Learning-Aided Source Identification and Power Quality Assessment for Edge-Integrated IoT

Akash Kumar Mandal and Swades De
Department of Electrical Engineering and Bharti School of Telecommunication
Indian Institute of Technology Delhi, New Delhi, India

***Abstract*—The rising heterogeneity and decentralization of industrial power sources demand intelligent, real-time monitoring at the utility edge to ensure operational continuity and system resilience. In this work, we propose a novel edge-integrated Internet-of-Things system that combines a learning-aided field sensor unit, termed source quality and interchange diagnosis (SQUID) unit, with a lightweight edge analytics node, power analysis and data aggregation (PANDA) processor. The SQUID unit performs adaptive voltage and current sensing, local anomaly detection, and wireless data transmission using a custom multi-threaded TCP/IP stack. The PANDA processor analyzes the incoming data using a sparse neural feature aggregator for power source classification and quality assessment under noise and compute constraints. Extensive lab tests show over 99% accuracy with limited field testing on three communication base station sites corroborating the lab test outcomes. Further, the performance remains robust under noise levels up to 10 dB with 38% improvement in classification precision over existing methods. These results confirm the system's robustness, scalability, and practical applicability in diverse industrial monitoring scenarios.**

***Index Terms*—Edge computing, learning-aided monitoring, power analysis and data aggregation (PANDA), real-time Internet-of-Things (IoT), source quality and interchange diagnosis (SQUID)**

I. Introduction

Rapid advancement of technology in industry 4.0 has witnessed a dense deployment of Internet-of-Things (IoT) nodes and mobile devices, which require many data centers and radio frequency (RF) base stations for their seamless operation. These utilities rely on multiple input power sources to ensure uninterrupted operation [1]. However, the load utility outages, such as 2017 New York city subway blackout, 2017 Atlanta airport blackout, 2011 AWS disruption, etc., highlight a critical need for reliable monitoring of their input power sources [2].

White papers from Rockwell Automation, YOKOGAWA, Affinity Energy, and Keysight, highlight that the quality of input power sources translates to user experience, operational efficiency of the utility, component life, and energy billing transparency [3]–[5]. Since the input energy supply is managed by different energy service providers (ESPs), gathering information on the quality of power sources is difficult for the utilities and impractical from the competitive business operation standpoint. *Thus, a real-time monitoring of the input power quality at the load utilities is crucial for their reliable operation, transparent billing, as well as powering source optimization towards green energy network design.*

A. Literature Review and Motivation

The existing IoT-based solutions for load utility monitoring primarily rely on ensuring situational awareness by recording parameters like temperature, pressure, load, and power across various sectors [6], [7]. These devices process data on-board, at the edge, or in the cloud for control and billing purposes [8], [9]. However, limited research addresses power source identification, input power quality monitoring, and switchover detection. Learning-based approaches can fill this gap by enabling predictive maintenance through source quality estimation and switchover identification [10].

Advanced devices such as power quality analyzers, intelligent electronic devices, and advanced measurement infrastructure can detect power disturbances and source changes [11], [12], but cannot trace the root cause of poor power quality. Utility-side monitoring tools like phasor measurement units and energy management systems assess overall source quality [13], [14], yet remain insensitive to load-specific needs and lack source identification capability.

Although IoT-based monitoring has advanced, learning-aided techniques for real-time source prediction and quality monitoring are still underexplored [15]. Traditional systems struggle with real-time analysis of large datasets, limiting their ability to detect disturbances and switchovers [16]. Techniques like neural networks and SVMs have shown effectiveness in event classification and fault forecasting [17], [18], while also enabling proactive maintenance [19]. A recent work [20] introduced an IoT solution for source and switchover detection but lacked on-board processing, an edge-based quality monitoring, and real-time communication. Moreover, many edge-based methods require significant compute resources (cf. Fig. 7(c)).

As an advance, this paper proposes a novel learning-aided source quality and interchange diagnosis (SQUID) unit for real-time monitoring of input power quality and detecting source-switchovers. SQUID unit performs on-board processing to mark switchover instants and transmits voltage, current, and event data to a remote power analysis and data aggregation (PANDA) edge server for logging and analysis. PANDA employs a novel sparse neural feature aggregator (SNFA)-classifier to identify source labels and assess power quality.

B. Contributions and Significance

The key contributions in this paper are as follows:

1) A compact, learning-aided IoT sensor node is developed for real-time power quality monitoring and

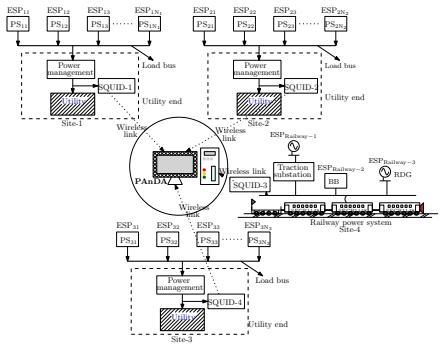


Fig. 1: System model for design and testing of SQUID and PANDA edge; PS: power source, RDG: railway-side distributed generation.

source switchover detection. SQUID incorporates onboard power supply, adaptive temperature-aware calibration, local analytics, and wireless connectivity.

