sensors

MDPI

Review

An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques

Rajalakshmi Krishnamurthi [1], Adarsh Kumar [2], Dhanalekshmi Gopinathan [1], Anand Nayyar [3,4,*] and Basit Qureshi [5]

[1] Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida 201309, India; k.rajalakshmi@jiit.ac.in (R.K.); dhanalekshmi.g@jiit.ac.in (D.G.)
[2] School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, India; adarsh.kumar@ddn.upes.ac.in
[3] Graduate School, Duy Tan University, Da Nang 550000, Vietnam
[4] Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam
[5] Department of Computer Science, Prince Sultan University, Riyadh 11586, Saudi Arabia; qureshi@psu.edu.sa
* Correspondence: anandnayyar@duytan.edu.vn

Received: 21 August 2020; Accepted: 22 October 2020; Published: 26 October 2020

Abstract: In the recent era of the Internet of Things, the dominant role of sensors and the Internet provides a solution to a wide variety of real-life problems. Such applications include smart city, smart healthcare systems, smart building, smart transport and smart environment. However, the real-time IoT sensor data include several challenges, such as a deluge of unclean sensor data and a high resource-consumption cost. As such, this paper addresses how to process IoT sensor data, fusion with other data sources, and analyses to produce knowledgeable insight into hidden data patterns for rapid decision-making. This paper addresses the data processing techniques such as data denoising, data outlier detection, missing data imputation and data aggregation. Further, it elaborates on the necessity of data fusion and various data fusion methods such as direct fusion, associated feature extraction, and identity declaration data fusion. This paper also aims to address data analysis integration with emerging technologies, such as cloud computing, fog computing and edge computing, towards various challenges in IoT sensor network and sensor data analysis. In summary, this paper is the first of its kind to present a complete overview of IoT sensor data processing, fusion and analysis techniques.

Keywords: Internet of Things; data processing; data analysis; data fusion; emerging technologies

1. Introduction

In the coming years of the Internet of Things (IoT), context-awareness bridges the interconnection between the physical world and virtual computing entities, and involves environment sensing, network communication, and data analysis methodologies [1]. Advancement enables several advanced IoT applications, such as intelligent healthcare systems, smart transport systems, smart energy systems and smart buildings. The IoT networks' unified architecture includes smart IoT-based application services and the underlying IoT sensor networks [2]. According to the Gartner forecast, the IoT global market envisions 5.8 billion IoT-based applications by 2020, with a 21% increase from 2019 [3]. Further, the IoT market's worldwide growth is propelled by wireless networking technologies and the adoption of emerging technologies such as cloud platforms. This trend leads to a drastic increase in demand for connected IoT devices and application services.

The primary objectives of IoT sensor networks include (i) sensing the critical information from the external physical environment, (ii) the sampling of internal system signals, and (iii) obtaining meaningful information from sensor data to perform decision-making [4,5]. It is to be noted that IoT-enabled applications involve a wireless sensor network (WSN). Further, these wireless sensors are randomly positioned and capable of establishing an ad hoc network without infrastructure requirements. The wireless sensor network is reinforced by low-cost and lower power devices, such as Wi-Fi, Bluetooth, Zigbee, Near Frequency Communication, etc. However, these wireless-based networks incur difficulties, such as inference, loss of data, redundancy of data and different data generation [6,7].

It can be observed that the raw sensor data from IoT sensors embed-large scale unclean and useless data. Thus, the raw sensor data need to undergo data cleaning processing, and then data analysis can be performed to obtain relevant information from this cleaned IoT sensor data [8,9]. Further, the large quantity of unwanted and useless data can lead to high computation costs and the overutilization of resources in a constrained IoT sensor network. The most common data processing techniques are data denoising, data imputation, data outlier detection, and data aggregation [10].

It is observed that the raw sensor data exhibit unwanted changes and modifications in the original signal. This raw data signal left untreated leads to expensive resource utilization and computation requirement. As such, the raw sensor signal's data processing is essential, and a variety of existing solutions are addressed in this paper.

In the IoT sensor network, the nodes are distributed, and several nodes are used to perform the same operation. Hence data integration or fusion from multiple sensors is required to improve accuracy in various IoT-based application services [11–13]. For example, in a real-time traffic monitoring system, data patching is useful for data fusion. The previous week's data are then fused to other data from the time of loss of data. This process involves count-and-classify data, loop-based data, vehicle speed measurement, automated number plate recognition (APNR), etc. As such, in this paper, the details of data fusion are also addressed.

A further dimension of the IoT sensor network addresses the fact that the IoT sensors' data possess complex properties, such as voluminousness, veracity and velocity. Thus, it is essential to store this data for performing data analysis and achieve the desired outcome for IoT sensor-based applications. IoT sensor network integration with emerging technologies provides efficient methods to handle sensor data's dynamic and complex nature. Furthermore, machine learning and deep learning techniques provide a promising solution towards the analysis of IoT sensor data [14–16]. Incorporating these data analysis techniques results in deep insights into sensor data, and provides good knowledge related to hidden data patterns and further decision-making. In this respect, this paper elaborates on various existing data analysis approaches.

It is observed that several works exist with each focusing on specific problems and issues associated with IoT sensor data. However, there is a lack of papers offering a complete overview of various IoT sensor data techniques, such as data processing, data analysis and data fusion.

The overall contributions of this paper are listed below:

- To provide an overview of various data analysis techniques for IoT sensor data;
- To explain the basic architecture for IoT sensor data processing, fusion, and analysis. Further, the interaction of these modules along with the IoT sensor network;
- To discuss the IoT sensor output characteristics, such as the voluminous IoT sensor data, heterogeneity, real-time processing, and scalability factors;
- To explain the mechanism of data processing techniques so as to address various issues in IoT sensor data, such as data denoising, missing value imputation, data outlier detection, and data aggregation;
- To address the importance of deep learning and machine learning models for IoT sensor data analysis. The need to integrate merging technologies such as cloud, fog, and edge computing towards the efficient computation of data analytical models.

Section 2 elaborates on the basic architecture of IoT sensor data processing, analysis and data fusion. Furthermore, the characteristic of sensor data is discussed in this section. Next, Section 3 details the different data processing methods, such as data denoising, missing data handling, data outlier detection, and data aggregation. Section 4 elaborates on the data fusion methods, and Section 5 presents the data analysis mechanisms. Finally, Section 6 concludes this paper.

2. Basic Architecture

In the technology era of the Internet of Things (IoT), the interconnection of physical things with virtual objects aims to enhance human life quality through advanced applications and growing sensor technology, communication networks and processing methodologies. A few examples of such advanced applications include connected smart city, intelligent transport systems, smart healthcare, smart building and smart grids. Towards enabling rapid advancement in IoT, the sensor networks play a vital role by sensing critical data from both the internal functional systems and the external environmental factors [17,18]. The wireless-based sensor network is much more popular, as these networks can be deployed ad hoc without the prerequisite of any infrastructure. The wireless sensor network is capable of self-organizing and can be deployed randomly.

Figure 1 depicts the basic architecture for IoT sensor data processing, fusion and analysis layers. The IoT sensor data layer primarily consists of various IoT sensors that can measure physical surroundings and capture real-time environment changes. The commonly used IoT sensors include temperature, pressure, humidity, level, accelerometer, gas, gyroscopes, motion sensors image, optical sensors, Radiofrequency Identifier (RFID) sensors, and Infra-Red (IR) sensors. The IoT sensors are mainly associated with the microprocessing unit, storage unit, control unit, power system and wireless communication interfaces. The IoT sensor devices are constrained in size, computing power, memory, networking capability and storage space. Wireless communication protocols, such as Wi-Fi, Zig Bee, Bluetooth, Near Frequency Communication (NFC) and LTE/4G mobile technologies, are commonly used for IoT sensor device communication.

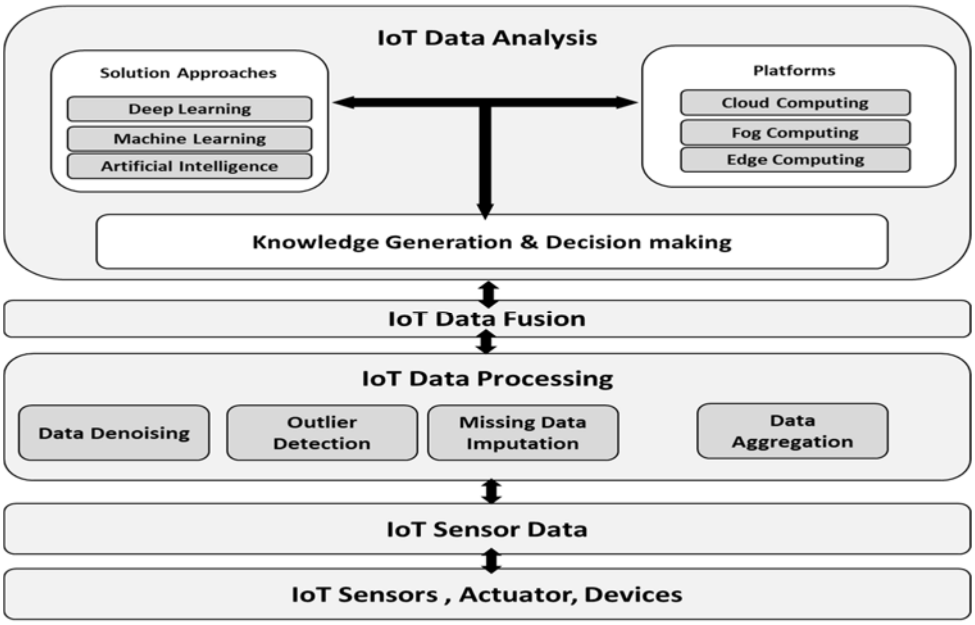

Figure 1. The basic architecture for IoT sensor data processing, data fusion and data analysis.

The majority of IoT sensor data incorporate real-time processing for industrial applications, healthcare, and scientific activities. For example, the healthcare body sensors to monitor the patients' critical conditions would generate massively voluminous data. These sensed data must be processed to remove uncertainties for further data analysis, so as to develop knowledge and decision-making. Thus, the data processing layer targets different functions, such as data denoising, data outlier detection, missing data imputation and data aggregation.

The data fusion layer is required to handle various sensor data challenges generated by several heterogeneous sensor devices. The data fusion data aim to integrate true sensor data from heterogeneous IoT sensor devices. The combined data from different sources are then passed to the data analysis layer for efficient knowledge generation and decision-making.

