Computer Communications 34 (2011) 793–802

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Prediction-based data aggregation in wireless sensor networks: Combining grey model and Kalman Filter

Guiyi Wei [a,*], Yun Ling [a], Binfeng Guo [a], Bin Xiao [b], Athanasios V. Vasilakos [c]

[a] *School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, China*
[b] *Department of Computing, Hong Kong Polytechnic University, Hong Kong*
[c] *Department of Computer and Telecommunications Engineering, University of Western Macedonia, Greece*

ARTICLE INFO

Article history:
Received 26 January 2010
Received in revised form 30 September 2010
Accepted 5 October 2010
Available online 12 October 2010

Keywords:
Wireless sensor networks
Data collection protocol
Data aggregation
Grey model
Kalman Filter

ABSTRACT

In many environmental monitoring applications, since the data periodically sensed by wireless sensor networks usually are of high temporal redundancy, prediction-based data aggregation is an important approach for reducing redundant data communications and saving sensor nodes' energy. In this paper, a novel prediction-based data collection protocol is proposed, in which a double-queue mechanism is designed to synchronize the prediction data series of the sensor node and the sink node, and therefore, the cumulative error of continuous predictions is reduced. Based on this protocol, three prediction-based data aggregation approaches are proposed: Grey-Model-based Data Aggregation (GMDA), Kalman-Filter-based Data Aggregation (KFDA) and Combined Grey model and Kalman Filter Data Aggregation (CoGKDA). By integrating the merit of grey model in quick modeling with the advantage of Kalman Filter in processing data series noise, CoGKDA presents high prediction accuracy, low communication overhead, and relative low computational complexity. Experiments are carried out based on a real data set of a temperature and humidity monitoring application in a granary. The results show that the proposed approaches significantly reduce communication redundancy and evidently improve the lifetime of wireless sensor networks.

1. Introduction

Wireless sensor networks consist of a large number of low-cost sensor nodes which form a multi-hop ad hoc network through wireless communication [19]. In general, sensor nodes rely on battery only and once deployed, they are usually unable to be recharged. Therefore, power is a critical resource in wireless sensor networks. Reducing energy consumption is of great importance in improving the lifetime of wireless sensor networks.

In wireless sensor networks used in environmental monitoring, a large number of sensor nodes collect information and return collected information to a base station(s) where it is processed, analyzed, and used. Since the sensor node is energy constrained and its valid communication distance is limited, it is infeasible for all the sensors to transmit data directly to the base station (or sink node). In most environmental monitoring applications, sensed data may be of high temporal or spatial correlation, and applications can tolerate some loss of data accuracy. Therefore, it is possible to use a data aggregation approach to process raw data at the sensor nodes or at intermediate nodes to reduce packet transmissions and save energy [3,20].

Since the data generated by sensor nodes during continuous sensing periods usually are of high temporal coherence, it indicates there are redundant data in the continuous data sequence, which causes unnecessary data transmission and energy consumption. In many environmental sensing applications, e.g. granary monitoring, data flow is many-to-one through a reverse multicast tree, from leaf sensor nodes to a small number of sink nodes. In these cases, transmitting redundant data will incur a serious waste of communication bandwidth and energy. The efficiency of data collection will decrease when each node sends all data to the sink node. Furthermore, the difficulty of scheduling at the link layer will increase and cause more frequent collisions [20]. Data aggregation techniques which exploit temporal correlation of the sensed data are needed to resolve these two problems.

Model-driven data aggregation approaches take advantage of data coherence to remove redundancy and reduce transmissions among sensor nodes [2]. They are effective in improving energy efficiency and extending the lifetime of the wireless sensor network [2]. In-network processing usually aggregates data at intermediate nodes between the sources and the sinks by using aggregation functions (such as maximum, minimum, sum and average). However, using aggregation functions causes a loss of data resolution. Furthermore, the difference between the processed data and the original data may be too large to tolerate in

[*] Corresponding author. Tel.: +86 136 06619504; fax: +86 571 28008303.
E-mail address: weigy@zjgsu.edu.cn (G. Wei).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.10.003

some environmental sensing applications. For example, granary monitoring must continuously gather temperature and humidity data from every sensor node with relatively small tolerated-error. Xin et al. [21] analyze some more complicated aggregation approaches, including data mining and multiple-source-queries routing. These approaches can provide higher accuracy. However, they consume a large amount of computational power and storage resources, because their pre-processing stages require $O(n^2d)$ transmissions, where n is the number of nodes and d is the diameter of the network [21]. Therefore, these complex approaches are infeasible for most environmental monitoring applications.

This paper proposes a novel prediction-based data collection protocol to reduce redundant data transmission. A double-queue mechanism is designed to synchronize the predicted data series in the sensor node and the sink node, and therefore, the mechanism avoids the cumulative error of continuous predictions. Based on this protocol, we design three prediction-based data aggregation approaches (GMDA, KFDA, and CoGKDA). The proposed approaches are used to predict the data of the next period at both sensor and sink ends based on the same small number of recent data items. When data of the next period is sensed, the sensor node compares the predicted data with the sensed data. The sensor node does not forward the sensed data to the sink node when the prediction error is less than a pre-configured threshold value. In this case, the sink node considers the predicted data as the sensed data in current sensing period. Therefore, unnecessary transmission is eliminated and energy is saved. The sensor node must send the sensed data to the sink node when the prediction error is out of the pre-configured threshold. The pre-configured threshold is a tunable parameter for users to control the accuracy of predicted data. It is inversely proportional to data accuracy. Experiments and evaluations demonstrate the proposed approaches can significantly reduce communication redundancy and improve the network lifetime in environmental monitoring applications.

Our contribution can be summarized as follows:

- A prediction-based data collection protocol is proposed to specify the cooperative processes between sensor node and sink node, in which a novel double-queue mechanism is designed to synchronize the prediction data series in the sensor node and the sink node, hence cumulative error in continuous predictions is avoided.
- By integrating the merits of the grey model in quick modeling with the advantages of Kalman Filter in processing data series noise, we have designed the CoGKDA algorithm for environmental monitoring wireless sensor networks. CoGKDA exhibits high data accuracy, low communication overhead, and relatively low computational complexity. Furthermore, CoGKDA can extend the sensor nodes' lifetime by reducing data transmissions redundancy and conserving power during continuous data collections.

The rest of the paper is organized as follows: In Section 2, state-of-the-art methods in data aggregation are reviewed. Section 3 presents a novel prediction-based data collection protocol. Section 4 describes the Grey-Model-based Data Aggregation approach. Another data aggregation approach based on the Kalman Filter is given in Section 5. In Section 6, a combined data aggregation approach and its concrete algorithm are presented in detail. Experiments and performance evaluation are presented in Section 7 and concluding remarks are made in Section 8.

2. Related work

There has been a lot of work done in the field of data-driven techniques for energy conservation in wireless sensor networks. Anastasi et al. [4] presents a systematic and comprehensive taxonomy of the energy conservation scheme in wireless sensor networks. Prediction-based data aggregation approaches are overviewed and classified into three types: stochastic approaches, time series forecasting, and algorithmic approaches.