2) The SQUID unit supports utility-defined transmission schedules and is robust to varying types and numbers of input sources - AC, DC, and switched supplies.
3) The PANDA processor aggregates data from distributed SQUID units and performs efficient source classification and quality assessment using a SNFA designed for high accuracy under low compute and noisy conditions.
4) A custom TCP/IP-based communication protocol is utilized for structured, low-latency, bidirectional data transfer between SQUID nodes and the PANDA edge server, supporting real-time diagnostics over wireless networks.

II. System Model

This section presents the system model comprising the SQUID unit and the PANDA edge node. The setup includes multiple monitoring sites (cf. Fig. 1), Site-1 to Site-4, each powered by multiple ESPs. In general, only one source supplies the utility at a time, with others taking over upon failure. The SQUID unit operates as a utility-end, retrofittable IoT device that captures real-time voltage and current measurements.

Captured data is timestamped in UNIX format and transmitted to the PANDA edge node at a utility-defined rate for centralized storage and analysis. A master file within the SQUID maintains updated measurements. To enhance responsiveness and reduce latency, the SQUID firmware runs three parallel threads: (i) data acquisition and logging, (ii) client-server communication, and (iii) board-level energy management.

The SQUID unit supports 2.4/5 GHz WiFi (IEEE 802.11b/g/n/ac) and communicates via a TCP/IP protocol. The PANDA edge server employs multi-threaded processing to handle concurrent SQUID connections and runs a lightweight, learning-aided source classification algorithm, SNFA, to identify the active power source from noisy signals and assess power quality metrics in real time. The following section details the SQUID design and its integration with PANDA.

III. Design of SQUID Unit

The SQUID unit has three modules: sensing, data processing, and wireless communication, which are detailed next.

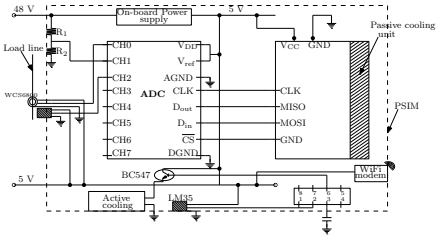


Fig. 2: Circuit diagram of the sensing unit.

A. Temperature-Sensitive Adaptive Sensing Module

The current is recorded using a Hall effect-based linear current sensor and the voltage measurements recorded using a tap-based potential divider assembly. Since the typical switching involved in DC utilities are in the range 50 kHz to 500 kHz, the selection of an appropriate current sensor involves an upper bound based on the switching frequencies, i.e., a least response time in the range 2-20 μs, or lower. Further, for observing the transients occurring in the utility, we consider switching, loading, and impulsive transients, the fastest of which occurs within $\tau = 100$ μs. Thus, with a safety margin ($\mathrm{s_m}$) of 5%, the operating bandwidth (BW) is given as: $\mathrm{BW} \approx \frac{1}{\mathrm{s_m}} \frac{0.35}{\tau} \approx 17.5$ kHz. To meet the requirement of sensing currents in the 0-10 A range and the previously calculated bandwidth, we employ the WCS6800 current sensor. The Hall effect sensor has three pins: Vcc, ground, and a central output pin (Vs) providing voltage proportional to the current. Under zero current, it outputs a bias voltage $\mathrm{V_{bias,0}}$, which varies with temperature and must be calibrated. Under standard conditions: $\mathrm{I_{load}} = \frac{\mathrm{V_s} - \mathrm{V_{bias,0}}}{\alpha}$, where α is the sensitivity of the sensor in mV/A. For recalibration as per the sensor temperature, we employ a low power linear temperature sensor LM35, with an operating range -55°C to 155°C. For a temperature-proportional output voltage $\mathrm{V_{temp}}$, a first order curve-fitting is used to learn the corresponding bias voltage: $\mathrm{V_{bias}} = \omega_0 \mathrm{V_{temp}} + \omega_1$, where ω_0 and ω_1 are the fitting constants. These values are determined as follows:

$$\mathrm{V_{temp}} \to 0, \quad \omega_1 = \mathrm{V_{bias,0}}$$

$$(\mathrm{V_{temp}}, \mathrm{V_{bias}}) \equiv (\mathrm{V_{temp,a}}, \mathrm{V_{bias,a}}), \quad \omega_0 = \frac{\mathrm{V_{bias,a}} - \mathrm{V_{bias,0}}}{\mathrm{V_{temp,a}}}$$

