IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 8, 2021, accepted February 27, 2021, date of publication March 26, 2021, date of current version April 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3069137

Edge Intelligence for Data Handling and Predictive Maintenance in IIOT

TAIMUR HAFEEZ, (Graduate Student Member, IEEE), LINA XU, AND GAVIN MCARDLE
School of Computer Science, University College Dublin, Dublin 4, D04 V1W8 Ireland
Corresponding author: Taimur Hafeez (taimur.hafeez@ucdconnect.ie)

ABSTRACT The use of IoT has become pervasive and IoT devices are common in many domains. Industrial IoT (IIoT) utilises IoT devices and sensors to monitor machines and environments to ensure optimal performance of equipment and processes. Predictive Maintenance (PM) which monitors the health of machines to determine the probable failure of components is one IIoT technique which is receiving attention lately. To achieve effective PM, massive amounts of data are collected, processed and ultimately analysed by Machine Learning (ML) algorithms. Traditionally IoT sensors transmit their data readings to the cloud for processing and modelling. Handling and transmitting massive amounts of data between IoT devices and infrastructure has a cost. Edge Computing (EC) in which both sensors and intermediate nodes can process data provides opportunities to reduce data transmission costs and increase processing speed. This article examines IIoT for PM and discusses how and where data can be processed and analysed. Initially, this article presents sampling and data reduction techniques. These techniques allow for a reduction in the amount of data transmitted to the cloud for processing but there are potential accuracy trade-offs when ML algorithms utilise reduced datasets. An alternative approach is to move ML algorithms closer to the data to reduce data transmission. There are three main techniques that utilise the EC paradigm to perform ML and data processing on intermediary nodes. These techniques are categorized according to where data processing occurs: *Device and Edge*, *Edge and Cloud* and *Device and Cloud (Federated Learning)*. In addition to exploring traditional approaches, these three state-of-the-art techniques are examined in this article and their benefits and weaknesses are presented. A novel architecture to demonstrate how EC can be utilized both for data reduction and PM in IIoT is also proposed.

INDEX TERMS Data reduction & analysis at the edge, machine learning for IoT, predictive maintenance in IIoT, edge computing.

I. INTRODUCTION

The Internet of Things (IoT) is envisioned to make our lives easier. Since its inception, almost every sector has somehow exploited it. For example, it is common to find the use of IoT for smart healthcare [1], [2], agriculture [3], smart homes [4], smart grid [5], [6] and smart industry [7]. Smart industry, also referred to as industry 4.0, makes use of information and communication technologies for efficient productivity [8]. In the context of industry, IoT is known as Industrial IoT (IIoT) and it has gained significant research attention recently [9], [10].

In IIoT, different sensors are employed to monitor the performance of equipment or even a complete production processes [20]. In IIoT, one technique called Predictive Maintenance (PM) has recently gained attention. The basic concept of PM is to monitor machine health with the help of sensing data to determine probable future degradation or failure of the machine. PM employs Machine Learning (ML) on the collected data to make predictions. Indeed, the accuracy of the ML models depends mainly on the collected data.

In IIoT, a traditional approach to collect data is to stream it from sensing devices to the cloud where it is processed and modelled. Sensing devices generate enormous amounts of data, continuously or periodically, often in a very short time frame. For example, within a second, thousands of records can be generated by a machine [16]. According to the Cisco cloud index (2013-2018), an automated facility can generate a terabyte of data every hour. To this end, approaches such as sampling, compression, filtering are used to reduce the data size. These techniques allow for a reduction in the amount of data forwarded to the cloud. However, there are potential accuracy trade-offs for the ML models which utilize reduced datasets.

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Verticale.

TABLE 1. A comparison of existing state-of-the-art surveys and this article differs.

Year	Reference	Edge Computing For IIoT	Machine Learning At Edge	Predictive Maintenance For IIoT	IoT Data Reduction At Edge
2018	[11]	✗	✗	✗	✗
2018	[12]	✓	✗	✓	✗
2018	[13]	✓	✓	✗	✓
2019	[8]	✓	✗	✗	✗
2019	[14]	✓	✓	✗	✗
2019	[15]	✗	✗	✓	✗
2019	[16]	✗	✗	✓	✗
2019	[17]	✓	✓	✗	✗
2020	[18]	✗	✗	✓	✗
2020	[19]	✓	✗	✗	✗
2020	This Article	✓	✓	✓	✓

In the face of cost and other challenges (latency, bandwidth and energy consumption) incurred by the traditional approach, a new computing paradigm called Edge Computing (EC) [31] has recently emerged. EC provides computation and processing nearer to the data source to reduce the data sent to the cloud for processing [19]. In EC, both sensors and intermediate nodes can process data and provide opportunities to reduce data transmission costs. In this respect, developers have options when wishing to reduce the data and associated costs and latency. They can use the limited processing of EC devices (e.g. sensors) to reduce the data being sent to the cloud using various sampling techniques or perform ML for PM on the EC device or even use a hybrid approach in which ML/PM is carried out using EC and the cloud. Another approach, proposed recently by Google is Federated Learning (FL) which seeks to train Deep Learning (DL) models on an edge device with the cloud serving as a global model aggregator [32]. All of the approaches have various trade-offs in terms of data size, transmission cost and accuracy of the ML/PM, which this article explores.

Literature shows that EC can help in meeting the real-time requirements for IIoT [33]. Authors in [34] have provided a range of applications for a smart factory where EC can play a role. There is interest in the research community to propose an optimum solution and so surveys on EC, PM, IIoT or ML have been conducted. However, they typically consider the technological [8], architecture [35], security [36], [37] and systems perspective [13] or focus on the analytics aspect [18] in the IIoT context. This article focuses more on the data and in particular, discusses the location within an IoT network where data can be processed (data reduction or analysis). The article presents the state-of-the-art by

- Reviewing traditional approaches which help in reducing data in IoT. This includes sampling, compression and fusion. These techniques help in reducing the data where generated and result in not only consuming fewer communication resources (e.g. network bandwidth) but also require less cloud resources for storage and computation.
- Presenting the research contributions which have been proposed to push ML closer to the data source. In this, ML training takes place in the cloud and only the model is pushed to the edge nodes. Such approaches can greatly benefit from powerful cloud resources for training complex ML models such as DL.
- Discussing the recently proposed techniques which exploit EC to implement ML for data processing and the PM in IIoT. In this, hybrid approaches in which frameworks use the sensing device and the Edge, are presented. Techniques based on such an architecture can help in meeting stringent latency requirements in some application domains.
- Reviewing the FL paradigm, recently proposed by Google, in which rather than aggregating raw data from devices, the cloud aggregates DL models trained locally on the edge. It is particularly beneficial in meeting privacy requirements when data are confidential that can not be shared with cloud providers in raw form.
- While some observations are presented at the end of each section, some future directions are proposed using reduced data for training ML models and where to implement what part of the PM framework in IIoT. We also propose an Edge-Cloud based architecture that utilises data reduction at the edge informed by network-level information (e.g. congestion) and PM analytics constraints from the cloud (e.g. accuracy) to reduce the data required for analysis tasks.

Figure 1 shows how the literature is categorized while Table 2 summaries the acronyms used in this article. The remainder of this article is organized as follows. Section II presents related surveys published since 2018. We build on the review of the literature from 1993 to 2018 presented in [18]. However, we also discuss PM in conjunction with EC which has emerged recently. In section III, data reduction approaches, which do not employ ML for IoT are discussed. We discuss sampling, compression and fusion techniques. Section IV focuses on the role of ML both for data reduction and PM analytics within EC. In particular, we review techniques that rely on Device-Edge interconnection in subsection IV-A, Edge-Cloud interconnection in subsection in IV-B and FL for IIoT in subsection IV-C. This unfolds where ML can be implemented in the overall architecture of IIoT systems. Section V presents some research challenges and provides future directions to address the need for data reduction and continuous retraining of ML models. Finally, a conclusion is given in section VI.

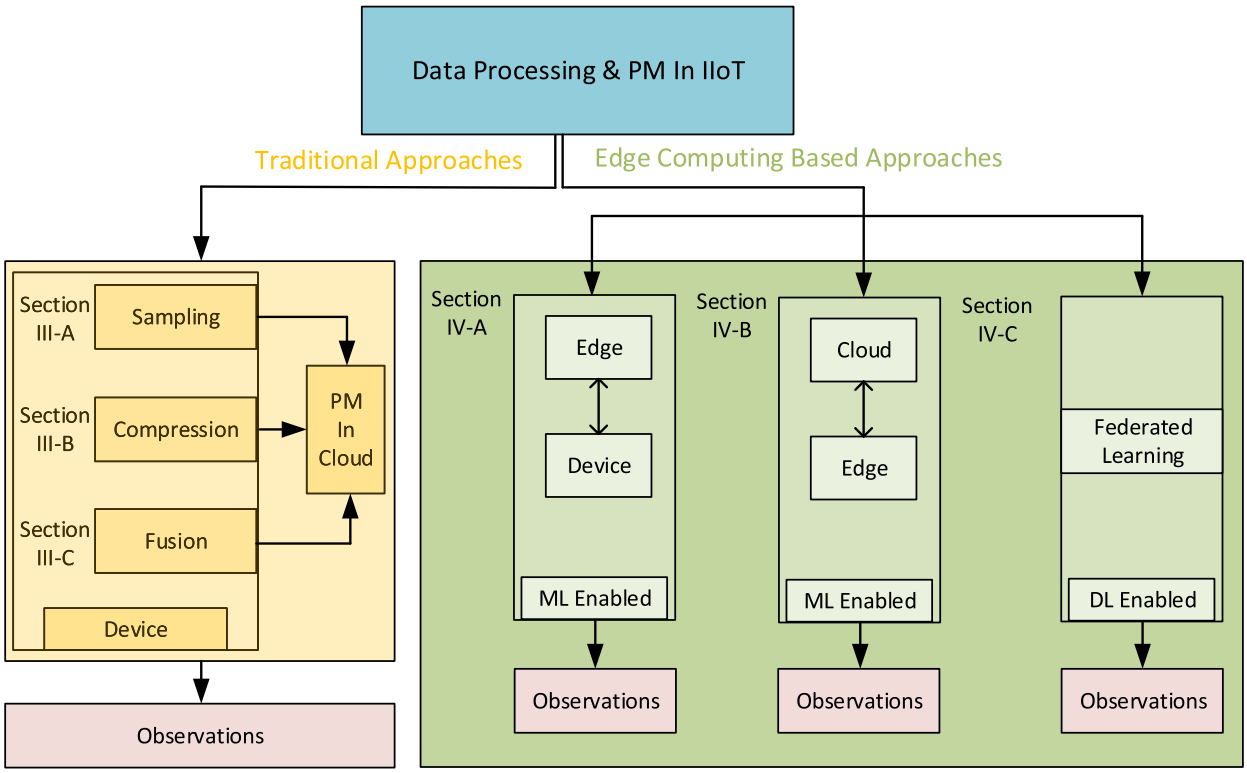

FIGURE 1. The structure and organisation of the literature in this article.

