Scenario 1 is therefore the natural choice for Wi-Fi or wired Ethernet environments, for laboratory validation, and for installations where rapid visibility of new events is more important than minimizing transmission count.

Scenario 2 corresponds to decoupled capture and deferred transfer. At the instant of detection, the LA stores the complete event window locally and publishes only a compact trigger that contains the event time, meter identifier, event class, and an optional step-magnitude estimate. Based on these triggers, the Aggregator decides which detailed records must be uploaded immediately, which can wait until the next reporting slot, and whether retransmission is needed for missing partners of a CM-SM pair. In practice, the detailed window can be fetched either by a dedicated MQTT request topic or by a scheduled upload from the LA after a configurable delay. This mapping smooths channel utilization, enables fairness among many meters sharing the same link, and is particularly attractive for MQTT over NB-IoT, PLC, or utility-operated networks where uplink opportunities are constrained or billed. The price paid for this flexibility is a moderate buffer requirement in the LA and a slightly more complex state machine at the Aggregator.

Scenario 3 is a periodic batch mode intended for the most bandwidth-limited or energy-constrained deployments. Each LA accumulates (q) consecutive accepted events and publishes them in one MQTT payload after a batching interval (T_{batch}) or when the buffer occupancy reaches a threshold. The Aggregator then reconstructs all CM-SM pairs offline within the batch and forwards only the validated merged records upstream. In this case, the effective protocol overhead per event is reduced approximately from (B_H) to (B_H/q), where (B_H) denotes the average MQTT and topic overhead of one publication and q represents the number of accumulated events. The method is therefore well-suited to low-duty-cycle wireless links or multi-hop IoT segments in which radio wake-ups dominate energy consumption. However, the batch approach introduces the highest end-to-end latency and requires the most careful design of local storage because the LA must preserve all not-yet-transmitted events across short communication interruptions and possible broker reconnections.

4.5. Evaluation of the Scenarios

In order to compare the three communication scenarios introduced in Figure 11 on a common basis, three performance metrics are considered, namely the uplink average

channel capacity, $\overline{C}_{u,x}$, the downlink average channel capacity, $\overline{C}_{d,x}$, and the local memory requirement, M_x, where the subscript $x \in \{1, 2, 3\}$ denotes the considered communication scenario. These metrics aim to quantify the communication load and the buffering resources necessary for the proposed data acquisition architecture, regardless of the electrical characteristics of the monitored installation.

The first metric, $\overline{C}_{u,x}$ can be analytically estimated using the uplink data volume, U_x. It represents the total uplink application-layer data volume generated during the observation interval T_{obs}. More specifically, U_x accounts for the amount of data transmitted from the local adapters (LAs) associated with the consumer meter and the sum meter toward the MQTT broker/Aggregator. Therefore, U_x estimates the communication effort required to deliver event notifications, complete event windows, or batches of events, depending on the adopted scenario. Since the uplink channel is the dominant communication direction in the proposed architecture, U_x is the main indicator used to compare the efficiency of the three solutions from the viewpoint of network utilization.

The second metric, $\overline{C}_{d,x}$, can be analytically estimated using the downlink data volume, D_x. It denotes the total downlink application-layer data volume transmitted during the same observation interval. In the considered system, downlink traffic is produced by broker acknowledgments, control messages, configuration updates, or explicit requests sent by the Aggregator toward the LAs. Although D_x is generally smaller than the corresponding uplink load, it remains relevant for scenarios in which the Aggregator actively controls the upload of detailed records, as in the deferred-transfer realization of Scenario 2. Accordingly, D_x provides a measure of the control overhead associated with each communication policy.

The third metric, M_x, expresses the local memory buffer requirement of one LA. This quantity corresponds to the maximum amount of data that must be stored locally in the LA in order to guarantee lossless operation under the considered communication strategy. Thus, M_x is directly related to the hardware requirements of the device and to its ability to tolerate delayed transmission, batching, or temporary communication outages. While Scenario 1 minimizes the need for local storage because events are transmitted immediately, Scenarios 2 and 3 require progressively larger buffers due to deferred upload and batch accumulation, respectively.

Formally, the three metrics can be defined over the observation interval T_{obs} as follows. Let $V_u^{(x)}$ be the cumulative uplink data volume generated in Scenario x, and let $V_d^{(x)}$ be the corresponding cumulative downlink data volume. Then,

$$U_x = V_u^{(x)}, \; D_x = V_d^{(x)}. \tag{20}$$

Moreover, let $Q^{(x)}(t)$ denote the buffer occupancy of one LA at time t in Scenario x. The required local memory is then defined as the peak queue occupancy over the observation interval,

$$M_x = \max_{t \in [0, T_{obs}]} Q^{(x)}(t). \tag{21}$$

According to these definitions, U_x, D_x and M_x are measured in bytes. Finally, the corresponding average channel capacity can be obtained by normalizing the cumulative data volumes by the observation interval:

$$\overline{C}_{u,x} = \frac{U_x}{T_{obs}}, \; \overline{C}_{d,x} = \frac{D_x}{T_{obs}}. \tag{22}$$

The adopted metrics are complementary. Specifically, $\overline{C}_{u,x}$ evaluates the channel capacity required on the upstream communication channel, $\overline{C}_{d,x}$ quantifies the control and acknowledgment traffic in the reverse direction, and M_x characterizes the buffering capa-

bility required at the edge. They provide a compact but comprehensive basis for comparing the three communication scenarios and for selecting the most appropriate implementation according to network constraints, device-memory limitations, and acceptable delivery latency. The three considered metrics are summarized in Table 5.

Table 5. Performance metrics of the communication scenario grading.

