symmetry

MDPI

Article

A Hybrid Prediction Model for Energy-Efficient Data Collection in Wireless Sensor Networks

Seyed Ahmad Soleymani [1], Shidrokh Goudarzi [2,*], Nazri Kama [3], Saiful Adli Ismail [3], Mazlan Ali [3], Zaini MD Zainal [3] and Mahdi Zareei [4]

[1] School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM), Johor Bahru 81310, Malaysia; asseyed4@live.utm.my

[2] Faculty of Information Science and Technology, Center for Artificial Intelligence Technology (CAIT), Universiti Kebangsaan Malaysia (UKM), Selangor 43600, Malaysia

[3] Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia (UTM), Kuala Lumpur 54100, Malaysia; mdnazri@utm.my (N.K.); saifuladli@utm.my (S.A.I.); mazlanali.kl@utm.my (M.A.); mdzaini@graduate.utm.my (Z.M.Z.)

[4] Tecnologico de Monterrey, School of Engineering and Sciences, Zapopan 45201, Mexico; m.zareei@tec.mx

* Correspondence: shidrokh@ukm.edu.my

Received: 6 November 2020; Accepted: 3 December 2020; Published: 7 December 2020

Abstract: Energy consumption because of unnecessary data transmission is a significant problem over wireless sensor networks (WSNs). Dealing with this problem leads to increasing the lifetime of any network and improved network feasibility for real time applications. Building on this, energy-efficient data collection is becoming a necessary requirement for WSN applications comprising of low powered sensing devices. In these applications, data clustering and prediction methods that utilize symmetry correlations in the sensor data can be used for reducing the energy consumption of sensor nodes for persistent data collection. In this work, a hybrid model based on decision tree (DT), autoregressive integrated moving average (ARIMA), and Kalman filtering (KF) methods is proposed to predict the data sampling requirement of sensor nodes to reduce unnecessary data transmission. To perform data sampling predictions in the WSNs efficiently, clustering and data aggregation to each cluster head are utilized, mainly to reduce the processing overheads generating the prediction model. Simulation experiments, comparisons, and performance evaluations conducted in various cases show that the forecasting accuracy of our approach can outperform existing Gaussian and probabilistic based models to provide better energy efficiency due to reducing the number of packet transmissions.

Keywords: clustering; prediction; energy consumption; wireless sensor network

1. Introduction

WSNs are spatially distributed autonomous sensory devices that control physical or environmental conditions. Their applications are wide-ranging, including disaster management, congestion monitoring in smart cities, and ecological supervision. The energy consumption and stability are essential considerations and critical challenges because of the limited capacity of the sensor nodes' batteries and the impracticality of frequent replacement of batteries in WSNs. Data extraction and transmission of the data packets are the most important factors impacting energy consumption. This is mainly because the node needs to acquire all sensor readings continuously and precisely. Such nodes utilize vast quantities of energy during the accurate extraction, aggregation, and transmission of data.

Data prediction can be a logical way to deal with these issues [1], wherein one performs prediction operations using the past data measured by sensors. By using this technique, there is no need to transmit the data measured by the sensor node continuously [2]. In some existing studies, such as [3,4], simple techniques to develop a predictor for the network of sensors to transfer the data from the entire array of sensors to a base station are utilized. However, the prediction methods employed in these works might not function appropriately when data values change significantly and continuously. To address this issue, local prediction based on clustering in sensor networks can be an effective. The local prediction model would be energy efficient, since the shorter length of the routing path is used to transmit sensor data. However, clustering-based local prediction is facing a couple of challenges. The first challenge is related to the high cost of training a predictor which is affected by the trade-off between communication and computation. The second challenge is the dynamic characteristics of the sensor data—in particular, when the prediction models are not working well for a set of less predictable data.

Filtering and clustering methods can be used to improve the spatial and temporal correlation between the sensor data to reduce the energy. To that end, we used a self-tuning approach based on Kalman filtering (KF) [5–8], which demonstrates high potential due to its non-biased and optimized estimation while minimizing covariance errors. The researchers in [9] aimed at improving link energy consumption during the transmission process by reducing the number of hops. Based on this, we developed a novel hybrid model for data sampling prediction of sensor nodes. The energy consumption of nodes can be balanced by adding the new link which can improve data transmission with low delay.

The main objective of the data prediction model for cluster-based WSNs is to reduce energy consumption, which is effected by radio transmissions, by decreasing the number of transmissions between the sender and the receiver. To that end, it needs to perform data sampling predictions in the WSNs effectively while clustering and data aggregating to each cluster head for overhead reduction.

These key considerations also motivated us to use cluster-based WSNs in this work. The primary contributions of this paper are as follows:

1 We designed a model based on decision tree (DT), autoregressive integrated moving average (ARIMA), and Kalman filtering (KF) methods for data prediction in order to reduce unnecessary data transmissions and as a result decrease energy consumption. This model employs a minimal set of sensor nodes for data collection based on intra-cluster prediction and processing of data. In the proposed model, DT is used to filter data associated with each node in order to derive a tree for clustering the sensor data. Additionally, a self-tuning approach based on KF is utilized to optimize estimation while minimizing covariance errors.

2 We provide the MATLAB simulation-based practical demonstration of the proposed model to measure the data packet transmission and energy consumption in sensor nodes under different numbers of distributed sensor nodes in the network.

The remaining sections of this paper are organized as follows: Section 2 explains the related works. Section 3 represents the preparation techniques and the primary modeling process. In Section 4, the performance of the proposed approach against other related methods has been evaluated, and lastly, Section 5 concludes the paper.

2. Related Works

Data prediction includes developing an intellectual phenomenon such as a model explaining data evolution. In meeting this objective, data prediction methods can be divided into three major classes, including stochastic, time series prediction, and algorithmic techniques. The major disadvantage of the stochastic classification approach [10] is the high cost of computation, which could be too much for sensor devices with limited analysis and power. Stochastic techniques tend to be more suitable to be used in the presence of several robust sensors [11]. The methods for time series forecasting [12] can offer accuracy that is satisfactory when using simple techniques (i.e., low order AR/MA).

Most current researchers in the WSN fields are concentrated on stability, efficiency in energy consumption, scalability, and improving the operational lifetimes of sensing nodes. Many clustering algorithms have been developed for a different types of applications [4,13,14]. Clustering techniques are used in WSNs for organizing and grouping nodes and specifying which specific node in each cluster undertakes the task of intra-cluster and inter-cluster data communications. This technique can be effective for reducing the number of data transmissions, so can reduce power consumption and improve the lifetime of the network. This technique for local prediction of sensor nodes is an example of one of these applications [15] where the cluster head acts as a sensor node and also keeps the historical data for every cluster's sensor node.