II. RELATED WORK

Since [18] provides a comprehensive review from 1993 to 2018 for PM, this article builds on that and provides a review of the PM primarily since 2018.

EC is a fundamental pillar of modern IIoT systems and has been discussed in many surveys [8], [12]–[14], [17], [19]. Research shows EC can assist in deploying ML models, analytics and data handling, however, existing surveys lack the discussion which realizes EC for these tasks. Table 1 depicts the focus of recent surveys.

Some recent surveys have discussed how ML can be deployed on Edge [13], [14], [17], [38]. [14] discussed the hardware and software frameworks for employing ML at the edge. Likewise, [17] discussed the ML and Artificial Intelligence(AI) implementation in the form of agents. The approach proposed in [13] also covered ML. However, PM analytics in IIoT systems are not considered. Authors of [38] have discussed the role of ML in offloading tasks to the edge. Table 1 summarises the various aspects of the IIoT paradigm that have been reviewed to date.

The importance of PM is seen in recent works [12], [15], [16], [18], however, these works do not consider the benefits or use of EC. In [12] authors focused on discussing the building blocks (such as the equipment, their integration in the system and analytics) of an IoT based smart factory. The articles in [15] explore techniques of PM analytics in IIoT. This includes knowledge-based approaches (ontology, rule-based etc.), techniques using ML models, and approaches involving DL models which help in inference and PM analytics. A comprehensive survey of the PM field is given in [18]. The authors selectively covered the literature from 1993 to 2018 in the field of PM. However, like other analytics, PM also involves data. Meeting real-time latency requirements depends on how data are being collected and processed.

Authors in [13] discussed the role of data. However, data reduction mechanisms are not considered. By searching the data reduction mechanism specifically for IIoT systems, it is clear there are few significant contributions. Therefore, we extended our literature review to consider data reduction techniques within EC in IoT and analytics within the EC paradigm in IIoT systems. Therefore, this article builds on the existing body of knowledge and reviews research efforts made using emerging technologies such as EC and ML to reduce the data as well as performing PM analytics in IIoT systems.

What differentiates this article from related surveys is that 1) it puts forward the available data reduction mechanisms which are very important for future IoT systems, especially in the case of redundant sensing. 2) Unlike related surveys which cover ML literature implementation from the *what* perspective in EC, this article instead focuses on the *where* perspective. It is important to unfold the location in the IoT network where a particular ML framework could be

TABLE 2. A table of acronyms used in this article.

Acronym	Full Form
EC	Edge Computing
IoT	Internet Of Things
IIoT	Industrial Internet Of Things
PM	Predictive Maintenance
AD	Anomaly Detection
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
FL	Federated Learning
LOF	Local Outlier Factor [21]
SZ	Squeeze [22]
AWBS	Adaptive Window Based Sampling [23]
LSTM	Long Short-Term Memory
PSO	Particle Swarm Optimization
GBM	Gradient Boosting Machine
MDCM	Mobile Data Cleaning Mode [24]
ABOD	Angle Based Outlier Detection [25]
CNN	Convolutional Neural Network
SNN	Siamese Neural Network
AE	Auto Encoder
ANN	Autoencoder Neural Network
HOG	Histogram Of Oriented Gradients
AMLC	Accuracy Maximization Offloading With Latency Constraints (AMLC) [26]
SLGT	Supervised Learning Of Genetic Tracking [27]
KNN	K-Nearest Neighbors
QL	Q-Learning
SDN	Software Defined Network
DNN	Deep Neural Network
SGD	Stochastic Gradient Descent
DDoS	Denial Of Service Attacks
RdS-ImS	Sensing-Data Reconstruction Algorithm Under Intelligent-Migration Strategy [28]
TBSS	Transfer By Subspace Similarity [29]
LDA	Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis
GNB	Gaussian Naive Bayes
SVM	Support Vector Machines
DT	Decision Tree
RF	Random Forest
DRL-DAP	Deep Reinforcement Learning Dynamic Adaptive Planning [30]

employed. For instance, if the application demands implementation of ML on a sensor node, the ML algorithm would need to be designed for low power devices.

In the next section, the focus is on data reduction approaches that do not employ ML. The techniques discussed are either implemented on a sensor node or an edge node.

III. TRADITIONAL APPROACHES

Since IIoT is new, not much attention has been given yet to data reduction mechanisms. Therefore, the search criterion for recent papers is expended from IIoT to the more general IoT domain. However, as compared to general IoT, IIoT applications have a variety of sensors and may push more data with higher velocity. In this section, traditional data reduction approaches are discussed. The term *traditional*, in this article, refers to those techniques which do not utilize ML for data reduction, nor are tested for complex IoT analytics such as PM. They are either implemented on a sensing device or the next immediate node which could be a gateway node. Their main purpose is to reduce the size of data that are forwarded to the cloud for analysis. Table 3 provides a summary of the reviewed approaches.

A. SAMPLING

Sampling refers to how frequently data points are taken from the incoming data. For instance, it describes the frequency sensed value(s) are being forwarded; every second, every minute or even every hour. This is generally used in applications having frequent redundant values. Constant temperature monitoring is an example of this. In this case, deduction or decision making can be done with a reduced number of samples. In this subsection, a few recent sampling techniques are reviewed.

ApproxIoT proposed in [40] works by applying reservoir and random sampling on the data stream and associates weights which indicate the significance of the data values at an aggregator. The problem with such an approach is that the multiplication of the weight with the data point eventually changes the values. This aspect makes it unsuitable for applications that demand actual values or values in a particular range. Unlike this approach, the technique proposed in [41], does not alter the values. It is based on two subsets, maximum and minimum values and the aggregating node uses those subsets to obtain an approximated stream. Such an approach may work well for basic queries but has no mechanism to deal with duplicate values which makes it unsuitable for some IoT applications.

The above approaches are designed and tested on basic queries such as average, sum, etc. However, the field of analytics is now matured and advanced analytics are required by today's IoT systems. Particularly in IIoT, advanced analytics include PM in which failure or maintenance of some equipment is detected/predicted ahead of time. Considering such a complex case, the authors in [23] have proposed an Adaptive moving average Window Based Sampling (AWBS) algorithm to reduce the data. The window size varies based on variation in the incoming data. When more variation, window size is reduced to forward more values to the cloud and vice versa. Using their proposed algorithm, they reduced one of NASA's datasets [42] to just 6.91% and passed it to an unsupervised Anomaly Detection (AD) algorithm, Local Outlier Factor (LOF) [21]. Results show that the reduced data have almost similar AD performance as compared to a case when the complete dataset is used or data are reduced using the approach of [40], [41] to 28.95% and 20%, respectively.

B. DATA COMPRESSION

Data compression is an important data reduction approach in which data are compressed before transmitting to the cloud for further processing/analysis. The authors in [43] proposed a compression approach that uses the edge storage concept. It exploits existing compressed data points and saves data either at an edge device or in the cloud without inter-device communication at the edge. Similarly, another

TABLE 3. A summary of data reduction methods used for different data types in an IoT architecture.

Year	Technique	Data Type	Reduction Type	Location of implementation	Performance Evaluation	Use Case /Dataset
2018	ApproxIoT [40]	Ride Fares	Sampling	Edge Server	Latency, Throughput Accuracy	NewYork Taxi [54]
2018	[41]	Temperature	Compression	Sensor	Cosine Similarity, Loss Or Similarity	NA*
2019	[55]	NA*	General	Edge Server	NA*	NA*
2019	[43]	Temperature	Compression	Sensor	Match Probability Match Gain	NA*
2019	[44]	Floating Point Integers	Compression	Sensor	Accuracy,Energy, Computation Time	Stress Detection [45]
2019	[50]	Temperature	Filtering	Network Devices	Data Reduction, Accuracy	Indoor Temperature Monitoring [56]
2019	[49]	Temperature, Humidity, Light	Filtering	Hybrid(Sensor And Edge Server)	Transmission Ratio, Processing Time, Energy Consumption	Indoor Environment Monitoring [56]
2019	TBSS [29]	Accelerometer, Gyroscope	Pre-process	Edge Server Or Cloud	Precision, Recall, F1 score	Human Activity Recognition [57]–[59]
2020	RdS-ImS [28]	Temperature	Compression	Edge Server	Reconstruction Error,	GreenOrbs
2020	AMDC [60]	Current, Energy, Frequency, Power Voltage	Compression	Edge Server	Bandwidth Saving, Normalized Root Mean Squared Error Error	Electric Meter For Household [61]
2020	Adaptive Compression [47]	Image, Floats, Integers,	Compression	Edge Server	Edge-to-Cloud Time, Compression Time, Compression Impact	Energy Meter [62] Cancer Images [63] Satellite Images [64] Oil Well Meter [65]

*NA: Not Available

compression approach called Sensing-data Reconstruction Algorithm under Intelligent-migration Strategy (RdS-ImS) to handle the data streams considering the whole network is proposed in [28]. The authors used the correlation of time-series from different nodes and compressed the data before forwarding. To achieve reliability, a re-transmission mechanism from sensors to edge server is employed. In case communication between an edge node and a sensor is not possible, an edge server estimates the value with the help of a predictive model and sends it to the cloud.

In [44], the authors proposed an energy-efficient approach for compressing multivariate time-series data for IoT devices. The approach tweaks the SZ compression algorithm [22], originally proposed for compressing data of high-performance computing applications. The sensor nodes compress the data using SZ and forward to the edge node which reconstructs the data. As the use case, they used the data set from [45], to determine the stress level of the driver given features such as electrocardiogram signals, respiration and heart rate, to mention a few. Results show that a DL model could effectively predict using compressed data without compromising accuracy. Moreover, the results also show the approach is efficient in terms of computation time and energy consumption of a smart device. However, this approach only works for floating-point values and is tested on labelled data which are not always available in general IoT applications [46].

However, given the heterogeneity of IoT data, compression designed for one case often performs poorly in the other. An adaptive approach for compression is proposed in [47]. The authors have proposed to equip an edge server with several compression approaches and adapt the compression according to the dataset. To determine which one to adopt, they proposed to take a sample of the data and apply compression approaches and select the one which offers a better compression ratio and rate. However, how to select the sample for comparing the approaches is not provided. Moreover, how accurately it compressed the data needs to be evaluated.

C. DATA FUSION

Apart from compressing an individual data stream [28], there is a technique called fusion in which data from various streams are fused to decrease data redundancy, increase data quality, improve reliability, handle missing data and more coverage of the area being monitored [48].

In the research proposed in [49] data are first reduced on the sensor node with the help of Lagrange Polynomials and then sent to the edge server. In the second stage, the edge server reconstructs the data and performs a Kolmogorov–Smirnov test to reduce the data aggregated from several neighbouring nodes and forwards to the cloud. Similarly, in-networking data reduction using two-layer architecture on the edge is proposed in [50]. In the first layer, the data are filtered based on the deviation between the actual value and estimated value, removing the redundancy. For estimating the value, Kalman filtering [51] is employed. This layer passes the data to a *Fusion layer* which is responsible for gathering data from several sensors, removing redundancy, filling the missing data and improving reliability. The quality of the data is still one of the challenges which data

heterogeneity presents in IoT systems [52]. The importance of data quality increases when the goal is to use reduced data in ML models.

