

Article

# Cloud Based IoT Solution for Fault Detection and Localization in Power Distribution Systems

Mussawir Ul Mehmood <sup>1</sup>, Abasin Ulasyar <sup>1,\*</sup>, Abraiz Khattak <sup>1</sup>, Kashif Imran <sup>1</sup>,  
Haris Sheh Zad <sup>2</sup> and Shibli Nisar <sup>3</sup>

<sup>1</sup> Department of Electrical Power Engineering, USPCAS-E, National University of Sciences and Technology, Islamabad 44000, Pakistan; musawir8012@gmail.com (M.U.M.); abraiz@uspcase.nust.edu.pk (A.K.); kashifimran@uspcase.nust.edu.pk (K.I.)

<sup>2</sup> Department of Electrical Engineering, Riphah International University, Islamabad 44000, Pakistan; haris.shehzad@riphah.edu.pk

<sup>3</sup> Military College of Signals (MCS), National University of Sciences and Technology, Islamabad 44000, Pakistan; shiblinisar@mcs.edu.pk

\* Correspondence: abasin@uspcase.nust.edu.pk

Received: 2 April 2020; Accepted: 23 April 2020; Published: 26 May 2020



**Abstract:** Power restoring time in power distribution systems (PDS) can be minimized by using efficient fault localization techniques. This paper proposes a novel, robust and scalable cloud based internet of things (IoT) solution for identification and localization of faults in PDS. For this purpose, a new algorithm is developed that can detect single and multiple simultaneous faults in the presence of single and multiple device or sensor failures. The algorithm has utilized a zone based approach that divides a PDS into different zones. A current sensing device (CSD) was deployed at the boundary of a zone. The function of CSD is to provide time synchronized current measurements and communicate with a cloud server through an edge device (ED). Another contribution of this research work is the unique implementation of context aware policy (CAP) in ED. Due to CAP, only those measurements are transmitted to cloud server that differ from the previously transmitted measurements. The cloud server performed calculations at regular intervals to detect faults in PDS. A relational database model was utilized to log various fault events that occur in PDS. An IEEE 37 node test feeder was selected as PDS to observe the performance of our solution. Two test cases were designed to simulate individual and multiple simultaneous faults in PDS. A third test case was implemented to demonstrate the robustness and scalability of proposed solution to detect multiple simultaneous faults in PDS when single and multiple sensor failures were encountered. It was observed that the new algorithm successfully localized the faults for all the three cases. Consequently, significant reductions were noticed in the amount of data that was sent to the cloud server. In the end, a comparison study of a proposed solution was performed with existing methods to further highlight the benefits of our technique.

**Keywords:** IoT; cloud computing; fault localization; power distribution systems; MQTT; fault identification; edge intelligence

## 1. Introduction

Due to recent advancements in smart grid technology, the installation of smart devices such as; smart meter (SM) and phasor measurement unit (PMU) have gained significant importance [1]. An information technology (IT) architecture is used to monitor and control distributed energy systems [2]. The function of the IT architecture is mainly dependent upon the data received from SMs and other smart home technologies. Different smart devices can be deployed in a power distribution system

(PDS) to measure phasor voltages and currents. However, if the number of nodes in a particular PDS increases, then utilization of smart devices becomes expensive. Considerable work has been carried out in terms of optimal placement and utilization of smart devices along with their communication infrastructure [3,4]. Maximum fault location observability was achieved by using a PMU placement technique [5]. In some cases, synchronized measurement units may be located at large distances from each other. For this purpose, a wide area monitoring system (WAMS) is used to connect sparsely located PMUs and SMs. Both the wired and wireless communication systems are utilized for WAMS in power systems [6,7]. However, due to expansion of a power system, different time parameters are also affected including: (i) transmission time, (ii) data handling time and (iii) processing time of generated data [8]. Additionally, different issues related with big data introduce new challenges that need considerable attention [9]. Therefore, a robust and scalable cloud based information and communication technology (ICT) infrastructure is required for power systems that can effectively address the problems of data handling and its processing. A fog computing layer was implemented in a cloud based architecture that increased the data handling capabilities [10]. Moreover, it reduced data storage requirement and power consumption. Recently, internet of things (IoT) has emerged as a key solution for various problems that are experienced in a smart grid environment [11,12]. An IPv6 based conceptual IoT model was implemented for a smart grid that provides a larger address space for connecting large number of devices [13]. According to this model, each device was assigned a unique IP address. An IoT based energy management system was designed for a standalone photovoltaic (SAPV) system that handled load shedding, source management, data acquisition and control of the SAPV networks [14]. An IoT solution was applied for the detection of glaze ice on high voltage (HV) and very high voltage (VHV) transmission lines [15]. On the other side, the cloud computing can provide different means of communication for large amounts of data that are generated by the IoT devices. Nowadays, various cloud service brokers are helping their users in the selection of desired cloud services [16]. The combination of IoT with cloud computing fulfills the needs of scalability and adaptability for smart grid infrastructure. However, this combination faces several challenges that include security, cost and privacy [17,18]. A detailed survey was performed on the role of cloud computing resources in engineering applications [19].

Generally, a PDS is not rigorously monitored due to its large size and complex nature that involves thousands of nodes, tapped lines and distribution transformers. The ability of a PDS to deliver continuous power at the user end without interruptions depends upon fast identification, localization and isolation of faults. Traditionally, the fault detection methods can be divided into three categories: (i) traveling wave method; (ii) high frequency measurement based methods and (iii) phasor measurement based methods. The travelling wave method relies on the transients generated by the faults. A fault is detected by measuring the arrival time of a traveling wave [20,21]. This can be further divided into two types; single-ended method [20] and double-ended method [21]. Unfortunately, the traveling wave method requires very specific measurements from complex devices (digital fault recorders) that are not common in distribution systems. Therefore, this method is most commonly used in transmission lines where digital fault recorders are only installed at critical substations. Furthermore, the method is not suitable for the PDS with complex features like tapped lines, distribution transformers, thousands of nodes and short lines with different values of resistances and inductances. In comparison, the high frequency measurement based methods make use of the high frequency components that are present in current and voltage signals. This approach can be divided into time domain and frequency domain methods. This type of fault detection is usually assisted by classification techniques such as artificial neural networks [22,23], fuzzy logic [24] and support vector machine methods [25]. Therefore, these methods can be categorized as knowledge based methods where large set of training data is required for their successful implementation. The data may contain information about the loads, fault locations and previous registered faults which is difficult to obtain in today's rapidly growing distribution systems. Nevertheless, phasor measurement based methods are most commonly used in PDS because they need less measurement data. These

methods usually estimate fault location as a function of line impedance by utilizing the current and voltage phasor measurements at one or both ends of power lines [26,27]. The double-ended methods use synchronized measurements that are more accurate than single-ended methods [28]. However, the available literature regarding these methods does not take into account the scalability requirements of a PDS.

A scalable infrastructure is desired that can simplify the addition of new devices to the network. This can be achieved by implementing a centralized scheme based on ICT infrastructure in a smart grid environment. In this article, a more generalized form of phasor measurement based method is adopted that can use the magnitudes of current phasors to detect and localize the faults in PDS. The method divides PDS into different zones. Accuracy of the method depends on the number of zones. As the number of zones are increased, the area covered by each zone becomes smaller and the faults are localized with high accuracy. Moreover, faults are detected by identifying only the faulty zones [29]. The scheme has low requirement of current measurements that can be provided by SMs or PMUs in PDS. Due to simplified nature of this method, it is combined with the IoT and cloud infrastructure that makes the scheme scalable for large networks [30]. The overall solution is built on important virtualization features of IoT [31]. It achieves detection of multiple simultaneous faults which was not feasible by using aforementioned fault detection methods. Since the installed devices in a large PDS generate huge amounts of data, edge intelligence has been introduced to filter and limit the amount of data being transferred over the internet which is a main characteristic of cloud based systems. The new solution can be applied in a particular smart grid environment where the PMUs are unavailable or the existence of WAMS infrastructure is limited. The performance of proposed solution is compared with the existing literature to further highlight the novelty and contribution of this research work. The current measurements at different buses were used to detect faulty zones but this was done locally by using local measurements between two immediate buses [32]. The information about the faulty zone was not communicated to a centralized location due to the absence of communication infrastructure [32]. As compared to [32], our scheme utilizes a centralized cloud based approach to detect faulty zones and the information is quickly communicated to relevant personnel at a centralized location. Additionally, no strategy was proposed to handle device failures which is another feature of our work. In [32], faults are detected by using artificial neural network (ANN) assisted wavelet transform method but this requires a specialized device that can take measurements at very high frequency. These devices need to be installed at each bus for the accurate localization of faults which is not practically feasible for distribution systems. Another disadvantage of [32] is that five separate ANNs were used to detect five different types of faults. Training of these ANNs requires large sets of training data which is difficult to obtain in ever growing PDS as discussed above. In addition to this, a wireless sensor network (WSN) based monitoring and fault detection scheme was realized for a medium-voltage power line [33]. This technique used synchronized measurements from wireless sensors. The fault data is communicated to a centralized location through edge devices for monitoring and storage. However, ref. [33] does not utilize any cloud services or edge intelligence to limit the amount of data sent over the network. In contrast, we have proposed a cloud based solution that uses edge intelligence which can lead to reduced infrastructure costs. Furthermore, ref. [33] uses highly specialized fault locators or wireless sensors. Installation of such devices for large distribution system can prove to be expensive. A new concept of edge mesh for distributed intelligence was introduced in [34]. This concept allows edge devices to perform computations and make decisions on edge. Therefore, the data required for continuous computations is no longer sent to the cloud. Cloud server performs the task of centralized monitoring and storage only. Therefore, the method provides the benefits of low latency and better scalability. This method along with our proposed fault localization scheme can be used in PDS. However, since the processing takes place in the edge device, more powerful edge devices will be required. In addition, more complex collaboration and networking schemes will be needed for the detection of faults in PDS. Ref. [35] proposed a ZigBee based WSN for its application in smart grid environment. The ZigBee devices or wireless sensors form a mesh network

