energies

MDPI

Article

OpenEdgePMU: An Open PMU Architecture with Edge Processing for Future Resilient Smart Grids

Nikolaos-Antonios I. Livanos [1,2], Sami Hammal [2], Nikolaos Giamarelos [1,2], Vagelis Alifragkis [2], Constantinos S. Psomopoulos [1] and Elias N. Zois [1,*]

[1] Department of Electrical and Electronic Engineering, University of West Attica, Thivon 250, Aigaleo, 122 41 Athens, Greece
[2] EMTECH SPACE P.C., Korinthou 32 & S. Davaki, Metamorfosi, 144 51 Athens, Greece
* Correspondence: ezois@uniwa.gr; Tel.: +30-21-0538-1559

Abstract: The increase in renewable energy sources (RESs) in distribution grids is a major driver for achieving green energy goals worldwide. However, RES power inverters affect power quality, increase power losses, and, in certain cases, may cause power interruptions due to harmonics, deterioration of the rate of change of frequency, and inability to rapidly react in grid faults. Today, phasor measurement units (PMUs) are the ultimate tools for real-time monitoring of distribution grids' health, and they enable several data-driven added-value services such as fast and automated fault detection, isolation, and recovery; state estimation; power quality monitoring; dynamic events analysis, etc. The present paper proposes an open hardware and software PMU platform, which is low cost, high performance, expandable, and, in general, suitable for research and innovation activities. The system is based on two processor modules (a digital signal processor from Texas Instruments TMS320c5517, and a microprocessor System-in-Package from Octavo Systems OSD3358), two local databases of 64 Gbytes each, GPS module, 5G modem interface, as well as analog and signal conditioning circuits to interface three-phase power voltage and current signals. The entire hardware design, schematics, and instrumentation components, as well as all firmware and software functions are completely open source. Pilot operation of the prototype design has been installed in three medium-/low-voltage substations in Cyprus, as well as twelve substations in Spain and Italy.

Keywords: phasor measurement unit; smart grids; microgrids; rate of change of frequency; frequency measurements; edge processing; power quality processing

Citation: Livanos, N.-A.I.; Hammal, S.; Giamarelos, N.; Alifragkis, V.; Psomopoulos, C.S.; Zois, E.N. OpenEdgePMU: An Open PMU Architecture with Edge Processing for Future Resilient Smart Grids. *Energies* **2023**, *16*, 2756. https://doi.org/10.3390/en16062756

Academic Editor: Abu-Siada Ahmed

Received: 13 February 2023
Revised: 3 March 2023
Accepted: 13 March 2023
Published: 15 March 2023

1. Introduction

Today, synchrophasor technology offers a valuable tool for monitoring and analyzing power systems. Phasor measurement units (PMUs) digitize voltage and current phasors with high frequency and synchronize acquired data with absolute timing via satellite GPS signals. By using advanced signal processing and correlation techniques on the data received from several PMUs, a high-quality analysis of the dynamic behavior of the electric power grid can be performed. Nowadays, this method is very valuable due to increasing penetration of renewable energy sources (RESs) and the increased stochasticity of distribution grid management. Apart from analysis purposes, today, PMUs are used for fast monitoring, protecting, and automating contemporary complicated distribution systems. Typical applications regarding wide area monitoring in distribution grids include: angle/frequency monitoring, post mortem analysis, voltage stability monitoring, improved state estimation, DG/IPP applications, power system restoration, etc. [1]. Further interesting applications of PMUs for distribution networks are: microgrid automation and operations, fault detection and localization, FACTS devices, grid's situational awareness, power quality analysis, etc. [2–7].

1.1. A Brief Review of Recent PMU Designs

During the last fifteen years, several research and development efforts toward implementing PMU experimental systems have been carried out. Following a chronological approach, we conducted a brief review on this subject.

Carta et al. [8], in 2008, developed an experimental arrangement of a flexible GPS-based system for synchronized phasor measurement in electric distribution networks and emphasized hardware and software reconfiguration and reprogramming according to different measurement requirements. The system was based on general-purpose digital acquisition boards that could achieve very high sampling rates, for example, up to 500 ksamples/sec, synchronized with an external GPS receiver. The experimental laboratory PMU arrangement achieved a mean value of phase difference less than 0.3 mrad with a standard deviation of 0.7 mrad.

Laverty et al. [9], between 2011 and 2014, developed the OpenPMU technology platform for synchrophasor research applications. The objective of the OpenPMU initiative was to establish a research community to work towards developing a fully featured PMU compliant with the IEEE standard C37.118 [10,11]. The intention was not to compete with commercial device manufacturers. Rather, they focused on the research community and aimed to provide a useful tool to support research and scientific activities. The OpenPMU technology platform is open source [12–15]; the software is licensed under the BSD license [12,13] and the hardware is licensed under the TAPR Open Hardware License (OHL) [14,15]. The platform is based on a COTS digital acquisition device (NI-DAQ). The system measures six channels (three voltage and three current) at a sampling frequency of 6.4 kHz. The TVE was measured to $\pm$ 0.15% and the frequency had a systematic error of -0.002 Hz. In general, the system meets many of the requirements of P class instruments.

The GridTRAK PMU was produced at Baltimore University by Stadlin [16] and the design has been published under an open-source license. The principal objective of GridTRAK was to produce a low-cost PMU, able to attract wide distribution among researchers and amateurs, that targeted widespread monitoring of distribution grids. The system measures magnitude, frequency, and phase for a single phase.

He and Liao [17], in 2012, developed an analog acquisition system for distribution automation. The hardware of the system is based on a dual-CPU (an ARM-based processor LPC2132 and a DSP TMS320F2812) and a high-precision analog-to-digital converter (AD7606). The combination of the two CPUs make the system powerful, as the ARM-based processor is focused on control and soft real-time tasks, while the DSP is completely focused on hard real-time processing of the acquired signals. The ADC provides 16-bit, simultaneous sampling for eight channels, with a sampling frequency of 200 ksamples/s and an antialiasing filter with a cut-off at 22 kHz. Although the authors did not provide signal processing performance results, it seems that the proposed platform has good performance, low-cost and expandability, able to support real-time monitoring, and control functionalities for distribution grids.

Another PMU based on DSP was proposed by Du et al. [18] in 2012. In this approach, the DSP is used as the main processor of the system. The design includes a 16-bit, high-speed A/D converter, with six conversion channels.

Romano and Paolone [19], in 2014, prototyped a PMU based on an FPGA. They proposed a complete approach that started from theory and continued to and implementation, and they performed a thorough validation of the prototype arrangement. The design demonstrated simulation of PMU-related algorithms (the e-IpDFT-based Synchrophasor Estimation) and implementation on a National Instruments Compact-RIO embedded control and acquisition system, embedding a reconfigurable Virtex-5 LX110 FPGA. The sampling of voltage and current waveforms was realized in two parallel 24-bit sigma-delta converters, with a sampling rate up to 50 ksamples/s. The arrangement included a GPS unit with time uncertainty of $\pm$100 ns. The prototype PMU could satisfy all the class P requirements together with the majority of class M ones.

Pinte et al. [20], in 2014, developed a distribution-level single-phase PMU for post-event analysis. They adopted a hardware module by National Instruments (myRIO-1900, cost USD 250) and a software platform based on a reconfigurable FPGA and flexible input and output module options. The sampling rate was programmed for 10 ksamples/s. The approach also included a compact uninterruptible power supply (UPS) module and, in general, the cost of the design was very compressed (estimated cost USD 350).

Rodrigues et al. [21], in 2016, designed and realized a PMU using a DSP that mainly focused on power distribution grid special requirements such as low cost, small size, multiple functionalities, etc. After comparing certain DSP-based development platforms, they adopted the STM32F4 Discovery [22], supporting the required 15.36 ksample/s sampling frequency and relevant signal processing performance. The achieved maximum verified TVE was 0.57%, and the frequency estimation errors were less than 0.032%.

Das and A. Pradhan [23], in 2016, developed a micro-PMU for distribution system applications. The design was based on COTS development modules and, specifically, an ultra-high-performance microcontroller (TIVA C Series TM4C123G from Texas Instruments), a 12 bit accuracy ADC with sampling frequency of 2.5 kHz, and a Wi-Fi module (ESP 8266 Node). The approach was mainly focused on low cost and the potential of a future commercial product.

Bhatti et al. [24], in 2016, developed a PMU aimed at minimum cost with compromise, that is, reduced available processing time and sampling rate. The device applied simplified signal processing and focused only on the basic harmonic frequency, i.e., the 50 Hz, to compute magnitude, frequency, and phase estimations. The implementation utilized a Raspberry Pi Model B+ microprocessor, the ADS1015 Adafruit analog to digital conversion module, and a GPS module, with total cost under USD 110.

Angioni et al. [25], in 2017, developed a low-cost PMU (LOCO) to monitor distribution grids. The development was the outcome of a European Union Horizon 2020 research and innovation action and the principal objectives were to build modular software components for automation of power systems that could be installed in heterogeneous, low-cost, hardware components such as Raspberry Pi and Beaglebone Black. The proposed device was based on three components: a data acquisition board (MCC USB 201), a measurement computation unit (Raspberry PI) and a GPS module (based on MTK3339, error less than 10 ns). The total hardware cost is approximately €250. The achieved max TVE was 1%, the phase angle error less than 1 mrad (0.057 degrees) and the frequency error 0.5 mHz.

Zhao et al. [26], in 2017, continued the OpenPMU approach and developed a second version that was designed using both open hardware and software platforms. The design was based on the Beaglebone Black hardware platform, and the software was realized on a Linux non-preemptive environment to achieve hard real-time sampling and processing constraints. By implementing a complicated software driver for the two programmable real-time units (PRUs) the approach achieved 12.8 kHz sampling and 1.75 μs maximum timing error which equated to phase angle estimation of 0.031 degrees. The analog to digital conversion supported synchronous acquisition of eight channels, with 16-bit resolution.

Romano et al. [19], in 2017, realized a low-cost PMU prototype for distribution networks based on FPGA. The approach focused on: specific requirements targeting distribution grids, such as accuracy, both regarding the TVE and phase angle; cost due to the fact that the application needs a very large number of devices; and size of the device since the available space in secondary distribution substations and medium-/small-sized RES plants is rather limited. The proposed design proved that FPGA realization could achieve high, but realistic sampling frequencies such as 20, 32, and 50 kHz by using the NovTech IoT Octopus board, and the total cost could drop to USD 500 in medium/high production volumes.

Cetina et al. [27], in 2018, proposed a low-cost power system metrology laboratory based on Raspberry Pi and an energy metering IC (ADE7878). The experimental platform targeted educational purposes for undergraduate students, including, among others, exercises for PMU algorithms.

Schofield et al. [28], in 2018, carried out a review of commercial implementation of PMUs, and then open-source-based implementations (open architecture hardware and software). They also demonstrated the realization of another low-cost design.

Bucci et al. [29], in 2019, designed and implemented a measuring system based on a high-performance microcontroller that could carry out power quality analysis of three phase power systems. The main features of the design were cost effectiveness, reduced dimensions, and low power consumption, which made it suitable for installation in existing electrical cabinets or civil and commercial users.

Femine et al. [30], in 2019, presented another low-cost implementation of a PMU. The key feature of the design was that the data acquisition, data processing, and data communication were integrated in a single low-cost microcontroller. A thorough metrological characterization of the realized prototype was conducted, using a high-performance PMU calibrator.

Seger et al. [31], in 2020, prototyped another low-cost design approach for a PMU. The device was built around an ARM board and commercially available modules. A comparison with a commercial PMU device demonstrated that similar results for monitoring low voltage networks and frequency events could be achieved, however, with a cost between USD 100 and 150.

Shankar et al. [32], in 2021, presented the development for a low-cost design and implementation of a PMU. The design followed the existing IEEE standards for synchrophasor measurement with the help of Arduino Uno and Arduino Due microcontrollers.

Rama Raju et al. [33], in 2021, developed a cost-effective PMU for wide area monitoring system applications, with open-source hardware, which could be easily modified as per the requirements of the applications.

The most recent approach comes from Schofield et al. [34], in 2022, with the low-cost PMU named PhasorsCatcher. The design is described in high detail, and the performance proved to be acceptable for research approaches in distribution grids. Last but not least, three PhD dissertations have designed and developed experimental PMU architectures [35–37].

Taking into consideration all the above-mentioned publications, it seems that, since 2008, several designs and implementations of experimental PMU arrangements have been researched and proposed. Most of the designs have focused on low cost, some of them are open source, and some others are based on COTS programmable evaluation modules and boards. In several cases, the experimental prototypes have achieved performances comparable to more expensive commercial devices.

Although several industrial PMU devices are currently available in the market, all the systems are based on closed proprietary technologies and are mainly targeted to commercial applications. In addition, although certain open-source PMU and laboratory/breadboard approaches are also available, there is an implementation maturity gap compared to off-the-shelf devices. These experimental devices are not suitable for pilot installation, something that is highly required in research and innovation actions and initiatives. In addition, recent advances in smart grid research towards future resilient distribution grids need to go a step further from simulation environments and need to apply proposed state-of-the-art methodologies and algorithms in real application fields. In such cases, the deployable systems should comply with a minimum of industrial hardware standards and specifications, as well as provide re-programmable open platforms that are able to easily deploy AI applications, following edge and fog approaches. Finally, both hardware and software platforms should be easy to be tailored according to specific application requirements.