D. OBSERVATIONS ON TRADITIONAL SENSOR-CLOUD ARCHITECTURE

Based on the reviewed research, there are some fundamental observations on traditional Sensor-Cloud architecture which are briefly described in this subsection.

Although Sensor-Cloud architecture reduces the data being sent, stored and processed in the cloud [53], traditional approaches such as sampling, compression and fusion are implemented mostly on the device (sensor) itself and analytics are performed in the cloud. However, they are not evaluated or tested for complex analytics such as PM. Even performance of the AWBS [23] is tested only for detecting abnormality of the data points. Therefore, data reduction at the edge needs further exploration. The limited resources of sensor nodes impact the implementation of sophisticated reduction algorithms on them. For example, perceptual importance point-based algorithms have complexity that a sensor can barely handle [50]. The situation becomes more severe with evolving applications of IoT which involve rich data types such as images. Furthermore, developing real-time analytics in the cloud is almost impossible to achieve.

Different IoT applications demand local analytics. For instance, in the IIoT context, based on local analytics, the decision to turn some equipment ON/OFF quickly in a production environment can avoid a catastrophic situation. Analytics depend on ML algorithms which are computationally expensive for some tiny sensors. Also, the energy consumption of tiny sensors has been one of the important concerns even before ML emerged in IoT. Thus, meeting a real-time goal with sensor-cloud architecture seems ambitious. This calls for EC which provides computation power near the data source (sensors), eliminating the latency issues of sensor-cloud architecture. Bringing EC into the architecture creates further possibilities of hybrid architectures that have been adopted in several research works recently. The following section examine recent efforts to increase intelligence at the edge through the use of ML. This can be done by using ML to produce intelligent data sampling or conducting ML/analytics on edge devices.

IV. EDGE COMPUTING BASED & MACHINE LEARNING ENABLED APPROACHES

AI and ML are now fundamental pillars of modern IoT applications. Recently, many efforts have been devoted to this research area. Given the different training time complexity of various ML models, the research community has explored which ML models work best for different IoT applications and contexts. However, not much attention is given to investigate the location *where* ML is most suitably implemented, which is of paramount importance for a few reasons. Firstly, training is a computationally expensive task for which the cloud can offer resources. Secondly, moving massive data volume to cloud storing the data and eventually training ML models using that consumes a lot of resources. Thirdly, where to deploy the prediction model is not consistent in every application. For instance, in a hazardous production facility, prediction on the device or the edge is more important than in the cloud [66] to combat any latency issues.

Unfortunately, deploying ML in an IoT system faces challenges due to constraints of the IoT system. For example, if ML is implemented in the cloud, real-time local decision-making [67] is almost impossible to achieve due to underlying limited bandwidth connectivity between sensing nodes and the cloud. To address the problem, ML can be deployed on the device. However, the limited computing capacity of the sensing nodes is a major challenge. Therefore, a hybrid architecture to implement computation intensive tasks such as training on the cloud and deploying models for prediction on the sensing node has emerged. However, this approach also presents challenges in the case when models require retraining based on new data. In this case, again all of the new data need to be moved to the cloud, incurring costs in terms of latency, energy consumption, and also the use of network resources [68].

Recently, EC which offers computation by residing between sensing nodes and the cloud has emerged. Some of the techniques presented in section IIIutilize EC for data reduction. However, research shows that EC which offers more computation ability than sensing nodes can be exploited to implement ML. Numerous research efforts have been proposed using EC for different IoT applications including PM. Therefore, this section reviews state-of-the-art of deploying ML for data processing and PM analytics in an IoT and IIoT network.

ML offers several advantages including accurate predictions, speed, automation and scalability [69]. Research shows that ML can greatly help with monitoring systems in IIoT [70]. Where on one hand complex DL models are being developed, on the other hand, research on EC is accelerating to provide more computing resources to DL models to support more applications [71]. Various ready-to-use ML frameworks with EC are presented by the authors in [14]. Before ML was used in IIoT, cognitive ability (to learn the environment) of the machines was merely predefined heuristics. However, sophisticated ML algorithms have enhanced cognitive ability by finding patterns in the data and making predictions [30].

This section will unfold the benefits and drawbacks of deploying ML models at different locations (device, cloud, edge or a hybrid) in an IoT network. Table 4 shows a summary of ML-based approaches. In particular, it reveals different aspects of particular approaches. Most importantly, it highlights the locations where specific parts of the implementation are performed such as pre-processing (also includes data reduction mechanisms, if used), training of the model and where the final analytics are performed. Figure 2 depicts the three-layer standard architecture of an IIoT employing EC. It shows that the edge layer which lies between the device and the cloud is a suitable location for deploying ML, therefore

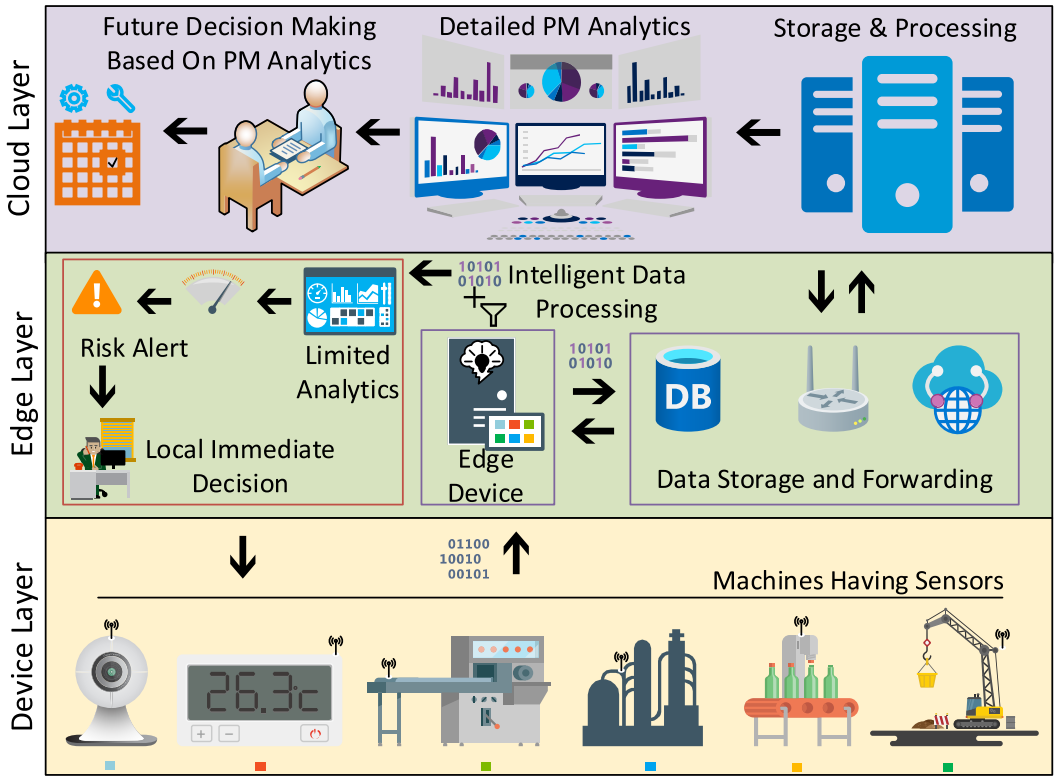

FIGURE 2. An architecture based on [39], having three layers namely device, edge and cloud for PM in IIoT.

provides opportunities to implement the frameworks using hybrid architectures.

A. TECHNIQUES USING DEVICE-EDGE ARCHITECTURE

In this section, techniques that rely on sensing nodes and edge nodes are reviewed.

In [80], authors considered the latency requirements in a proposed hybrid architecture. Their proposed hybrid architecture consists of an edge server and sensing device. Similar to [26] and [66], they also utilise image processing as a use case. Image data can also be converted to time-series text data with the help of EC. For instance, in [88], the edge server first performs pre-processing on the fetched image data and transforms it into text-based time-series data. Once the data are ready, they are passed to the Long Short-Term Memory (LSTM). The second component, as well as the novelty of the approach, is the parameter tuning of the LSTM through Particle Swarm Optimization (PSO).

Sometimes rather than using one model, several models are ensembled to achieve more accuracy. The authors in [83] proposed to deploy a lightweight ML algorithm, Light Gradient Boosting Machine (GBM), on the edge nodes, however, DL is deployed on the master nodes which are edge routers. One of the benefits of the proposed approach is that raw data are not pushed to the master nodes. Instead, Light GBM learns the features from the raw data and passes the learned features to the master node which further increases the accuracy with more computations (using DL). However, the architecture assumes that an edge router can be used to deploy computationally expensive DL models. To implement such an approach, reduced data can help in retraining DL models on the edge node without connecting to the cloud. Overall, ensemble approaches consume more resources as several models are trained.

An alternative approach to ensemble models is online training in which a single model is trained iteratively. Based on this idea, a big data cleaning technique, called Mobile Data Cleaning Model (MDCM) which utilises EC, is proposed in [24]. On the edge server, multidimensional data are cleaned with the help of first employing Angle Based Outlier Detection (ABOD) [25] and then training the ML model. However, MDCM outperformed the compared traditional cleaning model and ABOD, which are the baseline techniques. MDCM has used ML, so it needs to be compared with techniques that also use ML at the edge. Such techniques reduce data at the edge node, however, local transmissions from sensor nodes to edge nodes are not reduced.

To reduce the volume of data transmitted from sensor nodes to the edge server, in [87] authors have proposed to implement ML on the sensor node. The basic idea is to train the ML model on the computationally powerful device (in experiments they used the edge but argued that the cloud can be used too) and push the model to a sensor node. When a new value is sensed, it is passed to the ML model which predicts its label, which is only forwarded to the edge node if it has not been forwarded before. Comparison of different ML algorithm including Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Gaussian Naive

Bayes (GNB), Support Vector Machines (SVM), Decision Tree (DT) and Random Forest (RF). Results showed that SVM outperformed all of mentioned in reducing the data. Such an approach may work well when a single value makes a difference. However, similar to RdS-ImS [28], it does not support the use cases when rather than an individual value, a sequence of values (also called a pattern) is more important, such as IIoT data [72].

Another approach with the same goal as [87], has been proposed in [74]. The authors used the importance of the data as a measure to limit the transmission from sensors to the edge server. Two feature selection methods, namely Impurity and Perturb, gauge the importance of the block of data. The already trained ML model, deployed on the edge server, predicts the spatial information (from which sensor to collect) for the next slot of the time based on already aggregated data. The sensor node actively communicates with the edge server to determine if the next block is worth forwarding. However, this approach works on a distributed level i.e., data from which particular sensor are more important in the next time slot. More specifically it does not reduce the data at the stream level (being pushed by one sensor constantly).