similar to [34]. These devices can communicate with each other as well as to the WSN gateway. The WSN gateway provides the internet access to the wireless sensors. All communication with cloud is performed through WSN gateway. There are three main ways to connect WSN to the internet [36]: (i) independent network, (ii) hybrid network and (iii) access point network. Independent networks are most economical because they provide single WSN gateway between WSN and cloud. In case this gateway fails, all communication to the cloud is blocked. Therefore, this approach is not suitable for the application of fault detection. Hybrid and access point networks are more suited for fault detection because they provide more than one WSN gateways. Keeping in view all the above discussions, the main contributions of this research work are as follows.

- A centralized cloud based IoT solution is developed that can detect individual and multiple simultaneous faults in power distribution systems.
- The edge intelligence is implemented by using CAP that has significantly reduced the amount of data transfer to the cloud.
- The scheme has a novel feature to detect multiple simultaneous faults in the presence of single or multiple device failures in large and complex power networks that show the robustness and scalability of the solution.
- A detailed comparison of the proposed method is performed with existing fault detection methods and other smart solutions that further highlights the benefits and effectiveness of our technique.

The rest of the paper is organized as follows. Section 2 provides details about the algorithm to detect faults in PDS. Section 3 gives the overview of proposed IoT framework. The results and discussion are presented in Section 4. The conclusion of our work is provided in Section 5.

## 2. Fault Identification and Localization

The method that is used in our work for fault identification and localization is based upon the strategy used in [29]. The method divides PDS into different zones, where total number of zones is equal to  $P$ . The current measurements are taken at the boundary of a zone, where the total number of measurements is equal to  $Q$ . According to the proposed technique, the first step is to form a current identification matrix (CIM) with dimensions of  $P \times Q$  as given by Equation (1).

$$CIM = \begin{bmatrix} N_{11} & N_{12} & \cdots & N_{1Q} \\ N_{21} & N_{22} & \cdots & N_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ N_{P1} & N_{P2} & \cdots & N_{PQ} \end{bmatrix} \quad (1)$$

$$\text{Let } N_{kl} \in CIM$$

The next step is to define the values for  $N_{kl}$  by using following rules.

- If  $l^{th}$  current flows into  $k^{th}$  zone then set  $N_{kl}$  to 1.
- If  $l^{th}$  current flows out of  $k^{th}$  zone then set  $N_{kl}$  to -1.
- Otherwise set  $N_{kl}$  to 0.

In the third step, define a current vector  $I$  with  $Q$  number of elements as shown in Equation (2). The matrix  $B$  is calculated after multiplication of CIM and  $I$  as given by Equation (3).

$$I = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ \vdots \\ I_Q \end{bmatrix} \quad (2)$$

$$B = CIM \times I \quad (3)$$

The zone identification vector  $ZIV$  is determined by taking the magnitude of  $B$  as shown by Equation (4).  $ZIV$  is a column vector with dimensions of  $P \times 1$ . In addition to this, the per phase vector is represented as  $ZIV_p$  in Equation (5) where  $I_p$  represents individual phase currents for phase a,b and c and has dimensions of  $P \times 1$ .  $ZIV_p$  is calculated separately for each phase.  $ZIV_p$  is evaluated repeatedly at regular intervals. The  $\Delta ZIV$  is the difference of two consecutive  $ZIV_p$  vectors as given in Equation (6), where  $ZIV_p(t)$  is the vector calculated at time  $t$  and  $ZIV_p(t - T)$  is the vector found at  $t - T$ . The time interval between two successive measurements is denoted by  $T$ . Suppose a fault is identified and localized in zone 1, then the value of  $K_{11}$  in  $\Delta ZIV$  vector is greater than three times the  $K_{11}$  value of previously calculated  $ZIV_p$ .

$$ZIV = |B| = \begin{bmatrix} K_{11} \\ K_{21} \\ K_{31} \\ \vdots \\ K_{P1} \end{bmatrix} \quad (4)$$

$$ZIV_p = |CIM \times I_p| \quad (5)$$

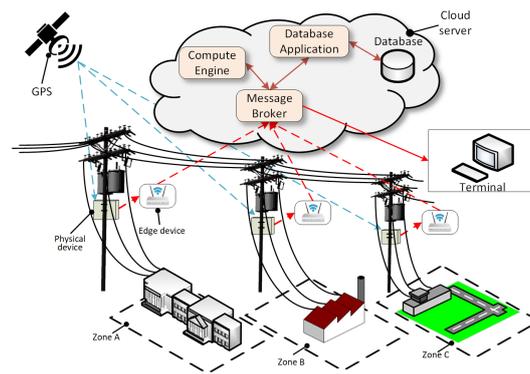
$$\Delta ZIV_p = ZIV_p(t) - ZIV_p(t - T) \quad (6)$$

$$|\Delta ZIV_p| = |ZIV_p(t) - ZIV_p(t - T)| \quad (7)$$

### 3. IoT Framework

The diagram of proposed cloud based IoT for the fault detection in PDS is shown in Figure 1. A current sensor is installed between the boundaries of two different zones. As the sensor represents the current sensing device (CSD), it measures the current during the operation of PDS. The sensor transmits the measured values of current to the edge device (ED) through transmission control protocol/internet protocol TCP/IP protocol. The ED acts as a data sink for sensor. ED provides a uniform interface to a cloud application, where the application running on the cloud can receive information without dealing with any complexities of CSD. CAP is introduced as additional feature in ED to filter the information based upon a predefined criteria. This enables ED to send only desired information to the cloud. A similar concept was adopted in [37], where edge intelligence was used to reduce the amount data transmitted by vessel monitoring system (VMS). Communication savings of about 70% to 90% were observed as only abnormal vessel behaviour was transmitted over the network. In our work MQ telemetry protocol (MQTT) protocol was used to transmit information to the cloud server [38]. ED compares the current measurements with its previously recorded values. Data is transmitted if the current data differs from its previous measurements by factor of  $F_o$  as given by Equation (8). The value of  $F_o$  is considered equal to 0.01 because 1% is the accuracy limit for phasor amplitude error as described by the standards [39,40]. The data is sent to the cloud through MQTT broker only if it satisfies Equation (8).

$$\left| \frac{C'(t) - C(t)}{C'(t)} \right| > F_o \quad (8)$$



**Figure 1.** Internet of things (IoT) framework for fault identification and localization application utilizing virtualization and edge intelligence features of IoT.