Stochastic approaches exploit the probabilistic and statistical properties of sensed data. Deshpande et al. [5] propose a data prediction scheme based on a probabilistic model to reduce data transmission and reduce the quantity of data acquisition. A representative stochastic approach, named KEN [6], uses dynamic probabilistic model to minimize communication from the sensor node to the base station. The data aggregation process does not require communication between the sensor node and the base station except when the sensor node senses anomalous data. KEN naturally accommodates applications that are based on event reporting or anomaly detection. An extension of KEN is presented in [7], where a Dynamic Probabilistic Model (DPM) is exploited to implement a probabilistic database view. The main drawback of this class of techniques is that they inherently have relative high computational cost. To improve compression of the data communicated, some stochastic models exploit sophisticated spatial correlations of data in neighboring nodes. However, the more sophisticated the model, the more communications are required among sensor nodes themselves for coordination [2]. Therefore, possible improvements in this direction may focus on deriving simplified distributed models for obtaining the desired trade-off between the energy efficiency and the data accuracy according to users' requirements.

The most representative time series methods include Moving Average (MA), Auto-Regressive (AR) and Auto-Regressive Moving Average (ARMA) models. These models are quite simple, and can be used in many practical cases. Probabilistic Adaptable Query system (PAQ) [8] uses a combination of AR models to probabilistically answer queries. This model is used globally to predict the readings of individual sensors at the sink node, and locally to detect when sensor nodes produce outlier readings or when the model ceases to properly fit the data at a sensor node. The Similarity-based Adaptive Framework (SAF) [9] uses a simple linear time series model that consists of a time-varying function, also called trend component, and a stationary *AR* component representing the divergence of the phenomenon from the time-varying function over time. SAF can detect both outliers and inconsistent data. Le-Borgne et al. [10] propose an adaptive multi-model selection mechanism, which uses a lightweight, online algorithm that allows a sensor node to autonomously determine a satisfactory model from a set of candidate models. As sensed data are collected, based on a weight metric, it is possible to select the model that offers at each instant the highest achievable communication savings. Time series forecasting methods can provide sufficient accuracy, and their implementation in sensor devices is simple and lightweight. However, it is difficult to find an appropriate model that can tackle the long-term trend and short-term noise of data sequences simultaneously while providing a tunable trade-off between energy efficiency and data accuracy.

Algorithmic approaches aggregate data by exploiting the heuristic or behavioral characteristics of the sensing phenomena. PREMON [10] views a snapshot of the sensor network as an image – the readings of individual sensors corresponding to the intensity value of pixels in the image. Monitoring operations are considered as receiving a sequence of the snapshots on a continuous basis. When the sink node gets the initial reading from a sensor node, it computes the model by evaluating correlations between macro-blocks and deriving a motion vector relative to each block. After obtaining the model, the sensor node sends the model back to the sink node. From this time on, the sensor node compares each sample with the prediction derived from the model. When sensed data are close to the prediction within a user-specified tolerance, the sensor node does not transmit the data to the sink node. The model is periodically updated. Goel et al. [11] propose a buddy protocol to extend the PREMON approach by establishing a collaborative buddy relationship between sensor and sink nodes. It is suitable for cluster structured wireless sensor networks. By including a periodic polling scheme in cluster operations, the proposed buddy protocol can guarantee that each node in the network is reachable within the specified maximum delay constraints. Han et al. [12] present an Energy Efficient Data Collection (EEDC) mechanism for data prediction. EEDC is effective in active inquiry-based applications, in which each node associates an upper and a lower bound, whose difference represents the accuracy of the sensed data. These bounds are sent to the sink node, which stores them for each sensor node in the network. These bounds can be updated according to source-initiated and sink-initiated requests. However, the algorithmic techniques are too complex in computation and may also incur a great deal of communication overhead [2].

Compared to the above mentioned data aggregation methods, the data collection protocol and data aggregation approaches proposed in this paper have the following advantages. (1) They can provide high prediction accuracy without a large amount of training data and *a priori* knowledge of the distribution of sensed data, and eliminate more redundant transmissions. (2) They are more adaptive to dynamic changes in the distribution of sensed data. In addition, they are more scalable and structure-free, therefore, they can be used to couple with other route or topology-based data aggregation protocols. (3) They are relatively lightweight in terms of computational complexity to resource-constrained sensor nodes.

3. Prediction-based data collection protocol

In the application layer of a wireless sensor network, data collection can be classified into three schemes: *Pull*, *Push*, and *Integration of Pull and Push*. In the *Pull* scheme, the sensor node acquires data from physical layer and caches it locally. The cached data is collected only when the sensor node receives a query from the sink node. In this case, the sensor network looks like a database. In the *Push* scheme, the sensor node periodically senses data and immediately delivers it to the sink node. The sink node acts as a passive data collector. The *Integration* scheme provides capabilities of active data pushing and passive data acquisition by integrating the *Pull* scheme with the *Push* scheme.

In this paper, a prediction-based data collection protocol is proposed for the *Push* scheme. The proposed protocol is different from data collection protocols in the MAC layer, since it only focuses on the prediction-based cooperation between the sensor node and the sink node without taking into consideration network topology, node density, link quality and radio transceiver parameters. In general, the main challenges in designing a prediction-based data collection protocol include: (1) how to keep the data series at the sink node and the sensor node synchronous. In our approaches, both sensor node and sink node must use the same data series and the same prediction algorithm. However, the sensor has real sensed data while the sink node does not. The reason is that some sensed data have not been sent to the sink node since related successful predictions are done previously; (2) how to avoid cumulative error in continuous predictions. Since the data used for performing predictions may contain predicted value, cumulative error will inherently be produced; and (3) how to differentiate successful prediction and data loss when the sink node does not receive the sensed data. When the sensed data is out of threshold, it must be sent to the sink node. Nonetheless, the sink node may fail to receive the sensed data due to packet loss induced by unreliable communication. From the viewpoint of the sink node, this case is very similar to the successful prediction scenario.

To solve above problems, the proposed cooperative data collection protocol is presented in detail as follows.

Prerequisites:

- (1.1) Each sensor node's lifetime is divided into equal periods. A sensor node produces only one sensed data in one period.
- (1.2) Both the sink node and the sensor node use the same prediction algorithm. The sink node is assumed to have sufficient computing power, storage, and energy.
- (1.3) A reliable data delivery is defined as an end-to-end data intercommunication in which the receiver must send an acknowledgement message back to the sender.