In [73], the authors presented two case studies of novelty detection on the edge of IIoT. In the first case, the condition of an electric motor placed on a kitchen hood (fan extractor to clean the air) is monitored. A microphone detects the vibrational signals and sends them to an IoT gateway where an ML classifier is deployed. Each classifier (or resulting label) has a novelty detection algorithm that is executed, and a novelty score is calculated. The second use case is fault detection of a water filtration plant. LOF [21] is used as an AD algorithm. These algorithms are trained on large datasets before detecting possible anomalous behaviour. The performance of these algorithms needs to be explored when reduced datasets are used.

Based on the discussion, sensor nodes are still part of most frameworks implementation. However, since they have reduced computation resources, they have not been used for computationally expensive tasks such as training. Although they have been used for analytics in some cases, they are mostly used for pre-processing tasks or compressing the data as depicted in Table 3.

1) OBSERVATIONS ON DEVICE-EDGE ARCHITECTURE

Frameworks designed on this architecture have several benefits. First, the architecture requires fewer communication resources as well as less burden on cloud resources. Second, if designed to work independently from the cloud, techniques can even work when there is no connection at all [66]. Third, the edge offers more context awareness as compared to cloud-based systems [19]. Fourth, with this architecture, meeting real-time requirement is possible. In the IIoT context, for example, real-time AD is very important to avoid a catastrophic situation. Fifth, it also meets security and privacy concerns as edge locally processes data.

While using a Device-Edge architecture decreases dependency on the cloud, it raises a few concerns which need to be considered if adopting this architecture for designing frameworks. Firstly, inherited from traditional sensor-cloud architecture, it also uses resources of sensor nodes for implementation. Secondly, even though the edge server is there and sensor nodes can offload computationally expensive tasks, offloading is still not matured and is being explored [90]–[92]. Thirdly, even when a reduction and analytics framework can be implemented using Device-Edge architecture, some of the data still need to be sent to the cloud. This becomes more important for IIoT where an organization can have production facilities in different locations and data from all facilities are required to have a broad picture of services.

To handle highly distributed scenarios in which sensing devices are located at different locations, generally, fusion is used. Research shows that the fusion of data from sensors deployed at different locations can impact the accuracy of further analysis or predictions [48] because fusion reduces data. Authors in [93] proposed a three-level architecture for the healthcare industry. The amount of processing of data at the bottom layer where data are being sensed is less than the middle layer where communication nodes are processing data. Similarly, a global aggregator in the cloud which has more computation power is processing more data than the middle layer from different locations.

Finally, even though an edge node has more resources than a sensor node, it still provides far less resources than the cloud. Research to deploy computationally expensive DL models on edge nodes is still being conducted [94]–[96]. This calls for another potential architecture that involves the edge and the cloud for implementation. The Edge-Cloud architecture is reviewed in the next section.

B. TECHNIQUES USING EDGE-CLOUD ARCHITECTURE

This section reviews research efforts in which edge devices serve the role of middle-ware. More specifically, on one side, an edge device is connected to sensor nodes that collect data. On the other side, the edge device is connected to the cloud. Figure 2 depicts the location of the Edge as a middle entity. This section reviews the approaches that have deployed ML partially on the cloud and the edge server.

Authors in [66] proposed to monitor the real-time data from sensors deployed on equipment in oil/petroleum wells. The goal is to use ML to monitor the equipment performance especially in cases where there is no connectivity available between the site and the back-end cloud, thereby solely depending on the edge. The edge gateway first retrieves sensor data, runs analytics, reports abnormal behaviour and periodically (subject to connectivity) connects with the back-end cloud to update the ML model. The approach uses an ensembler that contains different techniques such as Convolutional Neural Network (CNN), Siamese Neural Network (SNN), Autoendcoder Neural Network (ANN) and Histogram of Oriented Gradients (HOG). Once all models are trained in the

cloud, only one model is pushed to the edge gateway-to save its resources. Furthermore, the approach uses the cloud to pre-process the data which are then used to train ML models in the cloud. Thus, the initial data need to be pushed to the back-end to train the model.

When computing servers are deployed on the edge for real-time data processing, accuracy is also of paramount importance. Based on this concept, the authors in [26] proposed Accuracy Maximization offloading with Latency Constraints (AMLC) as an intelligent edge-cloud approach and a new metric called *service accuracy*. The overall computation offloading, and service involves the following three steps. First, IoT devices estimate the accuracy that edge servers can provide based on ML models deployed on them. The edge server having more accuracy of the ML model is prioritized. The second step involves the estimation of the delay which is involved in offloading the computation task. Then in the final step, when a mobile/sensor device is aware of the delays and accuracy of all the servers (edge + cloud), it sorts the servers in descending order based on accuracy and selects the first one. Similar to [66], it also works on image-based data.

For sensory time-series data, two techniques with the same use case (packaging industry) are proposed in [30] and [81]. In the former, the authors proposed cognitive ability and DL for better knowledge discovery and decision making in IIoT. Their framework is called Deep Reinforcement Learning Dynamic Adaptive Planning (DRL-DAP) has three layers including perception, transmission, and application. In the perception layer, data are collected from the devices using a RESTful application programming interface at the edge nodes. The transmission layer is responsible for using given technologies such as cellular, long-range wide area network and long-term evolution to transmit the data. The cognitive ability which helps in eradicating data ambiguity and building data semantics are also part of this layer. In the last layer, ML optimization models are deployed which help to take the intelligent decision of the production setting being monitored. In [81], the authors proposed Edge-AI which is implemented in a microcontroller as an edge device. Their purpose is to classify vibrational data of sensors placed on a power-train, used in the packaging industry. In both techniques, the ML model was trained on a massive amount of data collected and stored in the cloud earlier. However, in both techniques, EC is used for pre-processing and data analytics.

Techniques to monitor electric equipment have also been proposed. A distributed architecture to monitor an IIoT process is proposed in [82]. Temperature sensors are deployed at different locations of the transformer for measuring time-series temperature values and inferring the level of oil present in the transformer. An agent application gets the data from the IoT devices and makes decisions with the given knowledge. An agent also has a local data repository where results during the computation are saved. However, data are also forwarded to a back-end where an ML model is trained and updated. Similarly, in [84], the health of an electric induction motor is monitored by employing an accelerometer to collect vibrational data. The edge node first pre-processes the data by taking the temporal and spectral features and then passed to a CNN model which is also deployed on the edge node itself to classify if the object is working normally or faulty. The authors proposed to integrate prediction at the edge node to a marine vessel alarm system in case fault is predicted. Likewise, a solution to monitor the bearing health within a machine is proposed in [97] by using a traditional cloud-edge architecture. Different sensors such as temperature, rotation speed, vibration and humidity monitor the relevant parameters. The edge serves the purpose of initial data processing and forwards to the cloud where an ML algorithm predicts the future values and thus the possible equipment failure. However, the solution does not disclose how the edge server performs processing on data streams.

To this end, in [98] a framework called SERENA is proposed for PM using a hybrid cloud-edge computing approach. The sensor nodes send data to an edge gateway where statistical features of the raw data such as average, maximum, and minimum are calculated. Such data are called smart data. The model building module, based on historical smart data, generates the model. This AI/ML model is then used to predict the incoming new data which eventually helps in detecting the abnormal data and thus possible failures. It is worth noting that the ML model building and training take place in the cloud and then the model is pushed to the edge gateway.

The techniques described so far considered a fixed edge device. However, the authors in [27] proposed Supervised Learning of Genetic Tracking (SLGT) in an edge architecture(fixed + mobile) for an industrial park where resources are moved from the production phase to the next using trolleys. The architecture has three parts, namely front-end, near-end and far-end. As the name suggests, the front-end deals with the actuators and sensors deployed on the equipment such as trolleys moving logistics. EC is deployed in the near-end part in a divided way. To be specific, a fixed edge gateway passively receives the signal transmitted using Bluetooth low energy technology and performs pre-processing and forwards data to the back-end cloud. In addition to the fixed edge gateway, there is a mobile edge gateway that actively listens to the transmitted signal. A K-Nearest Neighbors (KNN) algorithm estimates the location zone where a trolley may be at any given time.

Support from network devices has also been investigated. Authors in [85] proposed condition monitoring of an industrial motor. They proposed to take the electric current and vibrational data from sensors, pre-process at the edge and send only the frequency spectra to the cloud. They assumed that network devices support storing of the raw data, which is not possible in many cases.

Based on the discussion and the summary given in Table 4, it is clear that DL algorithms are often used within IIoT. This comes from the fact that DL models are generally trained on the bulk of data and provide high accuracy. A review on deploying DL models on the edge is given in [71]. However,

TABLE 4. A summary of EC frameworks and associated ML algorithms for prediction use cases.

Year	Technique	Data Type	ML Algorithm(s)	Pre-Processing Location	Training Location	Analytics Location	Performance Evaluation	Use Case(s)
2018	[66]	Image	CNN, SNN ANN, HOG	Cloud	Cloud	Edge	NA	Wellhead Rod Pump
2018	[72]	NA*	Regression	NA*	NA*	NA*	NA*	NA*
2019	[73]	9 Sensors	LOF [21]	NA*	Edge	Edge	Classification & Detection Accuracy	Air Ventilation, Water Pump
2019	[74]	GPS Coordinates	Random Forest Regressor [76], [77]	Sensor	NA*	Edge	Root Mean Squared Log Error	Taxi Mobility Demand [75]
2019	UrbanEdge [78]	Time-series	LSTM	Edge	NA*	Cloud	Root Mean Squared Error, Mean Absolute Percentage Error	Building Occupancy, Traffic Volume, Electricity & Air Quality Index
2019	AMLC [26]	Image [79]	CNN	Cloud	Cloud +Edge	Edge	Service Accuracy	Images Classification
2019	DRL-DAP [30]	Sensors	DRL	Edge	Cloud	Edge	Job Completion Time Energy Consumption Variance in Usage	Packaging Candies
2019	Boomerang [80]	Image	CNN	Sensor	Edge	Sensor	Accuracy	Image Recognition
2019	Edge-AI [81]	Vibrational	CNN	Edge	Cloud	Edge	Confusion Matrix	Packaging
2019	[82]	Time-series Temperature	AE	Edge	Cloud	Edge	Time Of Anomaly Detection	Transformer Monitoring
2019	[83]	NA*	DL, LightGBM	Sensor	Edge	Edge Router	Accuracy Vs Anomaly Training Time	Intrusion Detection System
2019	[84]	Vibrational	CNN	Edge	Cloud	Edge	NA*	Induction Motor Health Monitoring
2019	[85]	Vibrational & Electric Current	NA*	Edge	Cloud	Cloud	NA*	Three Phase Induction Motor Health Monitoring
2019	[86]	Energy Demand	DNN & QL	Edge	Cloud	Edge	Delay & Energy Cost	Smart Energy Grid
2020	[24]	NA*	SVM	Edge	NA*	Edge	Delay, Energy	NA*
2020	[87]	Temperature, Humidity, Light	SVM, LDA QDA,RF GNB, DT	Sensor	NA*	Sensor	Number Of Transmissions Accuracy, F1 Score	Human Occupancy In A Room
2020	LSTM-PSO [88]	Image	LSTM	Edge	Edge	Edge	Mean Squared Error, Accuracy, Compute Time	Gold Mining
2020	SLGT [27]	Multiple Sensors	KNN	Edge	Cloud	Edge	Accuracy, Task Duration	Industrial Park
2020	[70]	NA*	QL	Edge	Cloud	Cloud	Number Of Failures	Equipment Failure
2020	[89]	NA*	RL	Edge	Cloud	Edge	NA*	Grid Sorter
*NA: Not Available								

using reduced data to train these models requires further exploration. If they do not provide high accuracy, other models need to be considered or designed. Furthermore, it is also clear that for data reduction, the edge or the device is mostly exploited. However, given the fact that initial training requires much computation, the cloud is still being used in most of the proposed techniques for training the models. In cases where a dedicated edge node is not available, network devices can be exploited too. For instance, authors in [78] describe an architecture called UrbanEdge. They proposed to use network devices such as routers to serve as edge nodes that pre-process the data and forward them to the back-end cloud where DL is used for PM analytics.