In the above equation  $C(t)$  represents phase current measurement at present time and  $C'(t)$  represents the last phase current measurement that was transmitted. By using the above approach, the overall data traffic over the network is reduced. Moreover, it decreases load on the MQTT broker which performs as a message broker between ED and cloud. In this research, an eclipse mosquito is utilized as a broker that is hosted on a cloud server [41]. The broker uses a publish/subscribe protocol to transfer messages between the clients. It coordinates between edge device and different clients such as; compute engine (CE) and database application. All the correspondence between the clients is handled by the broker in a useful manner. On the other hand, CE computes  $\Delta ZIV$  vector according to the strategy as given in Algorithm 1. Every sensor has its own topic to which it publishes messages. In Algorithm 1, the server waits for the messages from sensor. The Algorithm 1 does not follow any natural order or a particular sequence for receiving the messages from different sensors. For example at the start of a particular timestamp, if the  $rv$  is equal to  $\{0,0,0,0,0\}$  and the measurement of sensor 3 arrives before the other sensors, then  $rv$  is updated as  $\{0,0,1,0,0\}$ . Consequently, for example, if the measurement of sensor 1 arrives then  $rv$  becomes  $\{1,0,1,0,0\}$ . In the same way, the  $rv$  vector will be updated and becomes equal to  $\{1,1,1,1,1\}$  after the arrival of all measurements from all sensors for a particular timestamp. When server receives the message, it extracts the corresponding timestamp from the message packet and stores it in  $current\_t$ . The server then matches the  $current\_t$  with previously assigned timestamp in  $previous\_t$ . If both the variables become equal then the server updates phase current vector  $(ia, ib, ic)$  with phase current information that is extracted from the message packet. It also updates the flag of the sender in  $rv$ . On the other hand if the corresponding timestamp of the received message is greater than the previously stored timestamp in memory then it concludes that a new timestamp has arrived. In addition to this server checks to see if data for previous timestamp is complete or not. This task is performed by  $c\_state()$  function which takes  $rv$  as input parameter. The function returns the values of three variables such as;  $config\_id$ ,  $n\_m$  and  $m\_id$  where the  $config\_id$  represents the configuration of zones in PDS,  $n\_m$  represents the total number of failed sensors and  $m\_id$  contains the list IDs of faulty sensors. The  $config\_id$  and  $userdata$  are passed as input parameters to the  $algo()$  function. The  $userdata$  is a global object that consists of  $rv$ , phase current vectors  $(ia, ib, ic)$ , memory stores phase current vectors  $(m\_ia, m\_ib, m\_ic)$  and other required variables. The  $algo()$  function utilizes  $config\_id$  to evaluate CIM matrix according to the number of messages received from different sensors. This function also performs effectively when a message is not received from a particular sensor. This property is further discussed in Section 4.3. Using this CIM and current vectors, the task of fault detection and localization is performed according to the procedure described in Section 2. If fault is detected, then information about the fault is sent to the database application in JSON format. After all the necessary calculations the  $rv$  vector is reset to its initial state. The value of  $previous\_t$  is assigned to  $current\_t$ . On the other side, a cloud database such as; SQLite is used in our proposed solution. It helps to store information about the faults. The cloud database keeps a record of various parameters during the occurrence of faults. These parameters include time, per phase current

levels, fault type and faulty zone. The stored information of parameters in cloud database proves to be valuable while performing different types of analysis at a later stage.

---

**Algorithm 1:** Algorithm for CE
 

---

**Input:** Current measurements from Edge Devices  $\in \{I_1, I_2, I_3, I_4, I_5, \dots, I_n\}$   
**Output:** JSON object containing information about the fault

```

1 Function algo(config_id, userdata):
2   initialize CIM based on the config_id
3   Get ia, ib, ic, m_ia, m_ib, m_ic from userdata
4    $ZIV_x \leftarrow CIM \times ix^T$  // for every  $x \in \{a, b, c\}$ 
5
6    $memory\_ZIV_x \leftarrow CIM \times m\_ix^T$ 
7    $|\Delta ZIV_x| = ZIV_x - memory\_ZIV_x$ 
8   if  $|\Delta ZIV_x[k]| > 3 \times memory\_ZIV_x[k]$  then
9     Fault has occurred on phase  $x$  of zone  $k$ . Form a JSON object containing information
    about the fault occurred. Send information to the database application for storage
10  end
11  Update m_ia, m_ib, m_ic vectors in userdata with values from ia, ib, ic for use in next
    iteration
12 End Function
13  $rv \leftarrow [0, 0, 0, 0, 0, \dots]_{1 \times n}$  // initialize, position of every element represents a
    particular sensor
14  $ix \leftarrow [0, 0, 0, 0, 0, \dots]_{1 \times n}$  // for every  $x \in \{a, b, c\}$ . Phase current vector, where the
    length of vector is equal to number of sensors
15 userdata  $\leftarrow rv, ia, ib, ic, m_ia, m_ib, m_ic$  // User defined global data to be passed to
    callbacks
16 CE connects to the broker and waits for messages to arrive
17 Loop
18   CE receives a message in JSON object
19   info  $\leftarrow$  message received // CE decodes the received message and assigns it to
    variable info
20   userdata['current_t']  $\leftarrow$  info['t']
21   if userdata['current_t'] == userdata['previous_t'] then
22     userdata['ix'] [k]  $\leftarrow$  info['x'] // where 'k'  $\in \{1, 2, 3, 4, 5, \dots\}$  and  $x \in \{a, b, c\}$ 
23     userdata['rv'] [k]  $\leftarrow$  1
24   end
25   if userdata['current_t'] > userdata['previous_t'] then
26     [config_id, n_m, m_id] = c_state(rv)
27     Display(n_m, m_id) // Display information for missing sensors
28     algo(config_id, userdata)
29     userdata['rv']  $\leftarrow$  [0, 0, 0, 0, 0, ...]1×n // Reset rv for new timestamp
30     userdata['previous_t']  $\leftarrow$  userdata['current_t']
31     userdata['ix'] [k]  $\leftarrow$  info['x'] // extract information from message with new
    timestamp
32     userdata['rv'] [k]  $\leftarrow$  1
33   end
34 EndLoop

```

---

### 4. Results and Discussion

Figure 2 presents a flowchart that explains the complete operation of our proposed strategy for fault identification and localization in PDS. PDS is modeled in MATLAB/Simulink and data for sensor prototypes was obtained by simulating PDS at different operating conditions. This data was used by sensor prototypes to produce a steady stream of measurements that was expected from CSDs. The sensor prototypes send the measurements to edge devices in the form of JSON strings by using TCP/IP protocol. The edge device communicates with MQTT broker where the broker further passes the information to compute engine. In case a fault occurs, the information about the fault would be transmitted to the database application through MQTT broker. The database application writes the information to database file. Three test cases were designed evaluate performance of the proposed scheme. Case 1 and case 2 deal with individual and multiple simultaneous faults respectively. Both these cases are explained in the context of scenario when all the sensors are fully working for different zones in a PDS. On the other hand, case 3 shows the performance of proposed technique for a case when single or multiple sensor failure occurs. A system based on IEEE 37 node test feeder was simulated in MATLAB/Simulink as shown in Figure 3. IEEE 37 node test feeder is an actual system which is located in Arizona having a nominal voltage of 4.6 kV. It is a highly unbalanced system characterized by delta connected loads. It has been modeled using distributed line parameters to account for the effects of mutual capacitances and inductances. The base power ( $S_{base}$ ) and base voltage ( $V_{base}$ ) of the system are taken equal to 1 MVA and 4.16 kV, respectively.

The system was divided into 5 zones. Note that the accuracy of fault localization depends upon the number of zones. As the number of zones increases, the accuracy of fault localization also increases. However, the installation of number of CSDs grow with respect to rise in number of zones. This scheme works for radial as well as meshed networks but here radial feeder is used. The CIM for the IEEE 37 node test feeder is provided in Equation (9).