Initialization:

- (2.1) The sink node broadcasts its acceptable prediction error threshold ε and cumulative error threshold θ to all sensor nodes according to the requirement of specific application by using reliable data deliveries. ε and θ are tunable parameters, pre-configured at the sink node. When their values are modified, the fresh ε and θ must be re-broadcast to all sensor nodes.
- (2.2) Each sensor node constructs two data queues, *actual data queue* (*ADQ*) and *predicted value queue in sensor end* (PVQ_{sensor}). *ADQ* stores actual data series and is used to control cumulative error. PVQ_{sensor} stores the data series that is used to do the same predictions in both the sensor node and the sink node. PVQ_{sensor} may contain predicted data value. This is called the *Double Queue Mechanism*. The length of *ADQ* and PVQ_{sensor} are equal and both are specified by the applied prediction algorithm (denoted as l). The sink node constructs a corresponding queue for each sensor node, called PVQ_{sink}, $PVQ_{sink}(i) = PVQ_{sensor}(i)$ for sensor node i.
- (2.3) Each sensor node stores the first l sensed data into its *ADQ* and PVQ_{sensor}, and sends them to the sink node to construct PVQ_{sink} via reliable delivery. Let x_j denote the data item in a queue. In the initial stage, $ADQ(i) = PVQ_{sensor}(i) = PVQ_{sink}(i) = \{x_1, x_2, \ldots, x_l\}$ for an arbitrary sensor node i.

Prediction:

- (3) Let x_{l+1}, x'_{l+1} and x''_{l+1} denote the actual sensed data, predicted value using *ADQ*, and predicted value using $PVQ_{sensor}(i)$, respectively. It is noticeable that the sink node can also obtain x'_{l+1} from the $PVQ_{sink}(i)$ queue. If $abs(x''_{l+1} - x_{l+1}) < \varepsilon$, the prediction error is considered as in threshold; otherwise out of threshold. If $abs(x''_{l+1} - x'_{l+1}) < \theta$, the cumulative error is considered as in threshold; otherwise out of threshold. When prediction error and cumulative error are in their thresholds simultaneously, the prediction of this period is considered successful. For a successful prediction, the sensor node does not need to send x_{l+1} to sink node. The sink node considers the predicted value x''_{l+1} as x_{l+1} in this period. After a successful prediction, the queues are updated by following rules: (a) $ADQ(i) = \{x_2, x_3, \ldots, x_{l+1}\}$; (b) $PVQ_{sensor}(i) = \{x_2, x_3, \ldots, x''_{l+1}\}$; and (c) $PVQ_{sink}(i) = \{x_2, x_3, \ldots, x''_{l+1}\}$.

Exceptions:

- (4.1) The actual sensed data x_{l+1} must be sent to the sink node using reliable delivery in the following cases: (a) a failed prediction occurs; and (b) the number of continuous successful predictions exceeds a pre-configured number.
- (4.2) After an exceptional data delivery, the queues are updated by following rules: (a) $ADQ(i) = \{x_2, x_3, \ldots, x_{l+1}\}$, (b) $PVQ_{sensor}(i) = \{x_2, x_3, \ldots, x_{l+1}\}$, and (c) $PVQ_{sink}(i) = \{x_2, x_3, \ldots, x_{l+1}\}$.

4. Grey model based data aggregation (GMDA)

A system is called a white system if all information about it is known, and a black system if no information about it is known. A grey system is intervenient between the white system and the black system, in which poor, incomplete, or uncertain data is provided [1]. The grey model provides a powerful tool for modeling discrete series with a few data items and for forecasting based on determination of an exponential pattern. A sensor node can be treated as an uncertain grey system in the data aggregation process, since only a small sample and poor information is stored and provided. In this paper, the single variable first-order grey model $GM(1,1)$ [1] is used to capture the long-term trend of the sensed data sequence by exploring and extracting valuable information from recently sensed data.

Before predicting, a few historical sensed data should be stored in the sensor node to construct the initial data sequence for $GM(1,1)$ model, denoted as $Y^{(0)}$.

$$Y^{(0)} = \left(y^{(0)}(1), y^{(0)}(2), \ldots, y^{(0)}(t)\right). \tag{1}$$

In Eq. (1), $y^{(0)}(j)$, $j = 1, 2, \ldots, t$, represents a data element. t denotes the number of elements in the sequence. t is an invariant, which represents the length of the data sequence. The $GM(1,1)$ model uses data of most recent t periods. To eliminate the influence of oscillation in the initial data sequence, the natural logarithm and the exponential function are used to get the adjusted sequence for $GM(1,1)$ model, as described in Eq. (2).

$$\left(\ln Y^{(0)}\right)^{1/M} = \left\{\left(\ln y^{(0)}(1)\right)^{1/M}, \left(\ln y^{(0)}(2)\right)^{1/M}, \ldots, \left(\ln y^{(0)}(t)\right)^{1/M}\right\}. \tag{2}$$

In Eq. (2), M is an integer invariant. In general, $1 < M < 10$. Let $x^{(0)}(j) = (\ln y^{(0)}(j))^{1/M}$ and $X^{(0)}$ denotes the prediction data sequence, as described in Eq. (3).

$$X^{(0)} = \left(x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(t)\right). \tag{3}$$

Let $X^{(1)}$ be the 1-AGO (*accumulated generating operator*) sequence of $X^{(0)}$, as described in Eq. (4).

$$X^{(1)} = \left(x^{(1)}(1), x^{(1)}(2), \ldots, x^{(1)}(t)\right). \tag{4}$$

Therefore, the $GM(1,1)$ model can be established as Eq. (5) (a differential equation).

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = b. \tag{5}$$

Let $A = \begin{pmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \cdots \\ x^{(0)}(t) \end{pmatrix}$ and $B = \begin{pmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \cdots & \cdots \\ -z^{(1)}(t) & 1 \end{pmatrix}$, where $z^{(1)}(k) = \frac{1}{2}\left(x^{(1)}(k) + x^{(1)}(k-1)\right)$ when $k = 2, \ldots, t$. Therefore, $[\hat{a}, \hat{b}]^T = (B^T B)^{-1} B^T A$. Using the *Least Squares Method*, the values of the parameters a and b can be obtained. Therefore, $\hat{x}^{(1)}(k+1)$ can be obtained by using Eq. (6).

$$\hat{x}^{(1)}(k+1) = e^{-ak}\left(x^{(1)}(1) - \frac{\hat{b}}{\hat{a}}\right) + \frac{\hat{b}}{\hat{a}}. \tag{6}$$

Therefore, the predicted data $\hat{x}^{(1)}(k+1)$ of the data sequence $X^{(0)}$ can be computed by Eq. (7).

$$\hat{x}^{(0)}(k+1) = \hat{x}^{(1)}(k+1) - \hat{x}^{(1)}(k). \tag{7}$$

Finally, the final predicted data $\hat{y}^{(0)}(t+1)$ can be obtained by Eq. (8).

$$\hat{y}^{(0)}(t+1) = \left(e^{\hat{x}^{(0)}(t+1)}\right)^M. \tag{8}$$

Let $\Delta(t+1) = |\hat{y}^{(0)}(t+1) - y^{(0)}(t+1)|$ and ε represent the prediction error and the threshold of the prediction error, respectively. For simplicity, cumulative error is not taken into consideration here. After obtaining the predicted data and the prediction error, the sensor node compares the error with ε. If $\Delta(t+1) < \varepsilon$, the sensor node does not need to transmit $y^{(0)}(t+1)$ to the sink node. Otherwise, it must send $y^{(0)}(t+1)$ to the sink node. At the other end, the sink node runs the same prediction program with the same prediction data sequence. Therefore, it obtains the same predicted data as the sensor node predicted. However, the sink node can not compute the prediction error since it does not have $y^{(0)}(t+1)$. If there is no data coming from the sensor node in a fixed time T_0, the sink node sets $\Delta(t+1) < \varepsilon$ and considers $\hat{y}^{(0)}(t+1)$ as $y^{(0)}(t+1)$ in the current sensing period. T_0 should be longer than the maximum transmission latency, but shorter than the length of a sensing period. It is important to synchronize the prediction data sequences, PVQ_{sensor} and PVQ_{sink}. When $\Delta(t+1) \geqslant \varepsilon$, the sensor node must use the predicted data $\hat{y}^{(0)}(t+1)$ as the data of $(t+1)$th period in its next prediction sequence, because the sink node does have $y^{(0)}(t+1)$.