1.2. Motivation for the OpenEdgePMU Initiative

The motivation behind the present project is to establish an open community focused on high-end PMU applications for distribution grids. The principal novelty of the project is to provide a rigid set of open-source hardware and software tools that are useful for easy

and flexible implementation of pilot arrangements for research and innovation actions, with interest coming from academic entities, institutes, innovative SMEs, and startups.

Special interest in our initiative is anticipated by research and innovation projects (such as the European Union Research and Innovation Actions—RIA), where in most cases, system analyses based on simulations are not always adequately deliverable. These projects require proof-of-concepts and pilot installations in real-world applications and use case scenarios. A major necessity when developing and deploying such systems is to utilize programmable devices that have small knowledge entry barriers and comply with certain engineering standards (e.g., EMI, EMC, safety, etc.). Although development platforms and breadboards for laboratory use are widely available and adopted by academia, the hardware maturity is not adequate for industrial applications, even when targeting pilot cases. Certain development boards, such as Raspberry PI [38] and Beaglebone Black [39], can partially cover the applications' requirements; however, custom-made add-on cards, modules, and interfaces with external devices are also required. Thus, the total design fall (in most cases) into the breadboard/prototype status.

The proposed OpenEdgePMU initiative, bridges the aforementioned gap, and may help smart energy related academic entities to propose more mature hardware & software concepts for pilot cases and easily attract collaboration with potential end-users, such as electricity utilities, distribution system operators, energy generation plant owners, etc., in innovation activities. Targeting to increase the impact of our initiative in the research and innovation community, the OpenEdgePMU initiative provides open-source licensing for all software and hardware developments; the software is provided under the GPL license [34,35] and the hardware is provided under the TAPR Open Hardware license [14,15].

1.3. Challenges and Requirements for Distribution Grid PMUs

By focusing on PMU applications for power distribution grids, particular challenges must be taken into consideration. First, since most industrial PMUs are designed for transmission systems and contain a high number of functionalities, the cost is rather high. In distribution grids, the number of required PMU devices is high. The complexity of the distribution grid, having several distributed generation units and several different kinds of load, requires a much higher number of measuring devices. Even though the equipment cost is not the biggest part of the total cost of ownership for PMU applications [40], a possible high number of devices required in distribution grid applications could make the case unaffordable for massive exploitation.

In addition, transmission and distribution system measurements have key differences that must be taken into consideration [19,23]. For instance, the phase angle resolution regarding the distribution system must be as low as a few millidegrees, whereas in the transmission system a few degrees may be adequate. The resolution of the phasor magnitudes must be better in distribution grids as well as the accuracy of measurement, and the noise filtering and rejection in distribution grids need to be improved compared to transmission system cases. In general, combining both magnitude and phase, the accuracy of the TVE must be in the 0.0x% range.

The size of the devices is another factor that could affect deployment of PMUs in distribution grids. The available space for electronic equipment, especially in the secondary distribution substations, is very limited. Therefore, only compact apparatuses are possible to be installed and, in most cases, must comply to certain mechanical standards of the cabinets (e.g., DIN rail mount, etc.).

Furthermore, challenges regarding PMUs for distribution grids are relevant to high-fidelity data integration with utility distribution operations [3]. In a typical distribution grid case, several PMUs generate enormous numbers of data, which (at least) must be filtered, stored, transmitted, processed, correlated, and visualized in real time, in order to provide useful information and support operators' decisions. Several emerging trade-offs create a situation with extremely innovative potential. For instance, the dilemma of edge/fog vs.

cloud processing, or centralized vs. distributed data storage and processing, etc., seems to create huge research and scientific interest with respect to proposing new algorithms and innovative applications.

1.4. Novelty of the OpenEdgePMU Initiative

Based on all the above-mentioned challenges and applications' requirements, as well as the extensive review of recent PMU designs available in the scientific literature, our approach aims at aggregating the following high-level requirements, features, and functionalities into a novel integrated arrangement:

- Soft real-time edge/fog processing;
- Hard real-time processing of phasors;
- Local storage of all acquired data;
- Distributed storage management;
- Ability to support several different functionalities in a single processing system (e.g., typical PMU, fault detection and analysis, power quality analysis, state estimation, fast protection, smart grid automation, etc.);
- Phasors measurement accuracy adequate for distribution grids (0.0x%, millidegrees for phase);
- Compact size;
- Low cost;
- Open source (both regarding software and hardware design);
- Adequate project documentation to support adoption from the research and scientific community.

Our desire is to emphasize the edge/fog capabilities of the OpenEdgePMU system, as we vastly believe that the majority of future energy smart grid applications must follow similar concepts for several different reasons. First, a major trend today is to take advantage of the vast amount of generated data acquired by sensors installed in the field. These timeseries contain valuable information that can be extracted by artificial intelligence algorithms. Without edge processing capability, huge datasets must be transmitted to cloud software applications, in order to proceed to data processing, something that, in general, may be very challenging due to communication bandwidth limitations, cost of transmission, delays caused by data transfers, etc.

The proposed OpenEdgePMU system supports a novel edge processing approach. Edge processing together with the availability of local databases are used to eliminate the previously mentioned challenges. The following steps are applied:

- Vast amounts of locally generated data are processed by artificial intelligence agents running locally.
- The edge system generates only events to remote cloud software backends/frontends, thus reducing the necessity of extensive communication bandwidth and the relevant cost of transmission.
- The edge system stores the generated data timeseries locally, in case of a potential end-user's request for further detailed analysis. This distributed database may also vastly reduce the costs for cloud services.
- The edge system that processes the acquired data can communicate with minimum delays with other edge systems that are able to execute actions and control (e.g., protection, automation, etc.). The 5G infrastructure comes together to minimize communication latency and provide time-critical solutions.

The following sections present a brief description of all the system's components and related functionalities. The description begins with an overview of the system, continues with descriptions of the key components such as the DSP module, the edge processing module, the GPS, and the power supply module, and finishes with brief descriptions of certain application cases and projects that have been recently applied.

2. Materials and Methods

2.1. System Architecture

The OpenEdgePMU system is based on a configuration that uses two powerful processors: one processor that targets hard real-time deterministic responses and another processor for soft real-time functionalities. A GPS system is used to resynchronize the data acquisition with the global absolute time. Extensive on-system nonvolatile memory provides the opportunity for distributed databases. Fast Ethernet is used for interfacing 5G or ADSL communication means. More specifically, the high-level architecture of the proposed OpenEdgePMU system is depicted in Figure 1. The system consists of the following subsystems:

- The digital signal processor (DSP) module which is responsible for all hard real-time processing of the digitized phasor signals. It also contains the digitizer and analog conditioning electronic circuits that are responsible for interfacing with the external sensors of the phasor signals, as well as filtering and amplification suitable for the digitizer, which is responsible for the analog-to-digital conversion of the acquired phasor signals.
- The edge gateway module, which is responsible for all soft real-time processing, such as artificial intelligence, communication protocols, etc.
- The GPS module which is responsible for interfacing the satellite GPS signals, keeping absolute timing, and generating the time resynchronization signal for the digitizer.
- The power supply module which is responsible for the AC-to-DC conversion of one of the voltage phasors, to suitable low voltages for powering all electronic components.

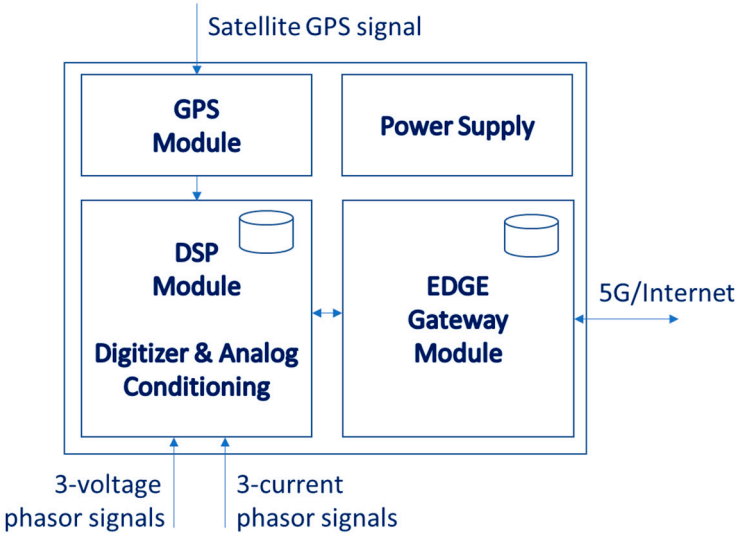

Figure 1. High-level architecture of the OpenEdgePMU system.

2.2. DSP Module

The DSP module handles all time-critical, hard real-time functionalities. The DSP module has been based on the design of the Texas Instruments TMDSEVM5517-C5517 Evaluation Module (EVM) [41]. The block diagram of the DSP module is depicted in Figure 2. The key components of the DSP module are:

- DSP TMS320c5517 (Texas Instruments, Dallas, TX, USA) [42] is the brain of the module running a firmware that handles all operations related to data acquisition, storage, processing synchronization, communication, and configuration. This is a high-performance, low-power, fixed-point digital signal processor from Texas Instruments, with the following specifications:

1. 200 MHz clock rate, 5 ns instruction cycle time, one or two instructions executed per cycle, dual multiply-and-accumulate units, two arithmetic and logic units (ALUs);
2. 320 KB on-chip RAM, zero wait state (64 kB dual-access RAM, 256 KB single access);
3. 128 KB on-chip ROM zero wait state;
4. Tightly coupled FFT hardware accelerator;
5. Peripherals: McSPI, McBSP, UHPI, EMIF, UART, USB, DMA, eMMC, SPI, I2C, and I2S.

- SDRAM (external) (Micron MT48H32M16LF) [43] is useful for firmware execution. The size of the SDRAM is 64 MB.
- Nonvolatile flash memory (external) (Infineon S29GL128S11DHIV20) [44] for storing the firmware of the DSP module. The size of the FLASH memory is 16 MB. The memory is in-system programmable; therefore, the developer can change and upload the DSP module's firmware.
- Nonvolatile SD memory (SD Card) [45] for storing the acquired data and other parameters for configuring the device. The default storage size is 64 GB and, depending on the sampling frequency, the storage period can last from some hours to several days.
- Analog-to-digital converter (ADS8586S, Texas Instruments) [46] for digitizing the analogue input signal. This is a 6-channel, high-speed, simultaneous-sampling ADC, with 16-bit resolution and fully differential bipolar inputs on a single supply, with the following key performance specifications:

  1. 250 kSPS max throughput per channel;
  2. Two selectable input ranges, $\pm$10 V and $\pm$5 V;
  3. Antialiasing low-pass filter to 24 kHz for $\pm$10 V range and to 16 kHz for $\pm$5 V range;
  4. DNL, $\pm$0.35 LSB;
  5. INL, $\pm$0.45 LSB;
  6. SNR, 96.4 dB
  7. THD, $-$114 dB.

- Analog conditioning electronic circuit for isolating the voltage input, reducing it to an acceptable level suitable for acquisition, and for amplifying the current input from the current sensors. The analog conditioning circuit consists of a total of six channels (three for voltage and three for current). The voltage input includes a differential isolation amplifier in combination with a voltage divider to reduce the input voltage and to isolate it from the digital circuit. The current input includes simple amplifier circuitry to amplify the input signal. The specifications of the voltage inputs are:

  1. Number of inputs, 3, one per phase;
  2. Input range, 230 VAC nominal;
  3. Isolation, optical isolation;
  4. Isolation voltage 4000 Vpeak;
  5. Transient immunity, 10 kV/us minimum;
  6. Input sensor type, Rogowski coil;
  7. Range, 100 mV/1000 A typical, but configurable for different sensors.

As depicted in Figure 2, the following interfaces with other subsystems are provided:

- Voltage analog input interface, to connect the external voltage phasor (230VAC nominal);
- Current analog input interface, to connect the external Rogowski coil;
- Full-duplex UART serial interface with the MCU/edge processing module, providing bandwidth 921,600 bps;
- GPS-PPS input signal interface, used for immediate external interrupt to the DSP, in order to resynchronize the internal sampling frequency timer;
- JTAG interface, used for connecting an external JTAG board to interconnect the DSP module with Texas Instruments Code Composer Studio environments, useful for firmware development and debugging;

- Full Duplex UART serial interface at 921,600 bps for direct debugging using a common terminal (common interface with MCU/edge processing module).

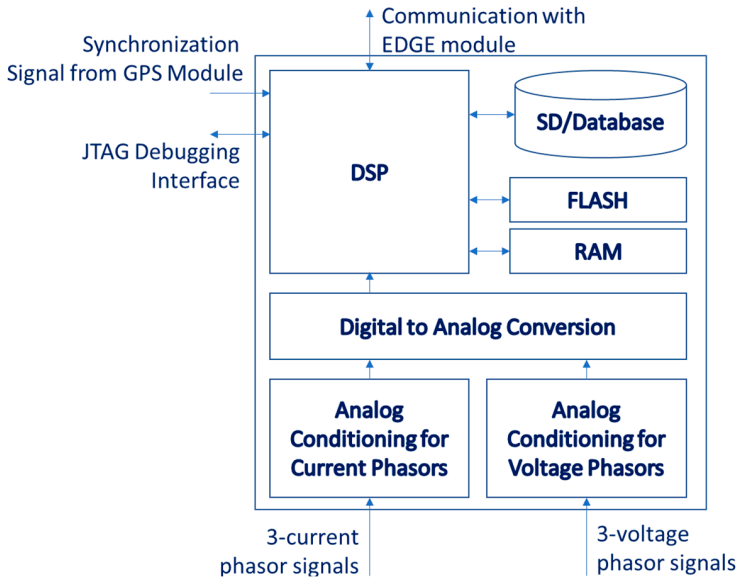

Figure 2. High-level architecture of the DSP module.