$$CIM = \begin{bmatrix} 1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

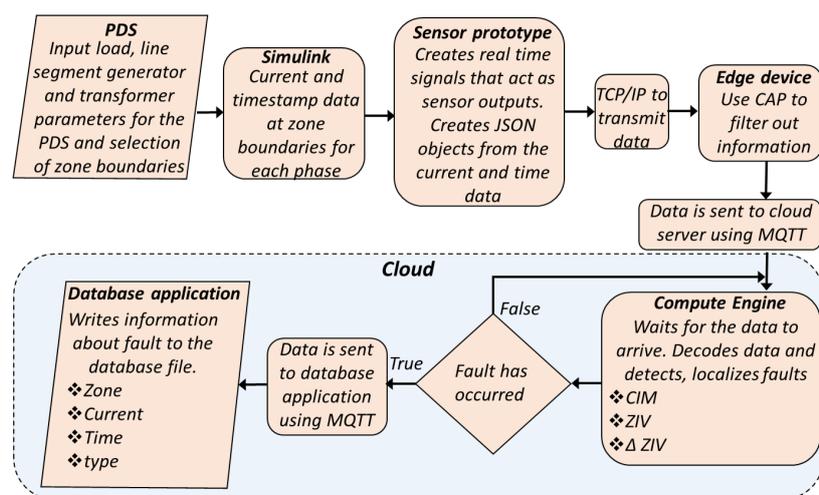
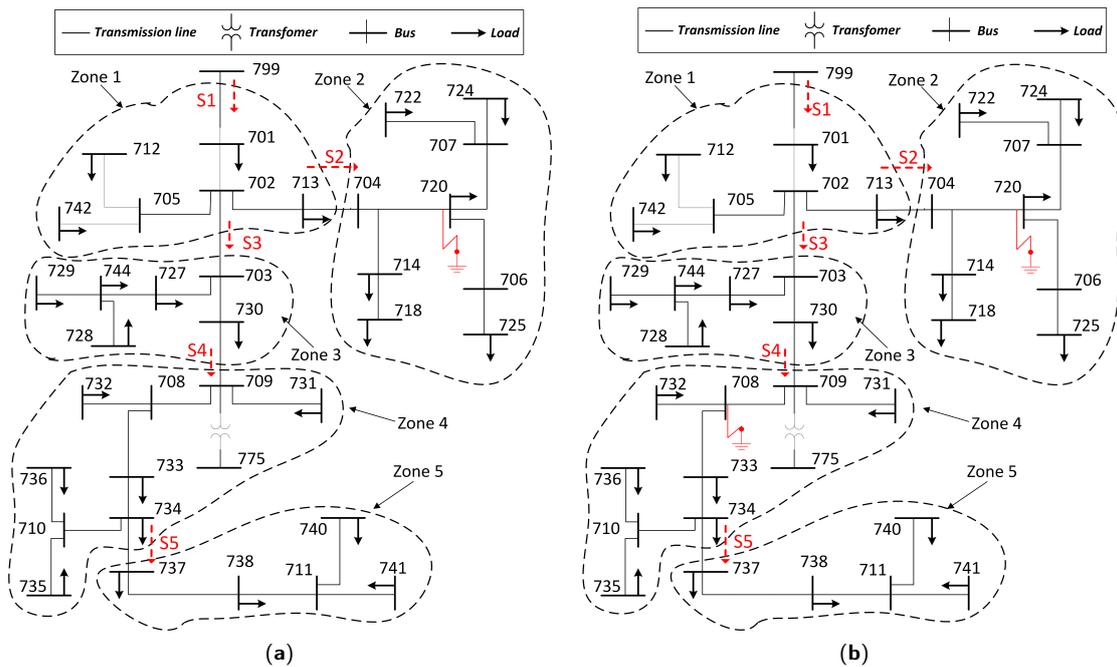


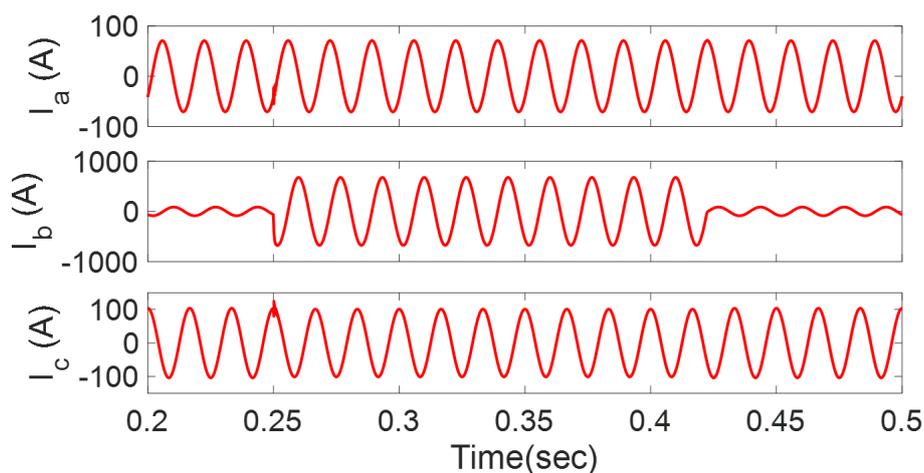
Figure 2. Flowchart of our proposed solution.



**Figure 3.** IEEE 37 node test feeder divided into 5 zones for fault detection and localization. Fault is induced at bus 720 (a). Fault is induced at bus 720 and 708 (b). (Current measurements at the zone boundaries are marked in red).

4.1. Case 1: Individual Faults

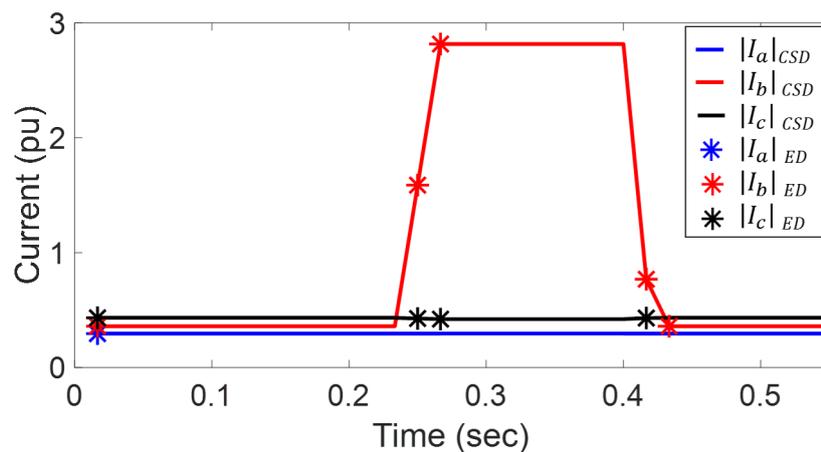
A Phase B fault at bus 720 is induced with a fault resistance of 5 ohms as shown in Figure 3a. The fault is induced from  $t = 0.25s$  to  $t = 0.416s$ . In this study, the time at the occurrence of fault is taken into consideration. Figure 4 presents the phase currents flowing through zone 2 before and after the occurrence of fault. It can be seen that current in phase B changes after the occurrence of fault while phase A and C show only small changes as compared to phase B. Current entering into zone 2 passes through sensor 2. As our study focuses on the analysis of the current magnitudes, CSDs are used for this purpose.



**Figure 4.** Per phase fault current levels between zone 1 and zone 2 (sensor 2).

Figure 5 provides the output data of CSD and ED for different phase current magnitudes. The solid lines indicate that the maximum number of measurements are taken at a constant rate of 60 Hz. This is the rate at which CSD give the data of different phase current magnitudes. The phase B current ( $|I_b|_{CSD}$ ) shows increasing trend when the fault is induced as shown in Figure 5. However, it is

observed that  $|I_a|_{CSD}$  and  $|I_c|_{CSD}$  is not affected. These measurements are further transmitted to ED. The ED filters the data and allows only to pass fault event data. In Figure 5, (\*) represents the data that is passed through ED. The ED successfully detects the increasing trend of phase B current as given by  $|I_b|_{ED}$ . The ED also captures small changes in phase C current as indicated by  $|I_c|_{ED}$ . Moreover, the phase A current  $|I_a|_{ED}$  remains constant and does not show any significant variation. Therefore, no current data is transmitted for phase A current during the occurrence of fault. It is found that number of transmitted phase current measurements are reduced significantly without any loss of accuracy in terms of fault detection and localization. Table 1 shows the  $ZIV$  values during the fault event. Note that the second value of the  $ZIV_b$  vector at time  $t = 0.25s$  is much greater than the value preceding the fault. The  $|\Delta ZIV_b|$  value for zone 2 indicates a fault on phase B which is the expected result.



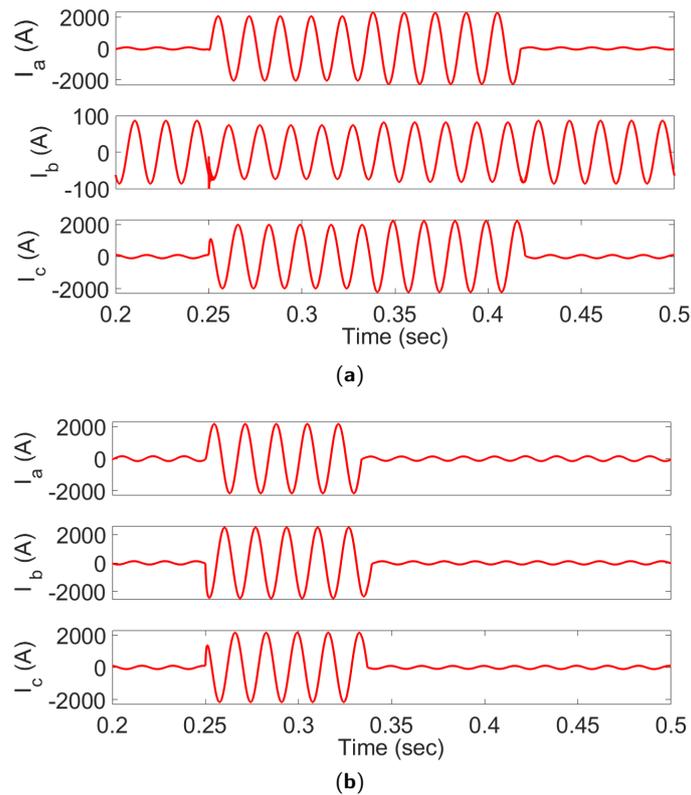
**Figure 5.** Measurements transmitted by the edge device with and without the use of context aware policy in edge device for sensor 2.