where $\mathrm{V_{temp,a}}$ and $\mathrm{V_{bias,a}}$ represent arbitrary value-pair for $\mathrm{V_{temp}}$ and $\mathrm{V_{bias}}$, respectively. Therefore, the auto-calibrated sensor output of WCS6800 is obtained as: $\mathrm{I_{load}} = \frac{\mathrm{V_s}}{\alpha} - \frac{1}{\alpha}(\omega_0 \mathrm{V_{temp}} - \omega_1)$. For voltage measurement, we use a resistor bridge for transforming the input voltage to a voltage-level suitable for the analog to digital converter (ADC). As shown in Fig. 2, for an input voltage $\mathrm{V_{in}}$, a desired output voltage $\mathrm{V_{desired}}$, the resistor values $\mathrm{R_1}$ and $\mathrm{R_2}$ are obtained as: $\mathrm{V_{desired}} = \mathrm{V_{in}} \mathrm{R_2} (\mathrm{R_1} + \mathrm{R_2})^{-1}$. The measured voltage and current values are fed to an ADC for quantization. The clock output and the quantized values of the voltage and current are communicated from the ADC to the microprocessor using a serial peripheral interface (SPI). The total power consumed by the sensing unit is given as follows:

$$\mathbf{x}(\tau) = \left[x_{1,1,f_s}\,\mathrm{rect}(0,\tau_1), x_{1,2,f_s}\,\mathrm{rect}\left(\tau_1+f_s^{-1},\tau_1+\tau_s\right),\cdots, x_{k-1,k,f_s}\,\mathrm{rect}\left(\tau_{k-1}+f_s^{-1},\tau_{k-1}+\tau_s\right), x_{k,k,f_s}\,\mathrm{rect}(\tau_{k-1}+\tau_s,\tau)\right]^T \quad (1)$$

$\mathrm{P_{sense}} = \mathrm{P_{cs}} + \mathrm{P_{vs}} + \mathrm{P_{ADC}} + \mathrm{P}_{\mu\mathrm{P}}$, where $\mathrm{P_{cs}}$, $\mathrm{P_{vs}}$, $\mathrm{P_{ADC}}$, and $\mathrm{P}_{\mu\mathrm{P}}$ are the power consumed by the current sensor, voltage sensor, ADC, and the microprocessor.

After capturing utility data, the following tasks are executed: (1) pre-operative calibration of the SQUID unit, (2) detection of source switchover instants at SQUID, and (3) identification of the active power source and its quality at PANDA processor.

B. Data Analysis at SQUID Unit

Let there be N input power sources. We need to calibrate the SQUID-PANDA system for appropriate source identification and quality analysis. Here, we use the source operation control in the power management block of the objective utilities to strategically record labeled data corresponding to each of the input power sources for a preset duration τ_d, with the data from ith source represented as $\mathbf{x}_i(\tau_d)$. The collective labeled dataset is transmitted to the edge using the communication protocol in Section III-C for training and analysis.

For a generic analysis, we assume an extreme scenario, where in a given recording interval k out of these N sources are recorded. Therefore, the recorded dataset $\mathbf{x}(\tau) \in \mathbb{R}^{\tau f_s \times 1}$ is represented as given in (1), where x_{j,j,f_s} denotes the voltage or current waveform of the jth input source sampled at frequency f_s, while $x_{j,j+1,f_s}$ represents the transient waveform during switching from source j to $j+1$, which remains zero during the switchover period. Hence, the switching frequency is selected to be higher than the inter-source switching frequency. The rectangular function $\mathrm{rect}(\tau_a,\tau_b) = u(t-\tau_a) - u(t-\tau_b)$ defines a window over $[\tau_a,\tau_b]$, where $u(\cdot)$ is the unit step function. From (1), a total of τf_s samples are recorded, with $(\tau_j - \tau_{j-1})f_s$ samples corresponding to the jth source. Differentiating (1), we get:

$$\begin{aligned}\dot{\mathbf{x}}(\tau) =& f_s\left[\Delta(x_{1,1,f_s},1)\left(\delta(t+\tau_1)-\delta(t)\right),\cdots,\right.\\ &\left.\Delta(x_{k,k,f_s},1)\left(\delta(t+\tau)-\delta(t+\tau_{k-1})\right)\right]^T \in \mathbb{R}^{\tau f_s\times 1}\end{aligned} \quad (2)$$

where T denotes the transpose operation and $\Delta(x_{1,1,f_s},1)$ is the difference between consecutive values of the waveform $x_{1,1,f_s}$ and $\delta(\cdot)$ denotes the Dirac delta function.

Proposition 1. *Derivative at the instants of switchover are characterized by impulse functions, indicating switchovers.*

Proof. The runtime derivative of source j is given by:

$$\dot{x}_{j,j} = f_s x_{j,j,f_s}\left(\delta(t-l) - \delta\left(t-l-f_s^{-1}\right)\right), \quad l \in (\tau_{j-1},\tau_j). \quad (3)$$

Due to the high sampling frequency f_s, the two delta terms in (3) nearly cancel, yielding $\dot{x}_{j,j} \to 0$. At the switchover point between sources j and $j+1$, the derivative becomes:

$$\dot{x}_{j,j+1} = f_s\left(x_{j,j,f_s}\delta(t-\tau_j) - x_{j,j+1,f_s}\delta\left(t-\tau_j-f_s^{-1}\right)\right). \quad (4)$$

For f_s exceeding the switching rate, $x_{j,j+1,f_s} = 0$, reducing (4) to: $\dot{x}_{j,j+1} = f_s x_{j,j,f_s}\delta(t-\tau_j)$, which is sharply peaked and of much larger magnitude than $\dot{x}_{j,j}$, due to the impulse nature over the brief interval f_s^{-1}. Thus, $\dot{x}_{j,j+1} \gg \dot{x}_{j,j}$. □

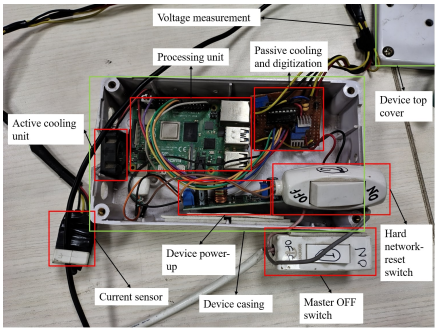


Fig. 3: Final SQUID unit design.