5. Kalman-Filter-based Data Aggregation (KFDA)

The Kalman Filter [13] is an efficient recursive filter that estimates the state of a linear dynamic system from a series of noisy measurements. It presents high prediction accuracy based on a small quantity of information. It has been used to design adaptive routing mechanisms in mobile wireless sensor networks [14,15]. Olfati-Saber [17] proposes a peer-to-peer continuous-time distributed Kalman Filter that uses local aggregation of the sensor data but attempts to reach a consensus on estimates with other nodes in the network. Yu et al. [18] also design a distributed consensus filter, in which each sensor can communicate with the neighboring sensors, and filtering can be distributed among nodes. By using a pinning control scheme, only a small fraction of sensors need to measure the target information. In this paper, the Kalman Filter is used to estimate the data sequence for each sensor node rather than to choose sensor nodes.

5.1. Kalman-Filter-based prediction model

In a sensor node, continuous data forms a discrete time data sequence, which can be modeled by the following *Linear Stochastic Difference equation*:

$$X(k) = A(k)X(k-1) + B(k)U(k) + W(k). \tag{9}$$

$X(k)$ represents the predicted data at the period k. $A(k)$ represents the state transition model which is applied to the data of the previous period $(k-1)$. $B(k)$ represents the control-input model applied to the control vector $U(k)$. $W(k)$ represents the noise of the prediction period, which is assumed to follow a zero mean multivariate normal distribution with the covariance $Q(k)$. Let $Z(k)$ denote the actual sensed data sequence at the period k:

$$Z(k) = H(k)X(k) + V(k), \tag{10}$$

$H(k)$ is the observation model which maps the predicted data space into the actual sensed data sequence. $V(k)$ is the noise which is assumed to be *Zero Mean Gaussian white Noise* with covariance $V(k)$.

5.2. Kalman-Filter-based prediction algorithm

The Kalman Filter has two distinct phases: prediction and update. The prediction phase uses the data estimated from the previous sensing period to produce an estimation of the data at the current period. The prediction model and its covariance model are illustrated as Eqs. (11) and (12), respectively. In the update phase, measurement information at the current period is used to refine this prediction to achieve a new, more accurate data estimate, again for the current period. The updated prediction model and its covariance model are illustrated as Eqs. (13) and (14), respectively. And the *Optimal Kalman Gain* can be computed using Eq. (15).

$$\widehat{X}(k+1|k) = A(k)X(k|k) + B(k)U(k), \tag{11}$$

$$P(k+1|k) = A(k)P(k|k)A(k)^T + Q(k), \tag{12}$$

$$\widehat{X}(k+1|k+1) = \widehat{X}(k+1|k) + Kg(k+1)(Y(k) - H(k+1)\widehat{X}(k+1|k)), \tag{13}$$

$$P(k+1|k+1) = (I - Kg(k+1)H(k+1))P(k+1|k), \tag{14}$$

$$Kg(k+1) = P(k+1|k)H(k)^T[H(k+1)P(k+1|k)H(k+1)^T + R(k)]^{-1}. \tag{15}$$

$\widehat{X}(n|m)$ represents the estimate of X at period n, given sensed data sequence of recent m periods. $P(n|m)$ represents the error covariance matrix according to $\widehat{X}(n|m)$.

In this section, the Kalman Filter is viewed as a single measurement of a single model for temperature prediction. To simplify the computation, we let $A(k) = 1$, $B(K) = Q(k) = 0$, $R(k) = H(k) = I$, and $P(0|0) = 1$. Let $Y(k) = \{y(1), y(2), \ldots, y(k)\}$ represent the historical sensed data sequence.

As a single-prediction-method-based approach, the KFDA is similar to the GMDA. The only difference is that they use different prediction algorithms.

6. CoGKDA

6.1. Modeling

In this section, GMDA and KFDA are combined to improve the accuracy of the prediction. Since the grey model is very effective for predicting data series with secular trends [1] and the Kalman Filter is useful for improving the prediction accuracy on the data series that may oscillate frequently in a short-term, the combination of the grey model and the Kalman Filter reduces randomness and improves prediction accuracy simultaneously. The main idea of CoGKDA is to combine the two independent prediction technologies to avoid potential deficiencies that a single prediction technology may lead to. In the combination, weights are used to leverage the two prediction approaches. The optimal weights can be computed by minimizing the sum of prediction error squares of the meta approaches.

Without loss of generality, let $W = (w_1, w_2, \ldots, w_m)^T$ represent the weight vector of m prediction models, $\sum_{i=1}^{m} w_i = 1$. Let $Y(t)$ represent the actual sensed data sequence of the recent t periods and $\widehat{Y}_j(t)$ ($t = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$) represent the corresponding predicted data sequences of the m prediction models. The predicted data sequence of the combined approach is $\widehat{Y}(t)$, $t = 1, 2, \ldots, n$. Therefore, the sum of prediction error squares can be denoted as $J(t)$ in Eq. (16).

$$J(t) = \sum_{j=1}^{m} w_j \left[(f(\widehat{Y}(t)))^p - (g(\widehat{Y}_j(t)))^p \right]^2, \tag{16}$$

where $P \neq 0$, f and g are continuous differentiable functions. To compute the extremum of W, we let $\frac{\partial J(t)}{\partial \widehat{Y}(t)} = 0$. The combined prediction model is defined as Eq. (17).

$$\widehat{Y}(t) = f^{-1}\left[\left(\sum_{j=1}^{m} w_j (g(\widehat{Y}_j))^p \right)^{1/p} \right]. \tag{17}$$

In this paper, with the consideration of constrained computing resources, we let $f(\widehat{Y}(t)) = \widehat{Y}(t)$, $g(\widehat{Y}_j(t)) = \widehat{Y}_j(t)$, and $p = 1$ to simplify the model. As a result, the combined prediction model changes to the weighted arithmetic average of the meta prediction models.