Performance Metrics	Description	Unit of Measure
$\overline{C}_{u,x}$	Uplink average channel capacity	kB/day
$\overline{C}_{d,x}$	Downlink average channel capacity	kB/day
M_x	LA unit memory buffer size	kB

The considered metrics can be analytically expressed directly from the MQTT mapping and event payload sizes introduced in Section 4.4. Let B_C and B_S denote the sizes of one CM and one SM event record, respectively, let B_H denote the average overhead of a single MQTT publication, including topic name and metadata, and let B_{ACK} denote the size of the MQTT ack packet, and B_R denote the size of a request packet sent by the Aggregator to the SM. For N_C consumer meters and N_S accepted power steps per consumer over the observation interval, the application-layer uplink data volume of Scenario 1 is approximately $U_1 = N_C N_S (B_C + B_S + 2B_H)$. When a power step is identified, the Aggregator sends an event-specific downlink message $M(T_{Ei})$ in order to tell SM which event data to deliver. In addition, the MQTT QoS 1 acknowledgment returned to each publisher should be considered. Since one matched CM–SM event pair generates one CM publication and one SM publication, the broker sends one PUBACK to CM and one PUBACK to SM. Thus, the application-layer downlink of Scenario 1 is approximately $D_1 \approx N_S(B_R + B_H) + 2N_C N_S B_{ACK}$, where B_R is the size (in bytes) of the Aggregator request packet.

Scenario 2 replaces immediate full upload by trigger-assisted transfer, so its total volume is of the same order, but its instantaneous rate is lower because detailed uploads are scheduled rather than simultaneous. For Scenario 2, the uplink data volume must include two phases: (i) trigger transmission from the LA when an event is detected; (ii) delayed upload of the full event windows when the Aggregator requests them, or when the LA uploads them after a delay. If every trigger leads to the upload of both detailed records, thus $U_2 = N_C N_S [B_C + B_S + 2B_T + 4B_H]$, where B_T is the size (in bytes) of one trigger message payload. Scenario 2 is the only case where downlink may become significant, because the Aggregator may request the detailed event windows after receiving the trigger messages. If the Aggregator explicitly requests a detailed upload from each side, then for each matched event pair, the downlink includes 2 PUBACKs for the trigger uplinks, one request to the CM-side LA, and one request to the SM-side LA. Thus, over the observation interval, the application-layer downlink of Scenario 2 is approximately $D_2 \approx N_C N_S [2B_{ACK} + 2(B_R + B_H)]$, where B_R is the size (in bytes) of the Aggregator request packet.

In Scenario 3, the LA accumulates q events and then publishes them in one MQTT batch. Thus, the uplink data volume can be analytically estimated as $U_3 \approx N_C N_S (B_C + B_S + 2B_H/q)$, which shows why the batching approach becomes attractive when the header overhead and network access cost dominate. Under normal operation, no per-event request is needed, so the systematic downlink is the QoS 1 PUBACK and the Aggregator request. However, the important difference is that one PUBACK now corresponds to a batch of q events, not a single event. Thus, the downlink data volume under Scenario 3 can be analytically estimated as

$$D_3 \approx \frac{N_S\,(B_H + B_R) + 2\,N_C N_S\,B_{ACK}}{q} \tag{23}$$

The required LA memory follows the opposite trend: $M_1 \approx B_C$ or B_S, $M_2 \approx N_{buf}\ (B_C$ or $B_S)$, and $M_3 \approx \lambda_E T_{batch}(B_C$ or $B_S)$ for one logical stream, where N_{buf} is the maximum number of locally stored events, λ_E is the mean event rate and T_{batch} the batching interval.

To support the analytical evaluation of the three communication scenarios under different conditions, the sizes of the main MQTT message components were estimated for a compact JSON application-layer mapping and for a binary application-layer mapping. Assuming a compact JSON serialization, the source role and meter identifier are conveyed by the MQTT topic described above, so that the payload contains only the event data and the minimal metadata required for correlation and replay management. Under this assumption, the CM event record can be estimated as $B_C = 401$ B and the SM event record as $B_S = 281$ B. For QoS 1 publishing, the MQTT topic and protocol overhead is $B_H = 34$ B per publication, while the corresponding acknowledgment size is $B_{ACK} = 4$ B. For the trigger-based realization of Scenario 2, a compact JSON trigger carrying the timestamp, sequence number, quality flag, event type, and quantized step magnitude can be estimated as $B_T = 53$ B. Hence, one complete CM-SM event pair generates approximately 750 B of MQTT application-layer traffic.

Assuming a binary MQTT mapping, and considering still valid the previous assumptions (see Equation (16) $\boldsymbol{X_{NN}}$ consisting of three quantities from SM and two quantities from CM, with 20 samples surrounding the power step moment for each electrical quantity), the CM event window has size $\mid X_{cEi} \mid = 20{\cdot}3{\cdot}2 + 8 = 128$ B, while the SM event window has size $\mid X_{sEi} \mid = 20{\cdot}2{\cdot}2 + 8 = 88$ B. By adding a 32-bit sequence counter and a 1-byte quality field, the complete event-record sizes become $B_C = 133$ B and $B_S = 93$ B, respectively. For QoS 1 MQTT, publishing the protocol and topic overhead is $B_H = 34$ B per publication, while the corresponding PUBACK size is $B_{ACK} = 4$ B. For the trigger-first realization of Scenario 2, a compact binary trigger can be encoded in $B_T = 16$ B by using one timestamp (8 B), one sequence counter (4 B), one status byte (1 B), one quality byte (2 B), and a quantized step indicator (2 B). The considered parameters are summarized in Table 6.