The primary data fusion involves the direct fusion of data sensor data from different sensor devices. It incorporates initial feature extraction which is followed by data fusion. The enhanced method involves feature extraction followed by identity declaration and data fusion [19,20]. This method allows for a high level of inferences of knowledge and much accurate decision-making. In recent years, the adoption of emerging technologies has revolutionized cloud computing, fog computing and edge computing towards IoT sensor data analysis. These enabling technologies provide a pervasive, reliable and convenient platform to handle IoT sensor data's dynamic, heterogeneous nature [21,22]. As such, the data analytic layer aims at developing smart functionality to address a wide variety of IoT-based applications. The objectives of these platforms are to reduce the computation and storage cost, improve network transmission reliability, reduce the network delay, enhance IoT network security and privacy, ensure scalability, and allow failure- and risk-free IoT solutions.

Characteristics of IoT Sensor Data

The IoT sensors generate data consecutively or upon the trigger of an external event. The other process involves data generated by sensor nodes that need to be gathered, aggregated, analyzed and visualized to obtain useful information. This information is then interpreted to produce the representable form, that is deliverable, and a reaction towards the external trigger. In addition to the data generated by sensor networks, other sources also have data streams. As such, the data generated are required to be aggregated and warehoused in an unprecedented manner, and streamed at a specific network data rate into remote locations for historical data analysis. However, there are several sensor data characteristics and problems associated with this. The authors in [23] discuss that sensor data exhibits information complexity due to factors like the huge volume, the dynamism of data, real-time updating, critical data aging, and interdependency between different data sources. Generally, the sensors are implanted into the human body, objects or locations. As such, the significant characteristics of IoT sensor data are as given below:

Technical Constraints—The limited size of the sensor leads to technical constraints such as computing power, battery power, networking capability, storage capacity and memory. As such, these sensors are highly vulnerable to failure, attacks, and easy breakdown, thus leading to losses of sensor data and inaccurate information;

Real-Time Processing—The sensor network will be capable of more complex networking tasks, and can perform the transformation of raw sensor data into more valuable and insightful information in real-time;

Scalability—In the physical world, the sensor network includes data sources from numerous sensors and actuators. Sensor networks must be scalable to accommodate the exponential growth of sensors and actuators, data handling, and meet the various objective of IoT-based applications;

Data Representation—The general format of sensor data is as a small-sized tuple with structured information. The various representations of sensor data are Boolean, binary, featured values, continuous data, and numeric values;

Heterogeneity—IoT sensor data are heterogeneous. There are different data sources, including rigidly structured data sets, real-time data-generating information networks, embedded systems with sensors, social network media data stream, and other participatory sensor networks.

3. IoT Sensor Data Processing

In IoT sensor networks, wireless communication protocols are popularly used for the information exchange process. These communication protocols work as unlicensed frequency bands that ease the flexibility and scalability of sensor deployments. However, the utilization of communication protocols for WSN under unlicensed frequency bands causes uncontrollable interference. The interference signals may lead to improper data transmission and sensor data with noise, missing values, outliers and redundancy. This section elaborates on the various data analyses performed to handle IoT sensor data issues such as denoising, missing data imputation, data outlier detection and data aggregation.

3.1. Denoising

The voluminous sensor data generated in the IoT network needs data analysis, mostly with real-time decision-making. The characteristics of sensor data are complex, involving high velocities, huge volumes, and dynamic values and types. Further, the sensor data pollute while perpetuating numerous obstacles until producing the required data analysis and real-time decision-making.

Noise is an uncorrelated signal component that enacts unwanted change and modification on the original vectors of the signal. The noise feature leads to the unnecessary processing and utilization of resources for handling the unusable data. The wavelet transform methods are capable of representing the signal and addressing the problem of signal estimation effectively. Significantly, the wavelet transformation preserves the original signal coefficients by removing the noise within the signal. This is achieved by thresholding the coefficient of noise signals, hence the perfect thresholding scheme is essential. The wavelet transformation is a prevalent method to analyze and synthesize continuous-time signal energy.

Let $e(t)$, $t \in$ R represent the signal energy, while it must satisfy the constraint defined as

$$\|e\|^2 = \int_{-\infty}^{\infty} \left| e(t) \right|^2 dt < \infty, \tag{1}$$

where the signal energy $e(t)$ that satisfies the constraint in Equation (1) belongs to the squared search space $L^2(R)$. The wavelet transformation is also used to analyze the discrete-time signal energy and eliminate the noise with energy signals. The wavelet transformation method allows us to investigate the signal characteristic by zooming at different time scales. The experimental results exhibited significant improvements in denoising the sensor signals.

There are two types of wavelet transforms, namely Continuous Wavelet Transform (CWT), which targets signal analysis on a time-frequency level, and Discrete Wavelet Transform (DWT) that targets signal analysis a time level.

Continuous Wavelet Transform (CWT): In CWT, the signal energy $e(t)$ is represented using a set of wavelet functions $C = \left\{ W_\psi(\alpha, \beta) \right\}, \alpha \in R^+; \beta \in R$, where α represents the dilation scaling factor, and β represents the shifting time localization factor, while ψ represents the wavelet function. The wavelet coefficient on the time-frequency plane is given by Equation (2).

$$W_\psi(\alpha, \beta) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\alpha}} \psi_0 \left(\frac{\lambda - \beta}{\alpha} \right) e(\lambda) d\lambda \tag{2}$$

where ψ_0 represents the shifted and dilated form of the original wavelet $\psi_0(t)$. The CWT is a function controlled by two parameters. The CWT targets to find the coefficients of the original signal $e(t)$ based on the shifting factor (β) and the dilation factor (α).

Discrete Wavelet Transformation (DWT): The DWT for continuous-time signals refers to transforming signals performed upon a discrete-time signal. The coefficients obtained from this transformation are defined in subset $D = W_{\psi}(2^{\alpha}, 2^{\alpha}\beta), \alpha \in \mathbb{Z},\ \beta \in \mathbb{Z}$. For a given continuous-time signal $e(\lambda)$, the coefficients of DWT are obtained using the integration of the subset D, as defined in Equation (3).

$$w(\alpha, \beta) = \left(\psi_{0(2^{\alpha}, 2^{\alpha}\beta)},\ e\right) = \int_{-\infty}^{\infty} 2^{-\alpha/2} \psi_0{}'(2^{-\alpha}\lambda - \beta) e(\lambda) d\lambda \tag{3}$$

where α indicates the scale factor and β indicates the localization factor. It is to be noted that this involves continuous-time signal $e(\lambda)$, and not the discrete signal.

The authors in [24] discuss that in some instances the signals received from the IoT sensor devices have a reasonable ratio value of Signal to Noise (SNR), but are unable to achieve the required Bit Error Rate (BER). To overcome such problems, the best solution is to eliminate the inferior wavelet coefficients. This elimination improves the SNR, based on a specific threshold limit. This is possible as the smaller coefficients tend towards more noise data than the desired signal data. Further, it is to be noted that the energy signals are concentrated on a particular part of the signal spectrum. As such, if that specific part of the signal spectrum has been transformed using wavelet coefficients, it improves the SNR value. Further, if the signal function has large regions of irregular noise and small smooth signal regions, then the wavelet coefficients play a vital role in improving the signal energy. Thus, if any signal function is polluted by larger noise, the wavelet coefficients are affected in the more significant part of the wavelet coefficients. The original signal will be contained within the small parts of wavelet coefficients. Thus, maintaining the right threshold limit would eliminate the majority of noise signals and retain the original signal coefficients. In this paper, the authors elaborate on the streaming of sensor data and raw sensor signals to recognize the characteristics and various issues related to noise associated with the sensor signals.

3.2. Missing Data Imputation

Imputation is an essential pre-processing task in data analysis for dealing with incomplete data. Various fields and industries like smart cities, healthcare, GPS, smart transportations, etc., use the Internet of Things as a key technology that generates lots of data [25]. The learning algorithms which analyze the IoT data generally assume that the data are complete. While missing data are common in IoT, the data analytics performed on missing or incomplete IoT data may produce inaccurate or unreliable results. Therefore, an estimate of the missing value is necessary for IoT. Three main tasks must be performed to solve this problem. The first step is finding the reason for missing data. Poor network connectivity, faulty sensor systems, environmental factors and synchronization issues are the various reasons for the incomplete results. The missing data are divided into three types: missing completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR). The further step involves studying the pattern of missing data. The two approaches are monotonous missing patterns (MMP) and random missing patterns (AMP). Finally, they form a missing value imputation model for IoT to use the model to approximate the value for the missing data. Within the literature, some missing value imputation algorithms include single imputation algorithms, multivariate imputation algorithms, etc. The traditional imputation algorithm is not suitable for IoT data. There are some algorithms which are mainly used for missing data imputation, and these are given in the next section.

Gaussian Mixture model: The Gaussian Mixture Model (GMM) is a clustering algorithm [26]. It is a probabilistic model that uses a soft clustering approach for distributing the data points to different clusters.

A Gaussian Mixture is defined as follows: G = $\{GD_1, GD_2, \ldots, GD_k\}$, where k denotes the number of clusters. Each GD_i is a Gaussian distribution function that comprises a mean μ, which represents its center, a covariance Σ, and a probability π, which denotes how big or small the Gaussian function will be. Assuming a data set D is generated using GMM with k components, the function f_k (GD_i)

represents the probability density function of the *k*th component. The probability of GD_i, $P(GD_i)$ generated by GMM, is as given in Equation (4) below.

$$P(GD_i) = \sum_{i=1}^{k} \pi_i\ f_i\ (GD_i|\mu_i, \Sigma_i) \tag{4}$$

To handle the missing data imputation of the IoT sensor data using the GMM model involves five steps, namely instance creation, clustering, classification, distance measuring and data filling. First, the instances in the data set D are divided into two separate data sets, as D_1 and D_2. D_1 contains all the instances without missing values, whereas D_2 contains all instances in the data set which has the missing values. Secondly, the GMM model based on the EM algorithm is used to cluster the complete data set D1. The cluster center for each cluster is determined. After that, the cluster for each instance in D1 is computed. Third, the incomplete data set D_2 is taken as a testing set. Each instance in D_2 is classified according to the clustering result. For instance, $\alpha_i \in D_{2,}$ α_i belongs to a cluster if it is closer to the cluster center of that cluster by Euclidean distance. In the fourth step, for each instance α_i in D_2, one must determine one or more complete instances that are closest to α_i in the same cluster, using Euclidean distance as the distance measure. Finally, one must fill in the missing value of the instance α_i by finding the mean value of the closest instance in the cluster.

Spatial and temporal correlation [27]: Sensor nodes periodically detect data. Since the sensor data are time-sensitive, different results would be obtained using other sensor data for analysis. The relationship between the sensor nodes in different periods is not the same, so it is necessary to select the correct data to analyze. According to authors, the appropriate sampling of data is required for accurate data analysis. The authors propose the Temporal and Spatial Correlation Algorithm (TSCA) to estimate the missing data. Firstly, it saves all the sensed data simultaneously as a time series and selects the most important series as the analysis sample, which significantly improves the algorithm's efficiency and accuracy. Secondly, it estimates missing temporal and spatial dimensional values. These two measurements are assigned different weights. Third, there are two strategies for dealing with severe data loss, which boosts the algorithm's applicability. The basic workflow of the TSCA model is illustrated in Figure 2.