6.2. Combination weights

For the ith prediction approach, its predicted data sequence is $\hat{y}_i = (\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{in})$, $i = 1, 2, \ldots, m$. Let e_{ij} denote the error of the ith approach in predicting the jth datum, e_i denote the error vector of the ith approach in Eq. (18).

$$e_i = (e_{i1}, e_{i2}, \ldots, e_{in}) = (y_1 - \hat{y}_{i1}, y_2 - \hat{y}_{i2}, \ldots, y_n - \hat{y}_{in}). \tag{18}$$

Using the weight vector W, the combined predicted data sequence can be described as Eq. (19).

$$\widehat{Y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n) = \left(\sum_{i=1}^{m} w_i \hat{y}_{i1}, \sum_{i=1}^{m} w_i \hat{y}_{i2}, \ldots, \sum_{i=1}^{m} w_i \hat{y}_{in} \right). \tag{19}$$

Therefore, the error metrics of the combined prediction $\{e_{ij}\}(e_{ij} = y_j - \sum_{i=1}^{m} w_i \hat{y}_{ij})$ and the sum of squares of the combined prediction errors is $J(t)$:

$$J(t) = \sum_{j=1}^{n} \left(y_j - \sum_{i=1}^{m} w_i \hat{y}_{ij} \right). \tag{20}$$

Using the *Least Squares Method* to minimize $J(t)$, the optimal weight vector can be obtained. $W = \frac{A^{-1}R^T}{RA^{-1}R^T}$, where $R = (1, 1, \ldots, 1)^T$ and

$$A = \begin{pmatrix} \sum_{i=1}^{n} e_{1i}^2 & \sum_{i=1}^{n} e_{1i}e_{2i} & \cdots & \sum_{i=1}^{n} e_{1i}e_{ni} \\ \sum_{i=1}^{n} e_{2i}e_{1i} & \sum_{i=1}^{n} e_{2i}^2 & \cdots & \sum_{i=1}^{n} e_{2i}e_{ni} \\ \cdots & \cdots & \cdots & \cdots \\ \sum_{i=1}^{n} e_{ni}e_{1i} & \sum_{i=1}^{n} e_{ni}e_{2i} & \cdots & \sum_{i=1}^{n} e_{ni}^2 \end{pmatrix}.$$

As described in Eq. (18), the combined prediction in CoGKDA needs the actual sensed data sequence to compute an optimal weight vector. Since the sink node does not have the actual sensed data sequence, it can not compute the optimal weight vector. In addition, computing an optimal weight vector in every period is too expensive for a sensor node. In this paper, an empirical and periodical update mechanism is proposed to solve this problem. Sensor nodes compute and send their optimal weight vector to the sink node in their initial prediction period. After a fixed number of periods (denoted as u), sensor nodes periodically re-compute and send the fresh weight vector to the sink node to synchronize their prediction parameters.

6.3. CoGKDA algorithm

The predictions should be based on the same data sequence (*PVQ*) at both the sensor node and the sink node. The global thresholds of the prediction error and cumulative error are set before the predictions, denoted as ε and θ, respectively. According to the proposed data collection protocol, we let the length of data queues $l = t$. For all sensor nodes, the data of the first t sensing periods must be transmitted to the sink node. Starting with the $(t+1)$th period, the sensor node and the sink node do predictions. For example, let $y(t+1), \hat{y}(t+1)$, and $\hat{y}'(t+1)$ represent the currently sensed data, the data predicted by using the prediction value queue in the sensor node (PVQ_{sensor}), and the data predicted by using the actual data queue (ADQ), respectively. In the $(t+1)$th period, $\Delta(t+1) = abs(\hat{y}(t+1) - y(t+1))$ is the prediction error and $\Delta'(t+1) = abs(\hat{y}'(t+1) - \hat{y}(t+1))$ is the cumulative error. The sensor node checks the prediction error with the pre-configured global thresholds. If $\Delta(t+1) < \varepsilon$ and $\Delta'(t+1) < \theta$, the sensor node does not send the actual $y(t+1)$ to the sink node. The sink node considers $\hat{y}(t+1)$ as $y(t+1)$. The sensor node must set $y(t+1) = \hat{y}(t+1)$ to keep the prediction data sequence PVQ_{sensor} synchronized with the sink node's PVQ_{sink} for future predictions. Otherwise, the sensor node sends $y(t+1)$ to the sink node. The CoGKDA algorithm in the sensor node is described in Table 1.

According to the protocol described in Section 3, when continuous and successful predictions are made in a sensor node, the sink node will not receive any data from the sensor node for a long time. In this case, the sensor node is very similar to a dysfunctional (or failed) sensor node. In addition, as the number of continuous and successful predictions increases, the cumulative error will increase correspondingly. To distinguish the two different cases and avoid excessive cumulative error, we use another threshold v for the number of continuous and successful predictions. When the number of continuous and successful predictions is out of v, the sensor node must send the actual data to the sink node. Therefore, when the sink node receives actual data from a sensor node in v periods, it knows the sensor node is functioning. Otherwise, the sensor node will be temporarily treated as a dysfunctional node since it has not sent back data in the pre-configured time. In CoGKDA, v is a pre-configured parameter, which should be determined by the trade-off between reducing concurrent error and increasing communication overhead. It is reasonable to deduce that as v decreases, CoGKDA decreases concurrent error but increases communication overhead.

Table 1
The CoGKDA algorithm.

```
Input:
  $Y(i)$: current prediction data sequence,   $Y(i) = (\hat{y}_{i-t+1}, \hat{y}_{i-t+2}, \ldots, \hat{y}_i),\ i \geqslant t$;
  $W$: static variable, the current optimal weight vector;
  $y_{i+1}$: the sensed data of the $(i+1)$ th period;
  $r$: static variable, the number of the continuous and successful predictions;
  $u$: static variable, the number of periods for re-computing weight vectors;
  $v$: static variable, a threshold for variable $r$. If $r \geqslant v$, sensor must send
    currently sensed data to sink node;
  $s$: static variable, the age of the current weight vector;
  $\varepsilon$: static variable, the threshold of prediction error;
  $\theta$: static variable, the threshold of cumulative error;
Output:
  $Y(i+1)$: next prediction data sequence;
CoGKDA ($Y(i), W, y_{i+1}, r, u, v, s, \varepsilon, \theta$)
{
  Perform GMDA prediction and obtain the predicted data $\hat{y}_g$ and its error $e_g$;
  Perform KFDA prediction and obtain the predicted data $\hat{y}_k$ and its error $e_k$;
  if $s < u - 1${
    $s = s + 1$;
  Perform the combination and obtain the predicted data $\hat{y}_c$, prediction error
    $pe_c$,
  and cumulative error $ce_c$;
  }
  else{
    Compute new optimal weight vector $W_{new}$;
    Send the new weight vector to the sink node;
    $W = W_{new}$;
    $s = 0$;
  }
  if $pe_c < \varepsilon$ and $ce_c < \theta$ and $r < v - 1${
    $r = r + 1$;
    $\hat{y}_{i+1} = \hat{y}_c$;//Synchronize the next period prediction data sequence with the
      sink node.
    $Y(i+1) = (\hat{y}_{i-t+2}, \hat{y}_{i-t+3}, \ldots, \hat{y}_{i+1})$; //Refresh the prediction data sequence
      for future predictions.
  }
  else{
    Send $y_{i+1}$ to the sink node;
    $r = 0$;
    $Y(i+1) = (\hat{y}_{i-t+2}, \hat{y}_{i-t+3}, \ldots, \hat{y}_i, y_{i+1})$;
  }
  return $Y(i+1)$;
}
```