Table 6. Summary of the parameters considered in the analysis.

Parameters	JSON Mapping (Bytes)	Binary Mapping (Bytes)
B_C	401	133
B_S	281	93
B_H	34	34
B_{ACK}	4	4
B_T	53	16
B_R	54	20

Figure 12 shows that in Scenario 1, both uplink and downlink network loads increase linearly with the number of consumer meters (NC) and the number of accepted power steps (NS). This behavior is expected because each accepted event generates an immediate CM publication, while the downlink includes the corresponding MQTT acknowledgments and the request from the Aggregator. Therefore, Scenario 1 provides the lowest delivery latency and the simplest communication logic, but also produces the highest message rate and the strongest daily traffic growth as the monitored installation scales up. Figure 13 highlights the behavior of Scenario 2, in which complete event records are not uploaded immediately but only after a trigger-first phase followed by deferred transfer. As a result, the total uplink load remains of the same order as in Scenario 1, but the communication

is smoother because detailed uploads can be scheduled in time. Consequently, Scenario 2 represents a balanced compromise between communication efficiency and implementation complexity, especially for constrained or utility-managed links. Figure 14 demonstrates that Scenario 3 is the most communication-efficient solution among the three considered cases. Since multiple accepted events are accumulated and transmitted in one batch, the effective MQTT overhead per event is reduced approximately from (B_H) to (B_H/q), where q is the number of events per day. This reduces both uplink and downlink traffic compared to the previous scenarios. The benefit becomes more evident as (N_C) and (N_S) increase, making batching particularly attractive for bandwidth-limited or duty-cycled networks. This gain in channel efficiency is achieved at the expense of longer delivery latency and higher local buffering requirements in the meter-side adapters.

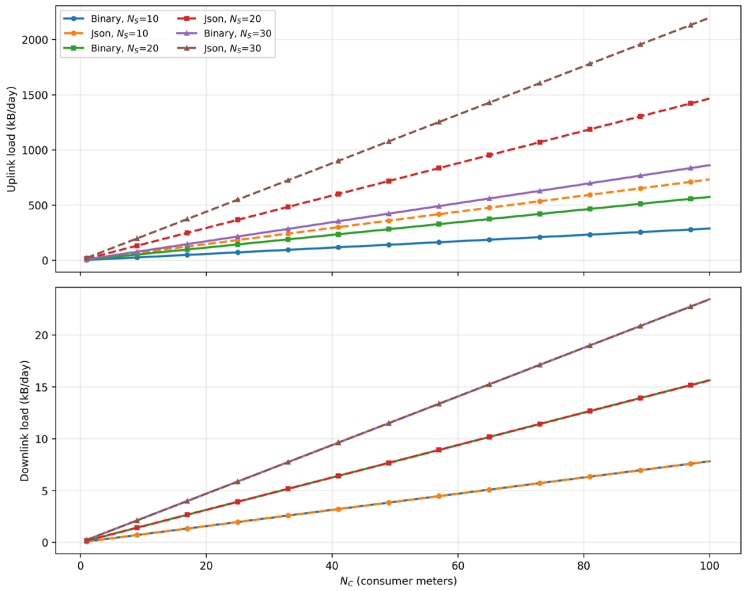

Figure 12. Network uplink and downlink load [kB/day] vs. Nc (number of consumer meters from 1 to 100) vs. Ns (Number of power steps in each CM 10, 20, 30 steps). One-dimensional plot with the family of lines. Scenario 1.

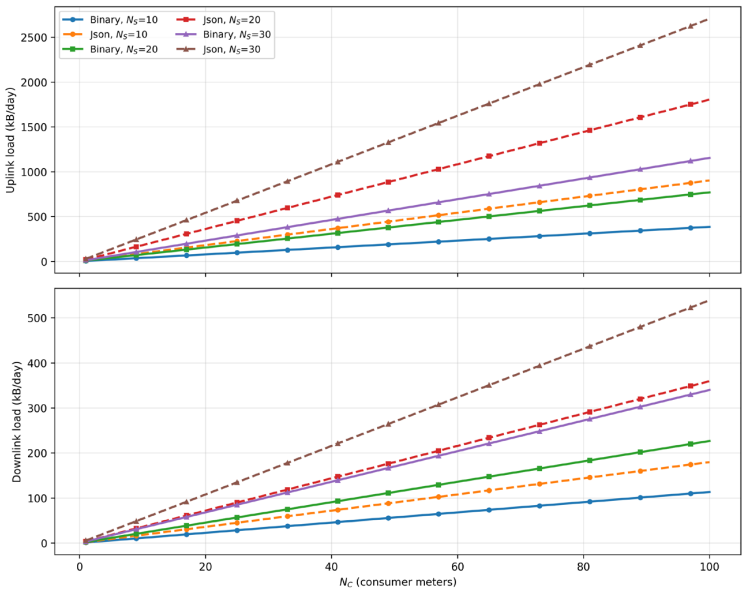

Figure 13. Network uplink and downlink load [kB/day] vs. Nc (number of consumer meters from 1 to 100) vs. Ns (Number of power steps at each CM 10, 20, 30 steps). One-dimensional plot with the family of lines. Scenario 2.