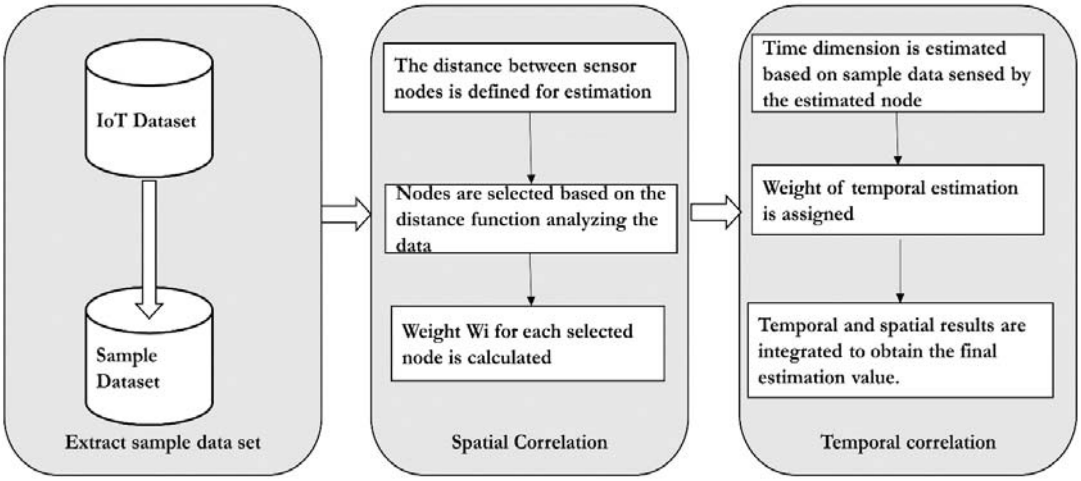
IoT Dataset
Sample Dataset
Extract sample data set
The distance between sensor nodes is defined for estimation
Nodes are selected based on the distance function analyzing the data
Weight Wi for each selected node is calculated
Spatial Correlation
Time dimension is estimated based on sample data sensed by the estimated node
Weight of temporal estimation is assigned
Temporal and spatial results are integrated to obtain the final estimation value.
Temporal correlation

Figure 2. The workflow of the Temporal and Spatial Correlation Algorithm.

The model as illustrated in Figure 2 assumes all the sensors are in the same communication range. The experiment was conducted on the air quality data set. As a first step, the target data set is extracted from the original data set. A missing data threshold is set, which differs from case to case. In the next step, compute the percentage of missing values. If it exceeds the threshold, then the imputation is ignored; otherwise, the spatial–temporal imputation is carried out. In the next step, the n proximity sensors are computed using the Haversine distance formula. The correlation between the n proximity sensors and the sensor with a missing value is calculated using the Pearson correlation co-efficient. Finally, the complete target data set is constructed and evaluated for accuracy.

The authors in [28] suggested a novel method of nearest neighbor imputation to impute missing values based on the spatial and temporal correlations between sensor nodes. The data structure kd-tree was deployed to boost search time. Based on the percentage of missing values, weighted variances and weighted Euclidean distances were used to create the kd-tree. Figure 3 illustrates the workflow of the spatial–temporal model. The algorithm defined in the proposed model follows the steps, as firstly, it sets the missing value threshold as T. The percentage P of the missing values in the chosen data set is then calculated. If P is within the threshold, then it finds the n proximate sensors through spatial correlation. The correlation between the sensors with the missing values and the n proximity sensors is computed using the Pearson correlation coefficient. The missing sensor data are then imputed by the readings of the proximity sensors corresponding to the time. The output data set is completed. The result is then compared with multiple imputation outcomes. Again, the accuracy is evaluated using Root Mean Square Error (RMSE).

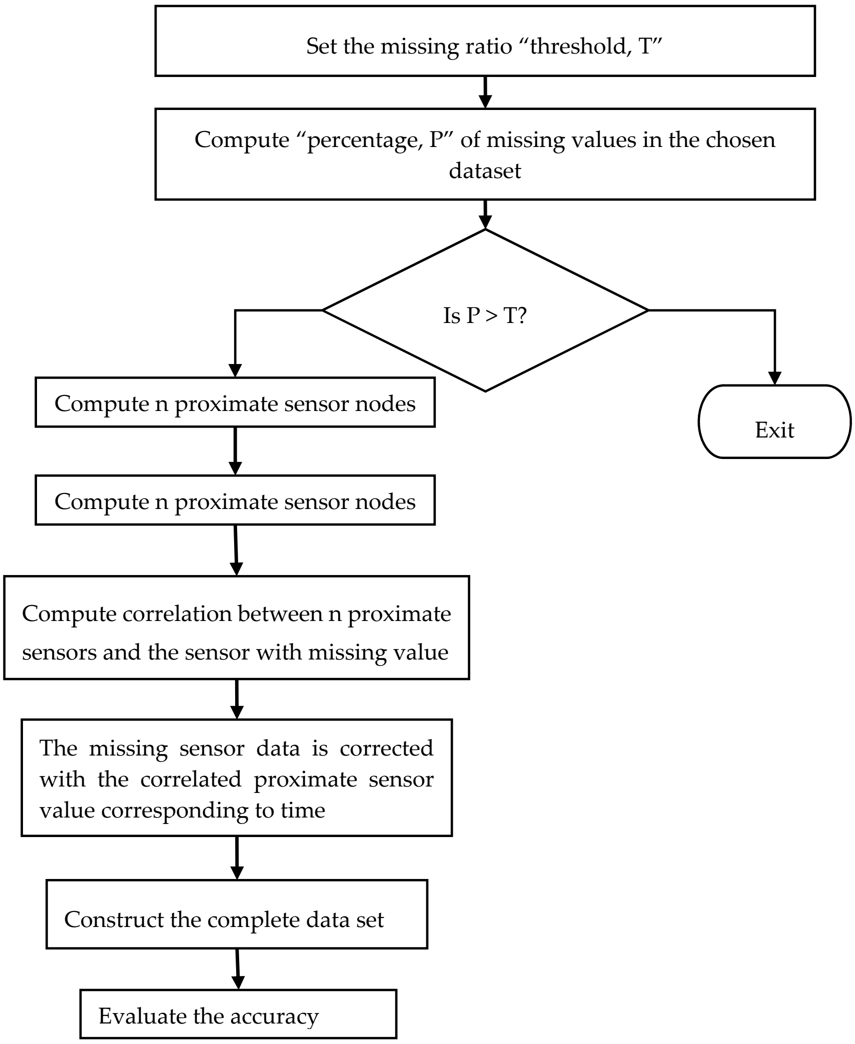
Set the missing ratio "threshold, T"
Compute "percentage, P" of missing values in the chosen dataset
Is P > T?
Compute n proximate sensor nodes
Exit
Compute n proximate sensor nodes
Compute correlation between n proximate sensors and the sensor with missing value
The missing sensor data is corrected with the correlated proximate sensor value corresponding to time
Construct the complete data set
Evaluate the accuracy

Figure 3. The workflow of the spatial–temporal model.

Incremental Space-Time Model (ISTM): The incremental model discussed in [29] is the model that updates the parameters of the existing model depending on the previous incoming data, rather than constructing a new model from scratch. The model uses incremental multiple linear regression to process the data. When the data arrive, this model is updated after reading the intermediary data matrix rather than accessing all the historical data. If any missing data are found, then the model provides an estimated value based on the nearest sensors' historical data and observations. The working of the ISTM model has three phases, which are initialization, estimation and updating. In the initialization phase, the model is initialized with historical data. For each sensor p, the historic data and the recording reference are represented as a data matrix. It also computes two intermediary matrices using these data matrices. Next, in the estimation phase, if the sensor's data contain one missing value, then the model generates an estimated value in real-time. It does this by referring to some data in the database. The estimated value will be then saved in the database. Finally, if the data arriving from the sensor do contain any missing value in the updating phase, then the model updates the new data. It then stores this in a reference database.

Probabilistic Matrix Factorization: There are two major advantages of using probabilistic matrix factorization (PMF) [30] for handling missing IoT sensor data. First is the dimensionality reduction, which is the underlying property of matrix factorization. The second is that the original matrix can be replicated using the factored matrices product. This method is used to recover the values missing in the original matrix. PMF is performed on the preliminarily assigned sensors. The neighboring sensors' data are examined for similarity, and are grouped into a different class of clusters using the K-means algorithm.

The K-means clustering algorithm groups the sensors into separate classes according to their measuring similarity. Analyzing the patterns of neighboring sensors helps to recover missing sensor data. Then, a probabilistic matrix factorization algorithm (PMF) is implemented in each cluster. PMF standardizes the data and restricts the probabilistic distribution of random features. The workflow of the algorithm is illustrated in Figure 4.

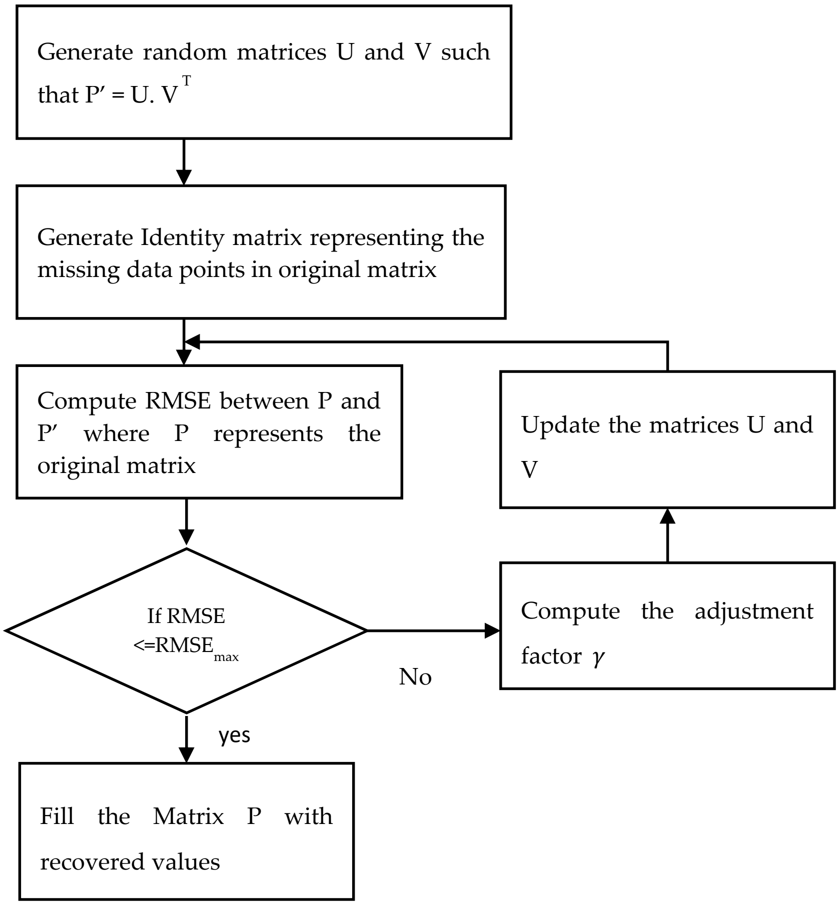
Generate random matrices U and V such that P′ = U. V^T
Generate Identity matrix representing the missing data points in original matrix
Compute RMSE between P and P′ where P represents the original matrix
If RMSE <=RMSE$_{max}$
No
Compute the adjustment factor γ
Update the matrices U and V
yes
Fill the Matrix P with recovered values

Figure 4. The workflow of the PMF model.