**Table 1.**  $|\Delta ZIV|$  vectors for phase a,b,c for test case 1 at time  $t = 0.25$  s. (The value shown in bold face indicates phase B fault in zone 2. All the values are calculated in p.u).

Zone	At Time $(t - T) = 0.233$			At Time $t = 0.25s$			Difference		
	$ZIV_a$	$ZIV_b$	$ZIV_c$	$ZIV_a$	$ZIV_b$	$ZIV_c$	$ \Delta ZIV_a $	$ \Delta ZIV_b $	$ \Delta ZIV_c $
1	0.599	0.346	0.535	0.599	0.301	0.541	0	0.045	0.006
2	0.295	<b>0.359</b>	0.433	0.295	<b>1.588</b>	0.427	0	<b>1.229</b>	0.006
3	0.294	0.176	0.221	0.294	0.176	0.221	0	0	0
4	0.241	0.186	0.284	0.241	0.186	0.284	0	0	0
5	0.395	0.303	0.143	0.395	0.303	0.143	0	0	0

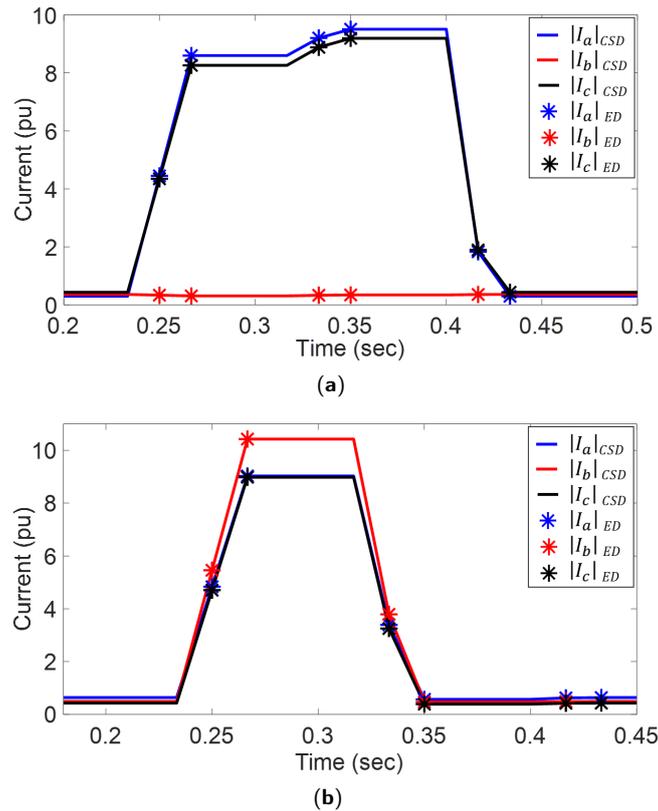
#### 4.2. Case 2: Multiple Simultaneous Faults

Two multiple simultaneous faults were induced for this case such that phase A and C faults at bus 720 and phase A, B and C faults at bus 708 as shown in Figure 3b. The faults were induced at time  $t = 0.25$  s Current measurements from sensor 2 and sensor 4 were used in this case. The per phase currents for case 2 are shown in Figure 6.



**Figure 6.** Per phase current levels between zone 1 and zone 2 i-e sensor 2 (a), per phase current levels between zone 3 and zone 4 i-e sensor 4 (b) for case 2.

Figure 7a shows data for the current that is entering into zone 2 due to the fault on bus 720. An increasing behavior in current magnitudes are observed as shown by  $|I_a|_{CSD}$  and  $|I_c|_{CSD}$ . The phase B current as indicated by  $|I_b|_{CSD}$  remains unaffected. Similarly, the three phase fault is induced on bus 708 as given in Figure 7b. All the three phase current magnitudes are increased at time  $t = 0.25$  s. This increasing trend is represented by  $|I_a|_{CSD}$ ,  $|I_b|_{CSD}$  and  $|I_c|_{CSD}$ . Similar to case 1, the data of current magnitudes is filtered by ED. The ED allows only to pass the fault current magnitudes. In addition to this, the ED also captures any variations in the data of current magnitudes. Although, both the faults are induced at the same time but the fault on bus 708 is removed at time  $t = 0.33$  s before the fault on bus 720. This causes the fault current flowing through bus 720 to increase further. The increase in fault current at bus 720 is evident from  $|I_a|_{CSD}$  and  $|I_c|_{CSD}$  in Figure 7a. ED for sensor 2 is able to recognize this change as shown by  $|I_a|_{ED}$  and  $|I_c|_{ED}$  in Figure 7a. A significant decrease in number of transmitted phase current measurements is observed due to CAP. Note that the decrease in number of transmitted phase current measurements does not mean that the number of overall messages transmitted will also decrease. In fact, the total messages transmitted will remain the same. Only the number of phase measurements contained in the message would change. For example, if the data for phase A changes while phase B and C remain the same then the message transmitted would contain timestamp and data for phase A only. If no phase changes then the message would be transmitted containing only the timestamp value.



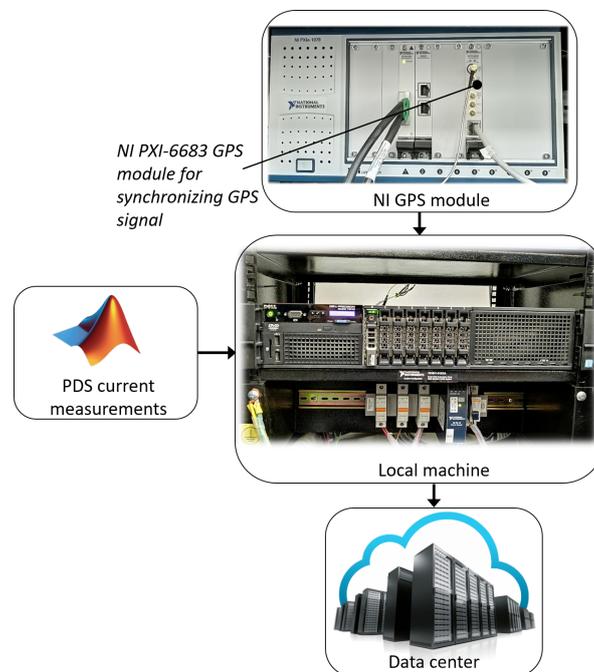
**Figure 7.** Measurements transmitted by the edge device with and without context aware policy (CAP) for sensor 2 (a), measurements transmitted by the edge device with and without CAP for sensor 4 (b).

Table 2 summarizes the results for test case 2.  $ZIV$  for before and after fault are shown along with the  $|\Delta ZIV|$ .  $|\Delta ZIV|$  values for each phase are listed separately. Values shown in bold font indicate fault detection. Note that the algorithm correctly predicts the faulty phases and zones in this case as well. To measure timestamp data in real time, an experimental setup is established as shown in Figure 8. As relative timestamp data from MATLAB/Simulink was used in previous cases, therefore the high resolution NI PXI-6683 GPS module is used to measure timestamp data in a synchronized manner. The PDS is simulated in MATLAB/Simulink to generate current data during the operation of PDS. The CSDs are prototyped in a local machine to perform operations on any current data that are received from PDS. The data of CSD is combined with the timestamp information of GPS module to form a JSON string. This combination results into a steady stream of synchronized current measurements that virtualizes real time current sensing. The combined data is fed to the ED that is also implemented in a local machine. The ED communicates with the CE which is the part of a data center (cloud server). This all results into the detection of fault. The faults are detected and the data is stored in a database file.

**Table 2.**  $|\Delta ZIV|$  vectors for phase a,b,c for test case 2 for fault detection at  $t = 0.25$  s. (The value shown in bold indicate fault on respective phases. All the values shown are calculated in p.u).