In [16], one of the distributed voting algorithms was developed. The sensors' tree structure acts like a tiny, robust computational device at the tree's root, built to overcome the issue of classification in using the algorithm. Some methods are presented in [3,4,13] for solving such failure issues. Reference [3] proposed a data prediction model that can reduce load on SNs to enhance network lifetime. Experimental results with the proposed model improved better accuracy and energy consumption in comparison with the traditional data prediction methods like linear regression model. In [4], the authors proposed a data reduction mechanism by building a model on both the edge node and the IoT devices. In [13], the authors proposed a data-conscious energy-saving method based on cluster head. The data are reduced by a prediction algorithm, which is done by the ARIMA model. Each round, the data model predicted is compared to the observed data. If there is dislodgement beyond the special threshold at that place, nodes send a difference of information to the cluster head. The data differences collected by the cluster head are compressed and the compressed information is sent to the sink node afterward. However, such prediction mechanisms have some disadvantages, especially for such things as a high frequency motion sensor during data collection. Two adapted protocols, namely, energy-LEACH and multi-hop-LEACH were proposed in [14]. Multi-hop LEACH protocol has offered to expand the communication trade-off among the cluster head and the sink. The term of sink refers to the cluster heads for the predicted model. Many of the advanced methods, such as those in [11], did not utilize the sensed data exchange until the model was available in the time series forecasting methods. An exciting direction includes a multi-model technique adoption, as used in [12]. In [15], the authors introduced the basic principle for a flexible method for discovering the event at constant delay y considering energy consumption reduction. The time-division multiple access (TDMA) cycle, which is a channel access method for shared medium networks, is used on the nodes under the same parent node. Therefore, a efficient data prediction technique on multivariate time series is required. The long range (LoRa) communication technology [16] has been proposed to solve the main issues of IoT applications, such as scalability and multiple sensor integration. This architecture contains gateways, sensors. Additionally, there is a network server for user applications in order to access data and a server for the application's data.

Prediction-based data-aware clustering (PDC) [17] is also a prediction method based on data-aware clustering to arrange a steady cluster of nodes. The presented prediction method shows high accuracy and low computational cost. Researchers in [18] aimed at implementing the multi-level route-aware clustering (MLRC) method for preserving energy in decentralized clustering protocols. The proposed protocol constructs a cluster and routing tree to decrease an unnecessary generation of routing control packets.

A data prediction transmission scheme based on clustering is proposed in [19]. In this model, the cluster aggregates data from the cluster head nodes and then transmits the data to the base station. However, this model has a limitation in solving the reconstruction problem of the lost data, and this issue leads to high energy consumption.

The correlation-based data collection algorithm was designed in [20]. In this algorithm, the network is divided into two main parts—clusters and sub-clusters. A amount of data needs to be collected during the data collection. Additionally, the multivariate Gaussian model is used for estimation of non-transmitted data.

Based on the available knowledge, there is a lack of an energy-efficient data collection model that not only collects and transmits data efficiently but also reduces both energy consumption and overheads in WSN.

3. The Proposed Model

In this work, we propose a model for energy-efficient data collection based on decision tree (DT), autoregressive integrated moving average (ARIMA), and Kalman filtering (KF) methods (see Figure 1). In the following sections, we first describe the algorithms and then describe the proposed hybrid model.

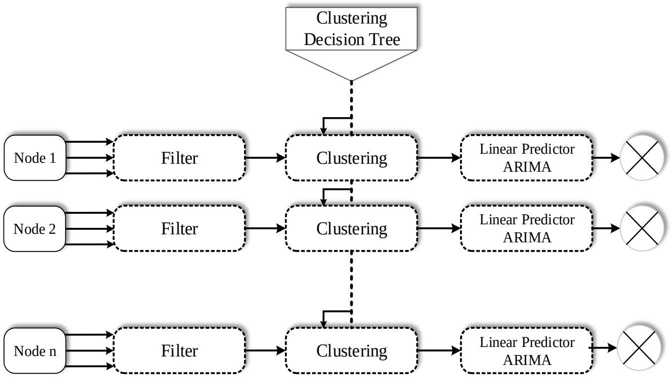


Figure 1. Proposed hybrid model.

3.1. The Algorithms Employed

- KF is an algorithm that provides estimates of some unknown variables given the measurements observed over time. Kalman filter is used to estimate states based on linear dynamical systems in state-space format. It has a relatively simple form and requires small computational power.
- DT is a popular classification algorithm to understand and interpret. The goal of DT is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data.
- ARIMA is an analysis model that uses time series data to predict future trends. It is a hybrid autoregressive model with the moving average model.

3.2. The Hybrid Method

This model predicts the data sampling requirements of sensor nodes. In the proposed model, Kalman filters are used to filter the sensor data streams associated with each node. This is based on the use of error covariance's self-tuning with fast adaptive capabilities under sudden signal input changes. The DT algorithm uses filtered data associated with each node to derive a tree for clustering the sensor data associated with each node. By combining the decision tree and Kalman filter, a high-precision prediction algorithm based on ARIMA assisted model is proposed. Compared with other machine learning methods like the deep convolutional network (DCN), the decision tree is intuitive and fast and is computationally efficient and suitable to be integrated into a real-time prediction system. Here, the hierarchical structure of the tree enables data within each node to be clustered to form some specific clusters where each cluster has its cluster head. A hierarchical structure is employed where the neighboring nodes should be clustered against a cluster head node. This hierarchical structure can decrease the communication cost and preserve energy by adaptively using the cluster head node for data collection in the cluster's coverage [21].

As explained above, DT is a classification algorithm that uses information about all the nodes to derive a tree. The cluster heads collect the information about the node, such as ID, residual energy, and position of all nodes in the cluster. The cluster heads store this information in a list form.

After collecting this information from all nodes, the DT algorithm for nodes clustering is performed. Then, the cluster heads can interact in a linear manner with the base station each time; the base station runs the DT algorithm and picks the nodes appropriate to be the next cluster head [22]. Temporal correlations between sensor readings can be presented in the time series comprising of the clustered tasks (represented by their centroids) associated with each sensor node. The correlation can be measured using mathematical models, including the linear ARIMA model [23]. Thus, the time series can be calculated using appropriate mathematical models, and the number of model parameters usually is significantly lower than the length of the whole series. The sensor node can then choose to transmit its values of data selectively. In Algorithm 1, the proposed hybrid model that includes the pseudocode description of the decision tree algorithm, cluster head procedure, and cluster nodes procedure is illustrated.