The DSP module is supported by two firmware: bootloader and application. The bootloader is used for bootstrapping the system, i.e., load the application firmware to the appropriate system memories, and then initiating execution. Based on certain jumper switches, the developer can use different boot loading sources, such as flash, SD, and McBSP. The standard boot loading method is via flash memory. The application firmware contains all the DSP module functionalities, which can be categorized as hard real-time, soft real-time, and system support functionalities:

- Hard real-time functionalities:
  1. Sampling of six (6) channels and conversion to digital signals;
  2. Storage of digital signals to the nonvolatile SD memory using EDMA transfers;
  3. Processing of the digital signals;
  4. Timing resynchronization with GPS.
- Soft real-time functionalities:
  1. Communication with the edge processing module;
  2. Measurements de-calibration.
- System support functionalities:
  1. Configuration via remote terminal;
  2. Measurement calibration;
  3. Debugging via JTAG and Texas Instruments CCS;
  4. In-system programming.

The application firmware of the DSP module is divided into four main layers: drivers, support libraries, processes, and application (shown in Figure 3). The structuring of all four layers is based on the pattern where higher level layers always utilize elements from lower level layers. The following list provides a description regarding each layer:

- The drivers layer is responsible for low-level driver implementation regarding certain internal DSP peripherals and external peripherals. The DSP internal peripherals are McSPI, I2C, UART, SD/MMC, EMIF, EDMA, GPIO, and ISR. The external peripherals are the ADC, NOR flash, and SDRAM memories.
- The support layer provides the application programming interface (API) to be used by processes to access low-level functionalities. This includes the terminal prompt

interface, the file system, and the XTERM support library, which is an implementation of the XMODEM protocol [47] used in In-System-Programming (ISP) to support future firmware updates, as well as potential file/data transfers.

- The process layer contains high and low priority processes related to device operation, including data acquisition (ACQ), data handling (DH), communication with the GW module (REMOTE), and the command prompt.
- The application Layer refers to the top-level application related to the actual functional operation of the DSP module regarding measurement. Essentially, this is a single entity that combines different processes to accomplish a certain task.

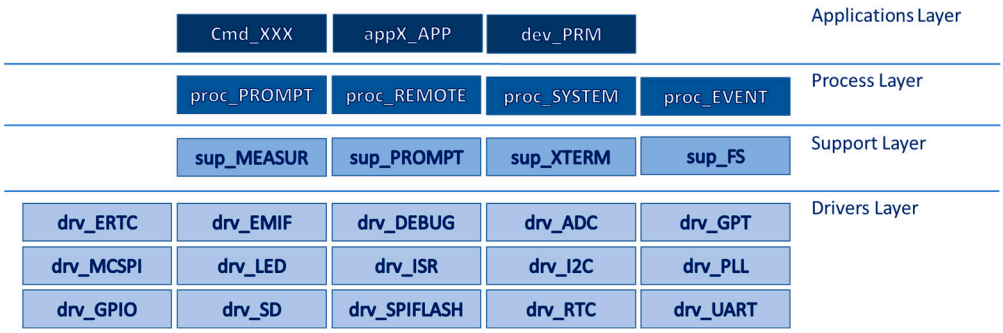

Figure 3. Layered architecture of the DSP module firmware.

The core functionality of the DSP module is depicted in Figure 4. The hard real-time operation is based on two interrupt service routines (ISR): one for TIMER-0 which is responsible for signals' sampling, and another for the GPS acquisition timing resynch. The ISR TIMER-0 calls the sampling function, which is responsible for transferring the ready sampled data from the ADC, via the McSPI peripheral. The clock speed for the transfer is 20 MHz, thus the time required for the total transfer of six channels with 16-bits for each channel is around 100 clock periods, i.e., 100×50 ns = 5 μs. Before doing the actual transfer, it signals the ADC to start a new signal sampling task. This can be handled by the ADC in parallel with the transfer of the previous sampled data to the DSP via the McSPI interface. The acquired data are stored into an intermediate memory buffer (Block_MEASUREMENT) as depicted in Algorithm 1. This memory buffer is separated into two identical parts. Each part contains the date and time of the first sample with millisecond accuracy, an array of 1363 samples for each one of the six channels, and finally a 32-bit checksum of the contained data into this structure (Figure 4). When the ISR-TMR0 completes filling one Block_Measurement buffer, it continues to the next one.

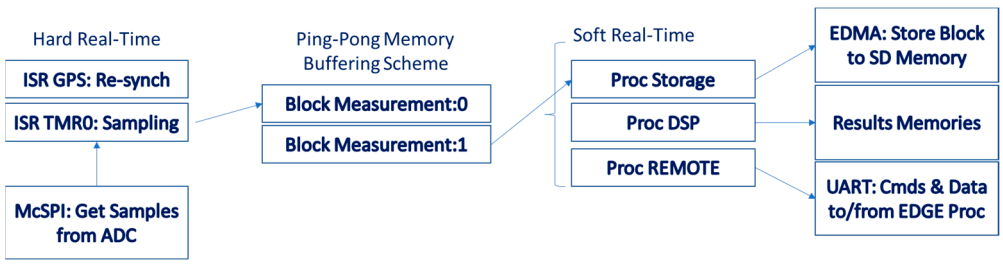

Figure 4. DSP module firmware key functionalities.

The soft real-time functionalities follow a round robin execution of three functions: *proc_STORAGE*, *proc_DSP*, and *proc_REMOTE*. All these processes are realized as state machines running in a cooperative manner, by means of sharing the available processing in reasonable time. The storage process checks if the hard real-time ISR-TMR0 has filled one Block_Measurement, to transfer the total block to the nonvolatile SD memory, via the MMC/SD peripheral of the DSP using EDMA transfers to achieve the best possible

performance. The implementation achieves continuous and uninterruptible storage of all six channels at 250 KHz sampling frequency, which means 6 channel × 250.000 samples/channel × 2 Bytes/sample = 3 Mbytes/s. The DSP process refers to any digital signal processing functionality that the developer needs to execute. The sample-by-sample and block processing options are both available to be applied according to the developer's request. The remote process is responsible for handling all data communication and commanding by using the edge gateway module. Again, this process follows a state machine approach, to avoid monopolizing the available processing time with this task.

Algorithm 1. Code Snippet-1

```
#define BLOCK_SIZE_MEASUREMENT              1363
#define NUMBER_BOCKS_MEASUREMENT            2
#define NUMBER_OF_CHNNELS_MEASUREMENT       6

struct Block_MEASUREMENT
{
Uint16 date_time [12];//ddmmyyhhmmssmmm
Int16 raw[BLOCK_SIZE_MEASUREMENT][NUMBER_OF_CHNNELS_MEASUREMENT];
Uint32 cs;//Check sum of the total FILE DataBlock structure
};
...
struct Block_MEASUREMENT block[NUMBER_BOCKS_MEASUREMENT];
```

The firmware of the DSP module also provides certain non-real-time processes (process_PROMPT, process_XTERM, etc.) that are responsible for the module's configuration, debugging, and other supporting tasks. The nonvolatile SD card of the DSP module provides a local database that is used for storage of:

- System configuration parameters, 8 kBytes;
- Events, 512 events × 64 Bytes = 32 kBytes;
- Signal data, 3.000.000 Blocks × 16.384 Bytes = 49,152 Gbytes;
- Extra, almost 14 GBytes for future use.

The biggest part of the memory is used for storage of the acquired signal data. Depending on the application requirements, the developer may change the default nonvolatile memory configuration accordingly. In addition, larger SD card sizes (e.g., 128 GBytes+) can probably be used; however, these have not yet been validated. Due to very high storage rates, very high-speed SD cards have been chosen (such as the SanDisk Extreme® microSD™ UHS-I Card) [45].

2.3. Edge Processing Module

The edge processing module is based on the Beaglebone Black design [39]. It can run a Linux operating system and is responsible for most soft real-time functionalities, including AI processing, communication protocols, IoT interfacing with the backend, etc. The brain of the module is the OSD3358 System-on-Chip (SoC) [48], which includes the following key features:

- Texas Instruments Sitara™ AM335x ARM® Cortex® A8 Processor running at 1 GHz;
- Ethernet controller and interface (supports 10/100/1000 Mbps communication rates);
- Nonvolatile storage controller for MMC, SD, and SDIO;
- Supports up to 1 GB DDR3L memory.
- On-chip peripherals: CAN, SPI, I2C, UART, GPIO, etc.

Figure 5 shows the high-level architecture of the edge processing module, including the interfaces with other subsystems. As depicted, the MCU (OSD3358) provides the following interfaces:

- MMC to interface with the SD memory card storing the OS and all the data;

- I2C to interface with the on-board EEPROM memory;
- UART to interface with the PMU and other external modules;
- Ethernet to provide an Ethernet interface to connect to the local network.

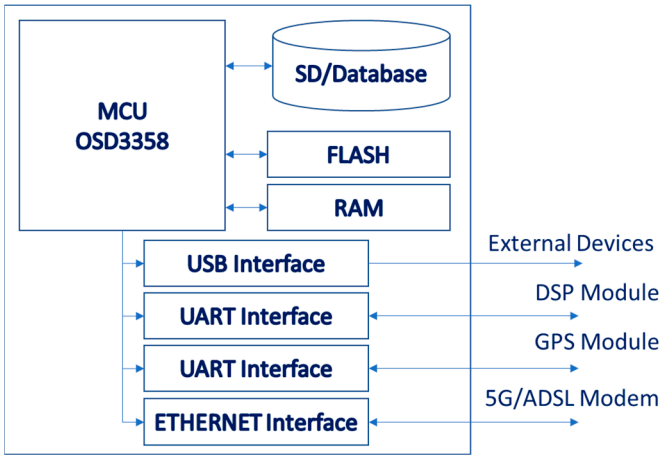

Figure 5. High-level architecture of the edge processing module.

The edge processing module software is based on the Linux Debian 10 operating system. This is important since critical functionalities, especially in terms of networking, become immediately available. These include all the TCP/IP protocol stack and security via firewalls. Additionally, the use of Linux OS allows the creation of software using high-level programming languages such as C++ and Python. Python is the main programming language used in the development of the provided software. Python was selected for its readily available scientific and mathematical libraries that allow quick development of AI functionalities. The following libraries come preinstalled in the edge processing module software:

- PandaPower [49], a library used for power system modeling;
- Pandas [50], a numerical and data analysis library;
- Scikit-learn [51], an AI construction and execution library;
- Flask [52], a library for the development of HTTP-REST interfaces;
- SQLite [53], a library for the interaction with the local SQLite database.

It is noted that the user can add libraries to cover other specific needs. Additionally, it is also noted that the user may rely on a completely different programming stack by using the Linux distribution's repositories. The current stack is depicted in Figure 6.

Figure 6. High-level architecture of the edge processing module software stack.

The application software is built on top of the aforementioned stack and its main functionalities are:

- Command and control of the PMU, including commands for the DSP module;
- Acquisition of data generated by the DSP module;
- Local database;
- Communication with backend/frontend applications;
- Communication with other OpenEdgePMUs;
- Execution of artificial intelligence functionalities.

Figure 7 depicts the high-level architecture of the edge processing module application software. It consists of the following main building blocks:

- The DSP module connector is responsible for connecting the edge processing module application with the DSP module via the serial port. After a successful connection, it can retrieve the current measurement of the DSP module. Additionally, it is responsible for periodically retrieving the latest measurements of the DSP module. It also stores the measurements to the database after successful retrieval.
- The REST server is a stand-alone service, based on the Flask framework, and is responsible for deploying a RESTful API on the edge processing module application, to serve HTTP requests. Such requests can refer to the latest measurements retrieved from the DSP module, historic measurements, or the status of the device (e.g., connection status, free storage space, MAC address, etc.). For the first two types of requests, the REST server connects directly to the database and retrieves the requested data, while for the third type of request, the REST server obtains the relevant information directly from the OS, by executing certain Linux commands. In all cases, the REST server will reply to the client with a REST protocol HTTP response, containing either the requested data upon success, or an error message upon failure.
- The real-time database component acts as a bridge between the edge processing module application and the database. Any measurements that must be stored are first sent to this agent, who is then responsible for storing them properly. Additionally, this agent handles write errors (e.g., duplicate values), database connectivity issues, and provides the state fetcher agent any measurements that might be requested. The latter functionality is crucial when storing rows across multiple tables, which must link their data by utilizing foreign keys.
- The SQLite database application, after retrieving the measurements, takes care of storing the measurements in the database that the device has in its storage disk. Since this database is installed on each device, it is important to consider some basic parameters, such as the storage frequency and the size of the tables. A fine balance between collecting adequate data samples for future analysis and properly utilizing the limited storage space is found in one-minute sampling of the real-time DSP module measurements. Regarding the storage frequency, the storage (by the gateway application) and the retrieval (by the main application) can differ. In addition, the database used is relational and is based on SQLite technology. It is noted that the database already has a predetermined structure that allows the continuous storage of PMU and RTU data.