7. Experiment and performance evaluation

7.1. Experiment setup

In this paper, experiments are based on an environmental monitoring system in a granary. Since grain is liable to mildew when the humidity and temperature in the storehouse are too high, it is very important to monitor real-time humidity and temperature. The data used in our experiments are derived from a real deployed sensor network. The sensor network is used to collect the temperature and humidity of the grain in a large granary, which consists of 30 storehouses. Each storehouse is a detached building, which is divided into 24 volumes. The grain is stored in the volumes. In each volume, there are four sensor nodes buried in the grain. The sensor field of a volume is divided into four zones: top, middle-top, middle-bottom, and bottom. All sensor nodes in a storehouse form a tree-structured network with three layers: sensor layer, intermediate layer and sink layer. Nodes in the intermediate layer and the sink layer are external powered, while nodes in the sensor layer are only powered by battery. Each intermediate node receives data from four sensor nodes in one volume and sends them to the sink node (intermediate nodes just relay data between sensor and sink nodes). One sink node is deployed in one storehouse to collect data from intermediate nodes. Sensor nodes sense and return temperature and humidity data every thirty minutes. This system has worked for three years and has collected a large volume of data.

To evaluate the proposed data aggregation approaches, these approaches are implemented in our test bed system, in which all sensor nodes are designed based on TinyOS 2 and the IEEE 802.15.4 protocol. All experiments are carried out based on a real data set. Since these prediction-based data aggregation approaches are structure-free and topology-free, a sensor node was randomly chosen for the experiments and only its temperature data is used. A data sequence (denoted as D) that includes 720 continuous data items (data for half a month) was randomly extracted from the original temperature data stream for the following experiments.

In the proposed approaches, ε and θ represent users' requirements on data accuracy. They are application-specific parameters. In the temperature monitoring of a granary, the tolerated error of the predicted data is relatively small, since the grain is sensitive to temperature changes. Therefore, in our experiments, we let $\varepsilon = 0.5$ and $\varepsilon = 1$ represent users' high and low on data accuracy requirements, respectively. For simplicity, we let $\varepsilon = \theta$.

7.2. GMDA

Compared to other prediction-based approaches, GMDA is very lightweight. The predicted data sequence of GMDA can be represented as $Y^{(0)} = (y^{(0)}(1), y^{(0)}(2), \ldots, y^{(0)}(t))$, where parameter t denotes the length of the sequence used for predictions. In general, longer sequences lead to more accurate predictions. However, greater length consumes more sensor storage and leads to higher computational complexity. To choose a suitable t value, we evaluate the growth rate of prediction accuracy while t changes from 3 to 9. The results are shown in Fig. 1. First, we randomly extracted three sub-sequences from the original data set. Each sub-sequence

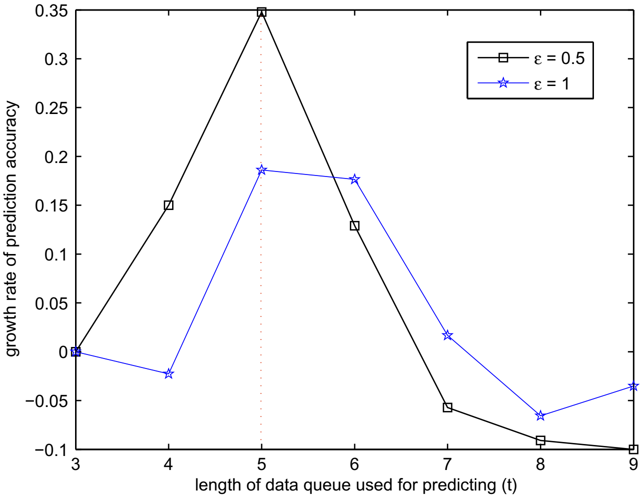

Fig. 1. Evaluation of parameter t in GMDA.

consists of thirty continuous data items. Second, using each t value, we performed GMDA on the three sub-sequences, respectively, and then we obtained the average prediction accuracy of each t value. Finally, we computed the growth rate of prediction accuracy while t increases. The prediction accuracy is measured by $\varepsilon = 1$ and $\varepsilon = 0.5$, respectively. Since the growth rate of prediction accuracy achieves its maximum value when $t = 5$, as shown in Fig. 1, we choose $t = 5$ in following $GM(1,1)$ algorithm experiments.

The experiments for GMDA are carried out based on the data sequence D. The results are shown in Figs. 2 and 3. As described in Section 3, the sensor node does not need to send actual data to the sink node when the error of the related prediction is within the threshold ε, thereby reducing transmissions and power consumption. In Fig. 2, the cumulative distribution function (CDF) of the prediction errors produced by GMDA is 25.22%. When we let $\varepsilon = 1$, the CDF value of GMDA's prediction errors achieves 43.77%.

Since GMDA causes no overhead communication, the CDF value of its prediction errors on the corresponding threshold is approximately equal to the percentage of communication energy saving, as shown in Table 2. As illustrated in Figs. 2 and 3, when the threshold of the prediction error decreases, energy consumption increases. The reason is that a smaller threshold causes a lower prediction success rate. In other words, more failed predictions cause more actual data communication.

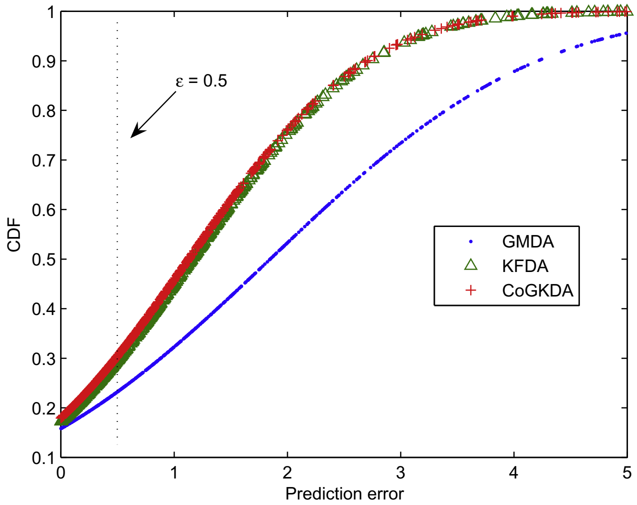

Fig. 2. Cumulative distribution functions of the prediction errors of GMDA, KFDA and CoGKDA when $\varepsilon = 0.5$.

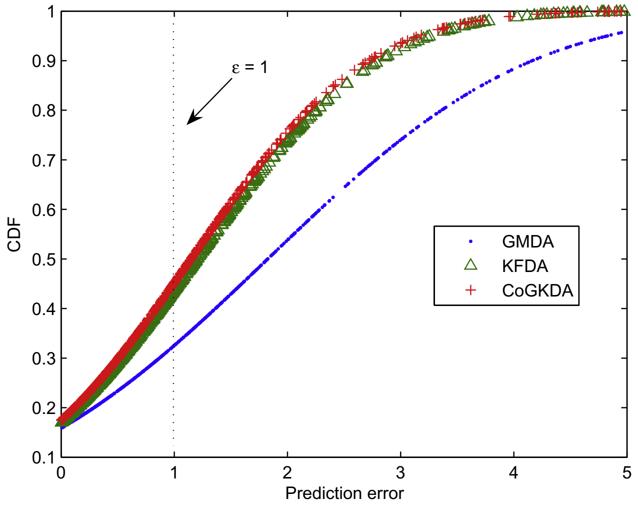

Fig. 3. Cumulative distribution functions of the prediction errors of GMDA, KFDA and CoGKDA when $\varepsilon = 1$.