The Kalman filter has a relatively simple form and requires small computational power. Applying an effective data filtering structure is needed for removing the redundant data at the sensors along with at the cluster-head nodes to reduce the number of data transmitting. This study proposes a distributed KF to process data series noise for extending the sensor nodes' lifetime by reducing data transmissions redundancy and conserving power during continuous data collections. The filtering of data to reduce the error of covariance is computed by the general theory on state estimation. Consequently, the adaption of the end filter to changes is established based on the consensus and its optimal estimator. It is assumed that a linear dynamical system is defined by the following Equation (1):

$$x(t+1) = Fx(t) + w(t) \tag{1}$$

whereby t represents the time index; $x(t)$ represents the state of a system; the F matrix $M \times M$ describes the way the system changes over time; and $w(t)$ calculates the uncertainty process as a function of the weight matrices, modeled by utilizing the white Gaussian noise at zero-mean with covariance matrix known as Q_w. The N sensors network performs the system's monitoring, engaged at specified areas randomly, and its observations using time t are as follows:

$$y(t) = Hx(t) + v(t) \tag{2}$$

The N-dimension vector $y(t)$ collects the N sensors' observations; the matrix $H(N \times M)$ measures time-varying channel fading which impacts every position's observation; and $v(t)$ is aligned with the observation noise, which is modeled with the Gaussian with Q_v covariance matrix zero-mean. This is commonly used in stating the linearity and Gaussian assumptions; the Kalman filter describes the scheme that is recursive [24] and offers a system state's optimum estimates. Every node i in a scenario that distributed only has access to its $y_i(t)$ observation. As such, every node should calculate the expressions using Equation (3) according to only its observation and through the exchange of information with just the one-hop neighbors. Due to the time (t) vector observations, the a priori estimation $\hat{x}^-(t)$ is defined by the filter to yield the estimation for posterior, as shown in the following:

$$\hat{x}^+(t) = \hat{x}^-(t) + G(t)\left[y(t) - H\hat{x}^-(t)\right] \tag{3}$$

where $G(t)$ represents the filter gain. The expression in (1) in this case, can be measured using all the distributed nodes, and every node can run the local filter version. The performance of the distributed Kalman filter is effective, adaptable, and strong for stochastic assessment on the WSN, which is influenced by various network nodes' unreliable connections. To deal with the missing data, it assumed that missing values were randomly distributed. Before making predictions, the Kalman filter marginalizes the lost data by merging the estimations of all the nodes that are active. We denote the missing parts of measurements as $\bar{z}$, and we assume the estimated information as $\tilde{z}$, i.e., $z = [\bar{z}, \tilde{z}]$. While $\bar{z}$ is not empty, $p(q \mid c, z)$ is used to marginalize over the missing data $\bar{z}$ to predict $p(q \mid c, \tilde{z})$.

Consequently, the Kalman filter keeps and sends the attributes with missing data as the essential attributes into the decision tree.

Algorithm 1: Hybrid algorithm.

```
Result: The output is the prediction of the ensemble trees for a given new input
        $x : \frac{1}{m} \sum_{m=1}^{M} T_m(x)$
The filter data by Kalman Filter generated a total number of data sample $n$ and $p$ input
  variables;
Initially, the total number ($M$) of trees to be generated and the number $v < p$ of variables used
  for each tree are determined:;
for $m = 1$ to $M$ do
    Draw a random sample $S^*$ with size $n$ with replacement from the original training data;;
    Grow a tree $T_m$ using the training sample $S^*$ through the following loop:;
    while the minimum node size $nod_{min}$ is reached do
        For the terminal node of the tree;
        Randomly select $v$ variables out of the $p$ variables;
        Select the best pair of split variable /point among the $v$ variables;
        Split the node into two daughter nodes;
    end
    Output the constructed tree $T_p(x)$;
end

Procedure Sub(Cluster Nodes):
    if the linear predictor (ARIMA) is $|x_t - \hat{x}_t| > \epsilon$ then
        Send the data to the cluster head;
        Update the history data;
    else
        Perform prediction to update the historic data;
    end
Procedure Sub(Cluster Head):
    for each data point, i in the cluster do
        if the linear predictor (ARIMA) is covariance stationery then
            active the prediction of node $i$;
        else
            Inactive the prediction of node $i$;
        end
    end
    for each data point, i in the cluster do
        if receive a data value i then
            Update the history data for the data point $i$;
        else
            Perform prediction to update the history data;
        end
    end
[Note]: The minimum node size $nod_{min}$ is the minimum number of observations in each end
  node.
```

3.3. Adaptive Update of Clustering by DT

A method for adapting the clustering is needed to measure the local operations. One option is a total re-clustering; however, this could be quite costly as it includes the setting up of a map for clustering of all the data points for each sensor node. The complete transformation in the cluster membership suggests that the total historical data and models should be built from the beginning. The DT algorithm presented in this section is for performing the initial clustering, and subsequently, the dynamic splitting and merging of clusters with low-cost communication. At the initial phase, the randomized node sets are chosen as the cluster heads. Upon receiving control information of every sensor node, the DT algorithm should run to select a new suitable cluster headset from the entire set of sensor nodes. First, it creates a cluster tree using distances between observations in the data by using Euclidean distance. After that, the list of cluster heads is transmitted to every sensor node. Next, each sensor node attaches itself to its cluster head.

Most algorithms for clustering can be utilized in such study; the update for adaptive clustering is normally needed to measure the change in the patterns of the locality. A total re-clustering is an option; however, this could be quite costly. That includes the setting up of a map for clustering of all the sensors. The complete transformation in the cluster membership suggests that the totality of historic data and models should be built from the beginning. The DT algorithm is presented in this section, for the clusters' dynamic splitting and merging with a low-cost communication. Clusters of sensor nodes that can be either active or inactive are taken into consideration. Active sensor nodes continuously monitor the x attribute while generating the x_t data values at every time tick t. At least one sensor is responsible for monitoring each point of the area. The connected nodes can create the set of active sensors, the result of the clustering mechanism. To assist in the computation of intersection points, each node keeps a table which contains the information of each neighbor (such as node ID, position and residual energy of all nodes, location, status: active/inactive, number of nodes before eliminating i-th node, number of nodes after eliminating i-th node, number of levels before eliminating i-th node, number of levels after eliminating i-th node) and periodically updates its current location and status. If the sensors are active, they monitor and generate a datum x_t at every time point t. A sensor node with no ability to make local predictions at cluster heads transmits the totality of values of the data to the cluster head and then calculates the distributed data accordingly. The sensor node can transmit its values of data to the cluster heads selectively using local prediction based on an error bound the error bound $\epsilon > 0$. In general, the proposed scheme in this study was developed according to the following factors: (i) distance of a node from the cluster centroid; (ii) the degree of mobility; (iii) the remaining battery power; and (iv) the vulnerability index. Obviously, the vulnerability index shows an effective way for computing the node vulnerability in sensors' tree structure. To compute the vulnerability factor of each node, the following equation can be used (4):