Zone	At Time $(t - T) = 0.233$ s			At Time $t = 0.25$ s			Difference		
	$ZIV_a$	$ZIV_b$	$ZIV_c$	$ZIV_a$	$ZIV_b$	$ZIV_c$	$ \Delta ZIV_a $	$ \Delta ZIV_b $	$ \Delta ZIV_c $
1	0.599	0.346	0.535	0.545	0.344	0.498	0.054	0.002	0.03
2	<b>0.295</b>	0.359	<b>0.433</b>	<b>4.448</b>	0.333	<b>4.348</b>	<b>4.152</b>	0.003	<b>3.915</b>
3	0.294	0.176	0.221	0.246	0.147	0.191	0.047	0.003	0.03
4	<b>0.241</b>	<b>0.186</b>	<b>0.284</b>	<b>4.510</b>	<b>5.202</b>	<b>4.589</b>	<b>4.265</b>	<b>5.017</b>	<b>4.304</b>
5	0.395	0.303	0.143	0.327	0.255	0.116	-0.068	0.004	0.027

Table 3 lists the results of the analysis carried out with GPS module. The module gives timestamps in terms of floating point numbers. According to NI PXI-6683 GPS module, the timestamp values represent number of seconds passed since 1st Jan 1904. Case 1 and case 2 are again implemented by using the experimental setup as shown in Figure 8. The setup works efficiently in terms of detection and localization of faults as evident from the values of  $\Delta|ZIV|$  as given in Tables 1 and 2. Additionally, the GPS module measure timestamp of any event occurred during the real time operation of PDS, which is another feature introduced in our proposed cloud based IoT scheme. On the other hand, a relative timestamp based strategy was utilized during the simulation studies of case 1 and case 2 in MATLAB/Simulink. During the simulation studies, prior information about the timestamp was available which is considered a drawback of this strategy. The user has information that at which time the fault is injected and occurred in PDS during the simulation. On the other side, the GPS module gives different values of timestamp as compared to the timestamps calculated during our simulation studies as shown in Table 3. Moreover, there is no prior information available about the timestamp at which the fault is injected and occurred. In addition, Table 3 represents the timestamp data and current measurements that are stored in the database for further analysis.



**Figure 8.** Experimental setup showing the GPS module to measure timestamp.

**Table 3.** Information stored in database when simulation with real GPS module was carried out.

	Timestamp	Type of Fault	Zone Number	$I_a(p.u)$	$I_b(p.u)$	$I_c(p.u)$	config-id
Case 1	3668306115.11743	Phase B	2	0.295	1.588	0.427	0
Case 2	3668305299.40703	Phase A and C	2	4.448	0.339	4.348	0
	3668305299.40703	Phase A,B,C	4	4.835	5.458	4.705	0

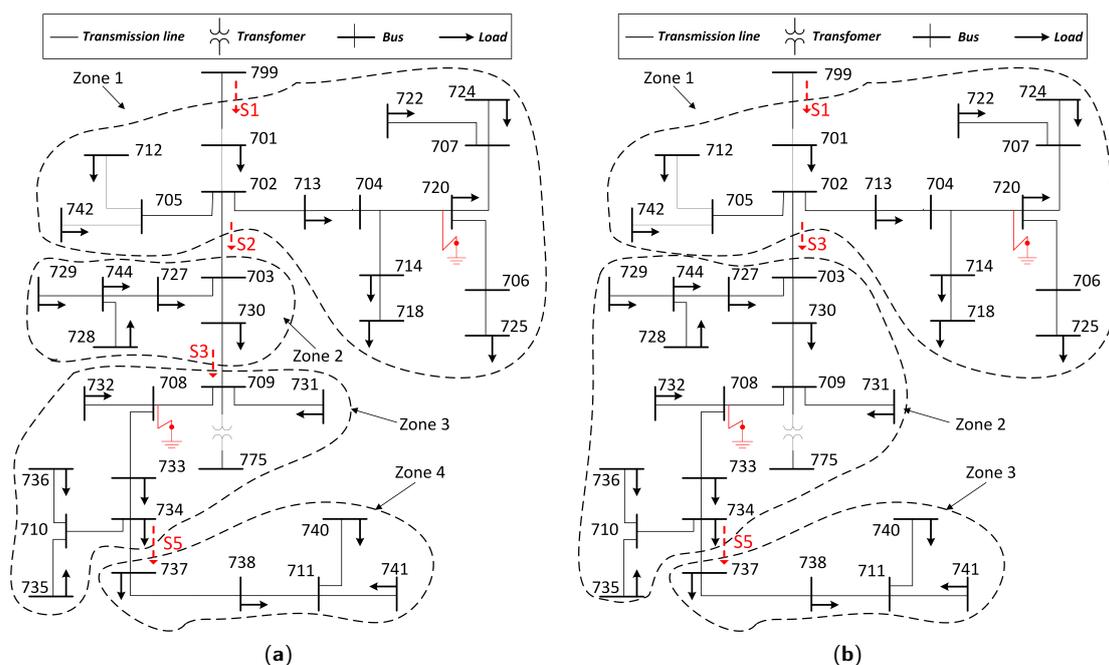
#### 4.3. Case 3: Multiple Simultaneous Fault with Sensor Failure

This section focuses on the performance of a proposed scheme for detection and localization of multiple simultaneous faults in the context of single or multiple sensor failure. A faulty sensor or edge device can result in incomplete data arriving at the cloud server. Here, the incomplete data means that the data from all the sensors in IoT network is not available at cloud server for a particular timestamp before the arrival of the new timestamp. A robust and reliable scheme should be capable of handling sensor failures and loss of data transmission in large networks. The proposed scheme is capable of

performing fault detection and localization if one or more than one sensor fails. This is achieved by the reconfiguration of the zones in PDS. A zone associated with a faulty sensor is merged with an adjacent zone having an operational sensor. It is important to mention that faulty sensor must lie between the two zones that will be merged for the fault detection in PDS. In addition to this, the CIM matrix is evaluated according to the new zone configuration. The calculation of CIM matrix is done automatically when the server detects incomplete data. A unique configuration ID is assigned to every zone configuration which is stored along with fault information in the database.

The simultaneous multiple faults on bus 720 and bus 708 from case 2 were used in this case to demonstrate performance of the proposed technique. Two scenarios were designed for this purpose. In scenario 1, the sensor 2 was shown faulty by disabling sensor 2 before the fault was introduced. This caused the zones to be reconfigured as shown in Figure 9a. Since previously in case 2, sensor 2 was located between zones 1 and 2. Therefore in case 3, they were merged into a single larger zone and renamed as zone 1. This new configuration was achieved by evaluating the CIM matrix again as given by Equation (10).

$$CIM = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$



**Figure 9.** Reconfiguration of zones after the failure of Sensor 2 (a). Reconfiguration of zones after failure of sensors 2 and 4 (b). (Current measurements at the zone boundaries are marked in red).

Table 4 shows the results for scenario 1. Note that the number of zones were reduced to 4 and the faults were successfully identified in the zone 1 and zone 3 as shown by the bold font values in Table 4. In scenario 2, two sensors are shown faulty by simultaneously disabling sensor 2 and sensor 4 before the fault was introduced. This caused the number of zones to be reduced to 3. The zones associated with sensor 2 were merged with zone 1 whereas the zone associated with sensor 4 was merged with zone 3 as shown in Figure 9b. After the new configuration of zones, the corresponding CIM matrix

was evaluated as given by Equation (11). Table 4 shows the results for scenario 2 and the faults are identified in zone 1 and zone 2 as represented by bold font values in Table 4.

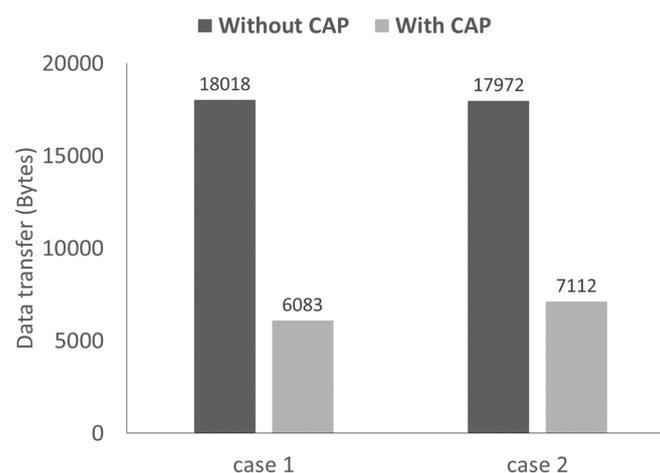
$$CIM = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

**Table 4.**  $|\Delta ZIV|$  vectors for phase a,b,c for test case 3 for fault detection under device failures. (The value shown in bold indicate fault on respective phases. All the values shown are calculated in p.u).