Table 2
Performance comparison between GMDA, KFDA, CoGKDA and Auto-Regressive method when the threshold $\varepsilon = 1$ and $\varepsilon = 0.5$, where ts denotes the number of data of the training set in Auto-Regressive method.

Approaches	Communication energy savings
GMDA: $t = 5$, $\varepsilon = 0.5$	25.22%
GMDA: $t = 5$, $\varepsilon = 1$	43.77%
KFDA: $\varepsilon = 0.5$	33.62%
KFDA: $\varepsilon = 1$	58.41%
Auto-Regressive method: $ts = 60$, $\varepsilon = 0.5$	27.35%
Auto-Regressive method: $ts = 60$, $\varepsilon = 1$	40.62%
CoGKDA: $\varepsilon = 0.5$	35.21%
CoGKDA: $\varepsilon = 1$	59.85%

7.3. KFDA

Using the same data sequence D for GMDA, the experiments on KFDA were carried out. The results show that KFDA obtains a higher prediction success rate than GMDA, as illustrated in Figs. 2 and 3. The CDF value of the prediction errors produced by KFDA is much higher than GMDA's CDF value for both $\varepsilon = 0.5$ and $\varepsilon = 1$. According to the data collection protocol described in Section 3, in the experiments on KFDA, no communication overhead is produced. Table 2 shows that the percents of communication energy savings are 33.62% and 58.41% when $\varepsilon = 0.5$ and $\varepsilon = 1$, respectively.

For data sequence D, KFDA behaves better than GMDA and better conserves power. Nonetheless, by comparing their prediction models (as described in Sections 6.1 and 6.2), KFDA requires more computation than GMDA in the process of data collection.

7.4. CoGKDA

To simplify the computation, we let $f(\widehat{Y}(t)) = \widehat{Y}(t)$, $g(\widehat{Y}_j(t)) = \widehat{Y}_j(t)$ and $p = 1$ in Eq. (16). Therefore, the CoGKDA model changes to a simple weighted arithmetic average of the two meta prediction models. The pivotal problem in CoGKDA is that the sensor node must compute optimal weights periodically and keep its weight vector synchronized with sensor nodes in predictions. The parameter u is used to control the interval of weight refresh. According to the proposed data collection protocol in Section 3,

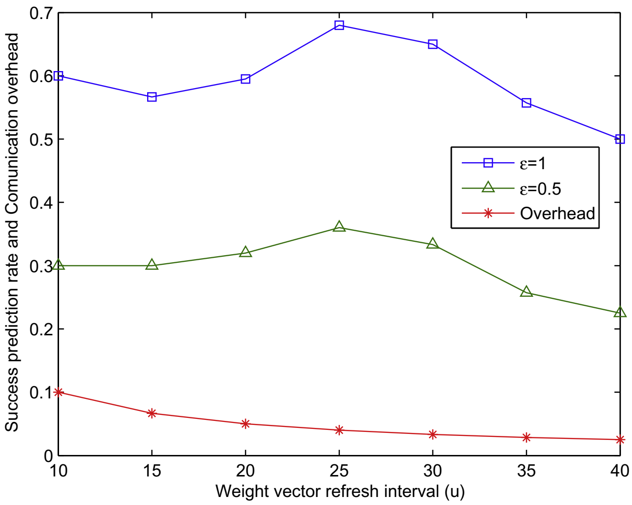

Fig. 4. Success prediction rate and communication overhead as the weight vector update interval (u) changes from 10 to 40.

the smaller u is, the higher the corresponding successful prediction rate will be and the more communication overhead CoGKDA will produce. To find a suitable u value, we carried out a simple experiment to evaluate the successful prediction rate and communication overhead while increasing u from 10 to 40 stepped by 5. In this experiment, we used the first $3u$ data items in D for predictions and computed prediction success rate and communication overhead for each u value. As illustrated in Fig. 4, the result shows $u = 25$ is the best choice for both $\varepsilon = 1$ and $\varepsilon = 0.5$. In CoGKDA, another parameter v is used to reduce long concurrent errors. To find a suitable v value, we carried out another simple experiment, in which we computed the percentage of overhead in communication energy consumption while changing v from 2 to 16. The result in Fig. 5 shows that communication overhead is 0 when ($\varepsilon = 1$, $v = 16$) and ($\varepsilon = 0.5$, $v = 8$). It is noticeable that the percentage of overhead in communication energy is less than 1% for both $\varepsilon = 1$ and $\varepsilon = 0.5$ when $v = 10$ (The percentage is approximate 0.87% when $\varepsilon = 1$, and 0 when $\varepsilon = 0.5$).

Based on above evaluations, we let $t = 5$, $v = 8$ and $u = 10$. Using the same data sequence D, we carried out experiments on CoGKDA. As illustrated in Figs. 2, 3 and Table 2, the results show that CoGKDA obtained a higher prediction success rate than GMDA and KFDA. The CDF value of the prediction errors produced by CoGDA is higher than those of KFDA and GMDA for both $\varepsilon = 0.5$ and $\varepsilon = 1$. From Figs. 2 and 3, it can be observed that performance of CoGDA is better than that of KFDA when the tolerable prediction error is in the low interval (range from 0 to 3), and close to that of KFDA when the tolerable prediction error is in the high interval (range from 3 to 5). Table 2 also shows that CoGKDA achieves energy savings of 35.21% and 59.85% while $\varepsilon = 0.5$ and $\varepsilon = 1$, respectively. It demonstrates that by combining KFDA and GMDA, CoGKDA improves communication energy saving with an insignificant increase in overhead.

State-of-the-art prediction-based approaches such as PAQ [8] and SAF [9] use Auto-Regressive or improved Auto-Regressive methods to aggregate data in wireless sensor networks. In SAF and PAQ, prediction models are built at each sensor to predict local readings. Sensor nodes transmit their local models to a sink node, which uses them to predict sensor values without need for downlinking communicating with sensor nodes. When needed, sensor nodes send information about outlier readings and model updates to the sink node. To compare CoGKDA with PAQ and SAF, we carried out the same data collection experiments on them by using the same data sequence D. Since PAQ, SAF and AR need a training set before prediction, we let their training set contain 60 data items (the first 60 data items in D). In this experiment, a third order AR algorithm was used as a benchmark. The experiment results are shown in Figs. 6 and 7. Fig. 6 shows that CoGKDA presents a higher success rate and more energy saving than the third order Auto-Regressive model $AR(3)$, PAQ, and SAF. To compare data accuracy of successful predictions, we analyze the mean square errors of successful predictions of these approaches. Fig. 7 shows that CoGKDA is also better than SAF and PAQ as the threshold ε changes from 0.2 to 1. Since SAF and PAQ do not have a cooperation mechanism to synchronize the data series used for prediction, they produce a lower communication overhead. However, as illustrated in Figs. 6 and 7, the prediction errors and mean square errors of SAF and PAQ are much higher than that of CoGKDA. The reason is that, in SAF and PAQ, the sink node and the sensor node use different data series to predict, while different data series produce different prediction errors which cause cumulative error increases.