$$V_{index} = \frac{n_i}{n_j} \times \frac{a_i}{a_j} \tag{4}$$

where n_i is number of nodes before eliminating the i-th node, n_j is the number of nodes after eliminating the i-th node, a_i is the number of levels before eliminating i-th node, and a_j is the number of levels before eliminating i-th node. The BS measures the distance in every node and the r cluster centroid. A shorter distance denotes a higher probability of the node to be a cluster head. Moreover, the higher the power of the battery, the higher the probability of the node becoming a cluster head. The mobility of the node affects the lifetime of the network significantly. Upon completion of the start-up phase, every sensor node transmits its data to a specific cluster head, which, in turn, broadcasts the members' list to other nodes [25]. The process of the cluster head selection is repeated using a predetermined interval or via meeting the criteria for the threshold value. The selective transmission model is the approximation of ϵ-loss: Due to the error bound $\epsilon > 0$, a sensor node transmits the x_t value to the cluster head if $|x_t - \hat{x}_t| > \epsilon$, whereby $\hat{x}_t$ represents a predicted data value. If the chosen value is closer to the predicted value, it would be pointless to be documented. Variation of the chosen value from the predicted value is an important consideration in measuring data distribution.

3.4. ARIMA Prediction Model

In this study, the methodology aimed at designing the best possible ARIMA-based model for predicting the energy consumption of sensor nodes during data collection. ARIMA models are univariate as they utilize the history of the time series for expressing how the variables react with an earlier stochastic variation. ARIMA might be executed through a four-step process after gathering historical data of the relevant parameters. The four steps include: (i) identifying the model; (ii) estimating the parameters; (iii) recognizing the model; and (iv) verifying and predicting the model [26]. A general ARIMA (p, d, q) model describing the time series expressed as follows:

$$\phi(B) = \bigtriangledown^d x_t = \theta(B) e_t \tag{5}$$

whereby x_t and e_t represent energy consumption and random error at time t, respectively. B refers to the backward shift operator described using $Bx_t = x_{t-1}$, and about $\bigtriangledown$; d represents the order of differencing; $\bigtriangledown = 1 - B$, $\bigtriangledown^d = (1 - B)^d$. $\phi(B)$ and $\theta(B)$ represent autoregressive (AR) and moving average (MA) operators of orders p and q separately, as described in the following:

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p \tag{6}$$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q \tag{7}$$

whereby $\phi_1, \phi_1, \cdots, \phi_p$ represent the autoregressive coefficients and $\theta_1, \theta_2, \cdots, \theta_q$ represent the moving average coefficients. The time series x_t can be represented using the linear transfer function of the noise series:

$$x_t = \mu + \varphi(B) e_t \tag{8}$$

where

$\varphi(B)$ can be computed as $\varphi(B) = \theta(B)/\phi(B)$. Figure 2 shows the Flow of proposed work.

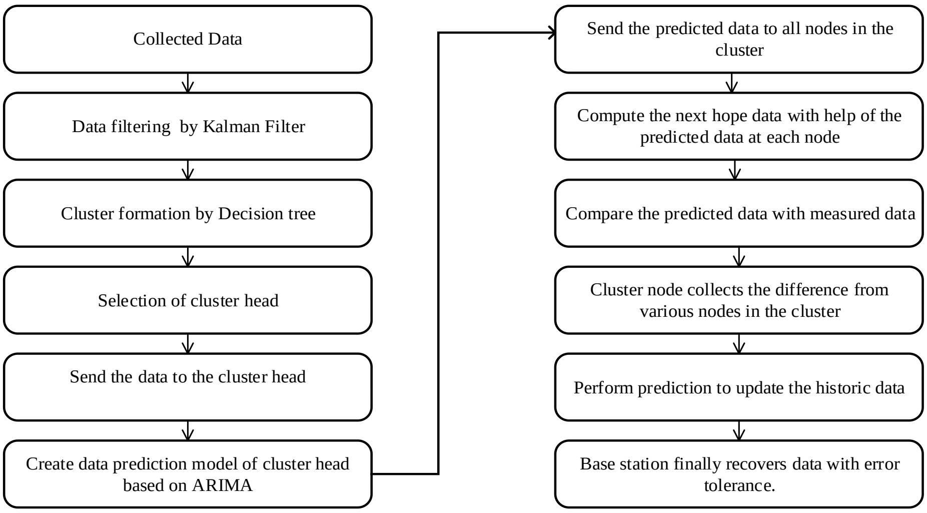

Collected Data
Data filtering by Kalman Filter
Cluster formation by Decision tree
Selection of cluster head
Send the data to the cluster head
Create data prediction model of cluster head based on ARIMA
Send the predicted data to all nodes in the cluster
Compute the next hope data with help of the predicted data at each node
Compare the predicted data with measured data
Cluster node collects the difference from various nodes in the cluster
Perform prediction to update the historic data
Base station finally recovers data with error tolerance.

Figure 2. Flow of proposed work.

4. Experiment Evaluation and Analysis

We conducted a series of experiments to compare the performance of the proposed algorithm with alternative techniques. We used the sensor datasets of the Intel Lab data [27] to measure the performance of our prediction method. The data collected from 54 nodes spread around their laboratory during one month. We filled in the missing data values with the averages of the values at different time epochs. We selected the entire temperature records for one week (15 September to 21 September). Every node contains an average of five nodes in the radio range. Data in each node i were modeled according to $x_t = \alpha_i x_{t-1} + e_t$, whereby $e_t \sim N(0, 0.01)$ and $\alpha_i \sim N(1, 0.01)$. All nodes were used to generate 2500 values of data. Each node initialized according to $\alpha_1 = 1$ and $x_0 = 0$. The distributed density estimation methods based on parametric techniques were chosen for comparison [28–30]. The method [31] stores the last datum sent at the sink and sensors. If the data value contains an error bound, then the sensor node cannot broadcast the data value. This model has a training phase with a probability density function (pdf) that refers to a set of obtained attributes. These consist of an advanced robust aggregation technique to extract statistical information from sensor networks described in [30], and that includes a distributed algorithm to compute a Gaussian mixture model (GMM), and a probabilistic model [32].