Zone	At Time $(t - T) = 0.233$ s			At Time $t = 0.25$ s			Difference		
	$ZIV_a$	$ZIV_b$	$ZIV_c$	$ZIV_a$	$ZIV_b$	$ZIV_c$	$ \Delta ZIV_a $	$ \Delta ZIV_b $	$ \Delta ZIV_c $
<i>Scenario 1 : Single sensor failure (sensor 2)</i>									
1	<b>0.894</b>	0.705	<b>0.968</b>	<b>4.993</b>	0.678	<b>4.846</b>	<b>4.10</b>	0.027	<b>3.878</b>
2	0.293	0.176	0.221	0.246	0.147	0.190	0.04	0.029	0.031
3	<b>0.241</b>	<b>0.186</b>	<b>0.284</b>	<b>4.506</b>	<b>5.203</b>	<b>4.589</b>	<b>4.265</b>	<b>5.017</b>	<b>4.304</b>
4	0.395	0.303	0.143	0.327	0.255	0.116	0.06	0.048	0.027
<i>Scenario 2: Multiple sensor failures (sensor 2 and sensor 4)</i>									
1	<b>0.894</b>	0.705	<b>0.968</b>	<b>4.993</b>	0.678	<b>4.846</b>	<b>4.10</b>	0.025	<b>3.878</b>
2	<b>0.705</b>	<b>0.362</b>	<b>0.505</b>	<b>4.752</b>	<b>5.350</b>	<b>4.780</b>	<b>4.217</b>	<b>4.998</b>	<b>4.273</b>
3	0.968	0.303	0.143	0.327	0.255	0.116	0.06	0.048	0.027

#### 4.4. Data Reduction

The CAP is implemented in ED to make the device intelligent. This intelligence reduces the amount of data transfer. Figure 10 shows the total amount of data in bytes for case 1 and case 2. The data is transferred to the cloud server via internet before and after implementation of the CAP in the ED. The total number of messages transmitted for each case remains the same, that is 300 messages per second. For both cases, it is found that reductions of 66.2% and 60.5% are observed in the amount of data transferred to CS. Variable length packets were transferred depending upon the number of phases for which the data is to be sent. On the other hand similar analysis can also be performed for case 3.



**Figure 10.** Amount of data sent to the network with and without a context aware policy.

#### 4.5. Latency Evaluation

The latency calculation of our proposed solution is necessary during the occurrence of faults in PDS. This evaluation determines the total time taken by the cloud based IoT solution to detect and localize the faults. Latency from ED to CE along with the processing time of CE are calculated in terms

of minimum, maximum and average delay. The latency evaluation for case 1, case 2 and case 3 are the same. Therefore, calculations are performed for the case 2 only. It is noticed that communication delays are observed due to the channel bandwidth and load of MQTT broker. In order to measure the delay accurately, a publish/subscribe loop-back strategy is used in our solution. In this strategy, ED sends messages to CE at regular intervals. CE retransmits those messages to ED as they are received at CE. Therefore, a message the arriving at ED has completed one round trip between ED and CE. The latency is calculated by taking half of the round trips performed between ED and CE. In the case 2 scenario, the minimum, maximum and average ED to CE delay are equal to 34.3 ms, 190.8 ms and 40.2 ms, respectively. When compared to a similar application for state estimation in [30], the delays obtained here are improved significantly. Average delay of 84.1 ms was obtained in [30]. Similarly, the minimum, maximum and average CE processing delay is calculated to be 2.03 ms, 6.86 ms and 3.65 ms, respectively. It is important to mention for case 2, that the average total delay of 43.85 ms is taken by our solution which is considered suitable for the applications like fault detection in PDS.

## 5. Conclusions

In this paper, a novel cloud based IoT solution is proposed for fault identification and localization in power distribution systems. In available literature, the fault detection and localization schemes are not suitable for distribution systems due to the requirement of large number of complex devices that need to be installed. Therefore, a zone based approach is suggested that uses current magnitudes acquired from sparsely located current sensing devices for the detection of faults. Existing smart meters have no data recording capabilities but they can report measurements using ICT infrastructure of smart grid environment. Therefore, a cloud based IoT solution is developed to acquire these measurements at a centralized location. Our solution utilizes a context aware policy in edge device to intelligently preprocess and filter the measurements so that only those measurements are sent to the cloud that indicate predefined levels of change to previously transmitted measurements. In large IoT based systems, device failures can cause major problems during the operation of PDS. For this purpose, a new device failure handling capability is added to the scheme for reconfiguring the zones that consequently detect individual or multiple simultaneous faults in the presence of single or multiple devices failure(s). This added feature demonstrates the robustness and scalability of our solution that is required in large distribution systems. As our generic solution has achieved significant results, it can be further generalized and extended in terms of real time application for a PDS. Due to the robustness and scalability of the solution, it can prove to be helpful in reduction of power infrastructure costs by using a centralized system based on cloud computing. The solution can use low cost devices or sensors to localize faults during the real time operation of PDS. The scheme can quickly localize multiple faults so that the crew or utility personnel do not have to physically search for the source or location of faults in PDS. Furthermore, the crew can use the information of any fault event to quickly isolate the faulty area and take necessary measures to restore the power in a distribution network. In addition, a successful comparison of proposed solution is performed with the existing fault detection methods that further highlights the main benefits of our scheme. Moreover, literature of different smart solutions such as; distributed computing technologies that include WSN and edge mesh are discussed. The characteristics of various solutions are compared with our centralized cloud based method. It is suggested that these solutions can be tested for PDS to enhance the performance of existing fault detection techniques, as future work.

**Author Contributions:** Writing—review and editing, conceptualization and supervision, A.U.; validation, A.K.; writing—review and editing, K.I.; visualization, H.S.Z.; writing—original draft, M.U.M.; software, S.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

## Abbreviations

The following abbreviations are used in this manuscript:

$ \Delta ZIV_a ,  \Delta ZIV_b ,  \Delta ZIV_c $	Difference between two consecutive ZIV vectors for phase A,B and C
CAP	Context Aware Policy
CIM	Current Identification Matrix
<i>config_id</i>	Configuration ID for a particular zone configuration
CSD	Current Sensing Device
<i>current_t</i>	Timestamp from the current message recieved
ED	Edge Device
$I_a, I_b, I_c$	Per phase current in P.u
$ia, ib, ic$	Phase current vectors
$m_{ia}, m_{ib}, m_{ic}$	Phase current vectors from previous iteration
<i>m_id</i>	List of IDs of failed sensors
<i>memory_ZIV</i>	Value of ZIV vector stored in memory during operation of CE
<i>n_m</i>	Total number of failed sensors
PDS	Power Distribution Systems
<i>previous_t</i>	Timestamp in memory from the previous message received
PMU	Phasor Measurement Unit
S1, S2, S3, S4, S5	Sensor 1, Sensor 2, Sensor 3, Sensor 4, Sensor 5
<i>t</i>	Instant of time
<i>T</i>	Time between successive measurements
VMS	Vessel Monitoring System
WAMS	Wide Area Monitoring Systems
ZIV	Zone Identification Vector
$ZIV_a, ZIV_b, ZIV_c$	ZIV vectors for separte phases

## References

- Hojabri, M.; Dersch, U.; Papaemmanouil, A.; Bosshart, P. A Comprehensive Survey on Phasor Measurement Unit Applications in Distribution Systems. *Energies* **2019**, *12*, 4552. [[CrossRef](#)]
- Kleineidam, G.; Krasser, M.; Reischböck, M. The cellular approach: Smart energy region Wunsiedel. Testbed for smart grid, smart metering and smart home solutions. *Electr. Eng.* **2016**, *98*, 335–340. [[CrossRef](#)]
- Ghasemkhani, A.; Monsef, H.; Rahimi-Kian, A.; Anvari-Moghaddam, A. Optimal design of a wide area measurement system for improvement of power network monitoring using a dynamic multiobjective shortest path algorithm. *IEEE Syst. J.* **2015**, *11*, 2303–2314. [[CrossRef](#)]
- Wen, M.H.; Xu, J.; Li, V.O. Optimal multistage PMU placement for wide-area monitoring. *IEEE Trans. Power Syst.* **2013**, *28*, 4134–4143. [[CrossRef](#)]
- Mohammadi, P.; Mehraeen, S. PMU Optimal Placement using sensitivity analysis for power systems fault location. In Proceedings of the 2015 IEEE Electrical Power and Energy Conference (EPEC), London, ON, Canada, 26–28 October 2015; pp. 244–249.
- Mahmood, A.; Javaid, N.; Razzaq, S. A review of wireless communications for smart grid. *Renew. Sustain. Energy Rev.* **2015**, *41*, 248–260. [[CrossRef](#)]
- Yigit, M.; Gungor, V.C.; Tuna, G.; Rangoussi, M.; Fadel, E. Power line communication technologies for smart grid applications: A review of advances and challenges. *Comput. Netw.* **2014**, *70*, 366–383. [[CrossRef](#)]
- Kezunovic, M.; Xie, L.; Grijalva, S. The role of big data in improving power system operation and protection. In Proceedings of the 2013 IREP Symposium Bulk Power System Dynamics and Control-IX Optimization, Security and Control of the Emerging Power Grid, Rethymnon, Greece, 25–30 August 2013; pp. 1–9.
- Zhuang, Y.t.; Wu, F.; Chen, C.; Pan, Y.H. Challenges and opportunities: From big data to knowledge in AI 2.0. *Front. Inf. Technol. Electron. Eng.* **2017**, *18*, 3–14. [[CrossRef](#)]
- Barik, R.K.; Gudey, S.K.; Reddy, G.G.; Pant, M.; Dubey, H.; Mankodiya, K.; Kumar, V. FogGrid: Leveraging fog computing for enhanced smart grid network. In Proceedings of the 2017 14th IEEE India Council International Conference (INDICON), Roorkee, India, 15–17 December 2017; pp. 1–6.