In the algorithm, PMF is used to factorize a single matrix into the product of two matrices. The dimensionality can be reduced by factorization. The ability to obtain the original matrix from the product of two factored matrices can be used to recover the missing values in the original matrix. The original matrix is represented as P $_{\text{NxM}}$. Now generate two random matrices, U and V, such that P′ = U. V $^{\text{T}}$, where U and V are of dimension NxK and KxM, respectively. K is an integer which represents the number of latent feature column-vectors in U and V. The missing data points in the original matrix are represented as an identity matrix, I having the same dimension as the original matrix P. The values in the I_{ij} matrix are defined using the following rule, as depicted in Equation (5).

$$I_{ij} = \begin{cases} 1,\ if\ P_{ij\ is\ not\ missing} \\ 0,\ if\ P_{ij\ is\ missing} \end{cases} \tag{5}$$

Next, compute the root mean square error (RMSE) between the P and P′, which is given in Equation (6) below.

$$RMSE = \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij}\left(P_{ij} - U_i V_j^T\right)^2 \tag{6}$$

The result obtained using Equation (1) is compared with $RMSE_{max}$, which is the maximum acceptable error. The algorithm will terminate if RMSE is ≤ $RMSE_{max}$. Otherwise, the values of U and V are updated using the Equations (7) and (8).

$$U' = U_i + \gamma.\frac{\partial RMSE_{ij}}{\partial U_i} \tag{7}$$

$$V' = V_j + \gamma.\frac{\partial RMSE_{ij}}{\partial V_j} \tag{8}$$

where γ denotes the adjustment factor that decides how much the U and V values need to be adjusted. Steps four to six are repeated until the RMSE is less than or equal to $RMSE_{max}$. A large value of γ may result in low precision, while a value that is too small may result in too many iterations.

The authors in [31] addressed the issue of missing value in IoT sensors. In IoT sensor networks, single-mode failures cause data loss due to the malfunction of several sensors in a network. The authors proposed a multiple segmented imputation approach, in which the data gap was identified and segmented into pieces, and then imputed and reconstructed iteratively with relevant data. The experimental results exhibited significant improvements over the conventional technique of root mean square.

3.3. Data Outlier Detection

In the IoT sensor network, the sensor nodes are widely distributed and heterogeneous. It is to be noted that, in a real physical environment, such a setup leads to enormous failure and risks associated with sensor nodes due to several external factors. This causes the original data generated from the IoT sensor network to become prone to modifications and produce data outliers [32]. Therefore, it is essential to identify such data outliers before performing data analysis and decision-making. For this purpose, spatial correlation-based data outlier detection is performed and carried using three popular methods, namely majority voting, classifiers, and principal component analysis.

Voting Mechanism: In this method, a sensor node is identified as functioning abnormally by finding the differences in reading with neighboring sensor nodes. According to authors [33], the Poisson distribution is the usual data generation method in various IoT sensor network applications. In the IoT sensor network, the data sets generated consist of outliers for short-term, non-periodic, and insignificant changes in data trends. The simple and efficient statistical methods for outlier detection in the Poisson distribution of the IoT sensor network data set are standard deviation and boxplot. Similarly, in a distributed environment, the Euclidean distance is estimated between the data generated by the faulty sensor node and the data from its immediate neighbor nodes. If the estimated difference in the data exceeds a specific threshold limit, then the data generated by this node are identified as an outlier. Although this technique is simple and less complicated, it is excessively dependent on the neighboring sensor nodes. Furthermore, the accuracy in the case of the sparse network is low.

Classifiers: This method involves two steps, firstly training the IoT sensor data using a standard machine learning model. Secondly, the data are detected using the classifier algorithms as either normal or anomaly [34]. The commonly used classifier algorithms is the support vector machine (SVM). Half of the data search space is trained to be standard data. Later, the data are analyzed and classified through SVM for the detection within the trained data of standard data, or otherwise of abnormal data. However, the demerits of the classifier algorithm involve its high complexity in the computation aspect.

Principal Component Analysis (PCA): The objective of PCA is to identify the residual value by extracting the principal components of the given data set. The residual values estimated by the data are evaluated through detection mechanisms such as T^2 score and squared prediction error (SPE) to identify the data outliers.

In [35], the authors addressed the outlier detection in the IoT sensor data using the Tucker machine and the Genetic Algorithm. Different sensor nodes are involved in the IoT sensor network, exhibiting spatial attributes and sensing data. Furthermore, the sensor data generated are dynamic for the time. Generally, the extensive sensor data contain anomalies due to the mode failure. The conventional means of detecting outliers involves vector-based algorithms. The demerits of vector-based algorithms disturb the original structural information of the sensor data, and exhibit the side effect of dimensionality. The authors proposed a tensor-based solution for the outlier detection using tucker factorization and genetic algorithms. The experimental results demonstrated improvements in the efficiency and accuracy of outlier detection without disturbing the intrinsic sensor data structure.

3.4. Data Aggregation

The data aggregation method is referred to as the method that collects and communicates information in a summary form. This can be used for statistical analysis. In IoT, heterogeneous data are collected from various nodes. Sending data separately from each node leads to high energy consumption, and needs a high bandwidth across the network, which reduces the lifetime of the network. Data aggregation techniques prevent these kinds of problems by summarizing data, which reduces the excessive transfer of data, increases the network's lifetime, and reduces network traffic. Data aggregation in the Internet of Things (IoT) helps to decrease the number of transmissions between objects. This lengthens the lifetime of the network and decreases energy consumption [36]. It also reduces network traffic.

The data aggregation methods in IoT are classified into the following:

(a) Tree-based approach [37–43]—This approach deploys the nodes in the network in the form of a tree. Here, the hierarchical and intermediate nodes perform the aggregation. The aggregated data are then transferred to the root node. The tree-based approach is suitable for solving energy consumption and lifetime of network problems;

(b) Cluster-based approach [44–48]—The entire network is organized as clusters. Each cluster will contain several sensor nodes with one node as the cluster head, which performs the data aggregation. This method aims to carry out the effective implementation of energy aggregation in large networks. This helps to reduce the energy consumption of nodes with limited energy in huge networks. Therefore, it results in a reduced overhead bandwidth due to the transfer of a limited number of packets. In the case of static networks where the cluster structure does not shift for a long time, cluster-based techniques are successful

(c) Centralized [49,50]—All sensor nodes in this system send the data to a header node, which is a strong node with all the other nodes. The header node is responsible for aggregating the data and sending out a single packet.

The authors in [51] addressed the issue of data uncertainty in IoT sensor data. Mainly, the data aggregation focused on device to device communication. The proposed technique involves the reconstruction of subspace-based data sampling. Next, the low-rank approximation is performed to identify the dominant subspace. Further, the robust dominant subspace is utilized for reliable sensor data, in an entirely supervised manner. The proposed method exhibits improvements in terms of accuracy and efficiency as regards removing the uncertainties and data aggregation of sensor data from the experimentation results.

4. Data Fusion

Data integration or fusion from multiple sensors is required to improve accuracy in various applications. For example, it can be used to trace a target in a military or surveillance system, trace an on-road vehicle's exact location, find the position of an obstacle in veins of the human body, etc. Various applications of data fusion in different domains are briefly explained as follows [52,53]:

- The multi-sensor data fusion approach can be applied in ships, aircraft, factories, etc., from a microscopic scale to a distance to hundreds of feet. In these systems, data from electromagnetic signals, acoustic signals, temperature sensors, X-rays, etc., can be integrated for validation and increased accuracy. This integration will increase the accuracy and build trust in the system, which is helpful in the timely maintenance of activities, system fault detection, and remote correction, etc.;
- Medical diagnosis is a critical system that involves the human body, and is used in disease identification, such as for tumors, lungs or kidney abnormalities, internal diseases, etc., through NMR, chemical or biological sensors, X-rays, IR, etc.;
- Many satellites, aircraft, and underground or underwater machines use seismic, EM radiation, chemical or biological sensors to collect accurate information or identify natural phenomena in environment monitoring from very long distances;
- Military and defense services use this technique in ocean surveillance, air-to-air or ground-to-air defense, battlefield intelligence, data acquisition, warning, defense systems, etc., using EM radiation from large distances.

Sensor fusion is the process of combining two or more data sources in a way that generates a more consistent, more accurate, and more dependable estimate of the dynamic system. This estimate gives better results than if the sensors were used individually. The objective of sensor fusion is to minimize cost, device complexity and the number of components involved, and improve sensing precision and confidence.

The data sources can be sensors or mathematical models, and the system state can be acceleration, distance, etc. Four different reasons for using sensor fusion include that (i) it increases the quality of data, (ii) it can increase reliability, (iii) it can measure unmeasured states, and (iv) it can increase the coverage area.

In general, the data fusion methods could be characterized as probabilistic, statistical, knowledge-based, and inference and reasoning methods. The probabilistic methods include Bayesian networks, maximum likelihood estimation methods, inference theory, Kalman filtering, etc. Statistical methods include covariance, cross variance, and other statistical analyses [54]. Knowledge-based methods include artificial neural networks, fuzzy logic, genetic algorithms, etc. [55,56]. Depending on the problem specification, the appropriate data fusion methods are to be chosen. Here, the basic Bayesian and Kalman filter methods are explained.