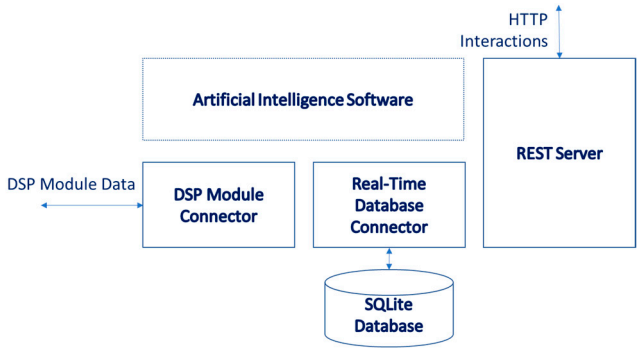

Figure 7. High-level architecture of the edge processing module application.

All of the above are designed to run as out-of-the-box features, meaning that even without developing an additional AI application, all PMU data are collected, stored, and exposed via a dedicated API. However, by using the above features and the power of the preinstalled libraries, one can create AI-powered applications without having to worry about the low-level technical issues such as data retrieval from the DSP module.

2.4. GPS Module

The OpenEdgePMU system contains a Global Positioning System (GPS) module to acquire the global absolute time and to use the GPS-PPS signal for resynchronizing acquisition. A UART interface is utilized for communicating with the edge processing module to configure system time. The following two GPS modules can be used in the OpenEdgePMU system:

- Lantronix GPS/GLONASS Receiver Module A5100-A [54];
- Adafruit Ultimate GPS [55].

The options both provide an uncertainty of the PPS signal less than 500 ns.

2.5. Power Supply Module

The power supply module (PSM) is responsible for providing power to all electronic components. The PSM input is the 230 VAC input from any of the three input voltage phases. Figure 8 depicts the high-level architecture of the PSM. The AC mains filter is used to protect the power supply from signal interference coming from the input voltage. The primary AC-to-DC converter is used to convert the 230 VAC to 5 VDC. Several other DC-to-DC converters generate all required voltages for the system's electronics. The main design specifications of the PSU are:

- Input range, 90–305 VACl;
- Isolation, 4000 VAC;
- Consumption, 5 Watt max;
- Other, option to power from phase A and neutral.

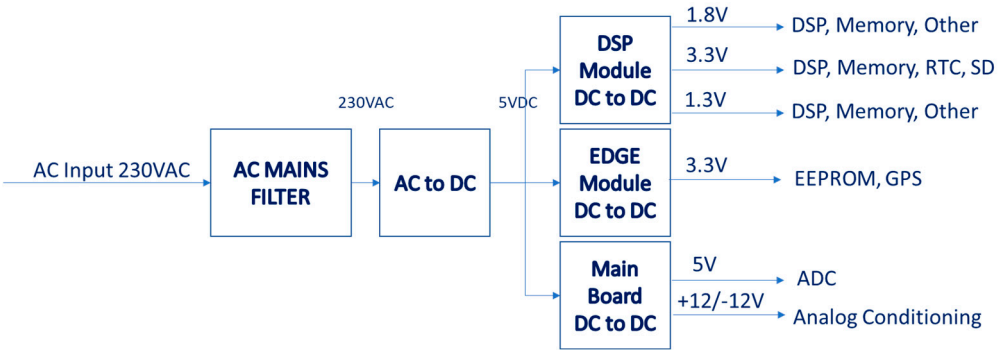

Figure 8. High−level architecture of the power supply module.

3. Results and Discussion

This section presents the results regarding the implementation and certain application cases of the OpenEdgePMU system.

3.1. Hardware Implementation

Hardware implementation of the OpenEdgePMU system consists of three (3) electronics boards:

- The DSP board is one of the key components in the OpenEdgePMU system. To simplify the overall design and utilization of the DSP in different devices, the DSP board is segregated by having its own design. In this way, a separate board was designed to include all the necessary components required for the DSP to operate with the aim to keep the digital high-speed electronics section segregated and to expose all the necessary interfaces via a common board-to-board mezzanine connector. The idea is that the DSP board is mounted onto the main board of the PMU module, which has a simpler design and uses only the necessary interfaces required. Figure 9 shows the top side of the DSP board, depicting its main components (DSP, external memory modules, power regulator, crystal etc.). The DSP Board has:

1. Dimensions, 67 mm × 52 mm;
2. Number of components, 203;
3. PCB layers, 8;
4. Schematic sheets, 12;
5. Estimated cost, €120.

- Regarding the edge board, as shown in Figure 10, the edge processing module is also segregated from the OpenEdgePMU system; it is essentially a separate board that is connected to the OpenEdgePMU base board. This board includes the most essential elements to make the MCU OSD3358 SoC operate properly, and all the interfaces are exposed to two mezzanine connectors to be used in other designs depending on the application. This is done to simplify the design process and to easily accommodate different designs utilized in different devices. Figure 10 depicts the top side of the edge board. The edge board has:
  1. Dimensions, 60 mm × 44 mm;
  2. Number of components, 144;
  3. PCB layers, 6;
  4. Schematic sheets, 7;
  5. Estimated cost, €200.

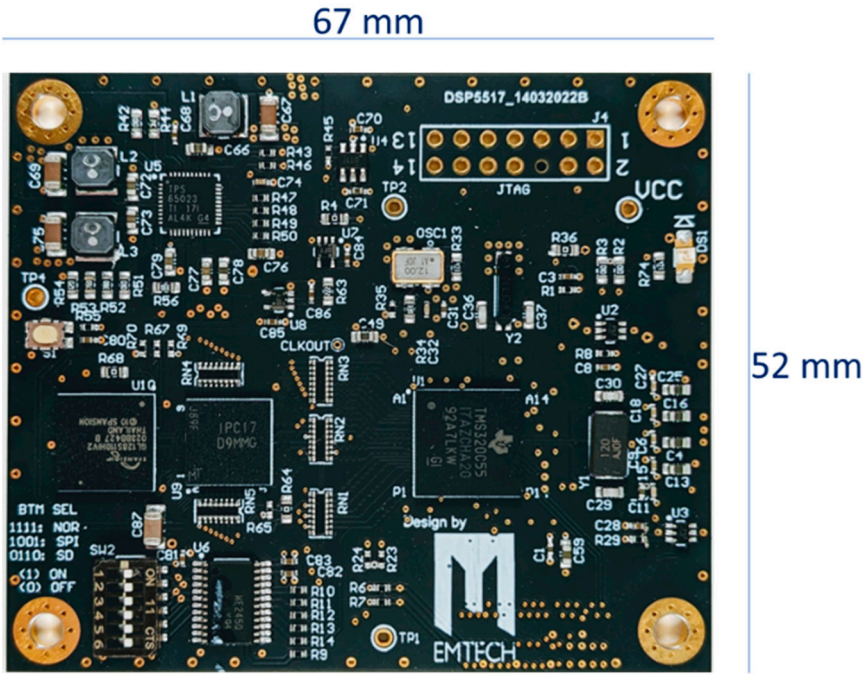

Figure 9. Implementation of the DSP module board, top side.

As an alternative to the developed edge processing module, a COTS Beaglebone Black can be used, Figure 11.

- The main board includes all other electronic components of the OpenEdgePMU system, such as the power supply, the analog conditioning and ADC part of the DSP module, input/output connectors, etc., as well as all other sockets suitable to attach the rest of the circuits, such as the DSP board, the edge board, and the GPS. Figure 12 shows the board without and with system add-on components. The main board has:
  1. Dimensions, 135 mm × 150 mm;
  2. Number of components, 201;
  3. PCB layers, 2;
  4. Schematic sheets, 10;
  5. Estimated cost, €150.

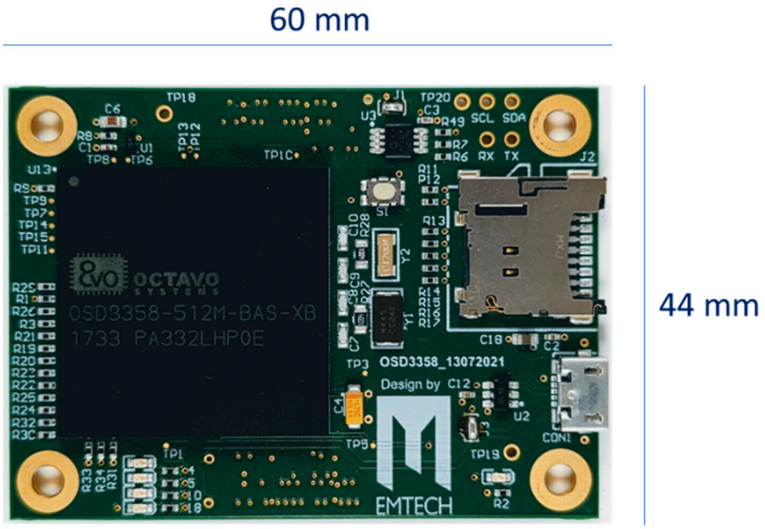

Figure 10. Implementation of the edge processing board.

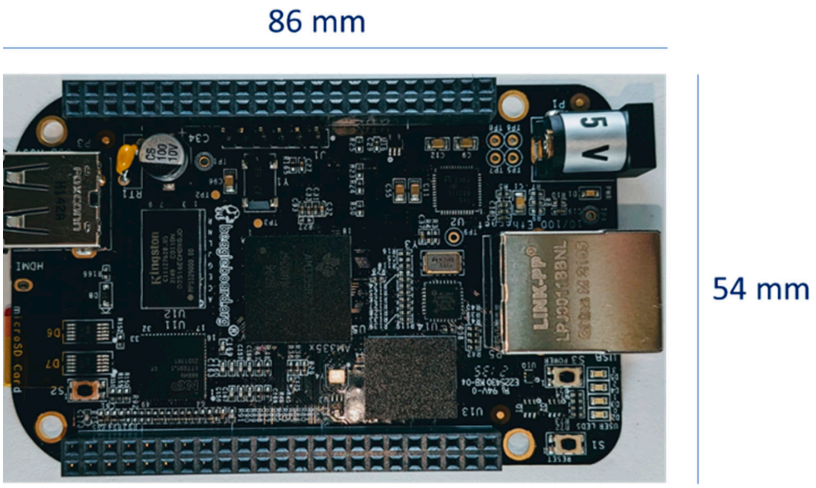

Figure 11. The Beaglebone Black can be attached as the edge processing module.

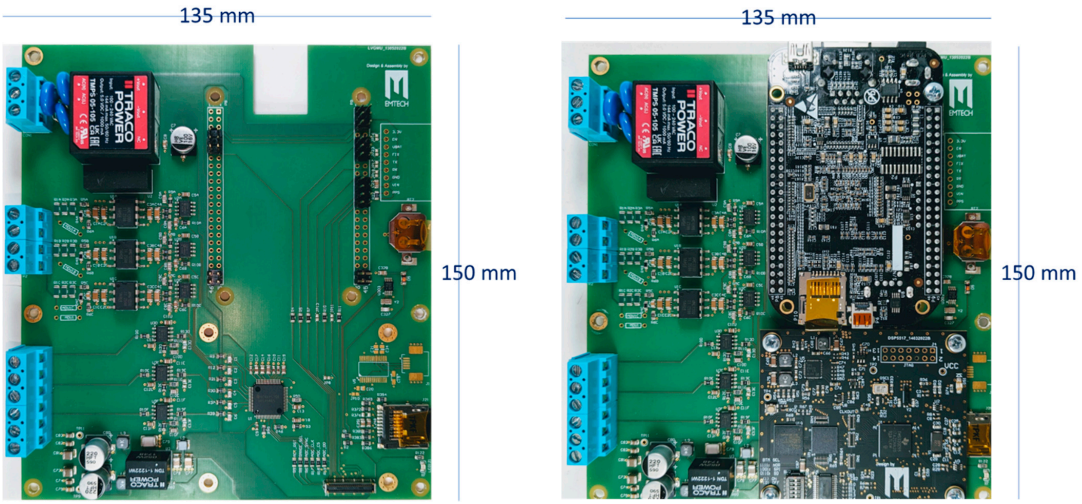

Figure 12. Implementation of the main board. **Left**: The main board. **Right**: The main board along with mounted DSP and EDGE processing units.