The techniques reviewed so far do not learn from positive or negative changes in the environment in which they operate. Fortunately, advances in ML have made it possible to adapt learning based on the environment. The type of ML which fits in such a scenario is Reinforcement Learning (RL). In RL, a learning agent takes an action in a given environment. The environment assigns a reward to the agent based on the outcome of actions. Agent repetitively acts to maximize the reward, which in other terms means the agent has learned the environment well and knows what action is correct to be taken in the next step. Authors in [70] have proposed to gather IIoT data using an edge node (called a gateway node) which forwards to the cloud where a well-known RL algorithm, called Q-Learning(QL), is used to detect failures. The RL algorithm is responsible for detecting the safety of the equipment in the factory. It generates a detection policy with high accuracy to ensure the safety of the equipment. Another approach based on RL is proposed in [89] for a grid sorter in IIoT. A grid sorter is a device that can move an object

in four directions e.g., left, right, up and down. The local edge node forwards sensor values to the cloud which trains a global model and returns the model to the edge in a factory. Each edge node in every factory then retrains an adaptive model based on local factory policy. When a grid sorter moves objects, the agent keeps learning in which direction a grid sorter can accurately move the objects.

RL has also helped in managing network resources in IIoT. Authors in [99] leveraged RL to assign actions to networking and control systems in a combined manner under a dynamic IIoT environment. More specifically, based on the data forwarded from sensing nodes about the system, an extended Kalman filter estimates a system's state which is forwarded to an RL based agent which decides commands for the networking and control. For the networking, it adjusts the modulation type, and for control systems, it tunes the sampling rate of the sensors (frequency of observations). Similarly in [100], the authors leveraged RL in combination with blockchain to manage resources of a distributed Software-Defined Network(SDN) framework for IIoT. In this, RL helps in optimizing computation resources which are shared by cryptography tasks of a blockchain-based distributed SDN network, and non-cryptography tasks. To manage the IoT network of smart energy management, RL is used in combination with EC is in [86]. A Deep Neural Network(DNN) model is trained in the cloud and QL is employed on the edge node. Devices from smart building send scheduling tasks to an RL agent at the edge server which makes decisions locally and if further training is required, forwards to the DNN model in the cloud. Authors in [101] provide a comprehensive review of how RL can be used in blockchain-based IIoT.

1) OBSERVATIONS ON EDGE-CLOUD ARCHITECTURE

Most of the frameworks reviewed in this article followed the Edge-Cloud architecture. This is also evident from Table 4.

Frameworks proposed using this architecture have their advantages. First, they can reap the benefits of both edge and cloud resources. This means that they can support more rich data types such as images. They can also train computationally expensive models such as DL, thanks to the abundance of cloud resources. Second, in this case, no burden on tiny sensor nodes is required as all the computation is either performed on the edge node or offloaded to the cloud. Third, the scalability of the system is easy as applications are globally managed in the cloud and adding another geographical site with the help of an edge node requires less effort. This is also important in the PM use case for IIoT as production/manufacturing facilities can be extended.

Although Edge-Cloud architecture addresses some of the concerns of Sensor-Cloud and Device-Edge architectures, it also presents a few concerns which can play an important role while designing frameworks. Firstly, deciding which part of the application needs to run where requires careful consideration. To be specific, part of the framework deployed on the edge node will meet the real-time requirement and those deployed in the cloud will leverage more computational power. Secondly, it also depends on the underlying networks which connect the edge and the cloud. Lastly, as data privacy and security have been hindering the adoption of IIoT [102], not all production facilities will be willing to store confidential data in the cloud. However, these concerns can be addressed to some extent in the IIoT context. For example, in IIoT, dividing an implementation based on analytics (real-time alarms, PM analytics) can help. The second challenge can be addressed by exploiting the network information such as congestion. A potential IIoT architecture proposed in this article uses these concepts and is discussed in the section V.

C. FEDERATED LEARNING AND IIoT

FL is another computing paradigm that was recently proposed by Google [32]. Rather than storing and training ML models on a centralized location (cloud), FL is used to train local models on the edge/end devices (called clients) where raw data are available. The clients then upload their model updates to a central server which computes a global model using a *Federated Averaging* algorithm. The new global model is then shared with all clients. This approach serves two purposes. Firstly, it meets privacy requirements because raw data are not leaving the source. Secondly, it minimizes the communication burden as only the model updates are forwarded, not the raw data. The approach relies on having sufficient computation resources to train models.

Since FL was proposed, efforts have been made to apply it in IIoT. For instance, for AD, authors in [103] used an FL approach to train an LSTM model for detecting the anomalous behaviour of a sensor in a smart building. Similarly, research by authors in [104] involved detecting anomalies in IIoT scenarios. In this case, an AD model is collaboratively trained on edge devices which is generalized later. Unlike the work in [103], the approach also captured the most important features, with an attention-based CNN model. These features are then passed to LSTM which predicts the future time-series data. Moreover, the approach also provides a mechanism to limit the number of Stochastic Gradient Descent (SGD) updates which the FL client can send to the server, improving communication efficiency. However, when training with FL, parameter selection for DL networks requires attention. For example, the authors of [105] made a specific effort towards such optimization using a PSO approach.

For the aeronautical industry, authors in [106], have combined FL with active learning. A DT model is trained on historic data of an aircraft. Then, during the flight, a local model is trained. During training, client nodes obtain labels for uncertain data from the server while maintaining the communication budget. In the air conditioning industry work has been done to use FL and blockchain to detect device failure in IIoT [107]. To alleviate the issue of class imbalance, a distance-based weighted federated averaging is also proposed. An incentive mechanism, to encourage clients to participate in the learning process is also provided which takes into account the size of the client data and class (normal or abnormal) of that data. RaspberryPi is used as an

edge node which is equipped with two ML models, logistic regression and neural network. This technique is not based on FL as initially proposed by Google where raw data do not leave the source. However, such an idea where there is a middle entity in form of an edge server between sensing devices and the cloud is also gaining attention. The research work proposed in [108] is another effort based on a new hierarchical version of FL. In this, the edge serves as a local aggregator and the cloud as a global aggregator. An edge node aggregates models from sensing nodes and forwards to the global aggregator. Therefore, the edge node serves as an FL server for sensing nodes and as an FL client to the global aggregator.

FL has also been used to increase security and for preventing IIoT from attacks. For instance, authors in [69] have used FL to avoid Denial of Service Attacks (DDoS) in IIoT. Models are trained locally and edge nodes contain detection and analyzing modules for DDoS. These modules have traffic policies and any network traffic must pass them. In case an attack is detected, it is blocked and an update is sent to the cloud. Similarly, authors in [109] proposed an FL based approach to defend against an attack on DNN models in IIoT while others have used FL to detect malware in applications of IIoT [110].

While FL promises privacy, it also faces security challenges. For instance, research shows that parameter sharing with the server is sufficient for an attacker to infer knowledge of underlying data. Moreover, since the models are being trained locally, an adversary can attack local models on a client, eventually affecting the global model at the server [111]. Therefore, several efforts are also made to address these issues. For instance, research work done in [112] is specifically focused on the idea of data mining from several sources while sharing the data in a cipher state. The authors claim that using their approach, a client can share the data in cipher state with the server and the server can train the model using data in cipher state. Similarly, the authors of [111] provide a secure gradient aggregation framework. The authors in [113], [114] provide a comprehensive survey on the security and privacy of FL.

1) OBSERVATIONS ON FL

Although FL has gained significant attention since it was proposed, some challenges are yet to be fully resolved. These are discussed below.

Research shows that a very large number of model updates between edge nodes and server could result in failure of model convergence [115]. In IIoT scenario, in particular, a scalability issue can arise if every IoT sensing node participates as a learning client. Furthermore, Google's Federated Average algorithm does not take into account heterogeneity in the data which exists in industrial data e.g., size and also the distribution of datasets on each edge device could differ [107]. Moreover, sometimes environmental conditions are heterogeneous which have a direct impact on data being recorded. In such a case, a local model update could report negative knowledge to the global aggregator as FL is based on data similarity of the participating clients [116]. However, a new version of FL in which a local server aggregates the data from IoT nodes and serves as a client to a global aggregator can help in overcoming such issues [108].

In FL, two further problems may arise. Firstly, when different nodes have heterogeneity in terms of quantity and quality of the data, it is difficult to decide the weight or importance to assign to an update from a particular node. Secondly, the convergence of the global model depends on the slowest node in the network. In the IIoT context, a sensing node can have poor network conditions which results in a delay in getting updates. In such as case, a naive node dropping approach can have a detrimental effect on the accuracy of the global model given the situation that a model of the slower node was trained on more or better data samples. In this direction, work needs to done, although some contributions are emerging [117], [118].

Finally, limited resources to deploy DL models on a device, unreliability of wireless channels during frequent client-server updates and the trust of the participating clients to share the trained models with the cloud limit the possible applications of FL [119]. Moreover, designing incentive-based FL frameworks to encourage clients to participate in the learning process is still a challenge [120]. Furthermore, FL is developed for only DL as it takes SGD updates from distributed clients. For using other models, it requires modifications [106]. Based on these observations, deploying DL models on tiny sensors in IIoT seems an inappropriate approach. However, the Edge-Cloud architecture can be leveraged to transmit data from sensors to the edge node (where a DL model can be trained) and pass on the model update to the cloud.

While we reviewed some of the state-of-the-art of FL based contributions, and more have been discussed in [121], [122], it is worth noting that FL has not been studied for the PM use case in IIoT except the contribution of authors in [123] in which they compared two algorithms with the FL and non-FL. They revealed that FL can preserve data locally and at the same time achieve similar performance to a traditional non-FL approach when an ML model is trained in a centralized manner. However, more work needs to be done to realize PM using FL in IIoT.

V. CHALLENGES & FUTURE DIRECTIONS

On one hand, given the heterogeneity of data types of IoT systems, one universal data reduction approach seems an infeasible option. On the other hand, an application or scenario-specific data reduction approach is also inconvenient. This is especially true given the growing variety of IoT domains (e.g. smart cities, agriculture, health and environment monitoring). A naive approach could be to design an algorithm based on data types. However, different applications having the same data type generate data at different velocity and volume. Therefore, a more realistic approach would be to design data reduction approaches based on data

types and using EC and ML technologies. When reducing the data, considering the accuracy of the prediction model is important.