11. Suryadevara, N.K.; Biswal, G.R. Smart Plugs: Paradigms and Applications in the Smart City-and-Smart Grid. *Energies* **2019**, *12*, 1957. [[CrossRef](#)]
12. Talari, S.; Shafie-Khah, M.; Siano, P.; Loia, V.; Tommasetti, A.; Catalão, J.P. A review of smart cities based on the internet of things concept. *Energies* **2017**, *10*, 421. [[CrossRef](#)]
13. Al-Ali, A. Role of internet of things in the smart grid technology. *J. Comput. Commun.* **2015**, *3*, 229. [[CrossRef](#)]
14. Swathika, O.V.G.; Hemapala, K.T.M.U. IOT Based Energy Management System for Standalone PV Systems. *J. Electr. Eng. Technol.* **2019**, *14*, 1811–1821. [[CrossRef](#)]
15. Maryska, M.; Doucek, P.; Sladek, P.; Nedomova, L. Economic Efficiency of the Internet of Things Solution in the Energy Industry: A Very High Voltage Frosting Case Study. *Energies* **2019**, *12*, 585. [[CrossRef](#)]
16. Vakili, M.; Jahangiri, N.; Sharifi, M. Cloud service selection using cloud service brokers: Approaches and challenges. *Front. Comput. Sci.* **2019**, *13*, 599–617. [[CrossRef](#)]
17. Zhang, J.; Huang, T.; Wang, S.; Liu, Y.J. Future Internet: Trends and challenges. *Front. Inf. Technol. Electron. Eng.* **2019**, *20*, 1185–1194. [[CrossRef](#)]
18. Chen, S.; Wen, H.; Wu, J.; Lei, W.; Hou, W.; Liu, W.; Xu, A.; Jiang, Y. Internet of Things based smart grids supported by intelligent edge computing. *IEEE Access* **2019**, *7*, 74089–74102. [[CrossRef](#)]
19. Chen, H.; Wu, S.; Jin, H.; Chen, W.; Zhai, J.; Luo, Y.; Wang, X. A survey of cloud resource management for complex engineering applications. *Front. Comput. Sci.* **2016**, *10*, 447–461. [[CrossRef](#)]
20. Zhang, G.; Shu, H.; Lopes, F.; Liao, Y. Single-ended travelling wave-based protection scheme for double-circuit transmission lines. *Int. J. Electr. Power Energy Syst.* **2018**, *97*, 93–105. [[CrossRef](#)]
21. Naidu, O.; Pradhan, A.K. A traveling wave-based fault location method using unsynchronized current measurements. *IEEE Trans. Power Deliv.* **2018**, *34*, 505–513. [[CrossRef](#)]
22. Aslan, Y.; Yağan, Y.E. Artificial neural-network-based fault location for power distribution lines using the frequency spectra of fault data. *Electr. Eng.* **2017**, *99*, 301–311. [[CrossRef](#)]
23. Ngaopitakkul, A.; Leelajindakraierk, M. Application of probabilistic neural network with transmission and distribution protection schemes for classification of fault types on radial, loop, and underground structures. *Electr. Eng.* **2018**, *100*, 461–479. [[CrossRef](#)]
24. Rafinia, A.; Moshtagh, J. A new approach to fault location in three-phase underground distribution system using combination of wavelet analysis with ANN and FLS. *Int. J. Electr. Power Energy Syst.* **2014**, *55*, 261–274. [[CrossRef](#)]
25. Bhalja, B.; Maheshwari, R. Wavelet-based fault classification scheme for a transmission line using a support vector machine. *Electr. Power Compon. Syst.* **2008**, *36*, 1017–1030. [[CrossRef](#)]
26. Bretas, A.S.; Salim, R.H. A new fault location technique for distribution feeders with distributed generation. *WSEAS Trans. Power Syst.* **2006**, *1*, 894.
27. Nikkhajoei, H.; Lasseter, R.H. *Microgrid Fault Protection Based on Symmetrical and Differential Current Components*; Power System Engineering Research Center: Tempe, AZ, USA, 2006; pp. 71–74.
28. Lin, T.C.; Lin, P.Y.; Liu, C.W. An algorithm for locating faults in three-terminal multisection nonhomogeneous transmission lines using synchrophasor measurements. *IEEE Trans. Smart Grid* **2013**, *5*, 38–50. [[CrossRef](#)]
29. Fan, W.; Liao, Y. Wide area measurements based fault detection and location method for transmission lines. *Prot. Control. Mod. Power Syst.* **2019**, *4*, 7. [[CrossRef](#)]
30. Meloni, A.; Pegoraro, P.A.; Atzori, L.; Benigni, A.; Sulis, S. Cloud-based IoT solution for state estimation in smart grids: Exploiting virtualization and edge-intelligence technologies. *Comput. Netw.* **2018**, *130*, 156–165. [[CrossRef](#)]
31. Farris, I.; Girau, R.; Militano, L.; Nitti, M.; Atzori, L.; Iera, A.; Morabito, G. Social virtual objects in the edge cloud. *IEEE Cloud Comput.* **2015**, *2*, 20–28. [[CrossRef](#)]
32. Koley, E.; Kumar, R.; Ghosh, S. Low cost microcontroller based fault detector, classifier, zone identifier and locator for transmission lines using wavelet transform and artificial neural network: A hardware co-simulation approach. *Int. J. Electr. Power Energy Syst.* **2016**, *81*, 346–360. [[CrossRef](#)]
33. Katyara, S.; Izykowski, J.; Chowdhry, B.S.; Musavi, H.A.; Hussain, R. WSN-based monitoring and fault detection over a medium-voltage power line using two-end synchronized method. *Electr. Eng.* **2018**, *100*, 83–90. [[CrossRef](#)]
34. Sahni, Y.; Cao, J.; Zhang, S.; Yang, L. Edge mesh: A new paradigm to enable distributed intelligence in internet of things. *IEEE Access* **2017**, *5*, 16441–16458. [[CrossRef](#)]

35. Batista, N.; Melício, R.; Matias, J.; Catalão, J. Photovoltaic and wind energy systems monitoring and building/home energy management using ZigBee devices within a smart grid. *Energy* **2013**, *49*, 306–315. [[CrossRef](#)]
36. Christin, D.; Reinhardt, A.; Mogre, P.S.; Steinmetz, R. Wireless sensor networks and the internet of things: selected challenges. In Proceedings of the 8th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, Hamburg, Germany, 13–14 August 2009; pp. 31–34.
37. Ferreira, J.C.; Martins, A.L. Edge Computing Approach for Vessel Monitoring System. *Energies* **2019**, *12*, 3087. [[CrossRef](#)]
38. Standard, O. MQTT Version 3.1. 1. 2014, Volume 1. Available online: <http://docs.oasis-open.org/mqtt/mqtt/v3> (accessed on).
39. Martin, K.; Hamai, D.; Adamiak, M.; Anderson, S.; Begovic, M.; Benmouyal, G.; Brunello, G.; Burger, J.; Cai, J.; Dickerson, B.; et al. Exploring the IEEE standard C37. 118–2005 synchrophasors for power systems. *IEEE Trans. Power Deliv.* **2008**, *23*, 1805–1811. [[CrossRef](#)]
40. Association, I.S. *IEEE Standard for Synchrophasor Measurements for Power Systems—Amendment 1: Modification of Selected Performance Requirements*; IEEE: Piscataway, NJ, USA, 2014.
41. Mosquitto, E. An Open Source MQTT Broker. Eclipse Mosquitto™. 2018. Available online: [Mosquitto.org](https://mosquitto.org) (accessed on 23 April 2018).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).