3.2. Current Funded Projects Using the OpenEdgePMU System

3.2.1. eBalance+ Project (European Union H2020)

The design of the OpenEdgePMU system has been used in the eBalance+ project [56] for a low voltage management unit (LVGMU). Figure 13 depicts a diagram of how the hardware units have been enclosed in a plastic box and interfaced with external sensors and subsystems. The PMU module, and eventually the LVGMU device, are integrated directly with the transformer. Specifically, the voltage inputs, on the one hand, are directly connected to the low-voltage side of the transformer; three inputs correspond to the three grid phases. The current inputs, on the other hand, are connected using Rogowski coil sensors that measure the current through each phase line. The device is designed to accommodate different types of Rogowski coil sensors by simply configuring a specific parameter to translate the output of the sensor to ampere units. The LVGMU is housed in a polycarbonate enclosure with the dimensions 300 mm × 200 mm × 90 mm. Figure 13 shows the connections between the LVGMU and the external elements, the types of cables used, and information about the connectors used. As depicted, the cable used to connect to the transformer is a 4 × 2.5 mm^2 cable, although other types with thinner inner wires can also be used, since the size of the cordgrip used (M20) can accept cables of 12 mm maximum diameter. Similar type cordgrips, but smaller size (M12), are used to hold the Rogowski coil sensor cables entering the enclosure. For the power input, an IEC C14 plug is used, which is part of the input AC mains filter that is included. For the network interface, a common panel mount RJ45 connector is used to provide the Ethernet interface required. Finally, an SMA connector is provided for the connection of the GPS antenna.

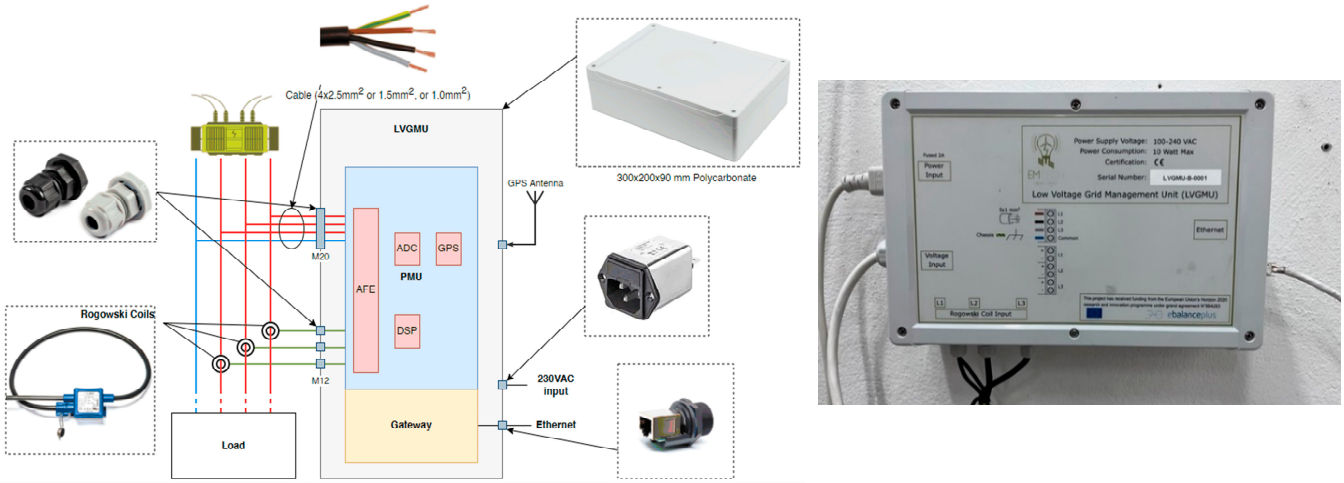

Figure 13. The LVGMU realization for the eBalance+ project, based on the design of the OpenEdgePMU system.

Within the context of the eBalance+ project, two different pilot demo sites are foreseen: the University of Malaga (UMA) and the University of Calabria (UNC)). A total number of 12 LVGMU devices are foreseen to be installed and operated during the demonstration phase of the project. Some devices have already been installed, and as planned, the remaining devices will be installed within the 2023 Q1 period.

3.2.2. TEXNOREYMETA Project (Greek National Funded)

The design of the OpenEdgePMU system has been utilized for the development of a secondary substation monitoring (STM) solution for a research and development project [57]. The objectives of the project were (a) the detection of non-technical losses in distribution grids and (b) the predictive maintenance of medium-voltage/low-voltage distribution transformers. The firmware of the OpenEdgePMU was modified in order to support:

- Real-time power quality analysis based on EN50160;
- Energy meter for secondary distribution substation;
- Phasor measurement unit.

The hardware design of the OpenEdgePMU system was adopted, and certain additional automation features were added, such as low sampling frequency measurements regarding the transformer health (temperature, vibrations, magnetic field, etc.). A mechanical enclosure similar to the eBalance+ project was used. The system has been installed in three (3) secondary power distribution substation in the Cyprus grid (operated by the Electricity Authority of Cyprus), Figure 14.

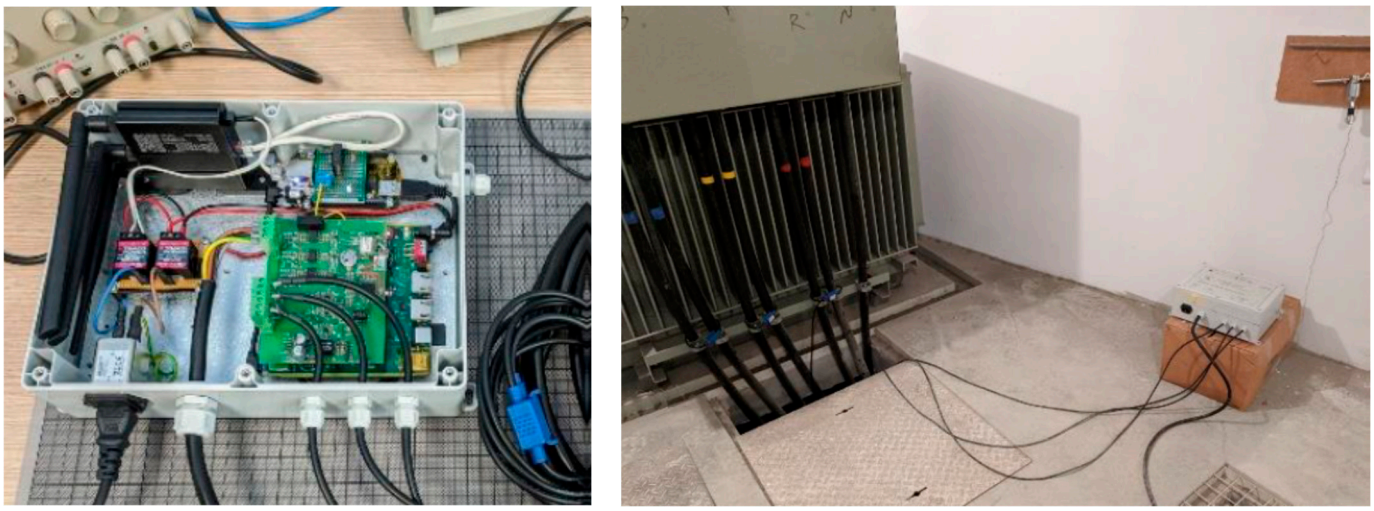

Figure 14. The STM realization for the TEXNOREYMETA project based on the OpenEdgePMU system.

The combination of the PMU with a power quality analysis (based on EN50160) is an ideal case to demonstrate the edge processing benefits. More specifically, the PMU continuously and uninterruptedly acquires the voltage and current phasors and saves the data to the local database of the DSP module. Using 10 kHz sampling frequency, the size of the data generated every day is:

$$10{,}000 \text{ samples/s} \times 6 \text{ channels} \times 2 \text{ bytes/channel} \times 86{,}400 \text{ s/day} = 10.4 \text{ GBytes/day}$$

Apparently, this size of data is very difficult to send to the backend even using 5G communication modems. Instead, the EN50160 power analysis algorithms are executed in real time in the DSP module and are responsible to mark the segments of data (data blocks) that a power quality event has detected (e.g., voltage sag, harmonics, voltage dip, etc.). Only the interesting blocks are selected to be automatically transmitted in the backend. For instance, the average daily events of each one of the three installations in Cyprus is around 50 events. The backend retries 10 data pages (five before and five after the event, however this is configurable) and each data page has 1363 samples of six channels, so the event related data generated every day is:

$$50 \text{ events} \times 10 \text{ pages/event} \times 1363 \text{ samples/page} \times 6 \text{ channels} \times 2 \text{ bytes/channel} = 8.2 \text{ Mbytes/day}$$

which is three orders of magnitude lower than transferring the total amount of data generated.

Today, the system is in pilot operation and detailed results will be published soon.

3.3. AI and Optimization Application Examples Using the OpenEdgePMU System

3.3.1. Power Flow (PF)

The power flow model of a power system is built using the relevant network, load, and generation data. Outputs of the power flow model include voltages at different buses, line

flows in the network, and system losses. These outputs are obtained by solving nodal power balance equations [58]. Since these equations are nonlinear, the Newton–Raphson technique is employed to solve this problem. By running PF, useful information about the voltages (magnitude and angle) for a given load, generation, and grid condition can be obtained at any time, and therefore appropriate actions can be taken to avoid surges and overloads. The power flow component is a wrapper for the python PandaPower library. PandaPower is extensively used for power system modeling, analysis, and optimization [59–62] as it presents a lot of advantageous features. PandaPower is implemented in Python, guaranteeing free availability and flexible expansion with other open-source libraries. The OpenEdgePMU system, and more specifically the edge processing module has been used to execute the above power flow approach for several distribution grid topologies, and the maximum execution time was limited to a few seconds (even milliseconds in certain low complexity grids).

3.3.2. Optimal Power Flow (OPF)

The optimal power flow (OPF) problem refers to the actions that should be taken to satisfy the power requirements of each consumption node in the most economical way, while obeying operational constraints [63]. During this task, the control variables (reactive power of PV inverter) are optimized, and then the control actions are applied within an iterative loop. By solving OPF, the instant optimal parameters of each generation bus and controllable bus of the grid are determined. The OPF problem, in its generic form, is a large-scale, nonconvex, mixed integer nonlinear programming (MINLP) optimization problem, belonging to the NP-hard class. This is partly due to the nature of the design variables that may be discrete, and to the non-continuity, non-differentiability, and non-convexity of the objective function. The optimization problem in hand can be described as:

$$\begin{gathered} \min f(\boldsymbol{x}, \boldsymbol{u}) \\ \text{s.t.} \\ \mathrm{g}(\boldsymbol{x}, \boldsymbol{u}) = 0 \\ \boldsymbol{h}_{min} \leq (\boldsymbol{x}, \boldsymbol{u}) \leq \boldsymbol{h}_{max} \end{gathered} \tag{1}$$

The vector $\boldsymbol{x}$ contains the state variables of the power grid, namely the active power P_i, voltage V_i, and phase δ_i of the nodes, while vector $\boldsymbol{u}$ contains the control variables, which can be the active power of generation nodes P_i^G, reactive power injections Q_i^G, transformers tap position T_{ik}, phase shifter position φ_{ik}, and capacitor bank switch position $Q_{c_{ik}}$. The underlying constraints of the OPF problem are derived from the physical limits of the system and Kirchhoff's laws [64] and are listed below:

- Voltage limits of i node

$$V_{i,min} \leq V_i \leq V_{i,max} \tag{2}$$

- Generation limits of generation nodes

$$\begin{gathered} P_{i,min}^G \leq P_i^G \leq P_{i,max}^G \\ Q_{i,min}^G \leq Q_i^G \leq Q_{i,max}^G \end{gathered} \tag{3}$$

- Tap limits on ik branch

$$T_{ik,min} \leq T_i \leq T_{ik,max} \tag{4}$$

- Loading limit of ik branch

$$S_{ik} \leq S_{ik,max} \tag{5}$$

- External grid active and reactive power injection limits

$$\begin{gathered} P_{min}^{in} \leq P^{in} \leq P_{i,max}^{in} \\ Q_{min}^{in} \leq Q^{in} \leq Q_{max}^{in} \end{gathered} \tag{6}$$

- Upper and lower limits of the curtailed power of each RES unit

$$\begin{matrix} P^c_{min} \leq P^c_i \leq P^c_{i,max} \\ Q^c_{min} \leq Q^c_i \leq Q^c_{max} \end{matrix} \quad (7)$$

- Active and reactive power balance at i node

$$\begin{matrix} P^G_i(x,u) - P^L_i - P_i = 0 \\ Q^G_i(x,u) - Q^L_i - Q_i = 0 \end{matrix} \quad (8)$$

where P^L_i and Q^L_i are the active and reactive load, and P_i and Q_i are the incoming or outgoing active and reactive power. Similar to PF, OPF is also executed with the help of PandaPower. The inputs of the algorithm consist of (a) the grid's topology and (b) the real-time state of the grid. The desired target objective must also be determined which may differ among:

- real power losses

$$P_{LOSS} = \sum_{ik=1}^{m} R_{ik}|I_{ik}|^2 \quad (9)$$

where P_{LOSS} is the total active power losses in the distribution network, m is the number of branches, R_{ik} is the resistance in the branch ik, and I_{ik} is the current in the branch ik.

- voltage deviation minimization

$$VD = \sum_{i=1}^{N} \left| V_i - V_i^{sp} \right| \quad (10)$$

where V_i^{sp} is the prespecified reference voltage value at i-th load bus, which is usually set at the value of 1.0 p.u. and N is the number of load buses,

- RES curtailment minimization

$$CP = \sum_{i=1}^{M} CF_i \times \left(P^c_{i,max} - P_i \right) \quad (11)$$

where CF_i is a weight factor of the curtailed power of the i-th RES unit, with $\sum_{i=1}^{M} CF_i = 1$.