Based on the different techniques reviewed in the previous section, the role of EC is important for the future of IIoT systems. The underlying reason is the location and computation ability of the edge. Since the edge is very close to the data source, the data do not need to be transmitted from the sensor to the cloud which adds latency and cost to the system. With EC, data will be processed in near real-time. The computation ability of edge devices means they provide more computation resources than the sensor nodes. This results in shifting more computation burden of the ML algorithms from the cloud to the edge. One such example is online/recurring training of the ML models based on new data.

Retraining is especially important when observations from the machine being monitored deteriorate but are less likely to cause it to fail. In this case, new observations which are still normal would need to be passed to ML for retraining. New ML algorithms can be designed which can be retrained based on environmental change. However, it can be costly (computation and energy) if all of the data are passed for retraining. This calls for data reduction approaches which can help in reducing the data size before passing it to the retraining phase. It would incur less computation and energy consumption while maintaining the latency requirement of PM analytics.

Research on data reduction approaches, based on the accuracy of the ML models, especially for complex analytics such as PM, are needed. Data reduction is yet to be explored given the fact that even state-of-the-art data reduction techniques have tested performance on the reduced data for very basic queries and do not support complex analytics such as PM. The accuracy of the models trained on reduced data should also be a concern when optimizing energy consumption and latency. This is more important in DL approaches which require more data to be trained on. We have seen from Table 4 that most of the time DL models are used for prediction. Moreover, the transmission cost can be reduced if a data reduction mechanism is deployed on the device itself. The literature shows that contextual information learning could be paramount for improving the performance of IoT systems [124], [125]. In IIoT, an approach proposed in [126] is an approach that learns context based on energy, backlog and conflict of participating nodes. Similarly, the authors of [127] proposed learning for task offloading in low latency and ultra-reliable communication scenarios. Therefore, in future, RL can greatly help in IIoT while leveraging EC.

Based on the reviewed PM work, it is evident that there is no de-facto architecture to be followed. From Table 4, different researchers have exploited different network entities for deploying frameworks for PM based on the application. This shows that the field has not fully matured and demands further exploration. The emergence of EC and an ability to deploy ML algorithms on the edge (sensors and EC), has also provided an opportunity. However, it is still unclear what is the best practice on where to implement the different parts of the application. Firstly, since most of the time DL is used, which requires large amounts of data for training, not much attention is given to other algorithms. Even FL only supports DL algorithms. Using other algorithms requires further efforts. However, if models can be trained on reduced data, then data reduction approaches would be helpful. Secondly, data types also demand attention. If there are observations from sensors or images, research needs to be done to determine which ML algorithms can produce the best results and where they can be implemented.

We propose an abstract level architecture of PM for IIoT, shown in Figure 3. This is based on the de-facto three-layer architecture involving EC. The three layers are Device Layer, Edge Layer, and Cloud Layer. Since this is based on Edge-Cloud architecture, the device layer merely consists of IoT devices that are forwarding data to the edge layer. For example, it can be assumed they are placed in a production environment to monitor some equipment. The cloud layer is responsible for detailed data analytics and defining accuracy constraints. However, the role of the edge layer is worth describing here.

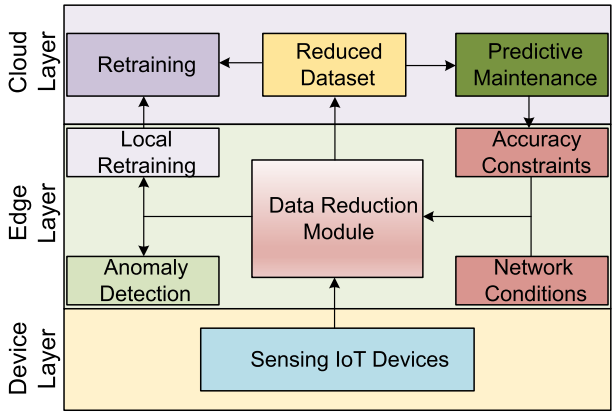

FIGURE 3. A novel IIoT architecture for local anomaly detection and PM cloud analytics.

The core of the edge layer is a data reduction module. This module 1) reduces data, 2) passes data to local analytics such as AD, and 3) forwards data to the cloud for detailed and long-term analytics such as PM. For intelligent data reduction, this module leverages certain information. Firstly, it gets information about the underlying network such as congestion from connected network nodes. For example, when there is more congestion in the network, a smaller number of data samples can be forwarded to the cloud system and similarly when network conditions improve, a greater number of samples can be forwarded. Exploiting network devices for data reduction has been proposed already [50], [78]. Secondly, the data reduction module gets accuracy constraints from the cloud for the phenomena under observation. When greater accuracy is required in the cloud, more samples can be forwarded and vice versa. However, a more intelligent

decision to adopt data reduction would be to use both PM accuracy and network information.

When reducing data, local retraining on reduced data is also possible. For instance, when the data reduction module is extracting a greater number of samples to forward to the cloud for the PM model, data can also be passed to retrain a local AD model to improve its accuracy. Retraining requires storing data at the edge which has also been proposed by authors in [128]. A retrained local model can be pushed to the cloud and can be integrated with a PM model to further improve PM accuracy.

VI. CONCLUSION

This article presents data processing and PM analytics in the IIoT context. Firstly, simple data reduction approaches which do not use ML including sampling, compression, and fusion, are discussed. Secondly, frameworks for data processing proposed specifically for IIoT are presented. The IIoT architecture is dissected and presented. In particular, three categories are discussed; 1) Device and Edge 2) Edge and Cloud 3) FL. In these approaches, we discuss what part of the frameworks is being implemented in which location of an IoT system. Finally, some challenges and future directions are presented. In this, a new architecture for implementing data reduction in conjunction with PM analytic is proposed. The proposed architecture is based on a three-layer EC architecture. It proposes to exploit the edge for data reduction, dynamic local short term decisions and forwarding data to the cloud for detailed data analysis and long-term decisions.

REFERENCES

[1] P. P. Ray, D. Dash, and D. De, "Edge computing for Internet of Things: A survey, e-healthcare case study and future direction," *J. Netw. Comput. Appl.*, vol. 140, pp. 1–22, Aug. 2019.

[2] H. Ahmadi, G. Arji, L. Shahmoradi, R. Safdari, M. Nilashi, and M. Alizadeh, "The application of Internet of Things in healthcare: A systematic literature review and classification," *Univ. Access Inf. Soc.*, vol. 18, no. 4, pp. 1–33, 2019.

[3] M. Lezoche, J. E. Hernandez, M. D. M. E. A. Díaz, H. Panetto, and J. Kacprzyk, "Agri-food 4.0: A survey of the supply chains and technologies for the future agriculture," *Comput. Ind.*, vol. 117, May 2020, Art. no. 103187.

[4] D. Mocrii, Y. Chen, and P. Musilek, "IoT-based smart homes: A review of system architecture, software, communications, privacy and security," *Internet Things*, vols. 1–2, pp. 81–98, Sep. 2018.

[5] H. Hui, Y. Ding, Q. Shi, F. Li, Y. Song, and J. Yan, "5G network-based Internet of Things for demand response in smart grid: A survey on application potential," *Appl. Energy*, vol. 257, Jan. 2020, Art. no. 113972.

[6] F. Al-Turjman and M. Abujubbeh, "IoT-enabled smart grid via SM: An overview," *Future Gener. Comput. Syst.*, vol. 96, pp. 579–590, Jul. 2019.

[7] E. Manavalan and K. Jayakrishna, "A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements," *Comput. Ind. Eng.*, vol. 127, pp. 925–953, Jan. 2019.

[8] G. Aceto, V. Persico, and A. Pescape, "A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3467–3501, Aug. 2019.

[9] E. Oztemel and S. Gursev, "Literature review of industry 4.0 and related technologies," *J. Intell. Manuf.*, vol. 31, no. 1, pp. 127–182, Jan. 2020.

[10] W. Z. Khan, M. H. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, "Industrial Internet of Things: Recent advances, enabling technologies and open challenges," *Comput. Electr. Eng.*, vol. 81, Jan. 2020, Art. no. 106522.

[11] T. Ruppert, S. Jaskó, T. Holczinger, and J. Abonyi, "Enabling technologies for operator 4.0: A survey," *Appl. Sci.*, vol. 8, no. 9, p. 1650, Sep. 2018.

[12] P. K. Illa and N. Padhi, "Practical guide to smart factory transition using IoT, big data and edge analytics," *IEEE Access*, vol. 6, pp. 55162–55170, 2018.

[13] H. Xu, W. Yu, D. Griffith, and N. Golmie, "A survey on industrial Internet of Things: A cyber-physical systems perspective," *IEEE Access*, vol. 6, pp. 78238–78259, 2018.

[14] M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine learning at the network edge: A survey," 2019, *arXiv:1908.00080*. [Online]. Available: http://arxiv.org/abs/1908.00080

[15] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," 2019, *arXiv:1912.07383*. [Online]. Available: http://arxiv.org/abs/1912.07383

[16] L. D. Xu and L. Duan, "Big data for cyber physical systems in industry 4.0: A survey," *Enterprise Inf. Syst.*, vol. 13, no. 2, pp. 148–169, Feb. 2019.

[17] A. Carvalho, N. O. Mahony, L. Krpalkova, S. Campbell, J. Walsh, and P. Doody, "At the edge of industry 4.0," *Procedia Comput. Sci.*, vol. 155, pp. 276–281, Jan. 2019.

[18] C. Krupitzer, T. Wagenhals, M. Züfle, V. Lesch, D. Schäfer, A. Mozaffarin, J. Edinger, C. Becker, and S. Kounev, "A survey on predictive maintenance for industry 4.0," 2020, *arXiv:2002.08224*. [Online]. Available: http://arxiv.org/abs/2002.08224

[19] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-computing-enabled smart cities: A comprehensive survey," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10200–10232, Oct. 2020.

[20] D. Sehrawat and N. S. Gill, "Smart sensors: Analysis of different types of IoT sensors," in *Proc. 3rd Int. Conf. Trends Electron. Informat. (ICOEI)*, Apr. 2019, pp. 523–528.

[21] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2000, pp. 93–104.

[22] S. Di and F. Cappello, "Fast error-bounded lossy HPC data compression with SZ," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2016, pp. 730–739.

[23] T. Hafeez, G. McArdle, and L. Xu, "Adaptive window based sampling on the edge for Internet of Things data streams," in *Proc. 11th Int. Conf. Netw. Future (NoF)*, Oct. 2020, pp. 105–109.

[24] T. Wang, H. Ke, X. Zheng, K. Wang, A. K. Sangaiah, and A. Liu, "Big data cleaning based on mobile edge computing in industrial sensor-cloud," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1321–1329, Feb. 2020.

[25] H.-P. Kriegel, M. S. Hubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2008, pp. 444–452.