The data sent to the edge includes: timestamp, voltage, current, and a 'source switchover' field, where switchovers are marked as 1 and runtime as 0. Next, we describe processing and calibration at the PANDA processor.

C. Design of Wireless Data Communication to Edge

As inter-source switching instants and source operation durations are time-critical, the SQUID controller operates three concurrent threads: data collection, data communication, and energy management. The data collection thread samples DC voltage and current at 1000 sps, storing timestamped measurements in separate files for fixed durations (default: 2 minutes, adjustable). The communication thread transmits compressed data files to the remote server using the TCP/IP protocol. The energy management thread optimizes onboard power and avoids network rate limiting by intermittently deactivating the WiFi module for a fraction λ during non-transmission windows. The power consumed is expressed as: $\mathrm{P_{com}} = \mathrm{P_{active}} + (1-\lambda)\mathrm{P_{idle}} + \lambda\mathrm{P_{off}}$, where $\mathrm{P_{active}}$, $\mathrm{P_{idle}}$, and $\mathrm{P_{off}}$ represent the power consumed by the communication unit in active, idle, and off state, respectively.

A server script is deployed at the edge node, where a socket is created and bound to an IP address and port. To receive data from SQUID units across different networks, port forwarding is configured via network address translation on the edge router. The PANDA processor runs a main thread that listens for incoming connections. Upon accepting a connection, it spawns a sub-thread to manage client communication while the main thread continues listening. Each sub-thread runs a writer script that processes incoming data and writes it to new files until the connection ends. The total power consumption is given as: $\mathrm{P_{cons}} = \mathrm{P_{supply}} + \mathrm{P_{sense}} + \mathrm{P_{com}}$, where $\mathrm{P_{supply}}$, $\mathrm{P_{sense}}$, and $\mathrm{P_{com}}$ are the power consumed by the supply, sensing, and communication modules, respectively. The final design of the SQUID client is demonstrated in Fig. 3. The advanced processing at the PANDA edge node is detailed next.

$$\mathbf{F}_i^T = [\text{mean}(\mathbf{x}_{i,V}), \text{std}(\mathbf{x}_{i,V}), \max(\mathbf{x}_{i,V}), \min(\mathbf{x}_{i,V}), \text{mean}(|\,\text{FFT}(\mathbf{x}_{i,V})|), \max(|\,\text{FFT}(\mathbf{x}_{i,V})|), \text{PSD}(|\,\text{FFT}(\mathbf{x}_{i,V})|), \text{skew}(\mathbf{x}_{i,V}), \text{kurt}(\mathbf{x}_{i,V})), \text{mean}(\mathbf{x}_{i,C}), \text{std}(\mathbf{x}_{i,C}), \max(\mathbf{x}_{i,C}), \min(\mathbf{x}_{i,C}), \text{mean}(|\,\text{FFT}(\mathbf{x}_{i,C})|), \max(|\,\text{FFT}(\mathbf{x}_{i,C})|), \text{PSD}(|\,\text{FFT}(\mathbf{x}_{i,C})|), \text{skew}(\mathbf{x}_{i,C}), \text{corr}(\mathbf{x}_{i,V}, \mathbf{x}_{i,C})] \in \mathbb{R}^{1\times 18}. \quad (5)$$

IV. Configuration of the PANDA Processor

A. Remote Classifier Calibration on PANDA Processor

The PANDA processor employs SNFA, which consists of three main components: *feature sparsification layer* (FSL) for reducing feature dimensionality, *modular sub-networks* (MSN) for processing feature groups independently, and the *aggregation layer* (AL) for combining the outputs of modular networks into a unified representation for final classification. The feature matrix $\mathbf{F} \in \mathbb{R}^{N\times d}$, where N is the number of samples and d is the feature dimension, is split into M groups, $\{\mathbf{F}_1, \cdots, \mathbf{F}_M\}$ for modular processing, such that $\mathbf{F}_i \in \mathbb{R}^{N\times d_i}$ and $\sum_{i=1}^{M} d_i = d$. The classifier layers are executed as follows:

$$\begin{aligned}
\textbf{FSL}: &\quad \mathbf{F}_i^{(s)} = \text{ReLU}(\mathbf{F}_i - \tau) \\
\textbf{MSN}: &\quad \mathbf{h}_m = \sigma(\mathbf{W}_m \mathbf{F}_m^{(s)} + \mathbf{b}_m) \\
\textbf{AL}: &\quad \mathbf{h}_{\text{agg}} = \sum_{m=1}^{M} \omega_m \mathbf{h}_m \\
\textbf{OL}: &\quad \hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_{\text{out}} \mathbf{h}_{\text{agg}} + \mathbf{b}_{\text{out}})
\end{aligned} \quad (6)$$

where $\mathbf{W}_m$, $\mathbf{W}_{\text{out}}$, and ω_m are the respective weights for the MSN, OL, and AL layers, and $\mathbf{b}$s represent the appropriate bias vectors. The feature vector for the ith group is denoted as $\mathbf{F}_i$, as given in (5), where $\text{mean}(\cdot)$, $\text{std}(\cdot)$, $\text{FFT}(\cdot)$, $\text{PSD}(\cdot)$, $\text{kurt}(\cdot)$, $\text{skew}(\cdot)$, and $\text{corr}(\cdot)$ denote the mean, standard deviation, fast Fourier transform, power spectral density, Kurtosis, skewness, and correlation operators, respectively. The aggregated embedding $\mathbf{h}_{\text{agg}}$ is passed through a linear classification layer to predict class probabilities in the OL for C classes. By introducing a sparsity threshold τ, the model prunes low-impact features reducing computational load. The SNFA model is trained using the below cross-entropy function:

$$(\mathbf{P}1): \quad \min_{\mathbf{W},\omega,\mathbf{b}} -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{1}[y_i = c] \log(\hat{y}_{i,c}) \quad (7)$$

where N is the number of training samples, y_i is the true label for the ith sample, and $\hat{y}_{i,c}$ is the predicted probability for class c. $\mathbb{1}[y_i = c]$ is an indicator function that evaluates to one when the true class label (label of the input power source) y_i of sample i is equal to c; otherwise zero. We utilize a standard stochastic gradient descent approach to optimize $(\mathbf{P}1)$.

B. Source Classification at PANDA Edge Server

The dataset received from the SQUID unit is divided between the instants of switchover. Each of these datasets undergo individual feature extraction to form feature vectors $\mathbf{F}_a$, where $a = a_1, \cdots, a_{M+1}$ with M denoting the total number of switchovers recorded in the consolidated dataset. Each of these feature vectors are compared to each vector from the calibration set using the following distance: $d_{i,j} = \|\mathbf{F}_i - \mathbf{F}_a\|$, $i = 1, \cdots, N, a = a_1, \cdots, a_{M+1}$. Then, the similarity between these vectors is computed as: $s_{i,j} = \frac{1}{1+d_{i,j}}$. $s_{i,j}$ is checked against a threshold s_{th} to compute the candidate sources for the final classification. Let candidate solutions be denoted as $\tilde{\mathbf{F}}_b$, where $b = b_1, \cdots, b_s$, with s being the total number of above-similarity feature vectors. The classified label for the recorded feature $\tilde{\mathbf{F}}_b$ is given as: $\hat{y}_b = \hat{f}(\tilde{\mathbf{F}}_b)$, where $\hat{f}(\cdot)$ is an SVM classifier. In case, two or more candidate vectors match the same source label, we use a contention resolution metric $c_b = \frac{A_b}{d_{\hat{y}_b,b}}$, which leverages the findings from the similarity matching and classification with $A_b = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$ is the accuracy maintained by the classifier $\hat{f}(\cdot)$ in classifying $\tilde{\mathbf{F}}_b$ while $d_{\hat{y}_b,b}$ denotes its distance from the classified label vector $\mathbf{F}_{\hat{y}_b}$. TP, TN, FP, and FN, represents true positives, true negatives, false positives, and false negatives, respectively.

Remark 1. *The first level isolation helps to select the candidate source signals while $\hat{f}(\cdot)$ matches the spatio-temporal trends of the recorded dataset with the calibration set.*

Next, we compute the key source quality defining metrics.

C. Source Quality Assessment at PANDA

To assess power quality at utility, we use the following:

1) Voltage Deviation (ΔV): Measures deviation from nominal voltage, $\Delta V = \frac{V_{\max} - V_{\min}}{V_{\text{nom}}}$, where $V_{\max}$, $V_{\min}$, and V_{nom} are the max, min, and nominal voltage values.

2) Voltage Regulation (ΔR): Represents per-unit voltage drop from no-load to full-load: $\Delta R = \frac{V_{\text{NL}} - V_{\text{FL}}}{V_{\text{FL}}}$, where V_{NL} and V_{FL} represent the no-load and full-load voltage values.

3) Voltage Sag/Swell (VS): Captures voltage dips or rises:

$$\text{VS} = \begin{cases} \frac{V_{\text{nom}} - V_{\text{sag}}}{V_{\text{nom}}}, & \text{during sag} \\ \frac{V_{\text{swell}} - V_{\text{nom}}}{V_{\text{nom}}}, & \text{during swell} \end{cases} \quad (8)$$

4) Frequency Deviation (Δf): $\Delta f = |f_{\text{act}} - f_{\text{nom}}|$, where f_{act} is estimated using a DFT-based method.