Regarding the optimization algorithm, several metaheuristic methodologies have been implemented, namely particle swarm optimization (PSO) [65] and genetic algorithms (GA) [66], while more can be easily integrated. The effectiveness and superiority of metaheuristics for solving the OPF problem have been thoroughly researched and demonstrated [67–69]. After the execution of the algorithm, two types of outputs are produced. The first concerns commanding and controlling the underlying elements of the power grid, specifically, for the RES elements. The second concerns a series of KPIs that define the performance of the algorithm. The architecture of the implemented algorithm is illustrated in Figure 15. The OPF approach has been realized and executed in the OpenEdgePMU system using the edge processing module. The results of OPF executions for the benchmark grid IEEE123 are presented in Table 1. As this section serves as an overview of the capabilities of OEPMU application, the results for real power losses minimization are only included, demonstrating the initial and optimized values. Looking at these values, it is obvious that the proposed Volt/Var algorithm achieves its goals.

The architecture of the implemented algorithm is illustrated in Figure 15. The OPF approach has been realized and executed in the OpenEdgePMU system, using the edge processing module. The results of OPF executions for the benchmark grid IEEE123 are presented in Table 1. As this section serves as an overview of the OEPMU application capabilities, the results for real power loss minimization are only included, demonstrating

the initial and optimized values. Looking at these values, it is obvious that the proposed Volt/Var algorithm achieves its goals.

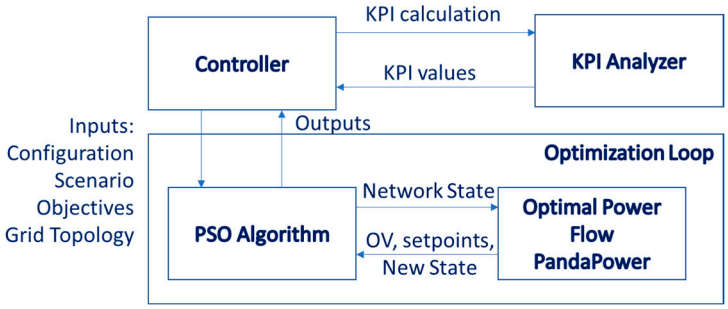

Figure 15. High-level architecture of the OPF framework implemented by the OpenEdgePMU system.

Table 1. OPF results subject to real power loss minimization.

Topology	IEEE 123
Objective	Real power losses
KPI initial value	3.360886005262138
KPI optimized value	0.1510907405190642

An important metric while trying to optimize an objective is always the speed of convergence. The convergence of the fitness function values versus the generations of the PSO algorithm is depicted in Figure 16.

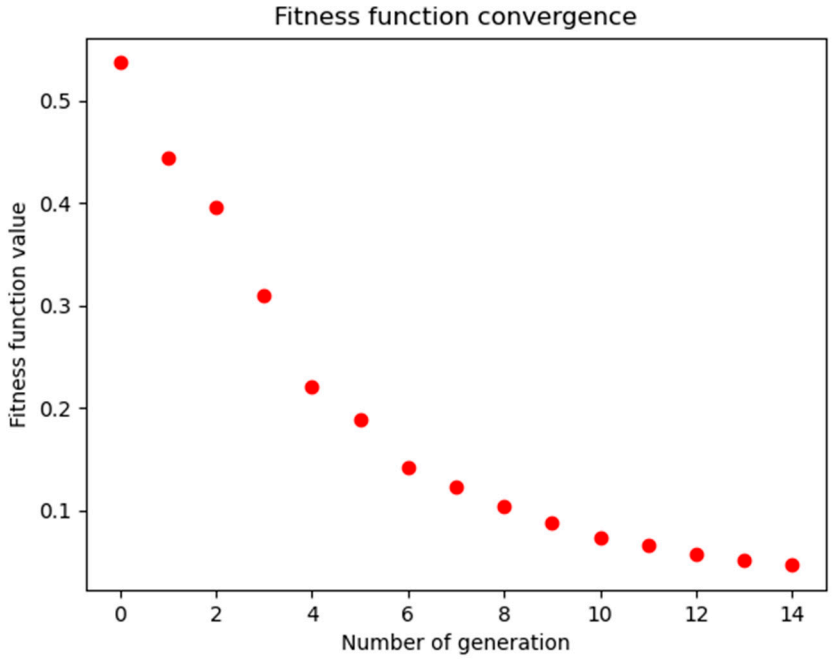

Figure 16. Fitness function value versus number of generations.

3.3.3. Load Prediction Based on AI

Electric load forecasting is, from a mathematical modeling point of view, a regression problem where the future value of the load demand has to be estimated by having a set of past values available. The field of computational intelligence, and more specifically, the field of machine learning (ML) [70] has proven to be an invaluable resource that provides a multitude of approaches to solve the aforementioned problem [71]. A number of the most

widely used methodologies, namely linear regression [72,73], support vector regression (SVR) [74,75], regression trees [76], random forests [77], MLP neural networks [78,79], and LSTM neural networks [80] have been developed in Python and wrapped in a single algorithmic framework that supports the following operations: data preprocessing, model training, model evaluation, extraction of the desired dataset, and results acquisition of an evaluated model. In a future version of the predictive framework, it is foreseen to add additional methodologies such as sparse coding [81] and convolutional neural networks [82]. It is useful to note that the integration of new models follows a standard and straightforward formalism. At the same time, the possibility of simulations is provided under different configurations, such as training scenario, evaluation scenario, prediction horizon, and reference year. Due to the high computational cost required to train the models, this process takes place at a central operational point (cloud application for power forecasting backend support), while the generation of predictions from the already trained models are performed seamlessly by the OpenEdgePMU system, and specifically using the edge processing module. Providing reliable and accurate load forecasts is an essential complementary feature of any modern power system, participating in a multitude of tasks such as scheduling generation sources [83], energy price forecasting [84], and secure operation [85]. The performances of some standard ML methods are evaluated based on the values of the MSE and R^2 indices obtained for forecasts with different horizons and different execution scenarios for each horizon are shown in Table 2. The execution times for the load forecasting models in the OpenEdgePMU system are in the range of a few seconds.

Table 2. Load forecasting results for multiple time horizons and ML methodologies.

Method/ Scenario	15 Min (sc013)		1 h (sc011)		6 h (sc034)		12 h (sc001)		24 h (sc035)	
	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE
LR	0.9731	0.9189	0.8805	2.2507	0.7261	3.4779	0.7571	2.9458	0.8562	2.6320
CART	0.9718	0.9303	0.8945	2.0556	0.7743	2.9189	0.7803	2.6624	0.8332	2.8246
RF	0.9720	0.9272	0.8948	2.0496	0.7741	2.9117	0.7806	2.6557	0.8658	2.5198
MLP	0.9734	0.8959	0.8898	2.0793	0.7533	3.1473	0.7461	2.8195	0.8569	2.6001
LSTM	0.9720	0.9564	0.8814	2.20008	0.7556	3.1903	0.7719	2.8539	0.8592	2.6377
SVR	0.9724	0.9155	0.8751	2.2201	0.7244	3.3945	0.7379	2.9913	0.8539	2.5498

The explanation of the simulation scenarios is set out below:

sc001, weekdays of one year;
sc011, daylight hours of one year;
sc013, daylight hours of weekdays only of one year;
sc034, spring days of one year;
sc035, summer days of one year.

3.3.4. Islanding

Intentional islanding is the purposeful sectionalization of the utility system during widespread disturbances to create power "islands". These islands can be designed to maintain a continuous supply of power during disturbances of the main distribution system. The distributed energy resources can then supply the load power demand of the islands created until reconnection with the main utility system occurs. These disturbances refer to cascading failures, of which the massive economic and social impacts have motivated a great deal of research effort on studying the vulnerability of power grids [86,87]. The aim of intentional islanding within the proposed approach is the resiliency of the network in terms of frequency balancing and continuous demand supply [88,89]. This optimization task is conceptualized by executing optimal power flow for several islanding setups of the network. The problem formulation includes the objective of load shedding and switches cost

minimization. The design variables comprise the setpoints of DERs, while the constraints of the intentional islanding problem coincide with those of OPF.

The execution of the algorithm implemented in Python can be done entirely in the field, in the edge board of the OpenEdgePMU system. First, a configuration must be defined that contains the network's topology, the optimization objectives, the type and point of failure (e.g., line tripping) that enables the islanding, as well as all the possible islanding setups. Then, the current state of each island is needed as input to the OPF algorithm which has already been described. The aim of this process is to extract the islanding setup with minimum cost, depending on the objective target set when OPF is executed. The application of the selected islanding setup is carried out by movements of smart switches that control the separation of the islands from the grid. It is also worth noting that communication between OpenEdgePMU nodes via 5G interface greatly reduces latency. Finally, the load level of the islands is monitored continuously, and load shedding is applied in the case of an imbalance. A test case is presented to validate the correct and efficient operation of the islanding algorithm carrying out the following steps:

1. Assume that failures are detected on Lines 5, 6, 12, and 16 of the grid topology shown in Figure 17.
2. Assume also that, initially, all switches are closed except for Switches 5, 9, 12, and 14.
3. The aforementioned failures lead to opening Switches 3, 4, and 7. Then, two possible islanding setups are obtained:
   a. Setup (a), closing Switch 5, Islands 1 and 2 are formed;
   b. Setup (b), closing Switches 5, and 15, Island 1 is formed.
4. Optimal power flow is solved for each islanding setup, Tables 3 and 4.
5. For each islanding setup, OPF is conducted, and the respective objective value is obtained, as shown in Table 3. Islanding setup (a) is selected as it achieves the minimum objective value.
6. Islanding setup (a) is applied, and then load shedding is conducted to maintain power supply to as many consumers as possible, as presented in Table 4. In this case, no load shedding is taking place, as the required demand does not exceed the available energy produced. The islanding algorithm is completed by solving the OPF problem for each island to ensure that power losses are minimized.

Table 3. Optimal power flow solution results.

Configuration	OPF Solution (RPL)
a	0.007605321647110751
b	3.360886005262138

Table 4. Optimal power flow solution detailed results.

Island	Generation Unit	Consumption	Load Shedding	OPF Results (OPF)
#1	SGEN 2 = 0.15	LOAD 3 = 0.01 LOAD 4 = 0.01 LOAD 5 = 0.01 Total = 0.03	No	0.00010976394887844073
#2	SGEN 4 = 0.15 SGEN 5 = 0.15 SGEN 6 = 0.15	LOAD 8 = 0.01 LOAD 9 = 0.01 LOAD 10 = 0.01 LOAD 11 = 0.01 LOAD 12 = 0.01 LOAD 13 = 0.01	No	0.0021422370982533434
	Total = 0.45	Total = 0.06		

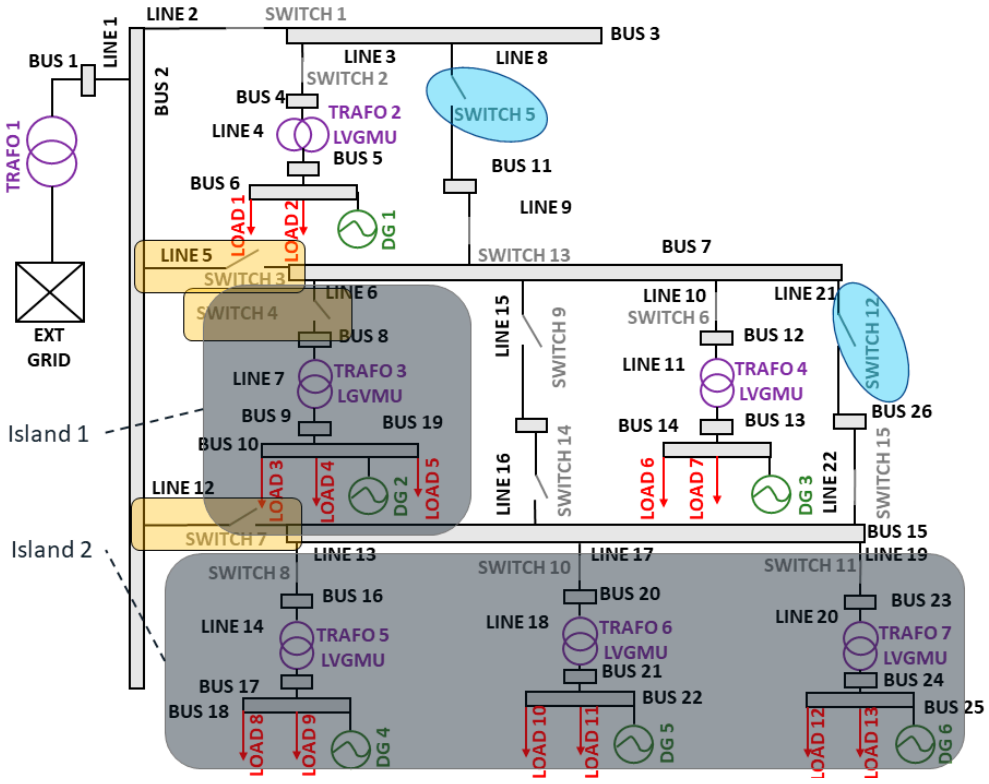

Figure 17. Intentional islanding configurations.