For a fair comparison, our model uses a GMM algorithm for cluster breakup or inter-cluster aggregation stage. Since we propose to use a hybrid DT optimized ARIMA model in our system, the two methods have similar message sizes to have a fair comparison. The cluster head preserves a circular array of historical data for each cluster member. Since our model assists cluster members in place of cluster heads, we quantify the energy consumption of cluster members. There exist 15 clusters and 45 cluster members on average. We show the sum of the energy consumption of all these 40 cluster members [16]. As energy consumption data involve non-stationary properties, different methods must be applied to change the non-stationary properties. An ARIMA model [ARMA (p, q)] for x time series involving n cases was predetermined through forecasting Equations (9) and (10) as follows:

$$x_t = \sum_{i=1}^{p} A_i x_{t-1} + \sum_{j=1}^{q} B_j \omega_{t-j} + \omega_t \tag{9}$$

$$\hat{x}_t = \alpha_1 x_{t-1} + \cdots + \alpha_p x_{t-p} + \omega_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \omega_t \tag{10}$$

where x_t indicates the original results, and $\hat{x}_t$ represents the forecasting data. The m-dimensional vector ω_t represents unmodified random data with a mean of 0 and covariance matrix of R, and $\theta = (p, q)$ shows the order of the forecast for which p is the number of auto-regressive expressions, q is the number of lagged prediction errors and $A_1, \cdots, A_p$ and $B_1, \cdots, B_q$ are the $m \times m$ coefficient matrices of the multivariate (MV) ARIMA model. Before implementing the ARIMA, the stationary process was tested in terms of the AR and MA coefficients. The coefficients of the AR and MA matrices are presented in Table 1. The data represent the optimal values of the AR and MA coefficients. Note that as measurement error increases, the AR and MA coefficients will tend to become zero. According to Table 1, all the AR and MA coefficients were weak. The coefficients should be precisely equal to one. Furthermore, their high standard deviations between 4.05 and 7.85 were not significant for all the cases. The heterogeneity of data distribution resulted in low precision (or relatively high standard deviation). After testing the AR and MA coefficients, it is concluded that the mixed ARIMA model is the desirable choice.

In $ARIMA(p, d, q)$, p corresponds to the number of auto-regressive terms, q corresponds to the number of lagged forecasting errors and d corresponds to the number of non-seasonal differences. It is assumed that random errors (ω_t) are independent with equivalent distribution with a steady variance. To evaluate the model's performance, main indices such as average relative error (ARE), (RMSE), and (MAE), were measured and revealed the prediction accuracy of the models. The basic ARIMA model parameters are shown in Table 2.

Table 1. Autoregressive (AR) and moving average (MA) coefficients of parameters.

ARIMA	Coefficients	St. Dev.
AR (1)	−0.7144	7.0748
AR (2)	−0.4466	4.0677
AR (3)	1.2873	5.0433
AR (1)	−0.7144	7.0748
AR (2)	−0.4466	4.0677
AR (3)	1.2873	5.0433
AR (1)	0.1765	4.753
AR (2)	0.1106	4.043
AR (3)	0.1076	7.753
MA (1)	−0.4131	7.0293
MA (2)	1.7011	5.0988
MA (3)	−0.1510	5.0728
MA (1)	0.4355	7.0981
MA (2)	1.2788	5.1067
MA (3)	0.1314	4.0433
MA (1)	−0.3081	6.233
MA (2)	0.2944	4.053
MA (3)	0.8733	5.012

Table 2. The values of main indices with different numbers of nodes.

	Number of Nodes		
	500	**1000**	**1500**
ARE	0.4	4.5	5.5
MAE	0.5	1.05	0.55
RMSE	0.8	1.2	0.52

Figure 3 shows how the proposed algorithm works on the member nodes and cluster head node of cluster 1. The error threshold is set to ±0.1°. The real temperature value and prediction value for 2000 samples are plotted in Figure 3a. Figure 3b is the larger version of Figure 3a for 600 samples from time 600 to 800. Figure 3 reveals the fitting and forecasting results. The optimized hybrid model decreases the forecasting error based on the Kalman filter–DT optimized ARIMA model in all the datasets. This represents an important improvement for energy consumption forecasting. The proposed model has been compared with [13] that used ARIMA for WSN data prediction. In [13] ARIMA has been applied to stationary data, which means the data series have no trend, with little variations of the mean that has a constant amplitude and has its short-term random patterns looking the same over time.

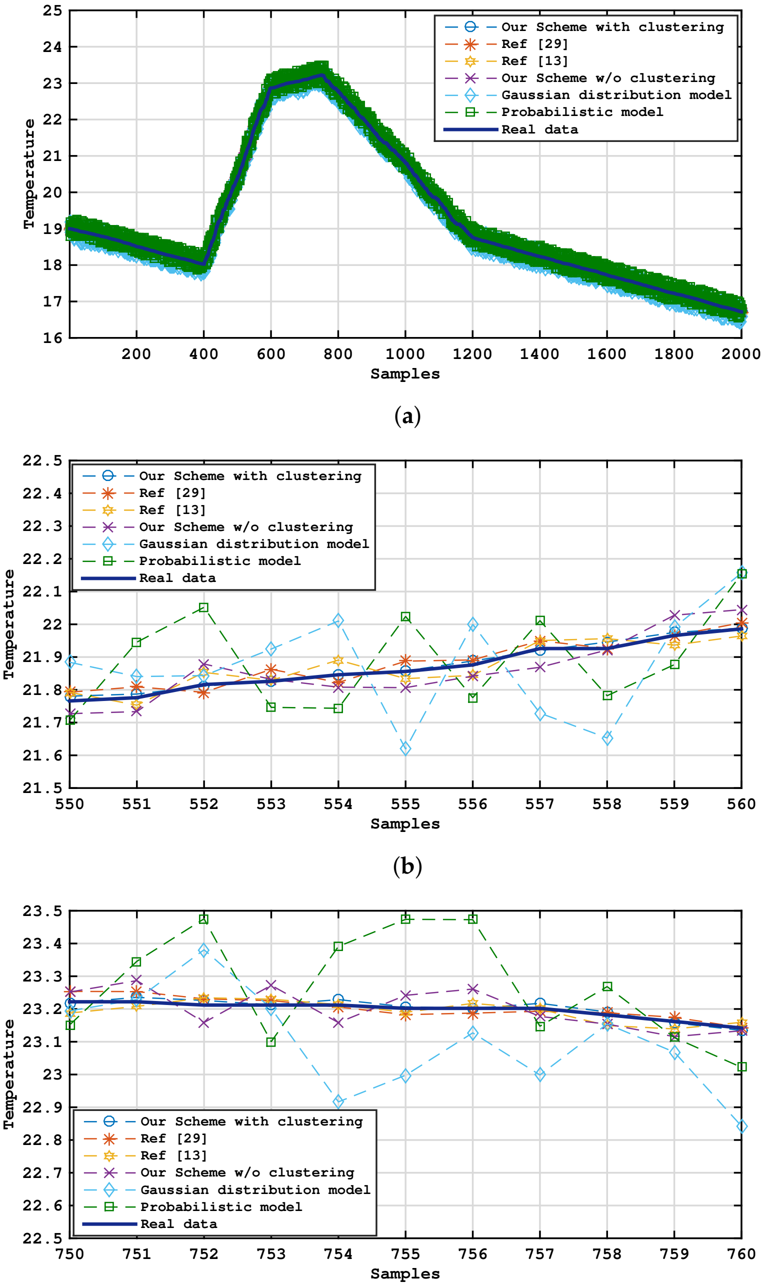
Traditional performance indices, such as (ARE), ($RMSE$), and (MAE), are used as measures for prediction accuracy. The number of nodes was varied to examine the performance sensitivity. Note that each one denotes the ratio between the transmission energy consumption and the prediction energy consumption. These indices are shown in Figure 4 as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{11}$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{12}$$