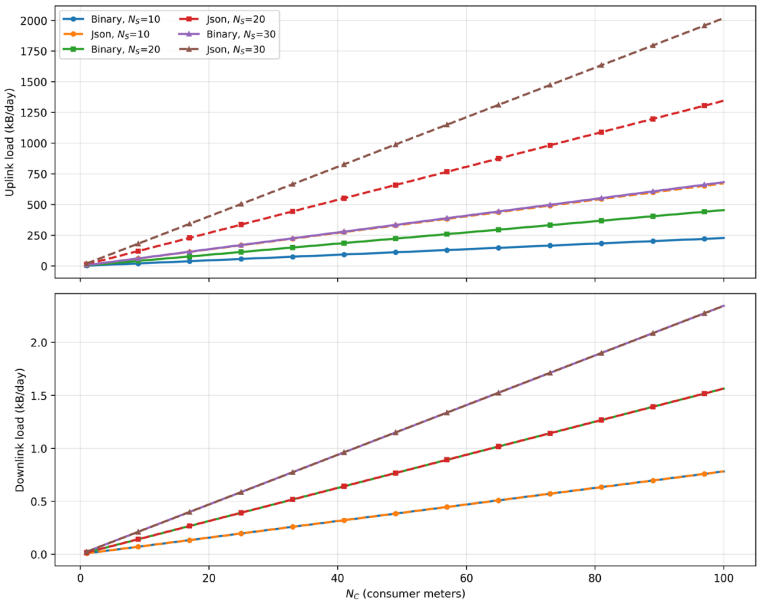

Figure 14. Network uplink and downlink load [kB/day] vs. Nc (number of consumer meters from 1 to 100) vs. Ns (Number of power steps at each CM 10, 20, 30 steps). One-dimensional plot with the family of lines. Scenario 3.

Figure 15 compares the LA memory buffer requirement on the CM side as a function of the number of power-step events (NS) accepted for the three communication scenarios and for binary and JSON packet representations. As expected, Scenario 1 gives the smallest memory requirement, because the CM-side adapter publishes the complete event window immediately after detection, and therefore only one event record, or at most a very small retransmission margin, must be kept locally. Scenario 2 requires a larger buffer, since the detected event window is first stored in the LA memory and the detailed data are uploaded only after a deferred transfer or an explicit request from the Aggregator. Scenario 3 yields the highest CM-side memory requirement, because multiple accepted events must be retained until the batching interval expires or the batch is transmitted. Consequently, the memory demand grows from Scenario 1 to Scenario 3, reflecting the trade-off between immediate transmission and communication efficiency. In all scenarios, the JSON mapping produces higher memory requirements than the binary mapping, because the serialized CM event record is larger, while the binary representation provides a more compact implementation.

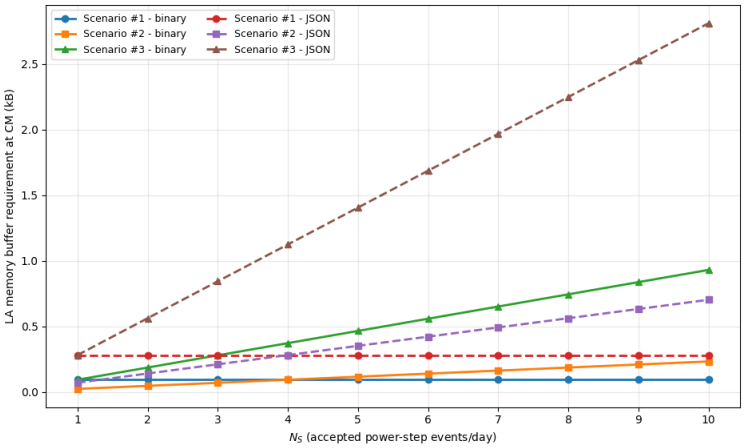

Figure 15. LA memory buffer requirements [kB] for CM vs. Ns, and then vs. communication scenario (#1, #2, and #3, Figure 11) vs. (binary/JSON packet).

Table 7 compares the SM-side LA memory buffer requirement as a function of the number of accepted power-step events N_S for the three communication scenarios and for binary and JSON data storage. Scenario 1 requires the smallest amount of memory, because the SM stores only the data associated with the last detected event before transmission; therefore, its buffer requirement is essentially constant and equal to one event record. In Scenario 2, the SM stores the last q events before transmitting them, so the required memory increases with N_S until the buffer reaches the maximum depth defined by q. Scenario 3 yields the highest memory demand because the SM stores the complete record of one day before transmission. As a consequence, the corresponding buffer requirement is dominated by daily data storage and is practically independent of the number of accepted events. In all three scenarios, the JSON representation requires more memory than the binary one, since the data are stored as ASCII characters rather than compact binary values.

Table 7. LA memory buffer requirements [kB] for SM vs. Ns, and then vs. communication scenario (1, 2, and 3, Figure 11) vs. (binary/JSON packet).

Scenario	LA JSON Mapping (kB)	LA Binary Mapping (kB)
#1	0.281	0.093
#2 [q = 10, N_S = 5]	1.405	0.465
#3	1044	345.6

5. Discussion

The findings of this study support the assumption that remote smart meter error estimation can be achieved using a sum meter serving as a reference instrument and deriving prediction models for the particular consumer meter. The methods we suggest and experimentally test in this study are different from other energy balance-based techniques published in scientific publications in a way that readings have to be acquired at the moment of power consumption event, rather than continuous 15 to 60 min intervals as in traditional energy revenue metering. Although such an approach raises specific requirements for data from smart meters collection, it is resilient to the challenges experienced by methods relying on interval energy consumption readings. Indeed, the two event-driven techniques investigated in this research show robustness to the presence of other loads monitored by the sum meter (electricity theft activities, as well as non-smart metering of some consumers during the transition from interval metering to smart energy metering in smart cities).