Bayesian method: In multi-sensor data fusion, the essential property of Bayesian statistics is that all unknown variables are considered random variables. The probability distribution function defines what is known about these unknown quantities. For a given probability distribution, the parameters are estimated from the prior distribution, and translates the uncertainty about the parameter values. The basic Bayes law states that

$$P(\alpha\big|\beta_1, \beta_2) \propto P(\alpha)L(\alpha;\beta_1)L(\alpha;\beta_2) \tag{9}$$

where prior density upon α is represented by $P(\alpha)$, the likelihood of α by β is represented by $L(\alpha;\beta)$, such that likelihood is proportional to $P(\alpha\big|\beta)$, and the conditional probability is given by the equation below.

$$L(\alpha;\beta) \propto P(\alpha\big|\beta) \tag{10}$$

This shows the probability of receiving sensor data β given the a priori value α. The fusion of two different sensor data measurements α and β, given a non-inform prior such as $P(\alpha)$, is then given

$$P(\alpha\big|\beta_1, \beta_2) \propto P(\alpha)L(\alpha;\beta_1)L(\alpha;\beta_2) \tag{11}$$

$$1 \times exp\frac{1}{2}\left(\frac{\alpha - \beta_1}{\delta_1}\right)^2 \times exp\frac{1}{2}\left(\frac{\alpha - \beta_2}{\delta_2}\right)^2 \tag{12}$$

Kalman Filter Method: This is used to estimate the system state when it cannot be measured directly. It is an iterative mathematical process that uses a set of equations and data inputs measured over time, containing noise and inaccuracies. It produces estimates of these unknown parameters that are more accurate than those taken from a single sensor measurement, by using a joint distribution function over each timestep variable.

It works as a two-step recursive algorithm, with a prediction and update step. In the prediction step, it estimates the current state of the variables along with the noises. Suppose the outcome of the new measurement is observed with some noise or uncertainty. In that case, the estimates are updated using a weighted average, assigning more weight to the estimates with higher certainty and less to estimates with low uncertainty. The estimated uncertainty of the predicted system's state is represented as a covariance matrix, and the weights are calculated.

The new estimate of the weighted average is obtained from the weighted average that lies between the predicted and measured state, giving better-estimated uncertainty. This process is recursive until the filter follows the model predictions more closely.

The Kalman filter model assumes the evolution of a state at time k from the state at (k − 1), according to the following equation.

$$x_k = Fx_{k-1} + Au_{k-1} + w_{k-1} \tag{13}$$

where F is the state transition matrix which is applied to previous state x_{k-1} vectors, and A is the control input matrix which is applied to the control vector u_{k-1}. w_{k-1} is the noise vector that is assumed to be the multivariate Gaussian distribution drawn from the zero mean with the covariance Q_k, where $w_{k-1} \sim N(0, Q_k)$. The measurement at time step k is observed as

$$z_k = Hx_k + v_k \tag{14}$$

where H is the measurement matrix, z_k is the measurement vector, and $\mathrm{v_k}$ is the measurement noise vector that is assumed to be a multivariate Gaussian distribution drawn from the zero mean with the covariance R, i.e., $v_k \sim N(0, R)$.

The objective of the Kalman filter is to provide an estimate of x_k at time k, given the initial estimate of x_0, the series of measurements $z_1, z_2, \ldots, z_k$ and the information of the system described by F, A, H, Q and R.

Three basic approaches can be used in multi-sensory data fusion or integration.

Direct fusion: In this approach, all sensors are associated among themselves for classification. Thereafter, data-level integration or fusion happens. Once the associated data sensor's data are integrated, then data features are extracted. These data features help in the identification of objects with sensors. This direct fusion process is also known as joint identity declaration, where multiple sensors' association identities are declared jointly. The direct fusion process is formally designed as shown in Equations (15)–(18).

$$\mathrm{O}:\ S_i \rightarrow S_j\ \forall i \neq j\ i \in \{1,2,3,\ldots,n\}\ and\ j \in \{1,2,3,\ldots,n\} \tag{15}$$

$$P:\ f_{fe}\left(f_{df}(f_A(O))\right) \tag{16}$$

$$ID_{data\_extraction}(S_i) = g(P) \tag{17}$$

$$Q : JID_{declaration}(S_i) \tag{18}$$

Here, S_i represents the ith sensor considered in the data fusion process, and O is the outcome of the ith sensor from jth sensor mapping. $f_A(.)$, $f_{df}(.)\ and\ f_{fe}(.)$ are the data association, internal data-level fusion, and feature extraction functions in the complete data fusion process. Individual sensor data identification is achieved by applying the identity declaration function (g) over the feature extraction outcome (P). Finally, Q is the outcome of joint identification in the direct fusion approach, using the function $JID_{declaration}(.)$ and outcome Q.

- *Feature extraction followed by fusion*: In this approach, the sensor's data features are extracted initially, followed by feature-based data association. These associations facilitate the fusion of the data based on the feature's association and identity declaration. Finally, feature-based data fusion and identity declaration result in the sensor's joint identify declaration and data integration. This approach is formally presented as shown in Equations (19)–(22).

$$R:\ f_{fe}(S_i)\ \forall\ i \in \{1,2,3,\ldots,n\} \tag{19}$$

$$U : g(f_A(\mathrm{R})) \tag{20}$$

$$V : H(f_A(R))) \tag{21}$$

$$Q : I(U,V) \tag{22}$$

Here, R is the outcome of feature extraction from each sensor. $H(.)$ is a function to integrate feature-level data fusion. Finally, the data fusion outcome Q for the joint identification declaration is computed by applying a parallel integration function $I(.)$ over the feature-level fusion outcome (U) and identity declaration outcome (V).

- *Feature extraction followed by identity declaration and fusion for high-level inferences or decisions*: In this approach, the sensor's data features are extracted initially. After that, individual sensor data identities are declared from extracted features. These unique identities help in data association or in searching the required data. A unique identity-based data association is used for declaration-level fusion and identity declaration. Both of these processes result in a joint identity declaration. This approach is formally presented as shown in Equations (23)–(28).

$$\mathrm{R}:\ f_{fe}(S_i)\ \forall\ i \in \{1,2,3,\ldots,n\} \tag{23}$$

$$U : g(\mathrm{R}) \tag{24}$$

$$S_i^{ID} = J(U) \tag{25}$$

$$V : H\left(f_A\left(U,\ S_i^{ID}\right)\right) \tag{26}$$

$$W : g\left(f_A\left(U,\ S_i^{ID}\right)\right) \tag{27}$$

$$Q : I(W,V) \tag{28}$$

The multi-sensor data fusion approach works in a multi-layered architecture as well. In multi-layered architectures [57], multi-sensory data fusion or integration approaches work at different layers. For example, the multi-layered components of the data fusion architecture, proposed by Joint Directors of Laboratories (JDL) [58], are briefly explained as follows.

- *Object Refinement*: This is considered as layer-1 processing in the JDL process model. Here, sensor-level data are combined to achieve better reliability, accuracy, position estimation, velocity measurement, attribute evaluations and identity exploration.
- *Situation Refinement*: This is a layer-2 concept in the JDL process model. According to this layer, a dynamic process is applied to describe the present relationships among various entities considered in the experimentation.

A list of associated events in the context of the environment has been prepared as well, as follows.

- *Threat Refinement*: This is a layer-3 concept in the JDL process model. In this layer, present situations are explored in such a way that future trends can be observed. For example, the current military and defense deployment can give an idea about what the implications will be of enemy threats. Who will become a friend or foe? What opportunities will arise if operations are planned soon or over a long time? etc.
- *Process Refinement*: This is a layer-3 concept in the JDL process model. In this layer, a meta-process is executed. In this meta-process, the overall data fusion processes for assessing and improving the real-time system or sub-system performances are performed.

The layering process used for data fusion in the JDL architecture, the source-preprocessing, database management systems, and human–computer interaction (HCI) are essential components. Source-preprocessing manages the resources and related data for any redundancy, duplicity or loss. Database management systems store the required and necessary information. HCI involves humans in controlling, monitoring and maintaining the systems.

5. Data Analysis

According to the authors in [59], the IoT sensor networks' key problems are the scalability and accuracy of sensor data. These challenges are addressed through sensor data mining and analysis techniques, such as data gathering, cleaning issues, data management, knowledge discovery, and data mining. In this aspect, machine learning and deep learning models play a vital role in obtaining the results, which include knowledge generation and decision-making.

Machine learning models: The authors in [60] addressed the issue that the IoT sensor data require an efficient mechanism for deriving meaningful information from data. In the case of IoT sensor data analytics, the machine learning models are required to be executed within the sensor embedded processor. This requires the customization of system programs and an efficient data structure to handle the features of real-time IoT sensor data. As such, the authors proposed the Gaussian Mixture Model (GMM) as a solution to handle the various sensor data features. Furthermore, the real-time cooperation of hardware and software, and continuous machine learning algorithms, were carried out for data training and the classification of the resultant data for IoT-based applications.

Deep Learning Models: The authors in [61] proposed feature learning for IoT sensor data using the deep learning mechanism. The IoT sensor data provide unclear features that need to be made accurate through real-time deep learning-based classification. However, the deep learning models are computationally expensive in terms of execution within resource-constrained sensor boards. As such, the authors proposed a preprocessing mechanism in the spectral domain, and then deep learning models were executed.

Neural Network for IoT Sensor Data Processing/Analysis: Artificial neural networks are well suited for function approximation and pattern recognition problems using supervised learning techniques. The architecture of ANN includes an input layer, an output layer, and one or more hidden layers.

Convolutional neural networks (CNN) consist of a network layer that is used for convolution operation applied to two-dimensional or one-dimensional sensor data [62]. The convolution operator is used to learn the local patterns from the data. Depending upon the available data, the right variant of CNN is used. CNN is mostly suited to the image data used for the analysis of image sensor data analysis. Most of the IoT devices, such as drones, smart cars, smartphones, etc., are equipped with cameras. Many IoT applications, like floods or landslides, forest fire prediction through drone images and traffic management, use vehicle cameras. CNN takes an image/speech/signal which is 2D or 1D, and the high-level features are extracted through a series of hidden layers. The hidden layers include the convolution layer and a fully connected pooling layer at the end. The convolution layer has a set of filters, known as a learnable parameter, which is the core of the CNN that filters the high-dimensional data to a lower dimension, which helps extract the most appropriate feature from the input image.

The recurrent neural network (RNN) is a neural network which learns from sequences or time series data [63]. Some tasks, such as detecting the driver's behavior in smart vehicles, identifying some movement patterns, or estimating the electricity consumption of the household environment, may depend on previous samples for prediction. In these cases, RNN would be the right choice. It can handle sequences of variable length. RNNs are well-suited for predicting the time series sensor data. The input to an RNN consists of the present sample as well as the sample described previously. In other words, the output of an RNN at step t−1 in time influences the output at step t in time. Each neuron has a feedback loop, which returns the current output as an input for the next step. This structure can be expressed so that each neuron in an RNN has an internal memory that preserves the computational information from the previous input. Long short-term memory (LSTM) is a variant of RNN that can look back for a long time before predicting. LSTM can efficiently retrieve and identify features for the prediction from the sensor data.