5) Total Harmonic Distortion (THD): Represents the harmonic distortion in voltage: $\text{THD} = \frac{\sqrt{\sum_{n=2}^{\infty} V_n^2}}{V_1}$, with V_n as the RMS of the nth harmonic, and V_1 for the fundamental.

6) Mean Time Between Failures (MTBF): Indicates the average time between two faults: $\text{MTBF} = \frac{1}{F} \sum_{j=1}^{F-1} (t_{j+1} - t_j)$, where t_j is the time of the jth fault and F is the total faults.

7) Recovery Time (T_{Rec}): Time taken to restore normal operation: $T_{\text{Rec}} = \frac{1}{F} \sum_{i=1}^{F} (T_{\text{ON},i} - T_{\text{OFF},i})$, with $T_{\text{OFF},i}$ and $T_{\text{ON},i}$ marking disturbance and recovery times.

8) Noise Level (P_{noise}): Evaluates the average power of high-frequency noise: $P_{\text{noise}} = \frac{1}{T} \int_0^T V_{\text{noise}}(t) dt$, where $V_{\text{noise}}(t)$ is extracted from $V(t)$ using a high-pass filter.

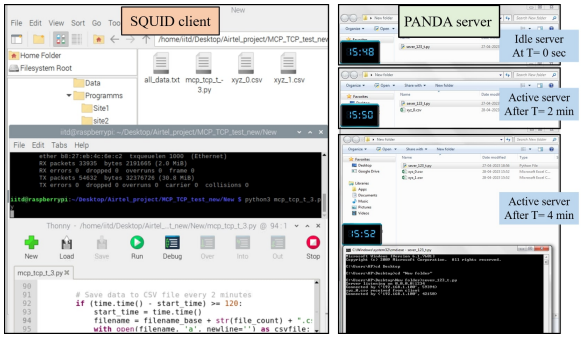
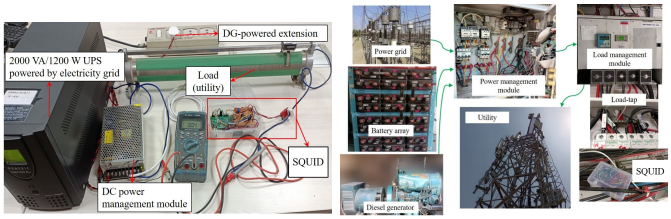


Fig. 4: PANDA edge server interface.



(a) Lab test setup (b) Field test setup

Fig. 5: SQUID unit test setups

9) Ripple Voltage (V_R): Assesses residual oscillations over DC: $V_R = \max(V(t) - V_{\rm DC}(t)) - \min(V(t) - V_{\rm DC}(t))$, where $V_{\rm DC}(t)$ is the average DC component of the voltage signal.

$\mathbf{M} = [\Delta V, \Delta R, \text{VS}, \Delta f, \text{THD}, \text{MTBF}, T_{\rm Rec}, P_{\rm noise}, V_R]$ with the threshold values in $\mathbf{M}_{th}$. Quality of experience (QOE) is calculated as: $\text{QOE} = \|\Phi(\mathbf{M} \succ \mathbf{M}_{th})\|$, where $\|\cdot\|$ denotes norm and $\Phi(\cdot)$ is the element-wise binary comparator.

PANDA server interfaced with SQUID is shown in Fig. 4. Performance of SQUID-PANDA system is presented next.

V. Results and Discussion

Specs for PANDA processor: Intel coreTM i9 14900K (36 MB cache, 24 cores, 32 threads, up to 6.0 GHz turbo, 125 W); 4×16 GB, DDR5, 4000 MT/s; C3: M.2 SSD Boot.

A. Key Performance Indicators (KPIs)

The KPIs used in this analysis are defined as follows:

1) Source identification accuracy: This is defined as the ratio of the correct predictions to the total predictions.

2) Source identification precision: TP/(TP+FP)

3) Log runtime complexity (LRC): This metric measures the ability of an algorithm to identify the operational source within a given recording interval τ. Mathematically, $\text{LRC} = 10\log_{10}\frac{\tau_r}{\tau}$, where τ_r is the time required in the identification.

B. Lab and Field Testing of SQUID and PANDA Processor

The lab setup for testing the SQUID-PANDA system is shown in Fig. 5(a), where the SQUID unit is mounted at the utility's load end (modeled as a resistive load). It measures nominal DC levels of -54 V and 1 A, with a rated capacity of -100 V and 100 A (AC/DC). To emulate field conditions, power grid, battery bank (BB) (2000 VA, 1200 W), and diesel generator (DG) are used as input sources. These are interfaced via a DC power management module that regulates their voltages to match the load requirements. Fig.5(b) shows the field test setup used to validate SQUID functionality and PANDA-based data processing. The power management module handles source switchover (grid, BB, DG), delivering power to the load management module, which allocates resources based on modeled load demand (e.g., call traffic). SQUID operates independently at the load side (without ESP), monitoring utility usage. Dual-feeder support ensures fault tolerance and maintainability. Tests were performed at 3 sites under diverse environmental conditions, with feeder-wise loading in Table I.