Cascading failures comprise a critical challenge in supply networks such as electric power grids, as they can lead to major economical and functional issues. The above supervision/control scheme can be used to address cascading failures and to increase the network resilience and stability [90,91].

3.4. Discussion and Remarks

Regarding hardware implementation of the system, we demonstrated that the OpenEdgePMU system has a modular realization, compact size, and very low cost, taking into consideration the processing power available for edge processing (DSP + MCU). By providing open-source licensing for both hardware and software, we encourage tailoring, modifications, and use of the platform in future smart grid projects.

Targeting to encourage adoption of the platform in this direction, we presented two funded projects, that have adopted and tailored the design according to projects' requirements. The scope was just to outline the application cases and demonstrate the suitability of the OpenEdgePMU system in similar research and innovation actions. Further details regarding these projects will be published in the future, as, currently, both projects are in pilot execution phases. In similar project circumstances, researchers would have to tailor the main board of the system and provide potential extensions that are required specifically for a project's objectives and needs. In contrast, the DSP module and the edge processing module are ready to be used without any modifications; thus, significantly reducing required project labor, cost, and risks.

Finally, we have presented selected algorithms and applications able to be executed according to the edge processing approach. Again, the scope of the description was to provide an outline of the application cases and to demonstrate the applicability of the OpenEdgePMU system. All application cases presented are related to future resilient smart grids where the major requirements are to analyze critical situations inside the grid, make fast decisions, and execute decisions in a fully automated manner; thus, the overall processing, communication, and actions' timing latency are minimized.

Comparing the OpenEdgePMU approach with all current experimental, academic, and open-source PMUs, we have demonstrated a more mature approach that can easily comply with strict industrial requirements, with a very expandable platform, both regarding software and hardware, in a very low-cost configuration.

4. Conclusions

The OpenEdgePMU system is an open innovation initiative [92] that is being used by an industrial and an academic partner in Greece, EMTECH, and the University of West Attica. Under open-source licenses, both for software and hardware components, the OpenEdgePMU system provides state-of-the-art knowledge, designs, and developments for the very interesting application field of phasor measurement units in power distribution grids.

Our motivation and contribution to the research and innovation communities is summarized as follows:

- Provide state-of-the art, low-cost hardware and software that can process future edge processing requirements in energy smart grids;
- Lower technology entry barriers, specially regarding academic and SMEs/startups;
- Fuse cross-disciplinary knowledge that potentially may come from further contributors;
- Investigate alternative business models and market opportunities in future resilient smart distribution grids;
- Promote open innovation with collaboration between SMEs and academia.

For dissemination and further application purposes, we have created an OpenEdgePMU portal [93] where all interested parties and individuals are very welcome to register and participate in our initiative. The software is also available in gitlab [94].

Our future actions are targeted to provide scientific papers with more detailed performance evaluations on specific applications of the OpenEdgePMU architecture. Our team is always available to support further research and innovation actions by tailoring both the hardware and the firmware for specific application needs. In addition, we intend to link our initiative with other open-source tools and applications (e.g., OpenPDC [95], PandaPower [49], Open61850 [96], etc.) in order to further support innovation in future resilient smart grids.

Author Contributions: Conceptualization, N.-A.I.L. and E.N.Z.; literature study, N.-A.I.L.; methodology, N.-A.I.L., S.H., V.A., N.G. and E.N.Z.; software, N.-A.I.L., S.H., V.A. and N.G.; validation, N.G., E.N.Z. and N.-A.I.L.; formal analysis, N.-A.I.L. and E.N.Z.; investigation, N.-A.I.L., C.S.P. and E.N.Z.; writing—original draft preparation, N.-A.I.L., V.A., S.H. and N.G.; writing—review and editing, N.G., E.N.Z. and N.-A.I.L.; supervision, E.N.Z.; project administration, N.-A.I.L. and E.N.Z.; funding acquisition, N.-A.I.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH—CREATE—INNOVATE (project code/grant agreement T2EDK-02497), and through the European Union's Horizon 2020 research and innovation programme, under grant agreement No. 864283, eBalance+ project.

Data Availability Statement: The data are not publicly available due to confidentiality and privacy reasons.

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

References

1. Wache, M.; Murray, D.C. Application of Synchrophasor Measurements for distribution networks. In Proceedings of the 2011 IEEE Power and Energy Society General Meeting, Detroit, MI, USA, 24–29 July 2011.
2. Hojabri, M.; Dersch, U.; Papaemmanouil, A.; Bosshart, P. A Comprehensive Survey on Phasor Measurement Unit Applications in Distribution Systems. *Energies* **2019**, *12*, 4552. [CrossRef]

3. Stewart, E.M.; Kiliccote, S.; Shand, C.M.; McMorran, A.W.; Arghandeh, R.; von Meier, A. Addressing the challenges for integrating micro-synchrophasor data with operational system applications. In Proceedings of the 2014 IEEE PES General Meeting, Conference & Exposition, National Harbor, MD, USA, 27–31 July 2014.
4. Dusabimana, E.; Yoon, S.-G. A Survey on the Micro-Phasor Measurement Unit in Distribution Networks. *Electronics* **2020**, *9*, 305. [CrossRef]
5. Phadke, A.G.; Thorp, J.S. History and applications of phasor measurements. In Proceedings of the 2006 IEEE PES Power Systems Conference and Exposition, Atlanta, GA, USA, 29 October–1 November 2006; pp. 331–335.
6. von Meier, A.; Culler, D.; McEachern, A.; Arghandeh, R. Microsynchrophasors for distribution system. In Proceedings of the 5th IEEE PES Innovative Smart Grid Technologies (ISGT) European 2014 Conference, Washington, DC, USA, 19–22 February 2014; pp. 1–5.
7. Rietveld, G.; Jongepier, A.; Seters, J.V.; Visser, M.; Liu, P.; Acanski, M.; Hoogenboom, D.; Brom, H.E.V.D. Application of PMUs for Monitoring a 50 kV Distribution Grid. In Proceedings of the CIRED Conference, Lyon, France, 15–18 June 2015; pp. 1–5.
8. Carta, A.; Locci, N.; Muscas, C.; Sulis, S. A flexible GPS-based system for synchronized phasor measurement in electric distribution networks. *IEEE Trans. Instrum. Meas.* **2008**, *57*, 2450–2456. [CrossRef]
9. Laverty, D.M.; Best, R.J.; Brogan, P.; Al Khatib, I.; Vanfretti, L.; Morrow, D.J. The OpenPMU Platform for Open-Source Phasor Measurements. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 701–709. [CrossRef]
10. *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*; IEEE Standard for Synchrophasor Measurements for Power Systems. IEEE Standards Association: New York, NY, USA, 2011; pp. 1–61.
11. *IEEE Std C37.118.1a-2014 (Amendment to IEEE Std C37.118.1-2011)*; IEEE Standard for Synchrophasor Measurements for Power Systems—Amendment 1: Modification of Selected Performance Requirements. IEEE Standards Association: New York, NY, USA, 2014; pp. 1–25.
12. The Open Source Initiative OSI—The BSE License. Available online: http://www.opensource.orge/licenses/bsd-license.php (accessed on 12 February 2023).
13. GNU General Public License. Available online: http://www.opensource.org/licenses/bsd-license.php (accessed on 12 February 2023).
14. Open Hardware Licenses, The Foundation for P2P Alternatives. Available online: http://p2pfoundation.net/Open_Hadware_licenses (accessed on 12 February 2023).
15. The TAPR Open Hardware License. Available online: http://www.tapr.org/ohl.html (accessed on 12 February 2023).
16. GridTrak—Home Page. Available online: http://www.gridtrak.com (accessed on 12 February 2023).
17. He, Z.; Liao, Y. The design of analog acquisition system in distribution automation. In Proceedings of the 2012 China International Conference on Electricity Distribution, Venice, Italy, 9–12 September 2012; pp. 1–4.
18. Du, L.; Huang, J.K.; Liu, Q.Y. A realisation of measurement unit for phasor measurement unit based on DSP. In Proceedings of the Asia-Pacific Power and Energy Engineering Conference (APPEEC), Shanghai, China, 27–29 March 2012.
19. Romano, P.; Paolone, M.; Chau, T.; Jeppesen, B.; Ahmed, E.A. A high-performance, low-cost PMU prototype for distribution networks based on FPGA. In Proceedings of the 2017 IEEE Manchester PowerTech, Manchester, UK, 18–22 June 2017.
20. Pinte, B.; Quinlan, M.; Reinhard, K. Low voltage micro-phasor measurement unit (μPMU). In Proceedings of the 2015 IEEE Power and Energy Conference at Illinois (PECI), 20–21 February 2015.
21. Rodrigues, R.N.; Cavalcante, L.H.; Zatta, J.K.; Schlichting, L.C.M. A Phasor Measurement Unit based on discrete fourier transform using digital signal processor. In Proceedings of the 2016 12th IEEE International Conference on Industry Applications (INDUSCON), Curitiba, Brazil, 20–23 November 2016; pp. 1–6.
22. Discovery Ket with STM32F407VG MCU. Available online: http://www.st.com/en/evaluation-tools/stm32f4discovery.html (accessed on 12 February 2023).
23. Das, H.P.; Pradhan, A.K. Development of a micro-phasor measurement unit for distribution system applications. In Proceedings of the National Power Systems Conference (NPSC), Bhubaneswar, India, 19–21 December 2016; pp. 1–5.
24. Bhatti, T.A.; Raheem, A.; Alam, T.; Malik, M.O.; Munir, A. Implementation of low cost non-DFT based Phasor Measurement Unit for 50 Hz power system. In Proceedings of the 2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), Quetta, Pakistan, 11–12 April 2016; pp. 120–125.
25. Angioni, A.; Lipari, G.; Pau, M.; Ponci, F.; Monti, A. A Low Cost PMU to Monitor Distribution Grids. In Proceedings of the AMPS 2017—IEEE International Workshop on Applied Measurements for Power Systems, Liverpool, UK, 20–22 September 2017.
26. Zhao, X.; Laverty, D.M.; McKernan, A.; Morrow, D.J.; McLaughlin, K.; Sezer, S. GPS-Disciplined Analog-to-Digital Converter for Phasor Measurement Applications. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2349–2357. [CrossRef]
27. Cetina, R.Q.; Roscoe, A.J.; Atoche, A.C. Low-cost power systems metrology laboratory based on raspberry Pi. In Proceedings of the 1st International Colloquium on Smart Grid Metrology (SmaGriMet), Split, Croatia, 24–27 April 2018; pp. 1–2.
28. Schofield, D.; Gonzalez-Longatt, F.; Bogdanov, D. Design and Implementation of a Low-Cost Phasor Measurement Unit: A Comprehensive Review. In Proceedings of the 2018 Seventh Balkan Conference on Lighting (BalkanLight), Varna, Bulgaria, 20–22 September 2018; pp. 1–6.
29. Bucci, G.; Ciancetta, F.; D'Innocenzo, F.; Fiorucci, E.; Ometto, A. Development of a low cost power meter based on a digital signal controller. *Int. J. Emerg. Electr. Power Syst.* **2018**, *19*, 20170280. [CrossRef]