Autoencoders (AE) is a neural network that consists of the same number of input and output layers connected through a series of hidden layers [64]. The autoencoders can construct an input at the output layer. Due to this feature, the autoencoders are used in industrial IoT for many applications, such as fault detection in hardware devices, anomaly detection in the performance of assembly lines, etc. Generative adversarial networks (GANs) contain two neural networks, namely the generative and discriminative networks, which mainly aim to produce synthetic and high-quality data [7]. They work on the principle of min–max games, where one network tries to maximize the value function, and the other network wants to minimize it.

Applying GAN for IoT applications involves generating entirely different data based on the available original data set. The general applications of GAN include localization formulation and wayfinding in graph-based networks [65,66]. In these applications, the generator network module of GAN estimates all the possible feasible paths that exist between two nodes. The discriminator module of GAN performs optimal path identification between the source and the target node. GANs are widely used in assistant systems for disabled persons. Further, the authors discuss GAN's implementation in converting the image to audio applications for visual impaired humans. Another example is the generation of descriptive information in text form out of a given original image. The authors applied GAN for image processing applications, where the objective is to identify the forged celebrity consisting of different pictures of the authentic celebrity.

The IoT sensor network-based applications involve dynamic factors, distributed services, and real-time responsive mechanisms [67]. Hence, there is a middleware layer requirement between IoT-based applications and the underlying IoT sensor data. Further, the scalability addresses huge volumes of data that are obtained from the IoT sensor network. To tackle dynamic, real-time processing and scalability issues, solutions through the integration of IoT sensor networks with other emerging technologies, such as cloud computing, fog computing and edge-based data analysis, are required.

Cloud-Based Data Analysis: The authors in [68,69] proposed cloud-based data analytics for the Internet of Things. To perform efficient data analysis requires a hierarchical and distributed model for the data processing systems. This method is replicated in several virtual machines that re asocial with a remote cloud data center. Then, without any prior data knowledge, the cloud-based analytics model can handle dynamic data processing and scalability issues. The cluster-based operations involve breaking more complex mathematical processes into simpler small tasks, and executing them in different clusters, thus reducing computation cost.

Further, to handle the huge volume of data generated by the IoT sensors, efficient solutions based on cloud computing necessitate the application of big data and massively parallel distributed system technologies. The authors in [70] address the storage of massive quantities of high-velocity and complex sensing data generated by IoT sensor systems. These papers presented data fusion through efficient curation techniques that integrate IoT and the huge sensor data on the remote cloud server, which thus enables the system to provide efficient services for IoT applications. Various issues, such as sensor network scalability, data cleaning, data compression, data storage, data analysis and data visualization, were addressed through the data curation of IoT sensor data into the cloud [71]. The authors in [72] addressed the data acquisition of resource-constrained, distributed IoT sensors.

Further, the possibility of in situ data analysis was discussed to overcome the network congestion problem, with context-aware and real-time processing, in the context of fog and edge computing. Here, surveillance camera sensors were considered for achieving end-to-end optimization. Multiple feature extraction techniques and various classification algorithms were considered, as were the proposed processing depth and amplification of gain through efficient methods. The experimentation result showed a 4.3-fold reduction in power consumption compared with other designs.

It is to be noted that the IoT sensor data consists of complex time series data, and thus requires efficient data analysis mechanisms. The authors in [73,74] presented a data analytic framework that involved a multi-dimensional feature handling, selection and extraction model. The papers also discussed the dynamic data analytic model for IoT sensor data prediction. The IoT sensor data are represented in time series, and involve complex data analysis mechanisms. The authors proposed a prediction model named the Adaptative Sliding window algorithm, and applied it to the sensor data for feature selection. The experiment was conducted on sensor data obtained from the Intel Berkley Research lab, and results exhibited more than 98% accuracy. The second experiment conducted on Chicago Park Weather Data exhibited more than 92% accuracy. The authors in [75,76] aimed to reduce the data transfer and processing in a remote cloud by efficient fog-based data analytics for IoT sensor data. The paper argued that uploading sensor data into a remote cloud over the Internet adds no value due to the underlying network complexity.

Further, the authors proposed homoscedasticity detection techniques for multi-dimension feature extraction in IoT sensor data. This method, along with neural classifiers, promised to be efficient in extracting important sensor data events in real-time processing. The authors in [77] discussed the pervasive nature of IoT sensor networks and generated a huge volume of sensor data. These sensor data, however, exhibited data redundancy and data outliers that largely degraded the overall performance of the IoT sensor networks. As such, the authors proposed a data analysis framework that aimed at recursively updating and adapting to the dynamic changes in the IoT system's surroundings. In this, the authors proposed data aggregation through principal component analysis, which involved principal component extraction. Further, the squared prediction error score was proposed in order to identify the data outliers. In real-time experimentation, the proposed data analytics framework exhibited efficient data aggregation and data outlier detection with high accuracy.

Fog-Based Data Analysis: The authors in [78–80] presented fog-level-based IoT sensor data processing. Here, the senor data features are extracted and processed to classify different signal patterns using a neural network. Consequently, based on the output of neural network classification, event identification and decision-making are performed at the fog level. These sensor signals are classified as normal or abnormal, and accordingly, the alert event is initiated by the fog nodes. The few example applications include smart home systems, fire alarm systems, surveillance systems and air monitoring systems. The objective of fog-based IoT sensor data analysis is to carry out real-time data analytics at the fog device-level, and decision-making as to the alerts used. The authors in [81,82] introduce a content orchestration mechanism that estimates an opportunistic fog node to offload data. The mechanism considers the available bandwidth delay, the cost and the quality of service metrics to determine the opportunistic fog node. Further, the decision-making module enables periodic reports from nodes that serve as input values for a real-time analytical process, and computes the factors for improving the quality of service.

Edge-Based Data Analysis: The authors in [83–85] series prediction, forecasting, and event-based data handling. For this purpose, many solutions, such as stochastic and regression models, can be performed at the edge level of the IoT sensor network. In this, the data analysis is carried out at the low-level sensor nodes. In the case of time series data forecasting, the two specific prediction models are carried out simultaneously in the sensor devices and the base station nodes. Based on the difference between the values predicted by some constant factor, the base station will be updated with new data. This ensures a low computation cost and resource utilization in an inefficient manner. The authors of [86] present trends in IoT-based solutions from the perspective of health-care. This research aims to provide efficient services within an Internet of Medical Things application, utilizing the edge computing paradigm. In this paper, the authors proposed a three tier IoT architecture that included edge devices, fog network and cloud servers. The sensors act as edge devices and are connected to fog networks for real time data analysis. For high-performance computation and increase storage capacity, these edge devices are connected to the cloud remote servers. The authors present an energy-efficient IoT data compression approach for edge-based machine learning. This research aims to reduce the cost of data transmission, therefore increasing the overall efficiency of data offloading in a cloud-based IoT network. The study applies a fast error-bounded lossy compressor on the collected data before transmission. Later, they apply supervised machine learning techniques to re-build the data. Their experimentation shows that the amount of data traffic is significantly reduced by a factor of 103.

Semantic Analysis of IoT Sensor Data: The increasing amount of sensory data arises from making data and applications readily accessible and understandable to future users [87,88]. The semantic enhancements structure and organize the data. This also allows interoperability between machines. The benefit of applying semantic technology to sensor data is the conceptualization and abstract interpretation of the raw data, making them computer-definable, and interlinking the data with existing data web resources. The role of semantics offers new methods for information processing and exploration. It also turns information into actionable knowledge. The most common approaches include (i) linked open data, (ii) real-time and linked stream processing, (iii) rule-based reasoning, (iv) machine learning-based approaches and (v) semantic-based reasoning. The semantic analysis method includes the IoT resources, services, and related processes described using semantic annotations. To add sense to the IoT data, various tools, including observation and measurement data, need to be connected. By applying significant inference and analysis mechanisms to the IoT data, one can also achieve data processing for different domains. This also allows access to domain information and related semantically enriched representations for other entities and/or existing data (on the web). Linked data is a process that connects to different resources and is currently being implemented on the Internet. The linked data approach allows for the interconnection of resources defined by other models and ontologies. The provision of automated resource tagging mechanisms using the principles available, such as linked data and the specification of automated mechanisms of association between various resources (e.g., place, style, provider, and other common properties), render IoT data accessible across different domains. Processing and evaluating semantic descriptions for information extraction, and enabling enhanced interactions with IoT resources, depends on the efficient querying, analyzing and processing of semantic data and resource linkages. Here, the primary objective is to implement the resource-constrained middleware to allow the semantic IoT linked data analytics. This dynamically injects semantics to enrich and make sense of the IoT raw sensor data. The Semantic Sensor Network (SSN) ontology is used as the gateway to represent the sensor's properties and observations.

6. Conclusions

The IoT sensor network's paradigm shift towards emerging technologies, such as cloud, fog and edge computing, leads to elevated sophistication in data processing, data fusion and sensor data analytics. This paper presents deep insight into the necessity of these processes. The basic architecture of IoT sensor data processing, fusion and analysis has been explained to elaborate on each of these processes' functionalities. Next, the characteristics of IoT sensor data, such as the voluminous size of the IoT sensor data, heterogeneity, real-time processing and scalability, were explained. The paper provides an overview of various data processing techniques of IoT sensor data, such as data denoising, missing data imputation, data outlier detection and data aggregation. In addition to the data processing techniques, the definitions and processes of data fusion in IoT sensor-based networks have been presented. Further, the paper elaborates the need for enhancing IoT sensor data analysis through emerging technologies, such as cloud, fog and edge computing. Next, this paper presented a novel case study on IoT sensor-based drone networks for centralized traffic monitoring, control and data analytics. The future scope of this work includes addressing the security and privacy challenges of IoT sensor data. Further, there is a need for distributed blockchain technologies for the failure- and risk-free computation of IoT sensor data, which can be addressed as an extension to this present work.

Author Contributions: Conceptualization, R.K., A.K. and D.G.; methodology, R.K., D.G. and A.K.; software, R.K. and A.K.; validation, A.N., A.K. and D.G.; formal analysis, R.K., A.N. and A.K.; investigation, A.N. and A.K.; resources, A.N. and B.Q.; data curation, A.K., R.K. and D.G.; writing—original draft preparation, R.K., A.K. and D.G.; writing—review and editing, A.N., A.K. and D.G.; visualization, R.K. and D.G.; supervision, A.N. and A.K.; project administration, A.N. and B.Q.; funding acquisition, B.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially funded by the Robotics and Internet of Things lab at Prince Sultan University.

Conflicts of Interest: The authors of this manuscript have no Conflicts of Interest.