Although detecting a malfunctioning CM is expected as fast as possible, a significant statistic has to be accumulated in order to ensure error estimation accuracy and reliability. Both presented techniques are designed to operate using accumulated power-event data caused by consumer load activities. Fewer power events should be necessary to achieve the same estimation accuracy. The conducted analysis enables one to predict how many power steps are mandatory to make a decision based on a chosen accuracy threshold (Table 4, Figures 7 and 8). It can be concluded that not a fixed period in days or months has to be dedicated to data accumulation, but a number of power steps, which are particularly consumer-dependent.

The method presented has no principal limitations for extension to an arbitrary number of CMs. A single SM observes the aggregate power consumption of all CMs connected to the feeder line, which is monitored by SMs. Every CM has to be attributed its individual regression-based or neural network-based prediction model, which will be used to monitor this particular CM error. From the practical perspective, the higher the number of CM observable by the same SM, the probability that power events from different customers overlap is higher (happens more often). When the overlapping occurs too often, acquiring data from non-overlapping power events will take longer due to the rejection of overlapping

events. Therefore, the location of SM installation is critical and subject to optimization seeking a trade-off between the number of required SMs and the duration of acquisition of a sufficient number of non-overlapping power events from CM. Also, the higher number of CMs demands higher uplink/downlink capacity in communication channels, such as CM Aggregator unit SM Aggregator units, dependent on the chosen scenario, as reported in Section 4.5.

The main limitation of the method is that currently, utilities do not acquire one-second resolution patterns of voltage, current and active/reactive power consumption. At this stage of development, the method still has to be proved to operate reliably in diverse configurations of distribution networks in residential areas, office or industry-type networks. The large-scale deployment will have a significant cost in contrast to selected meters error monitoring, aiming at method validation. Two main prerequisites for the method deployment are (i) presence of sum meter (which is not always available in distribution grids), (ii) communication equipment between CM, SM and Aggregator unit, (iii) computational resources in central server for training prediction models, storing models for every monitored meter, and running the algorithm of the method continuously upon acquisition of power events from CMs. In this publication, the implementation requirements for an IoT-based AMI-independent case of communication infrastructure were assessed, when no firmware modification in smart meters is necessary, but inexpensive LA has to be locally connected to CM. We think that the large-scale deployment of the method is too early to consider yet. Nevertheless, the method deployment can roll out gradually, starting with a small number of selected CM for error remote surveillance. This is the advantage of the method, that it can be piloted starting from a single CM monitored by SM, and extended to a larger number of monitored CM based on the initial results and feedback from stakeholders.

Considering the sensitive issue of consumer data privacy, we need to admit that the method requires the delivery of power events moments and corresponding power changes. This is not a full power consumption pattern, but the data could be tracked to the consumer behavior. On the other hand, after processing the samples of power events, they do not need to be stored permanently. Also, considering edge scenarios partially, the data from CM could be mapped for processing in the intermediate communication infrastructure (Aggregation unit); this way, screening the delivery of sensitive consumer data to the central server's database.

The study demonstrated that acceptable error detection accuracy (Table 4) can be achieved using 360 power steps (i.e., the size of the initial database), which are required for both the model training and decision-making phases. It was also observed that a subset of power steps adversely affects the accuracy of the methods. These include steps that overlap with the activity of other loads monitored by the sum meter, are of insufficient magnitude, or exhibit significant power oscillations before or after the event.

The removal of samples corresponding to such power events during the preprocessing stage, through the application of threshold criteria defined by parameters (s_{pMax}, dP_{min}, L_{nMax}), reduces the dataset size (Table 3) but leads to improved method performance. Regression-based and neural network–based methods exhibited different sensitivities to the preprocessing parameters (Figure 6).

Further research is required to determine the optimal values of these preprocessing parameters, along with t_m. However, given that specific load configurations are characteristic of individual consumers, and that the behavior of the network at particular meter installation points exhibits consistent patterns, it is reasonable to expect that parameter optimization for a given case can be achieved using the collected dataset for the model training. Consequently, similar patterns are expected to persist during the decision-making stage.

The research also considers cases in which the training and testing datasets partially overlap. The selection of these cases was guided by practical considerations. Both individual consumers and groups of consumers exhibit characteristic electricity usage patterns influenced by factors such as the day of the week, time of day, and other contextual variables. As each consumer typically operates a relatively limited set of electrical loads, device usage tends to recur under similar conditions. Consequently, similar power events are likely to be present in both the training and testing periods.

In the current study, the implementation of CM error estimation algorithms and models requires the delivery and processing of power-step event readings at a central server. Such an approach suffers from criticism due to consumer privacy concerns. Therefore, mapping of the investigated methods (training and inference procedures) on the IoT edge infrastructure is an important task for future research. The computational demands of the regression and NN-based methods are clearly different. Though the linear regression-based method underperforms in terms of accuracy, it could be a viable choice when targeting limited resources in the grid edge infrastructure for implementation [39]. Moreover, when tens to hundreds of smart meters are each supplemented by the data-driven model for metering error monitoring, the amount of resources required for implementation in a cloud server vs. an IoT edge scenario will become the criterion of utmost importance.

Adjustment of parameters of power events detection (s_{pMax}, t_m, dP_{min}, L_{nMax}) is the next research topic aiming at the tradeoff between computational resources/communication channel capacity and performance criteria such as estimation error (RMSE and MaxAE). The optimal set of power event detection parameters is method-dependent—the regression-based method benefits from better cleaned (preprocessed) data, whereas the NN-based method achieves better accuracy given more training data sampled by SM, even if the requirements of power step detection are less strict (larger fluctuations before and after the power event are allowed).