30. Femine, A.D.; Gallo, D.; Landi, C.; Luiso, M. A design approach for a low cost phasor measurement unit. In Proceedings of the I2MTC 2019—2019 IEEE International Instrumentation and Measurement Technology Conference, Auckland, New Zealand, 20–23 May 2019.
31. Seger, P.V.H.; Moreto, M.; Grando, F.L.; Lazzaretti, A.E.; Denardin, G.W.; Pastro, C.R. Monitoring Low Voltage Electrical Networks Using Low Cost PMUs: A Case Study. In Proceedings of the 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I and CPS Europe), Madrid, Spain, 9–12 June 2020.
32. Shankar, G.; Ray, S.; Goyal, S.R.; Srinivas, R. Development of a Low Cost Phasor Measurement Unit for Laboratory Studies. In Proceedings of the 2021 1st International Conference on Power Electronics and Energy (ICPEE), Bhubaneswar, India, ,.
33. Raju, V.V.R.; Shree, K.H.P.; Kumar, S.V.J. Development of cost-effective phasor measurement unit for wide area monitoring system applications. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 4731–4739. [CrossRef]
34. Schofield, D.; Mohapatra, D.; Chamorro, H.R.; Roldan-Fernandez, J.M.; Abdellah, K.; Gonzalez-Longatt, F. Design and Implementation of Low-Cost Phasor Measurement Unit: PhasorsCatcher. *Energies* **2022**, *15*, 3172. [CrossRef]
35. Mohapatra, D. Development and Hardware Implementation of a Phasor Measurement Unit Using Microcontroller. NIT Rourkela Ed. Available online: http://ethesis.nitrkl.ac.in/7621/1/2015_MT_Development_Debasis_mohapatra.pdf (accessed on 20 March 2022).
36. Sampson, O.V. Construction of a Phasor Measurement Unit (PMU) for Power System Applications. Ph.D. Thesis, University of Manitoba, Winnipeg, MB, Canada, 2015.
37. Culliss, J.A. A 3rd Generation Frequency Disturbance Recorder: A Secure, Low Cost Synchophasor Measurement Device. Ph.D. Thesis, Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA, 2015.
38. Raspberry Pi. Available online: https://www.raspberrypi.org (accessed on 12 February 2023).
39. Beagle Bone Black. Available online: https://beagleboard.org/black (accessed on 12 February 2023).
40. Marcus, Y.; Silverstein, A. *Factors Affecting PMU Installation Costs*; Tech. Rep.; Office of Electricity Delivery and Energy Reliability, U.S. Department of Energy: Houston, TX, USA, 2014.
41. TMDSEVM5517—C5517 Evaluation Module (EVM). Available online: https://www.ti.com/tool/TMDSEVM5517 (accessed on 12 February 2023).
42. Texas Instruments. TMS320C5517 Fixed-Point Digital Signal Processor, Data Sheet, SPRS727C. Available online: https://www.ti.com/lit/ds/symlink/tms320c5517.pdf (accessed on 12 February 2023).
43. Micron SDRAM MT48H32M16LF—Data Sheet. Available online: https://media-www.micron.com/-/media/client/global/documents/products/data-sheet/dram/mobile-dram/low-power-dram/lpsdr/y67m_512mb_mobile_lpsdr_at.pdf (accessed on 12 February 2023).
44. Infineon FLASH S29GL128S11DHIV20—Data Sheet. Available online: https://www.infineon.com/dgdl/Infineon-S29GL01GS_S29GL512S_S29GL256S_S29GL128S_128_Mb_256_Mb_512_Mb_1_Gb_GL-S_MIRRORBIT_TM_Flash_Parallel_3-DataSheet-v20_00-EN.pdf (accessed on 12 February 2023).
45. SanDisk Extreme®microSD™ UHS-I Card, Data Sheet. Available online: https://documents.westerndigital.com/content/dam/doc-library/en_us/assets/public/sandisk/product/memory-cards/extreme-uhs-i-microsd/data-sheet-extreme-uhs-i-microsd.pdf (accessed on 12 February 2023).
46. Texas Instruments. ADS8586S 16-Bit, High-Speed, 6-Channel, Simultaneous-Sampling ADC with Bipolar Inputs on a Single Supply, Data Sheet, SBAS833A. Available online: https://www.ti.com/lit/ds/symlink/ads8586s.pdf (accessed on 12 February 2023).
47. Xmodem Protocol. Available online: https://en.wikipedia.org/wiki/XMODEM (accessed on 12 February 2023).
48. Octavo Systems; Rev, I.N.D. 2022. Available online: https://octavosystems.com/octavosystems.com/wp-content/uploads/2017/08/OSD335x-SM-Datasheet.pdf (accessed on 12 February 2023).
49. Panda Power. Available online: www.pandapower.org (accessed on 12 February 2023).
50. Pandas: A Numerical and Data Analysis Library. Available online: https://pydata.org (accessed on 12 February 2023).
51. Scikit-Learn: An AI Construction and Execution Library. Available online: https://scikit-learn.org/stable/index.html (accessed on 12 February 2023).
52. Flask: A Library for the Development of HTTP-REST Interfaces. Available online: https://flask.palletsprojects.com/en/2.2.x/ (accessed on 12 February 2023).
53. SQLite: A Library for the Interaction with the Local SQLite Database. Available online: https://www.sqlite.org/index.html (accessed on 12 February 2023).
54. Lantronix, GPS/GLONASS Receiver Module A5100-A, Data Sheet. Available online: https://cdn.lantronix.com/wp-content/uploads/pdf/A5100_UG.pdf (accessed on 12 February 2023).
55. Adafruit Ultimate GPS, Data Sheet. Available online: https://cdn-shop.adafruit.com/datasheets/PA6B-Datasheet-A07.pdf (accessed on 12 February 2023).
56. eBalance+ European Union Project. Available online: https://www.ebalanceplus.eu (accessed on 12 February 2023).
57. Techoreymeta Greek National Project. Available online: https://tehnorevmeta.emtech.global/ (accessed on 12 February 2023).
58. Vaccaro, A.; Canizares, C. A knowledge-based framework for power flow and optimal power flow analyses. In Proceedings of the 2017 IEEE Power & Energy Society General Meeting, Chicago, IL, USA, 16–20 July 2017; p. 1. [CrossRef]

59. Thurner, L.; Scheidler, A.; Probst, A.; Braun, M. Heuristic optimization for network restoration and expansion in compliance with the single contingency policy. *IET Gener. Transm. Distrib.* **2017**, *11*, 4264–4273. [CrossRef]
60. Scheidler, A.; Thurner, L.; Braun, M. Heuristic optimisation for automated distribution system planning in network integration studies. *IET Renew. Power Gener.* **2018**, *12*, 530–538. [CrossRef]
61. Menke, J.-H.; Hegemann, J.; Gehler, S.; Braun, M. Heuristic monitoring method for sparsely measured distribution grids. *Int. J. Electr. Power Energy Syst.* **2018**, *95*, 146–155. [CrossRef]
62. Wang, H.; Kraiczy, M.; von Berg, S.W.-V.B.S.W.; Kämpf, E.; Ernst, B.; Schmidt, S.; Wirtz, F.; Braun, M. Reactive power coordination on strategies with distributed generators in distribution networks. In Proceedings of the 1st International Conference on Large-Scale Grid Integration of Renewable Energy in India, Delhi, India, 6–8 September 2017.
63. Frank, S.; Steponavice, I.; Rebennack, S. Optimal power flow: A bibliographic survey I. *Energy Syst.* **2012**, *3*, 221–258. [CrossRef]
64. Frank, S.; Rebennack, S. *A Primer on Optimal Power Flow: Theory, Formulation, and Practical Examples*; Colorado School of Mines: Golden, CO, USA, 2012; pp. 1–42.
65. Abido, M.A. Optimal Power Flow Using Particle Swarm Optimization. *Electr. Power Energy Syst.* **2002**, *24*, 563–571. [CrossRef]
66. Osman, M.S.; Abo-Sinna, M.A.; Mousa, A.A. A solution to the optimal power flow using genetic algorithm. *Appl. Math. Comp.* **2004**, *155*, 391–405. [CrossRef]
67. Papadimitrakis, M.; Giamarelos, N.; Stogiannos, M.; Zois, E.N.; Livanos, N.A.I.; Alexandridis, A. Metaheuristic search in smart grid: A review with emphasis on planning, scheduling and power flow optimization applications. *Renew. Sustain. Energy Rev.* **2021**, *145*, 111072. [CrossRef]
68. Mittal, U.; Nangia, U.; Jain, N.K. Computational Intelligence-Based Optimal Power Flow Methods-A Review. In Proceedings of the 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India, 11–13 February 2022; pp. 1–8. [CrossRef]
69. Maskar, M.B.; Thorat, A.R.; Korachgaon, I. A review on optimal power flow problem and solution methodologies. In Proceedings of the 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), Pune, India, 24–26 February 2017; pp. 64–70. [CrossRef]
70. Theodoridis, S. *Machine Learning: A Bayesian and Optimization Perspective*; Academic Press: Cambridge, MA, USA, 2015.
71. Mamun, A.A.; Sohel, M.; Mohammad, N.; Sunny, M.S.H.; Dipta, D.R.; Hossain, E. A Comprehensive Review of the Load Forecasting Techniques Using Single and Hybrid Predictive Models. *IEEE Access* **2020**, *8*, 134911–134939. [CrossRef]
72. Sun, X.; Ouyang, Z.; Yue, D. Short-term load forecasting based on multivariate linear regression. In Proceedings of the 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 26–28 November 2017; pp. 1–5. [CrossRef]
73. Azad, S.A.; Ali, A.B.M.S.; Wolfs, P. Daily average load forecasting using dynamic linear regression. In Proceedings of the Asia-Pacific World Congress on Computer Science and Engineering, Nadi, Fiji, 4–5 November 2014; pp. 1–7. [CrossRef]
74. Azad, M.K.; Uddin, S.; Takruri, M. Support vector regression based electricity peak load forecasting. In Proceedings of the 11th International Symposium on Mechatronics and Its Applications (ISMA), Sharjah, United Arab Emirates, 4–6 March 2018; pp. 1–5. [CrossRef]
75. Chen, W.; Chen, X.; Liao, Y.; Wang, G.; Yao, J.; Chen, K. Short-term load forecasting based on time series reconstruction and support vector regression. In Proceedings of the 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013), Xi'an, China, 22–25 October 2013; pp. 1–4. [CrossRef]
76. Chowdhury, D.; Sarkar, M.; Haider, M.Z.; Alam, T. Zone Wise Hourly Load Prediction Using Regression Decision Tree Model. In Proceedings of the 2018 International Conference on Innovation in Engineering and Technology (ICIET), Dhaka, Bangladesh, 27–28 December 2018; pp. 1–6. [CrossRef]
77. Lahouar, A.; Slama, J.B.H. Random forests model for one day ahead load forecasting. In Proceedings of the IREC2015 the Sixth International Renewable Energy Congress, Sousse, Tunisia, 24–26 March 2015; pp. 1–6. [CrossRef]
78. Singh, S.; Hussain, S.; Bazaz, M.A. Short term load forecasting using artificial neural network. In Proceedings of the 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, India, 21–23 December 2017; pp. 1–5. [CrossRef]
79. Akarslan, E.; Hocaoglu, F.O. Electricity demand forecasting of a micro grid using ANN. In Proceedings of the 2018 9th International Renewable Energy Congress (IREC), Hammamet, Tunisia, 20–22 March 2018; pp. 1–5. [CrossRef]
80. Zheng, J.; Xu, C.; Zhang, Z.; Li, X. Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network. In Proceedings of the 2017 51st Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 22–24 March 2017; pp. 1–6. [CrossRef]
81. Giamarelos, N.; Zois, E.N.; Papadimitrakis, M.; Stogiannos, M.; Livanos, N.-A.I.; Alexandridis, A. Short-Term Electric Load Forecasting With Sparse Coding Methods. *IEEE Accesss* **2021**, *9*, 102847–102861. [CrossRef]
82. Deng, Z.; Wang, B.; Xu, Y.; Xu, T.; Liu, C.; Zhu, Z. Multi-Scale Convolutional Neural Network With Time-Cognition for Multi-Step Short-Term Load Forecasting. *IEEE Access* **2019**, *7*, 88058–88071. [CrossRef]
83. Mi, Y.; Chen, Y.; Yuan, M.; Li, Z.; Tao, B.; Han, Y. Multi-Timescale Optimal Dispatching Strategy for Coordinated Source-Grid-Load-Storage Interaction in Active Distribution Networks Based on Second-Order Cone Planning. *Energies* **2023**, *16*, 1356. [CrossRef]
84. Daneshi, H.; Daneshi, A. Price forecasting in deregulated electricity markets—A bibliographical survey. In Proceedings of the 2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, Nanjing, China, 6–9 April 2008; pp. 657–661. [CrossRef]

85. Wu, L.; Shahidehpour, M.; Li, T. Stochastic Security-Constrained Unit Commitment. *IEEE Trans. Power Syst.* **2007**, *22*, 800–811. [CrossRef]
86. Ooi, B.T.; Mcgillis, D.; Galiana, F.D.; Marceau, R. The Potential of Distributed Generation to Provide Ancillary Services. In Proceedings of the 2000 Power Engineering Society Summer Meeting (Cat. No.00CH37134), Seattle, WA, USA, 16–20 July 2000; pp. 1762–1767.
87. Lalor, G.; Mullane, A.; O'Malley, M. Frequency control and wind turbine technologies. *IEEE Trans. Power Syst.* **2005**, *20*, 1905–1913. [CrossRef]
88. Brenna, M.; Foiadelli, F.; Petroni, P.; Sapienza, G.; Zaninelli, D. Distributed generation regulation for intentional islanding in Smart Grids. In Proceedings of the 2012 IEEE PES Innovative Smart Grid Technologies (ISGT), Washington, DC, USA, 16–20 January 2012; pp. 1–6. [CrossRef]
89. Corti, M.; Superti-Furga, G.; Tironi, E.; Pavan, A.M.; Sulligoi, G. Intentional islanding in medium voltage distribution grids. In Proceedings of the 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 7–10 June 2016; pp. 1–6. [CrossRef]
90. Conti, S.; Rizzo, S.A. Monte Carlo Simulation by Using a Systematic Approach to Assess Distribution System Reliability Considering Intentional Islanding. *IEEE Trans. Power Deliv.* **2015**, *30*, 64–73. [CrossRef]
91. Oboudi, M.H.; Mohammadi, M.; Rastegar, M. Resilience-oriented intentional islanding of reconfigurable distribution power systems. *J. Mod. Power Syst. Clean Energy* **2019**, *7*, 741–752. [CrossRef]
92. Mantas, V.; Soderquist, K.E. Open Innovation: Activating the Entrepreneurial Mindset. In Proceedings of the 5th European Conference on Innovation and Entrepreneurship, Athens, Greece, 16–17 September 2010.
93. OpenEdgePMU Portal. Available online: https://oepmu.emtech.global (accessed on 12 February 2023).
94. OpenEdgePMU Gitlab. Available online: https://gitlab.com/oepmu/open-edge-pmu (accessed on 12 February 2023).
95. "Open Source Phasor Data Concentrator" Grid Protection Alliance. Available online: https://github.com/GridProtectionAlliance/openPDC (accessed on 12 February 2023).
96. OpenIEC61850. Available online: www.openmuc.org/iec-61850 (accessed on 12 February 2023).