References

1. Liu, Y.; Dillon, T.; Yu, W.; Rahayu, W.; Mostafa, F. Missing value imputation for Industrial IoT sensor data with large gaps. *IEEE Internet Things J.* **2020**, *7*, 6855–6867. [CrossRef]
2. Chernick, M.R. Wavelet Methods for Time Series Analysis. *Technometrics* **2001**, *43*, 491. [CrossRef]
3. Gartner Inc. Available online: https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io (accessed on 10 April 2020).
4. Deng, X.; Jiang, P.; Peng, X.; Mi, C. An intelligent outlier detection method with one class support tucker machine and genetic algorithm toward big sensor data in internet of things. *IEEE Trans. Ind. Electron.* **2019**, *66*, 4672–4683. [CrossRef]
5. Sanyal, S.; Zhang, P. Improving quality of data: IoT data aggregation using device to device communications. *IEEE Access* **2018**, *6*, 67830–87840. [CrossRef]
6. Yang, C.; Puthal, D.; Mohanty, S.P.; Kougianos, E. Big-Sensing-Data Curation for the Cloud is Coming: A Promise of Scalable Cloud-Data-Center Mitigation for Next-Generation IoT and Wireless Sensor Networks. *IEEE Consum. Electron. Mag.* **2017**, *6*, 48–56. [CrossRef]
7. Cao, N.; Nasir, B.S.; Sen, S.; Raychowdhury, A. Self-Optimizing IoT Wireless Video Sensor Node with In-Situ Data Analytics and Context-Driven Energy-Aware Real-Time Adaptation. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2017**, *64*, 2470–2480. [CrossRef]
8. Rubin, D.B. Inference and missing data. *Biometrika* **1976**, *63*, 581–592. [CrossRef]
9. Dong, Y.; Peng, C.Y.J. Principled missing data methods for researchers. *SpringerPlus* **2013**, *2*, 222. [CrossRef]
10. Hua, C.; Yum, T.S.P. Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks. *IEEE/ACM Trans. Netw.* **2008**, *16*, 892–903. [CrossRef]
11. Bather, J. Tracking and Data Fusion. In Proceedings of the IEE International Seminar Target Tracking: Algorithms and Applications, Enschede, The Netherlands, 16–17 October 2001. [CrossRef]
12. Xiao, F. Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy. *Inf. Fusion* **2019**, *46*, 23–32. [CrossRef]

13. Ando, B.; Baglio, S.; Lombardo, C.O.; Marletta, V. A multisensor data-fusion approach for ADL and fall classification. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 1960–1967. [CrossRef]
14. Park, J.J.; Moon, J.H.; Lee, K.; Kim, D.I. Transmitter-Oriented Dual Mode SWIPT with Deep Learning Based Adaptive Mode Switching for IoT Sensor Networks. *IEEE Internet Things J.* **2020**, *7*, 8979–8992. [CrossRef]
15. Ravi, D.; Wong, C.; Lo, B.; Yang, G.Z. A Deep Learning Approach to on-Node Sensor Data Analytics for Mobile or Wearable Devices. *IEEE J. Biomed. Heal. Inform.* **2017**, *21*, 56–64. [CrossRef]
16. Kumar, A.; Rajalakshmi, K.; Jain, S.; Nayyar, A.; Abouhawwash, M. A novel heuristic simulation-optimization method for critical infrastructure in smart transportation systems. *Int. J. Commun. Syst.* **2020**, *10*, e4397. [CrossRef]
17. Chi, Q.; Yan, H.; Zhang, C.; Pang, Z.; Li, D.X. A reconfigurable smart sensor interface for industrial WSN in IoT environment. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1417–1425. [CrossRef]
18. Yonezawa, T.; Ito, T.; Nakazawa, J.; Tokuda, H. SOXFire: A Universal Sensor Network System for Sharing Social Big Sensor Data in Smart Cities. In Proceedings of the 2nd International Workshop on Smart, Delft, The Netherlands, 7–11 December 2020. [CrossRef]
19. Chen, D.; Wu, B.; Chen, T.; Dong, J. Development of distributed data sharing platform for multi-source IOT sensor data of agriculture and forestry. *Nongye Gongcheng Xuebao/Trans. Chin. Soc. Agric. Eng.* **2017**, *33*, 300–307. [CrossRef]
20. Milenkovic, M. A Case for Interoperable IoT Sensor Data and Meta-data Formats. *Ubiquity* **2015**, *2015*, 1–7. [CrossRef]
21. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
22. Sun, Q.B.; Liu, J.; Li, S.; Fan, C.X.; Sun, J.J. Internet of things: Summarize on concepts, architecture and key technology problem. *Beijing Youdian Daxue Xuebao/J. Beijing Univ. Posts Telecommun.* **2010**, *33*, 1–9.
23. Mao, Y.; Bhuse, V.; Zhou, Z.; Pichappan, P.; Abdel-Aty, M.; Hayafuji, Y. Applied mathematics and algorithms for cloud computing and iot. *Math. Probl. Eng.* **2014**, *2014*, 946860. [CrossRef]
24. Berkner, K.; Wells, R.O. Wavelet transforms and denoising algorithms. In Proceedings of the Conference Record of the Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1–4 November 1998; Volume 2, pp. 1639–1643. [CrossRef]
25. Yan, X.; Xiong, W.; Hu, L.; Wang, F.; Zhao, K. Missing value imputation based on gaussian mixture model for the internet of things. *Math. Probl. Eng.* **2015**, *2015*, 548605. [CrossRef]
26. Gao, Z.; Cheng, W.; Qiu, X.; Meng, L. A Missing Sensor Data Estimation Algorithm Based on Temporal and Spatial Correlation. *Int. J. Distrib. Sens. Netw.* **2015**, *2015*. [CrossRef]
27. Mary, I.P.S.; Arockiam, L. Imputing the missing data in IoT based on the spatial and temporal correlation. In Proceedings of the 2017 IEEE International Conference on Current Trends in Advanced Computing, ICCTAC 2017, Bangalore, India, 2–3 March 2017; Volume 2018, pp. 1–4. [CrossRef]
28. Li, Y.; Parker, L.E. Nearest neighbor imputation using spatial-temporal correlations in wireless sensor networks. *Inf. Fusion* **2014**, *15*, 64–79. [CrossRef] [PubMed]
29. Li, P.; Stuart, E.A.; Allison, D.B. Multiple imputation: A flexible tool for handling missing data. *JAMA—J. Am. Med. Assoc.* **2015**, *314*, 1966–1967. [CrossRef]
30. Vijayakumar, N.N.; Plale, B. Missing event prediction in sensor data streams using kalman filters. *Knowl. Discov. Sens. Data* **2008**, *149*, 170.
31. Halatchev, M.; Gruenwald, L. *Estimating Missing Values in Related Sensor Data Streams*; The University of Oklahoma: Norman, OK, USA, 2005; pp. 83–94.
32. Al-khatib, A.A.; Mohammed, B.; Abdelmajid, K. A Survey on Outlier Detection in Internet of Things Big Data. In *Big Data-Enabled Internet of Things*; IET: London, UK, 2020; pp. 265–272.
33. Shahraki, A.; Haugen, Ø. An outlier detection method to improve gathered datasets for network behavior analysis in IoT. *J. Commun.* **2019**, *14*, 455–462. [CrossRef]
34. Hasan, M.; Islam, M.M.; Zarif, M.I.I.; Hashem, M.M.A. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet Things* **2019**, *7*, 100059. [CrossRef]
35. Gaddam, A.; Wilkin, T.; Angelova, M.; Gaddam, J. Detecting Sensor Faults, Anomalies and Outliers in the Internet of Things: A Survey on the Challenges and Solutions. *Electronics* **2020**, *9*, 511. [CrossRef]
36. Nithyakalyani, S.; Gopinath, B. *Analysis of Node Clustering Algorithms on Data Aggregation in Wireless Sensor Network*; NISCAIR-CSIR: New Delhi, India, 2015.