From a methodological perspective, this study extends prior work by providing experimental verification results of SM error remote estimation accuracy using the energy balance principle and power events when SM and CM readings are acquired synchronously and delivered to the centralized server for training. The accuracies achieved are sufficient for practical detection of Class 1 accuracy meter power measurement error drifting beyond the limits of compliance.

The feasibility analysis of the implementation of data collection for the presented techniques includes a comparison of three scenarios. Each scenario is different in terms of maximizing criteria such as uplink and downlink communication channel capacity, and the required size of the memory buffer in local adapter devices connected directly to smart meters (SM and CM). Although smart meter firmware could be updated to manage the acquisition process and communication stacks, the concept of local adapters is considered in this preliminary research stage due to a much easier approach to decouple the standard smart meter from a device responsible for the functionality required by the presented observation techniques of smart meter error. Power step acquisition procedures were mapped on a well-established IoT protocol, MQTT, using an underlying WiFi network layer. In future research, alternative communication stacks could also be feasible to provide the necessary services for application layer requests according to the presented data acquisition from multiple CMs and SM.

In practical deployments, this LA-based architecture should be viewed as an auxiliary diagnostic layer coexisting with the existing AMI. It can read local or native meter interfaces and forward only normalized event windows to the edge/cloud broker, while billing, legal metrology, and fleet-management functions remain under the original AMI. This separation is particularly relevant in heterogeneous meter fleets because only the LA parser changes

when the meter protocol or manufacturer changes, whereas the Aggregator and error-estimation workflow remain unchanged.

6. Conclusions

Experimental verification of two earlier published methods for remote smart meter error monitoring has confirmed their suitability to estimate active power measurement error with the accuracy appropriate for the detection of non-compliance to measurement error requirements of Class 1 meter. Both methods, such as those (i) based on linear regression and (ii) on an artificial neural network application for the prediction of sum meter-measured power during the power event caused by consumer load activities, were found to ensure sub-1% estimation error for RMSE and maximum absolute error metrics given the experimental dataset collected from the real low voltage distribution network and using an accelerated controlled-load setup. The NN-based technique, in terms of RMSE and MaxAE, outperformed the regression-based technique. However, this is achieved at the cost of larger computational resources demanded by the NN-based method, making it harder to deploy on IoT edge devices with limited resources.

Selection of optimal power event detection parameters enables the minimization of the achievable estimation error of methods investigated: (i) the regression-based method exhibited 0.2% RMSE and 0.75% MaxAE for the set of parameters (dP_{min} = 250 W, t_m = 4 s and s_{pMax} = 10 W); (ii) the neural network-based method exhibited 0.09% RMSE and 0.31% MaxAE for the set of parameters (dP_{min} = 50 W, t_m = 4 s and s_{pMax} = 30 W).

The feasibility analysis of event-driven readings collection from a consumer smart meter and a distribution network sum meter was performed considering the MQTT protocol. Three data collection scenarios were suggested, each balancing factors like local adapter device buffer-memory volume and uplink/downlink channel loads. In particular, Scenario 1 minimizes latency and buffering, Scenario 2 provides a compromise through deferred transfer, and Scenario 3 achieves the highest communication efficiency by batching at the expense of larger local storage and longer delivery delay. Quantitatively, binary mapping reduced the CM event record from 281 B to 93 B and the SM event record from 401 B to 133 B compared to JSON, while the local buffer requirement of the SM-side decreased from 0.281 kB to 0.093 kB in Scenario 1, from 1.405 kB to 0.465 kB in Scenario 2, and from 1044 kB to 345.6 kB in Scenario 3. These results confirm the practical feasibility of the proposed MQTT-based architecture and indicate binary encoding as the most resource-efficient solution.

Author Contributions: Conceptualization, Ž.N. and S.R.; methodology, J.Š., Ž.N., M.S. and S.R.; software, V.D. and K.Z.; validation, R.L. and J.Š.; investigation, V.D.; data curation, J.Š.; writing—original draft preparation, Ž.N. and S.R.; writing—review and editing, M.S. and R.L.; visualization, V.D., J.Š., M.S. and S.R.; project administration, Ž.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research Council of Lithuania (LMTLT), agreement No S-MIP-24-42.

Data Availability Statement: The original data presented in the study are openly available on Zenodo at: https://doi.org/10.5281/zenodo.19203797 (accessed on 2 May 2026).

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.

Abbreviations

The following abbreviations are used in this manuscript:

AI/ML	Artificial Intelligence/Machine Learning
AMI	Advanced Metering Infrastructure
AMR	Automated Meter Reading
ASCII	American Standard Code for Information Interchange
CM	Consumer Meter
COSEM	Companion Specification for Energy Metering
DLMS	Device Language Message Specification—protocol that facilitates communication between utility control centers, data concentrators and smart meters
DSMR	Dutch Smart Meter Requirements—protocol used by smart energy meters
IoT	Internet of Things
IQR	Inter Quartile Range
JSON	JavaScript Object Notation
LA	Local Adapter
LAN	Local Area Network
LoRa	A physical proprietary radio communication technique based on spread spectrum modulation
M-Bus	A European standard for the remote reading of water, gas or electricity meters
MaxAE	Maximum Absolute Error
MQTT	Message Queuing Telemetry Transport
NB-IoT	Narrowband Internet of Things
NN	Neural Network
NTP	Network Time Protocol
NTPv4	Network Time Protocol version 4
NTS	Network Time Security
OMSs	Outage Management Systems
P1	A standardized interface on the smart meter that allows it to communicate real-time energy consumption data to other devices
PLC	Power Line Communication
ReLU	Rectified linear unit activation function in the context of artificial neural networks
RESTful	Communication protocol used over Wi-Fi for Internet of Things devices
RMS	Root Mean Square
RMSE	Root Mean Square Error
RS485	A standard serial communication interface
SM	Sum Meter
SNTP	Simple Network Time Protocol
UTC	Coordinated Universal Time
WAN	Wide Area Network—a large-scale network spanning cities or countries, commonly used to connect corporate branch offices and provide internet access
Zigbee	A wireless protocol