$$ARE = \frac{\sum_{i=1}^{n}(|y_i - \hat{y}_i|\,/y_i)}{n} \tag{13}$$

Figure 4 shows the performance indices (ARE), (RMSE), and (MAE) over the number of nodes i.e., 500, 1000, and 1500, in which the our hybrid model involved the lowest RMSE, MAE, and ARE, as assessed for various numbers of nodes. This finding indicated the capability of the hybrid model in successfully reducing the error of the predicted values. With consideration of the different suggested models in the present work, acceptable outcomes can be demonstrated with great correctness utilizing superior linear and non-linear methods, particularly when both models exhibited precise and great forecasting robustness. Due to the ARIMA model's capability to satisfactorily estimate the linear part of data, it is demonstrated that the use of the DT and KF algorithms for non-linear parts of the model could increase predicting results more efficiently.



(**a**)

(**b**)

(**c**)

Figure 3. The total number of packets versus the number of nodes.

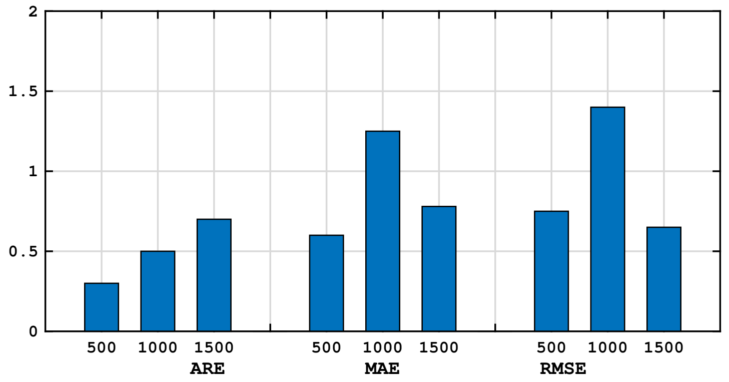


Figure 4. Performance indices ARE, RMSE, and MAE vs. number of nodes, i.e., 500, 1000, and 1500.

A MATLAB simulator was implemented where simulation parameters were fixed based on hardware configurations of MICA2 [31]. For conciseness, only the representative results were reported. A real-world sensor dataset, Intel Lab data [27] dataset was used for scalability analysis of the algorithms which is shown in Figure 5. Greater scalability of the proposed framework could be observed by integrating data aggregation. This is due to the ability of the distributed technique to complete the local updating of data and local prediction, whereas the centralized system sustains a high-communication cost to transmit the data to the sink. The cluster head collects data within each cluster. Then, the cluster head should complete local prediction on the data distribution. The members of each cluster need to execute prediction, and then predicted data should transmit to the cluster head. Hence, each cluster head has a clear perspective of all sensor data through the cluster. Thus, the communication cost significantly decreased. The energy consumed for a fixed number of frames and different head-sizes compared in this section. We assess the number of transmitted packets by all nodes in the models [27,28].

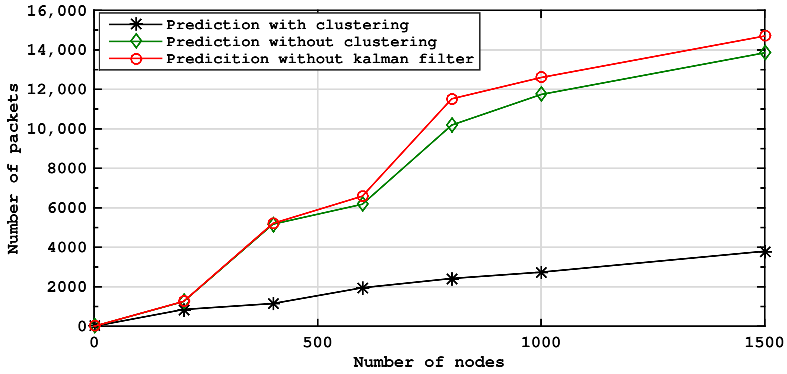


Figure 5. The total number of packets versus the number of nodes.

Figure 6 demonstrates the scalability of the different methods with the size of the network using a real-time dataset [32]. The Gaussian distribution scheme in [28] suffers a high-communication cost for spreading the data to the sink. Additionally, the probabilistic model described in [24,29], is complicated by high-level computations such as aggregating, with the costly computational costs. This is because of local and global probabilistic prediction models that are constructed at the sensor nodes and sink respectively. After comparison, we can realize that the increased scalability of our algorithm using the aggregation of data. Our model can perform local data updates and local data prediction. Thus, our proposed algorithm has better performance, as it is possible to utilize our data aggregation approach to process raw data at the sensor nodes or at middle nodes to decrease packet transmissions and save energy. In this section, the energy consumption for a fixed number of frames and diverse head-sizes is compared.