TABLE I: Loading conditions for three power sourced load feeders

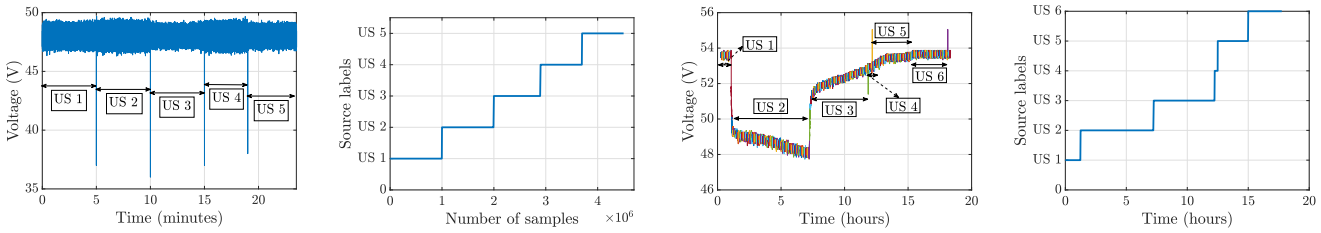
Site	Source type	Load statistics	
		Feeder 1	Feeder 2
Site 1	Power grid	54.2 V, 2.28 kW, 39.92 A	54.2 V, 1.445 kW, 28 A
	Battery	49.2 V, 2.15 kW, 43.6 A	49.6 V, 1.49 kW, 29.66 A
	DG	51.2 V, 2.14 kW, 42.8 A	51.4 V, 1.5 kW, 29.1 A
Site 2	Power grid	51.8 V, 4.3 kW, 81.1 A	Not available
	Battery	48.3 V, 4.4 kW, 92.6 A	Not available
	DG	50.5 V, 4.5 kW, 92.5 A	Not available
Site 3	Power grid	54.2 V, 1.445 kW, 28 A	54.2 V, 2.28 kW, 39.92 A
	Battery	49.6 V, 1.49 kW, 29.66 A	49.2 V, 2.15 kW, 43.6 A
	DG	51.4 V, 1.5 kW, 29.1 A	51.2 V, 2.14 kW, 42.8 A

C. Performance Analysis of SQUID Unit

Fig. 6 shows the performance of the SQUID unit in the lab and field test setups shown in Figs. 5(a) and (b), respectively. Fig. 6(a) and Fig. 6(c) show the voltage profiles recorded by the SQUID unit in lab and field setups. From Fig. 6(b) and Fig. 6(d) we note that, the SQUID unit correctly identifies the instants of inter-source switchovers in the lab as well as field test setups, which can also be noted through the sudden voltage spikes. However, at this instant, the source labels are unknown and hence the ith detected source switchover leads to an unknown source (US) $i+1$, as shown in Fig. 6(a) and Fig. 6(c). Further, from Fig. 6(a) and Fig. 6(c), we observe that, even in DC utilities, owing to controlled voltage generation, the field data shows higher variability. As a consequence, it is naturally difficult to identify the source labels in lab testing setup. *In the subsequent PANDA-end analysis, we use the lab dataset for studying the robustness of the proposed framework.*

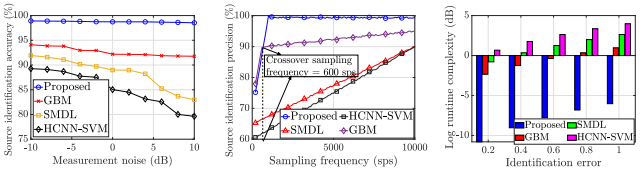
D. Data Analysis Performance of PANDA Edge Processor

For comparison, we assess gradient boosting machine (GBM), stacked machine and deep learning (SMDL), and hierarchical CNN with SVM (HCNN-SVM). In Fig. 7(a), where white noise is added, the proposed method maintains 99% identification accuracy and stays robust up to 10 dB noise. In contrast, GBM, SMDL, and HCNN-SVM show accuracy drops with noise, with range variations of 2%, 9%, and 10%, respectively, and about 7% lower peak accuracy. Fig.7(b) shows that sampling rates ≥ 600 sps yield near-100% source identification precision. At 800 sps, the proposed approach surpasses GBM, SMDL, and HCNN-SVM by 10%, 33%, and 38%, respectively. In Fig.7(c), the proposed PANDA-based analysis yields consistently negative LRC values, indicating accurate identification within the recording interval. For an



(a) Voltage vs. time (lab) (b) Switchover vs. # samples (lab) (c) Voltage vs. time (field) (d) Switchover vs. # samples (field)

Fig. 6: Voltage profile and switchover analysis at SQUID unit using data from lab and field test setup; US i: ith unknown source.



(a) (b) (c)

Fig. 7: (a) Accuracy vs. measurement noise, (b) precision vs. sampling frequency, and (c) runtime complexity at 800 sps.

identification error of 0.1, it reduces LRC by 81.82% and 95.45% compared to GBM and SMDL. HCNN-SVM's positive LRC implies failure to identify sources in time.