References

1. Silva, N.; Castro, R.; Ferrão, P. Smart Grids in the Context of Smart Cities: A Literature Review and Gap Analysis. *Energies* **2025**, *18*, 1186. [CrossRef]
2. Ibrahim, Q.; Qassab, M. Smart City and Smart Metering: A Technological Infrastructure for Future Urban Development. *Preprints* **2025**, 2025030741. [CrossRef]
3. Haddara, M.; Johnsen, I.; Løes, J.; Nanda Kumar, K. Transforming Energy Management with IoT: The Norwegian Smart Metering Experience. *Smart Cities* **2025**, *8*, 84. [CrossRef]
4. Rajaguru, S.; Johansson, B.; Granath, M. Exploring Smart Meters: What We Know and What We Need to Know. In *Perspectives in Business Informatics Research*; Hinkelmann, K., López-Pellicer, F.J., Polini, A., Eds.; Springer: Cham, Switzerland, 2023; Volume 493, pp. 105–120. [CrossRef]

5. Mohanty, A.; Mohanty, S.; Satapathy, A.S.; Soudagar, M.E.M.; Shahapurkar, K.; Cuce, E. Empowering smart city through smart grid communication and measurement technology. *Int. J. Low-Carbon Technol.* **2025**, *20*, 404–420. [CrossRef]
6. Esfandi, S.; Tayebi, S.; Byrne, J.; Taminiau, J.; Giyahchi, G.; Alavi, S.A. Smart Cities and Urban Energy Planning: An Advanced Review of Promises and Challenges. *Smart Cities* **2024**, *7*, 414–444. [CrossRef]
7. Sivadevuni, S.S.; Maiya, R.; Ganesh, D.; Krishna, A.R. Intelligent Metering Innovations for Urban Utility Management: A Review. In Proceedings of the 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 11–13 December 2023. [CrossRef]
8. Śmiałkowski, T.; Czyżewski, A. Detection of Anomalies in the Operation of a Road Lighting System Based on Data from Smart Electricity Meters. *Energies* **2022**, *15*, 9438. [CrossRef]
9. Morgoev, I.D.; Klyuev, R.V.; Morgoeva, A.D. Literature Review of Methods for Detecting Non-Technical Electricity Losses in Distribution Grids. *Artif. Intell. Appl.* **2026**, 1–15. [CrossRef]
10. Wang, J.; Liu, W.; Zhang, Y.; Liu, Z.; Zheng, X.; Wang, Y.; Hao, J.; Dai, X. Research on the Error Estimation Method for Electric Energy Meters of Electric Vehicle Charging Piles based on Deep Learning. *Meas. Sci. Rev.* **2025**, *25*, 40–47. [CrossRef]
11. Wang, C.; Pu, L.; Zhao, Z.; Zhang, J. A Method for Analyzing the Operating Data of Electric Energy Meters Based on Data Mining Analysis. *Int. J. Image Graph.* **2026**, *26*, 2650001. [CrossRef]
12. Duan, J.; Tang, Q.; Li, N.; Qiu, W.; Yao, W. Predictive Health Management of Smart Meters: Daily Measurement Error Forecasting Under Complex Environmental Conditions. *IEEE Trans. Smart Grid* **2025**, *16*, 2429–2438. [CrossRef]
13. Chen, X.; Zhang, H.; Zhang, C.; Wang, T.; Yan, Y.; Xu, Y. A Cloud-Based Adaptive Convolutional Network for Real-Time Error Detection in Energy Meter Systems. *Informatica* **2025**, *49*, 347–360. [CrossRef]
14. Yu, C.; Sun, N.; Gao, J.; Hong, F.; Guo, Y. A measurement error prediction framework for smart meters in typical regions. *Measurement* **2025**, *242*, 116254. [CrossRef]
15. Wu, J.; Hu, L.; Wang, J.; Fu, G. Research on Smart Meter Fault Diagnosis Based on Improved Hybrid Sampling and BiLSTM-CNN. In *Proceedings of the 2025 4th International Conference on Advanced Sensing, Intelligent Manufacturing (ASIM), Changzhou, China, 31 October–2 November 2025*; IEEE: New York, NY, USA, 2025; pp. 1–5. [CrossRef]
16. Al Khafaf, N.; Song, H.; Kamoona, A.; Sabar, N.; McGrath, B.; Yu, X.; Jalili, M. Smart meter data intelligence for sustainable distribution network operations: State-of-the-Art applications and pathways toward net-zero. *Renew. Sustain. Energy Rev.* **2026**, *231*, 116723. [CrossRef]
17. Nakutis, Ž.; Saunoris, M.; Vaičiukynas, E.; Daunoras, V.; Šaltanis, J.; Zulonas, K.; Lukočius, R. Novel Consumer Power Event-Driven Methods for Remote Estimation of Smart Meter Error. *IEEE Access* **2025**, *13*, 179504–179522. [CrossRef]
18. Liu, M.; Liu, D.; Sun, G.; Zhao, Y.; Wang, D.; Liu, F.; Fang, X.; He, Q.; Xu, D. Deep Learning Detection of Inaccurate Smart Electricity Meters: A Case Study. *IEEE Ind. Electron. Mag.* **2020**, *14*, 79–90. [CrossRef]
19. Kong, X.; Zhang, X.; Lu, N.; Ma, Y.; Li, Y. Online Smart Meter Measurement Error Estimation Based on EKF and LMRLS Method. *IEEE Trans. Smart Grid* **2021**, *12*, 4269–4279. [CrossRef]
20. Chen, L.; Lao, K.W.; Ma, Y.; Zhang, Z. Error Modeling and Anomaly Detection of Smart Electricity Meter Using TSVD + L Method. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3521314. [CrossRef]
21. Xia, T.; Liu, C.; Lei, M.; Xia, S.; Li, D.; Ming, D. Measurement Error Estimation for Distributed Smart Meters Through a Modified BP Neural Network. *Front. Energy Res.* **2022**, *10*, 928681. [CrossRef]
22. Zhong, Y.; Yang, G.; Liang, J. Rapid Identification Method for the Sudden Change of Operating Errors in Smart Electricity Meters. In Proceedings of the 2023 5th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), Chengdu, China, 19–21 May 2023. [CrossRef]
23. Liu, F.; Huang, H. Smart Meter Error Estimation in Topological Low Voltage Energy System. *IEEE Access* **2024**, *12*, 147422–147437. [CrossRef]
24. Yang, Z.; Wang, Y.; Chen, J.; Zhou, Z. Smart Meter Fault Diagnosis based on Directional Gradient KNN. In Proceedings of the 2023 5th International Conference on System Reliability and Safety Engineering (SRSE), Beijing, China, 20–23 October 2023. [CrossRef]
25. Tong, X.; Ma, J.; Ma, L.; Yan, S.; Tang, Q.; Teng, Z.; Cheng, D. A Novel Prediction Method for Smart Meter Error Using Multiview Convolutional Neural Network. *IEEE Sens. J.* **2024**, *24*, 42009–42017. [CrossRef]
26. Ronaghi, S.; Ferrero, A.; Salicone, S.; Jetti, H.V. Data-Driven Anomaly Detection for Smart Energy Meters. In *Proceedings of the 14th IEEE International Workshop on Applied Measurements for Power Systems (AMPS), Graz, Austria, 25–27 September 2024*; IEEE: New York, NY, USA, 2024; pp. 1–6. [CrossRef]
27. Amirkhanova, G.; Aidynuly, A.; Adilzhanova, S.; Fu, Y.; Dina, B.; Alipbeki, O.A. Comparative Analysis of Machine Learning Models for Anomaly Detection in Industrial Smart Meter Time-Series Data. *Information* **2026**, *17*, 131. [CrossRef]
28. Nakutis, Ž.; Rinaldi, S.; Kuzas, P.; Lukočius, R. A Method for Noninvasive Remote Monitoring of Energy Meter Error Using Power Consumption Profile. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 6677–6685. [CrossRef]
29. Liu, F.; Liang, C.; He, Q. Remote Malfunctional Smart Meter Detection in Edge Computing Environment. *IEEE Access* **2020**, *8*, 67436–67443. [CrossRef]