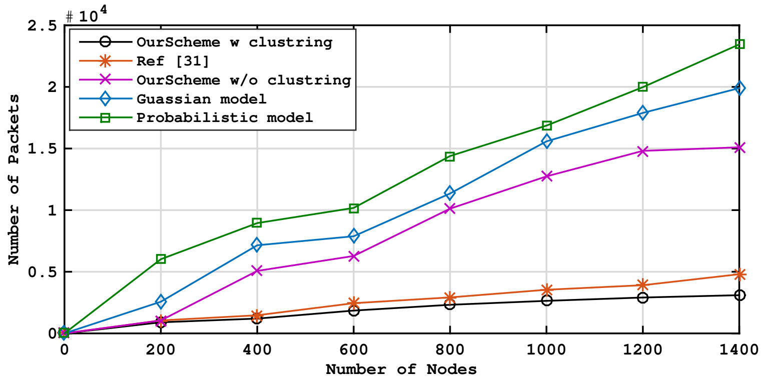


Figure 6. Comparison of total number of packets and the number of nodes.

Figure 7 displays the variation in the energy consumed per node in terms of the number of clusters and network diameter. The clusters' number and the amount of energy consumption in one round are designated on the x-axis and y-axis, respectively. The energy consumed in one node is estimated by Equation (9). Energy (E) is the energy of a node at a fixed time. This energy should be appropriate for at least one round. Each node becomes a member of the headset for one time during one round and a non-cluster head for ($\frac{n}{km} - 1$) times. There are k clusters and k nodes. In each iteration of DT, k nodes were selected for each cluster. Thus, km nodes were selected in each iteration as members of the headsets. Some iterations needed for all of the selected n nodes using ($\frac{n}{km}$), which is the number of iterations needed in one round. Since there are m nodes in a headset, ECH/inter/cluster is uniformly shared between the headset members, as follows:

$$E_{CH/node} = \frac{E_{CH/inter/cluster}}{m} \tag{14}$$

The graph demonstrates that energy consumption decreased where the number of clusters is improved. The graph illustrates that the optimal sort of clusters lies between 35 and 60 for 1500 nodes. There exist 15 clusters and 45 cluster members on average. Additionally, the graph illustrates that the sensor nodes should send data to distant cluster heads when the number of clusters is lower than the optimal range, for example, 20. In contrast, when the number of clusters is larger than the optimal set there will be further communications to the distant base station.

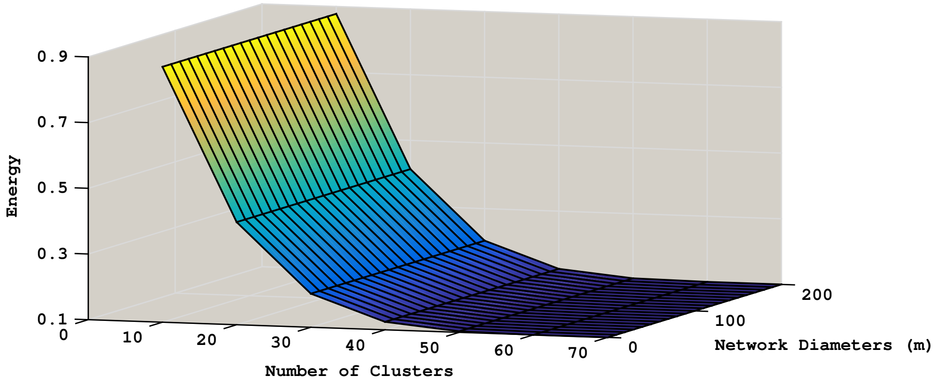


Figure 7. Energy consumption based the on number of nodes.

Figure 8 illustrates an energy dissipation model used by the transmitter and receiver. Computation of energy required is carried out using the model [9] during data transmission. The transmitter dissipates energy to perform radio electronics and amplification. The receiver dissipates energy to execute radio electronics only. Transmission (T_{trans}) is the time required to transmit the message through the transmission channel given by (14).

$$T_{trans} = \frac{length\,of\,message\,(bits)}{transmission\,rate\,of\,a\,node\,(kbps)} \tag{15}$$

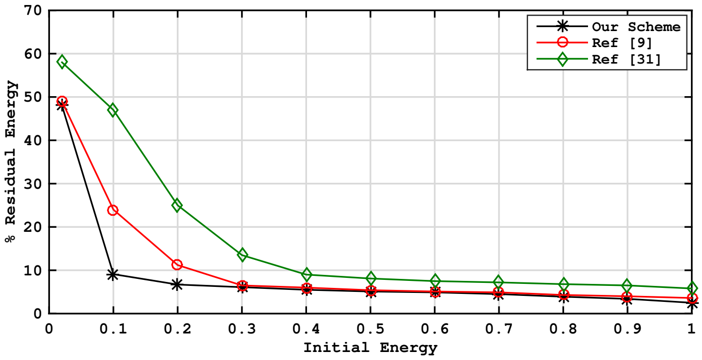


Figure 8. The computed percentage of residual energy across the network.

5. Conclusions

In this paper, an energy-efficient data collection model based on clustering and prediction has been proposed. In the clustering phase, sensor nodes form the clusters while the cluster heads collect and store the data measured by sensor nodes. The proposed hybrid prediction model was utilized to examine the trade-off between the communications and prediction. The performance of the model has been evaluated by using various numbers of nodes at different periods. Based on the simulation experiment, it was proven that the proposed model significantly outperformed the other related approaches in terms of prediction accuracy and energy efficiency. As a result, it can reduce the energy consumption used for data collection in hierarchical networks and extends the network lifetime—incredibly, even when a high number of clusters are allocated. The traffic generator, such as a LoRa network operator, can be integrated into the proposed application as future work. We plan to use the LoRa traffic generator to easily field test the network distribution of proposed method on a large-scale geographical area, to reduce the initial deployment costs.

Author Contributions: Methodology, S.A.S. and S.G.; software, S.A.S., S.G. and M.Z.; validation, S.A.S. and S.G.; investigation, S.A.S., S.G. and M.Z.; writing–original draft preparation, S.G. and S.A.S.; writing—review and editing, S.A.S. and S.G.; supervision, N.K., S.A.I. and M.A.; funding acquisition, N.K., S.A.I., M.A. and Z.M.Z. Resources: S.A.I. and Z.M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially funded by Universiti Teknologi Malaysia, under Fundamental Research Grant with Vote Number 21H12, and Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (grant ID: GGPM-2020-029 and grant ID: PP-FTSM-2020).