37. Zhong, H.; Shao, L.; Cui, J.; Xu, Y. An efficient and secure recoverable data aggregation scheme for heterogeneous wireless sensor networks. *J. Parallel Distrib. Comput.* **2018**, *111*, 1–12. [CrossRef]
38. Liu, Y.; Gong, X.; Xing, C. A novel trust-based secure data aggregation for Internet of Things. In Proceedings of the 9th International Conference on Computer Science and Education, ICCCSE 2014, Vancouver, BC, Canada, 22–24 August 2014; pp. 435–439. [CrossRef]
39. Schimbinschi, F.; Nguyen, X.V.; Bailey, J.; Leckie, C.; Vu, H.; Kotagiri, R. Traffic forecasting in complex urban networks: Leveraging big data and machine learning. In Proceedings of the 2015 IEEE International Conference on Big Data, Santa Clara, CA, USA, 29 October–1 November 2015; pp. 1019–1024. [CrossRef]
40. Khattak, H.A.; Hussain, R.; Ameer, Z. Internet of vehicles: Integrated services over vehicular Ad Hoc Networks. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*; LNICST 2018; Springer: Berlin/Heidelberg, Germany, 2018; Volume 224, pp. 61–73. [CrossRef]
41. Dalgleish, M.; Hoose, N. Highway traffic monitoring and data quality. *Traffic Eng. Control.* **2009**, *50*, 29–30.
42. Alinia, B.; Hajiesmaili, M.H.; Khonsari, A.; Crespi, N. Maximum-quality tree construction for deadline-constrained aggregation in WSNs. *IEEE Sens. J.* **2017**, *17*, 3930–3943. [CrossRef]
43. Sicari, S.; Grieco, L.A.; Boggia, G.; Coen-Porisini, A. DyDAP: A dynamic data aggregation scheme for privacy aware wireless sensor networks. *J. Syst. Softw.* **2012**, *85*, 152–166. [CrossRef]
44. Wu, W.; Cao, J.; Wu, H.; Li, J. Robust and dynamic data aggregation in wireless sensor networks: A cross-layer approach. *Comput. Netw.* **2013**, *57*, 3929–3940. [CrossRef]
45. Xu, J.; Yang, G.; Chen, Z.Y.; Chen, L.; Yang, Z. Performance analysis of data aggregation algorithms in wireless sensor networks. In Proceedings of the 2011 International Conference on Electrical and Control Engineering, ICECE 2011, Yichang, China, 16–18 September 2011; pp. 4619–4622. [CrossRef]
46. Satapathy, S.S.; Sarma, N. TREEPSI: Tree based energy efficient protocol for sensor information. In Proceedings of the 2006 IFIP International Conference on Wireless and Optical Communications Networks, Bangalore, India, 11–13 April 2006. [CrossRef]
47. Messina, D.; Ortolani, M.; Re, G.L. A network protocol to enhance robustness in tree-based WSNs using data aggregation. In Proceedings of the 2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, MASS 2007, Pisa, Italy, 8–11 October 2007. [CrossRef]
48. Tang, F.; You, I.; Guo, S.; Guo, M.; Ma, Y. A chain-cluster based routing algorithm for wireless sensor networks. *J. Intell. Manuf.* **2012**, *23*, 1305–1313. [CrossRef]
49. Guo, W.; Xiong, N.; Vasilakos, A.V.; Chen, G.; Cheng, H. Multi-source temporal data aggregation in wireless sensor networks. *Wirel. Pers. Commun.* **2011**, *56*, 359–370. [CrossRef]
50. Rajkamal, R.; Ranjan, P.V. Energy efficient aggregation for continuous monitoring applications of wireless sensor network. *J. Comput. Sci.* **2012**, *8*, 55–60. [CrossRef]
51. Dehkordi, S.A.; Farajzadeh, K.; Rezazadeh, J.; Farahbakhsh, R.; Sandrasegaran, K.; Dehkordi, M.A. A survey on data aggregation techniques in IoT sensor networks. *Wirel. Netw.* **2020**, *26*, 1243–1263. [CrossRef]
52. Ding, W.; Jing, X.; Yan, Z.; Yang, L.T. A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion. *Inf. Fusion* **2019**, *51*, 129–144. [CrossRef]
53. Qi, J.; Yang, P.; Newcombe, L.; Peng, X.; Yang, Y.; Zhao, Z. An overview of data fusion techniques for Internet of Things enabled physical activity recognition and measure. *Inf. Fusion* **2020**, *55*, 269–280. [CrossRef]
54. De Paola, A.; Ferraro, P.; Gaglio, S.; Re, G.L.; Das, S.K. An Adaptive Bayesian System for Context-Aware Data Fusion in Smart Environments. *IEEE Trans. Mob. Comput.* **2017**, *16*, 1502–1515. [CrossRef]
55. Wang, M.; Perera, C.; Jayaraman, P.P.; Zhang, M.; Strazdins, P.; Shyamsundar, R.K.; Ranjan, R. City data fusion: Sensor data fusion in the internet of things. *Int. J. Distrib. Syst. Technol.* **2016**, *7*, 15–36. [CrossRef]
56. Diez-Olivan, A.; Del Ser, J.; Galar, D.; Sierra, B. Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. *Inf. Fusion* **2019**, *50*, 92–111. [CrossRef]
57. Alkhamisi, A.; Nazmudeen, M.S.H.; Buhari, S.M. A cross-layer framework for sensor data aggregation for IoT applications in smart cities. In Proceedings of the IEEE 2nd International Smart Cities Conference: Improving the Citizens Quality of Life, ISC2 2016, Trento, Italy, 12–15 September 2016. [CrossRef]
58. Misbahuddin, S.; Zubairi, J.A.; Saggaf, A.; Basuni, J.; Sulaiman, A.; Al-Sofi, A. IoT based dynamic road traffic management for smart cities. In Proceedings of the 2015 12th International Conference on High-Capacity Optical Networks and Enabling/Emerging Technologies, HONET-ICT 2015, Islamabad, Pakistan, 21–23 December 2015. [CrossRef]

59. Aggarwal, C.C. An introduction to sensor data analytics. In *Managing and Mining Sensor Data*; Springer: Boston, MA, USA, 2013; pp. 1–8. [CrossRef]
60. Kanawaday, A.; Sane, A. Machine learning for predictive maintenance of industrial machines using IoT sensor data. In Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS 2018, Beijing, China, 24–26 November 2017; Volume 2017, pp. 87–90. [CrossRef]
61. Yu, T.; Wang, X.; Shami, A. Recursive Principal Component Analysis-Based Data Outlier Detection and Sensor Data Aggregation in IoT Systems. *IEEE Internet Things J.* **2017**, *4*, 2207–2216. [CrossRef]
62. Balakrishna, S.; Thirumaran, M.; Solanki, V.K. IoT sensor data integration in healthcare using semantics and machine learning approaches. *Intell. Syst. Ref. Libr.* **2020**, *165*, 275–300.
63. Ye, J.; Dobson, S.; McKeever, S. Situation identification techniques in pervasive computing: A review. *Pervasive Mob. Comput.* **2012**, *8*, 36–66. [CrossRef]
64. Xie, S.; Chen, Z. Anomaly detection and redundancy elimination of big sensor data in internet of thing. *arXiv* **2017**, arXiv:1703.03225.
65. Qanbari, S.; Behinaein, N.; Rahimzadeh, R.; Dustdar, S. Gatica: Linked Sensed Data Enrichment and Analytics Middleware for IoT Gateways. In Proceedings of the 2015 International Conference on Future Internet of Things and Cloud, FiCloud 2015 and 2015 International Conference on Open and Big Data, OBD 2015, Rome, Italy, 24–26 August 2015; pp. 38–43. [CrossRef]
66. Sekiyama, M.; Kim, B.K.; Irie, S.; Tanikawa, T. Sensor data processing based on the data log system using the portable IoT device and RT-Middleware. In Proceedings of the 2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2015, Goyang, Korea, 28–30 October 2015; pp. 46–48. [CrossRef]
67. Hromic, H.; Le Phuoc, D.; Serrano, M.; Antonić, A.; Žarko, I.P.; Hayes, C.; Decker, S. Real time analysis of sensor data for the Internet of Things by means of clustering and event processing. In Proceedings of the IEEE International Conference on Communications 2015, London, UK, 8–12 June 2015; pp. 685–691. [CrossRef]
68. Krishnakumar, K.; Gubbi, J.; Buyya, R. *A Framework for IoT Sensor Data Analytics and Visualisation in Cloud Computing Environments*; University of Melbourne: Parkville, Australia, 2012; p. 12.
69. Das, R.B.; Bozdog, N.V.; Bal, H. Cowbird: A Flexible Cloud-Based Framework for Combining Smartphone Sensors and IoT. In Proceedings of the 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2017, San Francisco, CA, USA, 6–8 April 2017; pp. 1–8. [CrossRef]
70. Ding, Z.; Xu, J.; Yang, Q. SeaCloudDM: A database cluster framework for managing and querying massive heterogeneous sensor sampling data. *J. Supercomput.* **2013**, *66*, 1260–1284. [CrossRef]
71. Zhu, T.; Dhelim, S.; Zhou, Z.; Yang, S.; Ning, H. An Architecture for Aggregating Information from Distributed Data Nodes for Industrial Internet of Things. *Comput. Electr. Eng.* **2017**, *58*, 337–349. [CrossRef]
72. Sharma, S.; Chen, K.; Sheth, A. Toward practical privacy-preserving analytics for IoT and cloud-based healthcare systems. *IEEE Internet Comput.* **2018**, *22*, 42–51. [CrossRef]
73. Zhang, C.; Liu, Y.; Wu, F.; Fan, W.; Tang, J.; Liu, H. Multi-Dimensional Joint Prediction Model for IoT Sensor Data Search. *IEEE Access* **2019**, *7*, 90863–90873. [CrossRef]
74. Shyamalagowri, M.; Rajeswari, R. Unscented Kalman filter based nonlinear state estimation case study-Nonlinear process control reactor (Continuous stirred tank reactor). In Proceedings of the 10th International Conference on Intelligent Systems and Control, ISCO 2016, Coimbatore, India, 7–8 January 2016. [CrossRef]
75. Kumarage, H.; Khalil, I.; Alabdulatif, A.; Tari, Z.; Yi, X. Secure Data Analytics for Cloud-Integrated Internet of Things Applications. *IEEE Cloud Comput.* **2016**, *3*, 46–56. [CrossRef]
76. Patni, H.; Henson, C.; Sheth, A. Linked sensor data. In Proceedings of the 2010 International Symposium on Collaborative Technologies and Systems, Chicago, IL, USA, 17–21 May 2010; pp. 362–370. [CrossRef]
77. Qin, Y.; Sheng, Q.Z.; Falkner, N.J.; Dustdar, S.; Wang, H.; Vasilakos, A.V. When Things Matter: A Survey on Data-Centric Internet of Things. *J. Netw. Comput. Appl.* **2016**, *64*, 137–153. [CrossRef]
78. He, J.; Wei, J.; Chen, K.; Tang, Z.; Zhou, Y.; Zhang, Y. Multitier Fog Computing With Large-Scale IoT Data Analytics for Smart Cities. *IEEE Internet Things J.* **2018**, *5*, 677–686. [CrossRef]
79. Yoon, G.; Choi, D.; Lee, J.; Choi, H. Management of IoT Sensor Data Using a Fog Computing Node. *J. Sens.* **2019**, *2019*, 5107457. [CrossRef]

80. Raafat, H.M.; Hossain, M.S.; Essa, E.; Elmougy, S.; Tolba, A.S.; Muhammad, G.; Ghoneim, A. Fog Intelligence for Real-Time IoT Sensor Data Analytics. *IEEE Access* **2017**, *5*, 24062–24069. [CrossRef]
81. Singh, S.P.; Nayyar, A.; Kumar, R.; Sharma, A. Fog computing: From architecture to edge computing and big data processing. *J. Supercomput.* **2019**, *75*, 2070–2105. [CrossRef]
82. Kaur, A.; Singh, P.; Nayyar, A. Fog Computing: Building a Road to IoT with Fog Analytics. In *Fog Data Analytics for IoT Applications*; Springer: Singapore, 2020; pp. 59–78.
83. Qureshi, B. Profile-based Power-aware Workflow Scheduling Framework for Energy-Efficient Data Centers. *Future Gener. Comput. Syst.* **2019**, *94*, 453–467. [CrossRef]
84. Cai, H.; Xu, B.; Jiang, L.; Vasilakos, A.V. IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges. *IEEE Internet Things J.* **2017**, *4*, 75–87. [CrossRef]
85. Djedouboum, A.C.; Abba Ari, A.A.; Gueroui, A.M.; Mohamadou, A.; Aliouat, Z. Big Data Collection in Large-Scale Wireless Sensor Networks. *Sensors* **2018**, *18*, 4474. [CrossRef] [PubMed]
86. Qureshi, B. An affordable Hybrid Cloud based Cluster for Secure Health Informatics Research. *Int. J. Cloud Appl. Comput.* **2018**, *8*, 27–46. [CrossRef]
87. Bytes, A.; Adepu, S.; Zhou, J. Towards Semantic Sensitive Feature Profiling of IoT Devices. *IEEE Internet Things J.* **2019**, *6*, 8056–8064. [CrossRef]
88. Abu-Elkheir, M.; Hayajneh, M.; Ali, N.A. Data Management for the Internet of Things: Design Primitives and Solution. *Sensors* **2013**, *13*, 15582–15612. [CrossRef] [PubMed]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.