30. Utomo, D.; Hsiung, P.-A. A Multitiered Solution for Anomaly Detection in Edge Computing for Smart Meters. *Sensors* **2020**, *20*, 5159. [CrossRef] [PubMed]
31. Reis, M.J.C.S. Lightweight Signal Processing and Edge AI for Real-Time Anomaly Detection in IoT Sensor Networks. *Sensors* **2025**, *25*, 6629. [CrossRef] [PubMed]
32. Xu, C.; Chao, Z.; Zhang, H.-M.; Yu, Y.; Cheng, Z.-Q.; Xu, Y.-Z. Design of an online monitoring and traceability optimization system for measurement errors of instrument transformers and energy meters based on edge computing architecture. *EURASIP J. Wirel. Commun. Netw.* **2025**, *2025*, 102. [CrossRef]
33. Saunoris, M.; Šaltanis, J.; Lukočius, R.; Daunoras, V.; Zulonas, K.; Vaičiukynas, E.; Nakutis, Ž. A Method for Assessment of Power Consumption Change in Distribution Grid Branch After Consumer Load Change. *Appl. Sci.* **2025**, *15*, 8299. [CrossRef]
34. *DSMR 5.0.2 P1*; Companion Standard: Dutch Smart Meter Requirements, Version 5.0.2. Netbeheer Nederland: The Hague, The Netherlands, 2016. Available online: https://www.netbeheernederland.nl/publicatie/dsmr-502-p1-companion-standard (accessed on 1 September 2025).
35. Tightiz, L.; Yang, H. A Comprehensive Review on IoT Protocols' Features in Smart Grid Communication. *Energies* **2020**, *13*, 2762. [CrossRef]
36. Ngcobo, T.J.; Ghayoor, F. An Overview of DLMS/COSEM and G3-PLC for Smart Metering Applications. *Int. J. Smart Sens. Intell. Syst.* **2022**, *15*, 1–15. [CrossRef]
37. Matz, A.P.; Fernandez-Prieto, J.-A.; Cañada-Bago, J.; Birkel, U.A. Systematic Analysis of Narrowband IoT Quality of Service. *Sensors* **2020**, *20*, 1636. [CrossRef] [PubMed]
38. Ferrari, P.; Flammini, A.; Sisinni, E.; Rinaldi, S.; Brandão, D.; Rocha, M.S. Delay Estimation of Industrial IoT Applications Based on Messaging Protocols. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 2188–2199. [CrossRef]
39. Arcas, G.I.; Cioara, T.; Anghel, I.; Lazea, D.; Hangan, A. Edge Offloading in Smart Grid. *Smart Cities* **2024**, *7*, 680–711. [CrossRef]