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

References

1. Arbi, I.B.; Derbel, F.; Strakosch, F. Forecasting methods to reduce energy consumption in WSN. In Proceedings of the 2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Turin, Italy, 22–25 May 2017; pp. 1–6.
2. Jiang, H.; Jin, S.; Wang, C. Prediction or not? An energy-efficient framework for clustering-based data collection in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2010**, *22*, 1064–1071. [CrossRef]
3. Agarwal, A.; Dev, A. A Data Prediction Model Based On Extended Cosine Distance For Maximizing Network Lifetime of WSN. *Wseas Trans. Comput. Res.* **2019**, *7*, 23–28.
4. Tayeh, G.B.; Makhoul, A.; Laiymani, D.; Demerjian, J. A distributed real-time data prediction and adaptive sensing approach for wireless sensor networks. *Pervasive Mob. Comput.* **2018**, *49*, 62–75. [CrossRef]
5. Peksa, J. Prediction Framework with Kalman Filter Algorithm. *Information* **2020**, *11*, 358. [CrossRef]

6. Ferrero, R.; Pegoraro, P.A.; Toscani, S. Synchrophasor Estimation for Three Phase Systems Based on Taylor Extended Kalman Filtering. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 6723–6730.
7. Zhou, J.; Gu, G.; Chen, X. Distributed Kalman filtering over wireless sensor networks in the presence of data packet drops. *IEEE Trans. Autom. Control.* **2018**, *64*, 1603–1610. [CrossRef]
8. Wang, J.; Zhu, R.; Liu, S. A differentially private unscented Kalman filter for streaming data in IoT. *IEEE Access* **2018**, *6*, 6487–6495. [CrossRef]
9. Pandey, O.J.; Hegde, R.M. Low-latency and energy-balanced data transmission over cognitive small world WSN. *IEEE Trans. Veh. Technol.* **2018**, *67*, 7719–7733. [CrossRef]
10. Yan, R.; Sun, H.; Qian, Y. Energy-aware sensor node design with its application in wireless sensor networks. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 1183–1191. [CrossRef]
11. Goel, S.; Passarella, A.; Imielinski, T. Using buddies to live longer in a boring world [sensor network protocol]. In Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), Pisa, Italy, 13–17 March 2006; p. 5.
12. Jang, W.S.; Healy, W.M.; Skibniewski, M.J. Wireless sensor networks as part of a web-based building environmental monitoring system. *Autom. Constr.* **2008**, *17*, 729–736. [CrossRef]
13. Diwakaran, S.; Perumal, B.; Devi, K.V. A cluster prediction model-based data collection for energy efficient wireless sensor network. *J. Supercomput.* **2019**, *75*, 3302–3316. [CrossRef]
14. Xiangning, F.; Yulin, S. Improvement on LEACH protocol of wireless sensor network. In Proceedings of the 2007 International Conference on Sensor Technologies and Applications (SENSORCOMM 2007), Valencia, Spain, 14–20 October 2007; pp. 260–264.
15. Ouni, S.; Gherairi, S.; Kamoun, F. Data aggregation and pipelining scheduling protocols for real–time wireless sensor networks. *Int. J. Hoc Ubiquitous Comput.* **2013**, *12*, 56–64. [CrossRef]
16. Lavric, A.; Petrariu, A.I.; Coca, E.; Popa, V. LoRa Traffic Generator Based on Software Defined Radio Technology for LoRa Modulation Orthogonality Analysis: Empirical and Experimental Evaluation. *Sensors* **2020**, *20*, 4123. [CrossRef] [PubMed]
17. Ashouri, M.; Yousefi, H.; Basiri, J.; Hemmatyar, A.M.A.; Movaghar, A. PDC: Prediction-based data-aware clustering in wireless sensor networks. *J. Parallel Distrib. Comput.* **2015**, *81*, 24–35. [CrossRef]
18. Sabet, M.; Naji, H. An energy efficient multi-level route-aware clustering algorithm for wireless sensor networks: A self-organized approach. *Comput. Electr. Eng.* **2016**, *56*, 399–417. [CrossRef]
19. Yang, X.; Wen, Y.; Yuan, D.; Zhang, M.; Zhao, H.; Meng, Y. 3-D compression-oriented image content correlation model for wireless visual sensor networks. *IEEE Sens. J.* **2018**, *18*, 6461–6471. [CrossRef]
20. Gedik, B.; Liu, L.; Philip, S.Y. ASAP: An adaptive sampling approach to data collection in sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2007**, *18*, 1766–1783. [CrossRef]
21. Zhang, D.; Li, G.; Zheng, K.; Ming, X.; Pan, Z.H. An energy-balanced routing method based on forward-aware factor for wireless sensor networks. *IEEE Trans. Ind. Inform.* **2013**, *10*, 766–773. [CrossRef]
22. Shokouhifar, M.; Jalali, A. Optimized sugeno fuzzy clustering algorithm for wireless sensor networks. *Eng. Appl. Artif. Intell.* **2017**, *60*, 16–25. [CrossRef]
23. Olofsson, T.; Ahlen, A.; Gidlund, M. Modeling of the fading statistics of wireless sensor network channels in industrial environments. *IEEE Trans. Signal Process.* **2016**, *64*, 3021–3034. [CrossRef]
24. Imani, M.; Dougherty, E.R.; Braga-Neto, U. Boolean Kalman filter and smoother under model uncertainty. *Automatica* **2020**, *111*, 108609. [CrossRef]
25. Yin, Y.; Shi, J.; Li, Y.; Zhang, P. Cluster head selection using analytical hierarchy process for wireless sensor networks. In Proceedings of the 2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, Helsinki, Finland, 11–14 September 2006; pp. 1–5.
26. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
27. Madden, S. *Intel Lab Data*; Intel Research Lab: Berkeley, CA, USA, 2006.
28. Jiang, H.; Jin, S. Scalable and robust aggregation techniques for extracting statistical information in sensor networks. In Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06), Lisboa, Portugal, 4–7 July 2006; p. 69.
29. Wu, M.; Tan, L.; Xiong, N. Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications. *Inf. Sci.* **2016**, *329*, 800–818. [CrossRef]

30. Chu, D.; Deshpande, A.; Hellerstein, J.M.; Hong, W. Approximate data collection in sensor networks using probabilistic models. In Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), Atlanta, GA, USA, 3–7 April 2006; p. 48.
31. Pala, Z. Effects of Mica2-based discrete energy levels on the lifetime of cooperation neighbor sensor networks. *Turk. J. Electr. Eng. Comput. Sci.* **2016**, *24*, 2671–2678. [CrossRef]
32. Barak, S.; Sadegh, S.S. Forecasting energy consumption using ensemble ARIMA–ANFIS hybrid algorithm. *Int. J. Electr. Power Energy Syst.* **2016**, *82*, 92–104. [CrossRef]

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