[26] W. Sun, J. Liu, and Y. Yue, "AI-enhanced offloading in edge computing: When machine learning meets industrial IoT," *IEEE Netw.*, vol. 33, no. 5, pp. 68–74, Sep. 2019.

[27] Z. Zhao, P. Lin, L. Shen, M. Zhang, and G. Q. Huang, "IoT edge computing-enabled collaborative tracking system for manufacturing resources in industrial park," *Adv. Eng. Informat.*, vol. 43, Jan. 2020, Art. no. 101044.

[28] Z. Sun, X. Zhang, T. Wang, and Z. Wang, "Edge computing in Internet of Things: A novel sensing-data reconstruction algorithm under intelligent-migratoin stragegy," *IEEE Access*, vol. 8, pp. 50696–50708, 2020.

[29] Y. Zhong, S. Fong, S. Hu, R. Wong, and W. Lin, "A novel sensor data pre-processing methodology for the Internet of Things using anomaly detection and transfer-by-subspace-similarity transformation," *Sensors*, vol. 19, no. 20, p. 4536, Oct. 2019.

[30] B. Chen, J. Wan, Y. Lan, M. Imran, D. Li, and N. Guizani, "Improving cognitive ability of edge intelligent IIoT through machine learning," *IEEE Netw.*, vol. 33, no. 5, pp. 61–67, Sep. 2019.

[31] S. Greengard, "Ai on edge," *Commun. ACM*, vol. 63, pp. 18–20, Aug. 2020.

[32] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Stat.*, 2017, pp. 1273–1282.

[33] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.

[34] J.-H. Huh and Y.-S. Seo, "Understanding edge computing: Engineering evolution with artificial intelligence," *IEEE Access*, vol. 7, pp. 164229–164245, 2019.

[35] I. Sittón-Candanedo, R. S. Alonso, S. Rodríguez-González, J. A. G. Coria, and F. De La Prieta, "Edge computing architectures in industry 4.0: A general survey and comparison," in *Proc. Int. Workshop Soft Comput. Models Ind. Environ. Appl.*, 2019, pp. 121–131.

[36] Y. Zhang, H. Huang, L.-X. Yang, Y. Xiang, and M. Li, "Serious challenges and potential solutions for the industrial Internet of Things with edge intelligence," *IEEE Netw.*, vol. 33, no. 5, pp. 41–45, Sep. 2019.

[37] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[38] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102781.

[39] Edge Computing Task Group. *Introduction to Edge Computing in IIoT*. Accessed: Aug. 2, 2021. [Online]. Available: https://www.iiconsortium.org/pdf/Introduction_to_Edge_Computing_in_IIoT%_2018-06-18.pdf

[40] Z. Wen, D. L. Quoc, P. Bhatotia, R. Chen, and M. Lee, "ApproxIoT: Approximate analytics for edge computing," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 411–421.

[41] V. Alieksieiev, "One approach of approximation for incoming data stream in IoT based monitoring system," in *Proc. IEEE 2nd Int. Conf. Data Stream Mining Process. (DSMP)*, Aug. 2018, pp. 94–97.

[42] *Prognostics Center—Data Repository*. Accessed: Dec. 23, 2020. [Online]. Available: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/

[43] N. Yazdani and D. E. Lucani, "Protocols to reduce CPS sensor traffic using smart indexing and edge computing support," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[44] J. Azar, A. Makhoul, M. Barhamgi, and R. Couturier, "An energy efficient IoT data compression approach for edge machine learning," *Future Gener. Comput. Syst.*, vol. 96, pp. 168–175, Jul. 2019.

[45] J. A. Healey and R. W. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 156–166, Jun. 2005.

[46] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 624–635, Apr. 2018.

[47] T. Lu, W. Xia, X. Zou, and Q. Xia, "Adaptively compressing IoT data on the resource-constrained edge," in *Proc. USENIX 3rd Workshop Hot Topics Edge Comput. (HotEdge)*, 2020.

[48] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, and B. Qureshi, "An overview of IoT sensor data processing, fusion, and analysis techniques," *Sensors*, vol. 20, no. 21, p. 6076, Oct. 2020.

[49] H. Harb, C. A. Jaoude, and A. Makhoul, "An energy-efficient data prediction and processing approach for the Internet of Things and sensing based applications," *Peer Peer Netw. Appl.*, vol. 13, no. 3, pp. 780–795, 2020.

[50] W. Ismael, M. Gao, A. Al-Shargabi, and A. Zahary, "An in-networking double-layered data reduction for Internet of Things (IoT)," *Sensors*, vol. 19, no. 4, p. 795, Feb. 2019.

[51] C. Yukun, S. Xicai, and L. Zhigang, "Research on Kalman-filter based multisensor data fusion," *J. Syst. Eng. Electron.*, vol. 18, no. 3, pp. 497–502, Sep. 2007.

[52] E. Adi, A. Anwar, Z. Baig, and S. Zeadally, "Machine learning and data analytics for the IoT," *Neural Comput. Appl.*, vol. 32, pp. 16205–16233, Oct. 2020.

[53] K. Villalobos, B. Diez, A. Illarramendi, A. Goñi, and J. M. Blanco, "I4TSRS: A system to assist a data engineer in time-series dimensionality reduction in industry 4.0 scenarios," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1915–1918.

[54] Z. Jerzak and H. Ziekow, "The DEBS 2015 grand challenge," in *Proc. 9th ACM Int. Conf. Distrib. Event-Based Syst.*, Jun. 2015, pp. 266–268.

[55] P. Korambath, H. Malkani, and J. Davis, "Streaming workflows on edge devices to process sensor data on a smart manufacturing platform," in *Proc. 15th Int. Conf. eScience (eScience)*, Sep. 2019, pp. 621–622.

[56] P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin, and R. Thibaux. (Feb. 2004). *Intel Lab Data*. Accessed: Dec. 23, 2020. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html

[57] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and MitigatingMobile sensing heterogeneities for activity recognition," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2015, pp. 127–140.

[58] B. Kaluža, V. Mirchevska, E. Dovgan, M. Luštrek, and M. Gams, "An agent-based approach to care in independent living," in *Proc. Int. Joint Conf. Ambient Intell.*, 2010, pp. 177–186.

[59] F. Palumbo, C. Gallicchio, R. Pucci, and A. Micheli, "Human activity recognition using multisensor data fusion based on reservoir computing," *J. Ambient Intell. Smart Environ.*, vol. 8, no. 2, pp. 87–107, Mar. 2016.

[60] M. R. Chowdhury, S. Tripathi, and S. De, "Adaptive multivariate data compression in smart metering Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1287–1297, Feb. 2021.

[61] N. Batra, M. Gulati, A. Singh, and M. B. Srivastava, "It's different: Insights into home energy consumption in India," in *Proc. 5th ACM Workshop Embedded Syst. Energy-Efficient Buildings (BuildSys)*, 2013, pp. 1–8.

[62] *NSSDCA Photo Gallery*. Accessed: Dec. 23, 2020. [Online]. Available: https://nssdc.gsfc.nasa.gov/photo_gallery/

[63] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *J. Digit. Imag.*, vol. 26, no. 6, pp. 1045–1057, Dec. 2013.

[64] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the accuracy of appliance identification based on distributed load metering data," in *Proc. Sustain. Internet ICT Sustainability (SustainIT)*, 2012, pp. 1–9.

[65] R. E. V. Vargas, C. J. Munaro, P. M. Ciarelli, A. G. Medeiros, B. G. D. Amaral, D. C. Barrionuevo, J. C. D. D. Araújo, J. L. Ribeiro, and L. P. Magalhães, "A realistic and public dataset with rare undesirable real events in oil wells," *J. Petroleum Sci. Eng.*, vol. 181, Oct. 2019, Art. no. 106223.

[66] B. Boguslawski, M. Boujonnier, L. Bissuel-Beauvais, F. Saghir, and R. D. Sharma, "IIoT edge analytics: Deploying machine learning at the wellhead to identify rod pump failure," in *Proc. Soc. Petroleum Eng. Middle East Artif. Lift Conf. Exhib.*, Nov. 2018.

[67] S. Yin, J. Bao, J. Li, and J. Zhang, "Real-time task processing method based on edge computing for spinning CPS," *Frontiers Mech. Eng.*, vol. 14, no. 3, pp. 320–331, Sep. 2019.

[68] M. De Donno, K. Tange, and N. Dragoni, "Foundations and evolution of modern computing paradigms: Cloud, IoT, edge, and fog," *IEEE Access*, vol. 7, pp. 150936–150948, 2019.

[69] J. Li, L. Lyu, X. Liu, X. Zhang, and X. Lyu, "FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT," 2020, *arXiv:2012.06150*. [Online]. Available: http://arxiv.org/abs/2012.06150

[70] Y. Dong, G. Qin, and H. Tian, "Enhancing data monitoring scheme based on reinforcement learning in IIoT systems," in *Proc. 12th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2020, pp. 69–72.

[71] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep learning for edge computing applications: A state-of-the-art survey," *IEEE Access*, vol. 8, pp. 58322–58336, 2020.

[72] J. Park, H. Park, and Y.-J. Choi, "Data compression and prediction using machine learning for industrial IoT," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 818–820.

[73] G. Burresi, A. Rizzo, M. Lorusso, S. Ermini, A. Rossi, and F. Cariaggi, "Machine learning at the edge: A few applicative cases of novelty detection on IIoT gateways," in *Proc. 8th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2019, pp. 1–4.

[74] Y. Inagaki, R. Shinkuma, T. Sato, and E. Oki, "Prioritization of mobile IoT data transmission based on data importance extracted from machine learning model," *IEEE Access*, vol. 7, pp. 93611–93620, 2019.

[75] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. *CRAWDAD Dataset EPFL/Mobility (V. 2009-02-24)*. Accessed: Dec. 23, 2020. [Online]. Available: https://crawdad.org/epfl/mobility/20090224

[76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[77] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[78] X. Fan, C. Xiang, L. Gong, X. He, C. Chen, and X. Huang, "UrbanEdge: Deep learning empowered edge computing for urban IoT time series prediction," in *Proc. ACM Turing Celebration Conf.*, May 2019, pp. 1–6.