VI. Concluding Remarks

This paper presented a learning-enabled, edge-integrated IoT framework for real-time source identification and power quality assessment in utility environments with multiple energy sources. The proposed SQUID-PANDA system integrates an intelligent, field-deployable SQUID node, capable of autonomous voltage/current sensing, adaptive calibration, and local switchover detection, with a communication-efficient PANDA edge processor that performs low-compute machine learning-based classification and diagnostics using a novel SNFA algorithm. Leveraging wireless TCP/IP communication and a modular multi-threaded design, the system supports scalable deployment across distributed utility sites. Results show that the system achieves high source identification accuracy, even under noisy conditions, and outperforms state-of-the-art learning techniques. Its robust performance across diverse scenarios and real-time diagnostics with minimal computational overhead enable predictive maintenance and ensure operational continuity. *Extensive field tests are underway to further validate system performance under broader variability.*

Acknowledgment

This work was supported in part by the Tehri Hydroelectric Development Corporation under Grant THDC/RKSH/R&D/F-2076/1036; and in part by the Science and Engineering Board under Grant CRG/2023/005421.

References

[1] A. Muhtadi *et al.*, "Distributed energy resources based microgrid: Review of architecture, control, and reliability," *IEEE Trans. Indus. Appl.*, vol. 57, no. 3, pp. 2223–2235, 2021.

[2] A. Kwasinski *et al.*, "Telecommunications power plant damage assessment for hurricane katrina–site survey and follow-up results," *IEEE Syst. J.*, vol. 3, no. 3, pp. 277–287, 2009.

[3] K. Olikara, "Power quality issues, impacts, and mitigation for industrial customers," *Test & Measuerement*, pp. 1–13, 2020. [Online]. Available: https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/power-wp002_-en-p.pdf

[4] A. Evora, "5 reasons to consider a power monitoring system," pp. 1–1, 2018. [Online]. Available: https://www.automationworld.com/home/blog/13318206/5-reasons-to-consider-a-power-monitoring-system

[5] "Electronic load fundamentals," pp. 1–19, 2018. [Online]. Available: https://www.keysight.com/in/en/assets/7018-06481/white-papers/5992-3625.pdf

[6] A. Pasdar *et al.*, "Cybersecurity solutions and techniques for Internet of Things integration in combat systems," *IEEE Trans. Sust. Comput.*, 2024.

[7] H. Xu *et al.*, "Smart mobility in the cloud: Enabling real-time situational awareness and cyber-physical control through a digital twin for traffic," *IEEE Trans. Intel. Transport. Syst.*, vol. 24, no. 3, pp. 3145–3156, 2023.

[8] G. S. Wagh *et al.*, "A distributed privacy-preserving framework with integrity verification capabilities for metering data and dynamic billing in a malicious setting," *IEEE Trans. Smart Grid*, 2024.

[9] A. Galasso *et al.*, "A real smart digital current simulator prototype for BESS power management," *IEEE Trans. Indus. Appl.*, 2024.

[10] A. Bakdi, N. B. Kristensen, and M. Stakkeland, "Multiple instance learning with random forest for event logs analysis and predictive maintenance in ship electric propulsion system," *IEEE Trans. Indus. Informat.*, vol. 18, no. 11, pp. 7718–7728, 2022.

[11] M. T. Ahammed and I. Khan, "Ensuring power quality and demand-side management through IoT-based smart meters in a developing country," *Energy*, vol. 250, p. 123747, 2022.

[12] J. Garrido-Zafra *et al.*, "IoT cloud-based power quality extended functionality for grid-interactive appliance controllers," *IEEE Trans.n Indus. Appl.*, vol. 58, no. 3, pp. 3909–3921, 2022.

[13] P. Shah and X. Zhao, "Network identification using μ-PMU and smart meter measurements," *IEEE Trans. Indus. Informat.*, vol. 18, no. 11, pp. 7572–7586, 2022.

[14] Z. Zhao *et al.*, "Distributed robust model predictive control-based energy management strategy for islanded multi-microgrids considering uncertainty," *IEEE Trans. Smart Grid*, vol. 13, no. 3, pp. 2107–2120, 2022.

[15] X. Chen *et al.*, "Reinforcement learning for selective key applications in power systems: Recent advances and future challenges," *IEEE Trans. Smart Grid*, vol. 13, no. 4, pp. 2935–2958, 2022.

[16] J. Bian *et al.*, "Machine learning in real-time internet of things (IoT) systems: A survey," *IEEE IoT J.*, vol. 9, no. 11, pp. 8364–8386, 2022.

[17] C. Cui *et al.*, "Detection and classification of multiple power quality disturbances using stockwell transform and deep learning," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[18] A. Sepehr *et al.*, "Employing machine learning for enhancing transient stability of power synchronization control during fault conditions in weak grids," *IEEE Trans. Smart Grid*, vol. 13, no. 3, pp. 2121–2131, 2022.

[19] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[20] R. Choudhury *et al.*, "DC signature analysis aided power source identification," in *in Proc. IEEE 17th Int. Conf. Indus. Informat. Syst. (ICIIS)*, 2023, pp. 299–304.