[79] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[80] L. Zeng, E. Li, Z. Zhou, and X. Chen, "Boomerang: On-demand cooperative deep neural network inference for edge intelligence on the industrial Internet of Things," *IEEE Netw.*, vol. 33, no. 5, pp. 96–103, Sep. 2019.
[81] S. Akhtari, F. Pickhardt, D. Pau, A. D. Pietro, and G. Tomarchio, "Intelligent embedded load detection at the edge on industry 4.0 powertrains applications," in *Proc. IEEE 5th Int. Forum Res. Technol. Soc. Industry (RTSI)*, Sep. 2019, pp. 427–430.
[82] L. Thangiah, C. Ramanathan, and L. S. Chodisetty, "Distribution transformer condition monitoring based on edge intelligence for industrial IoT," in *Proc. IEEE 5th World Forum Internet Things (WF-IoT)*, Apr. 2019, pp. 733–736.
[83] H. Yao, P. Gao, P. Zhang, J. Wang, C. Jiang, and L. Lu, "Hybrid intrusion detection system for edge-based IIoT relying on machine-learning-aided detection," *IEEE Netw.*, vol. 33, no. 5, pp. 75–81, Sep. 2019.
[84] V. Asalapuram, I. Khan, and K. Rao, "A novel architecture for condition based machinery health monitoring on marine vessels using deep learning and edge computing," in *Proc. IEEE Int. Symp. Meas. Control Robot. (ISMCR)*, Sep. 2019, pp. C1–C3.
[85] V. De Leon, Y. Alcazar, and J. L. Villa, "Use of edge computing for predictive maintenance of industrial electric motors," in *Proc. Workshop Eng. Appl.*, 2019, pp. 523–533.
[86] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent edge computing for IoT-based energy management in smart cities," *IEEE Netw.*, vol. 33, no. 2, pp. 111–117, Mar. 2019.
[87] A. P. Singh and S. Chaudhari, "Embedded machine learning-based data reduction in application-specific constrained IoT networks," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Mar. 2020, pp. 747–753.
[88] J. Yin, X. Luo, Y. Zhu, W. Wang, L. Wang, C. Huang, and J. Wang, "An edge computing-based predictive evaluation scheme toward geological drilling data using long short-term memory network," *Trans. Emerg. Telecommun. Technol.*, p. e3888, Feb. 2020.
[89] K. Kim and Y.-G. Hong, "Industrial general reinforcement learning control framework system based on intelligent edge," in *Proc. 22nd Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2020, pp. 414–418.
[90] L. Lei, H. Xu, X. Xiong, K. Zheng, and W. Xiang, "Joint computation offloading and multiuser scheduling using approximate dynamic programming in NB-IoT edge computing system," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5345–5362, Jun. 2019.
[91] J. Kang and D.-S. Eom, "Offloading and transmission strategies for IoT edge devices and networks," *Sensors*, vol. 19, no. 4, p. 835, Feb. 2019.
[92] M. Wang, L. Zhu, L. T. Yang, M. Lin, X. Deng, and L. Yi, "Offloading-assisted energy-balanced iot edge node relocation for confident information coverage," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4482–4490, Jun. 2019.
[93] R. Dautov, S. Distefano, and R. Buyya, "Hierarchical data fusion for smart healthcare," *J. Big Data*, vol. 6, no. 1, pp. 1–23, Dec. 2019.
[94] F. Liang, W. Yu, X. Liu, D. Griffith, and N. Golmie, "Toward edge-based deep learning in industrial Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4329–4341, May 2020.
[95] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan. 2018.
[96] A. M. Ghosh and K. Grolinger, "Edge-cloud computing for Internet of Things data analytics: Embedding intelligence in the edge with deep learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2191–2200, Mar. 2021.
[97] C. Cheng, B.-K. Zhang, and D. Gao, "A predictive maintenance solution for bearing production line based on edge-cloud cooperation," in *Proc. IEEE Chin. Autom. Congr. (CAC)*, Nov. 2019, pp. 5885–5889.
[98] D. Bowden, A. Marguglio, L. Morabito, C. Napione, S. Panicucci, N. Nikolakis, S. Makris, G. Coppo, S. Andolina, A. Macii, and E. Macii, "A cloud-to-edge architecture for predictive analytics," in *Proc. Workshops EDBT/ICDT 2019 Joint Conf.*, 2019.
[99] H. Xu, X. Liu, W. Yu, D. Griffith, and N. Golmie, "Reinforcement learning-based control and networking co-design for industrial Internet of Things," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 885–898, May 2020.
[100] J. Luo, Q. Chen, F. R. Yu, and L. Tang, "Blockchain-enabled software-defined industrial Internet of Things with deep reinforcement learning," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5466–5480, Jun. 2020.
[101] F. Jameel, U. Javaid, W. U. Khan, M. N. Aman, H. Pervaiz, and R. Jäntti, "Reinforcement learning in blockchain-enabled IIoT networks: A survey of recent advances and open challenges," *Sustainability*, vol. 12, no. 12, p. 5161, Jun. 2020.
[102] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial Internet of Things security: Requirements and fog computing opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2489–2520, 4th Quart., 2020.
[103] R. A. Sater and A. Ben Hamza, "A federated learning approach to anomaly detection in smart buildings," 2020, *arXiv:2010.10293*. [Online]. Available: http://arxiv.org/abs/2010.10293
[104] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, early access, Jul. 24, 2020, doi: 10.1109/JIOT.2020.3011726.
[105] B. Qolomany, K. Ahmad, A. Al-Fuqaha, and J. Qadir, "Particle swarm optimized federated learning for industrial IoT and smart city services," 2020, *arXiv:2009.02560*. [Online]. Available: http://arxiv.org/abs/2009.02560
[106] N. Aussel, S. Chabridon, and Y. Petetin, "Combining federated and active learning for communication-efficient distributed failure prediction in aeronautics," 2020, *arXiv:2001.07504*. [Online]. Available: http://arxiv.org/abs/2001.07504
[107] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu, "Blockchain-based federated learning for device failure detection in industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5926–5937, Apr. 2021.
[108] Y. Ye, S. Li, F. Liu, Y. Tang, and W. Hu, "EdgeFed: Optimized federated learning based on edge computing," *IEEE Access*, vol. 8, pp. 209191–209198, 2020.
[109] Y. Song, T. Liu, T. Wei, X. Wang, Z. Tao, and M. Chen, "FDA3: Federated defense against adversarial attacks for cloud-based IIoT applications," *IEEE Trans. Ind. Informat.*, early access, Jun. 30, 2020, doi: 10.1109/TII.2020.3005969.
[110] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "FED-IIoT: A robust federated malware detection architecture in industrial IoT," *IEEE Trans. Ind. Informat.*, early access, Dec. 9, 2020, doi: 10.1109/TII.2020.3043458.
[111] A. Fu, X. Zhang, N. Xiong, Y. Gao, H. Wang, and J. Zhang, "VFL: A verifiable federated learning with privacy-preserving for big data in industrial IoT," *IEEE Trans. Ind. Informat.*, early access, Nov. 6, 2020, doi: 10.1109/TII.2020.3036166.
[112] L. Kong, X.-Y. Liu, H. Sheng, P. Zeng, and G. Chen, "Federated tensor mining for secure industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2144–2153, Mar. 2020.
[113] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.
[114] M. Asad, A. Moustafa, and C. Yu, "A critical evaluation of privacy and security threats in federated learning," *Sensors*, vol. 20, no. 24, p. 7182, Dec. 2020.
[115] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data," 2018, *arXiv:1811.11479*. [Online]. Available: http://arxiv.org/abs/1811.11479
[116] T. Hiessl, D. Schall, J. Kemnitz, and S. Schulte, "Industrial federated learning–requirements and system design," in *Proc. Int. Conf. Practical Appl. Agents Multi-Agent Syst.*, 2020, pp. 42–53.
[117] F. Malandrino and C. F. Chiasserini, "Federated learning at the network edge: When not all nodes are created equal," 2021, *arXiv:2101.01995*. [Online]. Available: http://arxiv.org/abs/2101.01995
[118] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 1698–1707.
[119] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Trans. Ind. Informat.*, early access, Aug. 18, 2020, doi: 10.1109/TII.2020.3017668.
[120] L. U. Khan, S. R. Pandey, N. H. Tran, W. Saad, Z. Han, M. N. H. Nguyen, and C. S. Hong, "Federated learning for edge networks: Resource optimization and incentive mechanism," *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 88–93, Oct. 2020.

[121] J. C. Jiang, B. Kantarci, S. Oktug, and T. Soyata, "Federated learning in smart city sensing: Challenges and opportunities," *Sensors*, vol. 20, no. 21, p. 6230, Oct. 2020.
[122] Q. V. Pham, K. Dev, P. K. R. Maddikunta, T. R. Gadekallu, and T. Huynh-The, "Fusion of federated learning and industrial Internet of Things: A survey," 2021, *arXiv:2101.00798*. [Online]. Available: http://arxiv.org/abs/2101.00798
[123] N. Ge, G. Li, L. Zhang, and Y. L. Y. Liu, "Failure prediction in production line based on federated learning: An empirical study," 2021, *arXiv:2101.11715*. [Online]. Available: http://arxiv.org/abs/2101.11715
[124] L. I. Carvalho, D. M. A. da Silva, and R. C. Sofia, "Leveraging context-awareness to better support the IoT cloud-edge continuum," in *Proc. 5th Int. Conf. Fog Mobile Edge Comput. (FMEC)*, Apr. 2020, pp. 356–359.
[125] M. A. Hasnat, M. J. Hossain, A. Adeniran, M. Rahnamay-Naeini, and H. Khamfroush, "Situational awareness using edge-computing enabled Internet of Things for smart grids," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.
[126] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S. H. Ahmed, and A. K. Bashir, "Learning-based context-aware resource allocation for edge-computing-empowered industrial IoT," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4260–4277, May 2020.
[127] Z. Zhou, Z. Wang, H. Yu, H. Liao, S. Mumtaz, L. Oliveira, and V. Frascolla, "Learning-based URLLC-aware task offloading for Internet of Health things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 396–410, Feb. 2021.
[128] Y. Yang, Q. Cao, and H. Jiang, "EdgeDB: An efficient time-series database for edge computing," *IEEE Access*, vol. 7, pp. 142295–142307, 2019.

TAIMUR HAFEEZ (Graduate Student Member, IEEE) received the B.S. degree from the COMSATS Institute of Information Technology, Islamabad, Pakistan, in 2016, and the M.S. degree from the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, in 2018. He is currently pursuing the Ph.D. degree with the School of Computer Science, University College Dublin, Ireland. He was one of the 100 selected young Computer Science Researchers at Heidelberg Laureate Forum, Germany, 2020. His research interests include the Internet of Things (IoT) and machine learning. He is a Student Member of ACM, ACM SIGCSE, and the Irish Computer Society.

LINA XU received the B.E. degree from Software Engineering School, Fudan University, China, the B.Sc. degree in computer science from University College Dublin (UCD) through a joint program, and the Ph.D. degree in computer science from UCD, in 2014. She was a Research Scientist with HP Labs, from 2014 to 2016. She has been an Assistant Professor with the School of Computer Science, UCD, since 2016. She is currently working on a smart living project aiming to apply the IoT technologies to smart living. Her research interests include the Internet of Things (IoT), 5G networks, smart networking, and machine learning.

GAVIN MCARDLE is currently an Assistant Professor with the School of Computer Science, University College Dublin (UCD), and a Collaborator with the CeADAR–Centre for Applied Data Analytics Research. He has an extensive publication record, including edited books, book chapters, articles, and research articles. He has received several grants from the National and European funding agencies to support collaboration with researchers in academia and industry. His research interests include location-based services, user profiling, geo-visual analysis, smart transport, smart city technology, and urban dynamics. He has been a keynote speaker at several international conferences and institutes.