It is remarkable that: (1) CoGKDA provides high successful prediction rate by using the adjusted predicted data sequence PVQ, rather than the actual data sequence ADQ; and (2) CoGKDA reduces energy consumption caused by redundant communications with insignificant overhead.

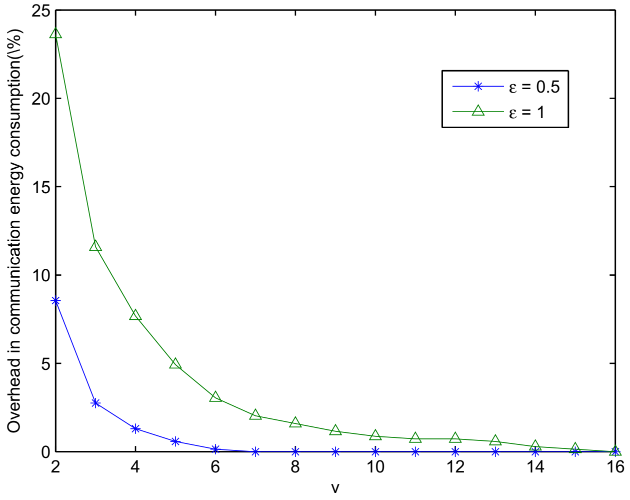

Fig. 5. Communication overhead as v changes from 2 to 16.

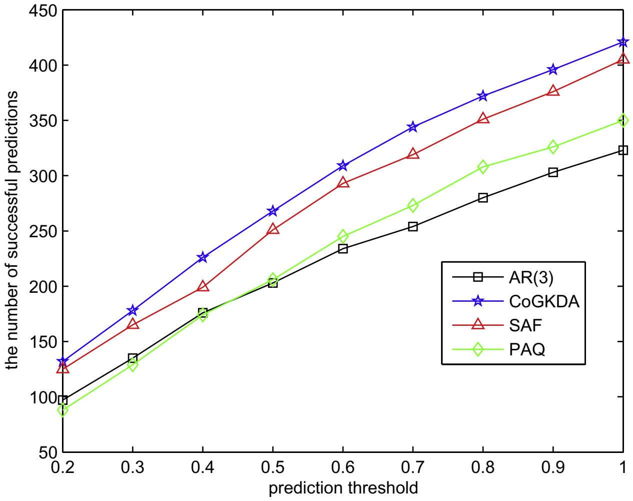

Fig. 6. The comparison of the success rates as ε changes from 0.2 to 1.

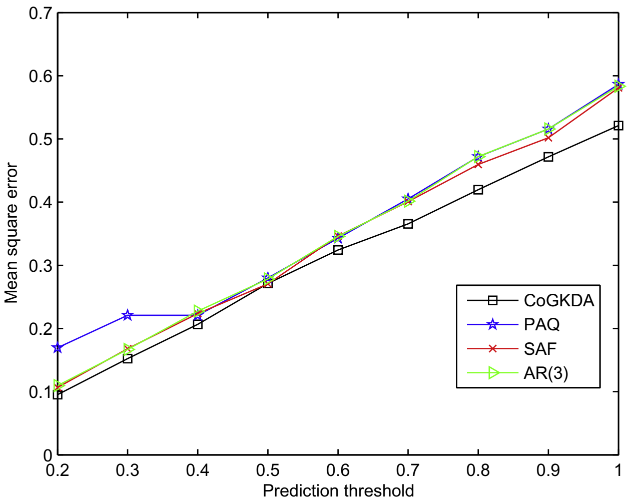

Fig. 7. The comparison of the mean square errors of all successful predictions as ε changes from 0.2 to 1.

7.5. Complexity and scalability

Main computation in GMDA derives from the $GM(1,1)$ algorithm. According to Eqs. (1)–(8), it can be deduced that the computational complexity of GMDA is $O(t)$, where t is the number of data items for the prediction algorithm. For most applications, the computational complexity of GMDA is very low when $t = 5$.

In the KFDA approach, as described in [16], the order of the Kalman Filter is $O(2m^2n) + O(2mn^2) + O(m^3) + O(n^3)$. In this paper, m is the number of dimensions of data prediction sequence at the prediction phase, and n is the number of predicted data items for one recursion. If $n = am$, where $a > 1$, then the number of computations can be transformed to $O[(1 + 2a + 2a^2 + a^3)m^3]$. As described in Eqs. (9)–(15), the computational complexity becomes $O(3m^3)$, when using the equivalent system with $A(k) = 1$, $B(k) = Q(k) = 0$, $R(k) = H(k) = I$ and $P(0|0) = 1$. Therefore, the computational complexity of KFDA is $O\left(\frac{3}{1+2a+2a^2+a^3}\right)$ times the normal KF algorithm. For example, let $a = 2$, the computational complexity of KFDA is only 0.14 times the normal KF algorithm. By analyzing the CoGKDA algorithm in Table 1, the computation of CoGKDA approach mainly consists of three parts: GMDA, KFDA and computing weight vector W. As described in Eqs. (16)–(20), the complexity for computing W is $O(mn^2)$, in which m is the number of meta predictions model and n is the number of data items in the prediction data sequence. In CoGKDA, the computational complexity for computing W is $O(2n^2)$. To unify measurements, we let m represent the number of data items in prediction data sequence. Therefore, adding the three parts, the computational complexity of CoGKDA is $O(m) + O(3m^3) + O(6m^2)$, and the order of CoGKDA's computation complexity is approximately $O(m^3)$.

It can be seen that CoGKDA is the most complex, KFDA is simpler, and GMDA is the simplest. Although the computational complexity of CoGKDA seems high, in practice, it is acceptable for most applications when the length of data queues (ADQ and PVQ) is small, e.g. $m = 5$.

In the proposed approaches, all computations for data aggregation are only performed in the sensor node and the sink node. Data processing in intermediate nodes is not needed. Therefore, the proposed approaches are independent of network scale. Their performance is only determined by user's requirements which are jointly controlled by parameters ε, θ, t, u and v. Furthermore, the proposed approaches can be used in tree-structured, cluster-structured and peer-structured wireless sensor networks, since the data collection protocol they used is structure-free. Therefore, they can be used to couple with other route-based or topology-based data aggregation protocols. Above analysis indicates the proposed approaches are scalable for most environmental monitoring applications.

8. Conclusion

Prediction-based data aggregation is a fundamental data-driven energy conservation approach. The prediction-based approach saves energy by reducing redundant data communications. Since the prediction-based approach is structure-free, it can be used to couple with other route or topology-based data aggregation approaches. By analyzing energy efficiency and data accuracy, a novel prediction-based data collection protocol is proposed to specify the cooperations between the sensor node and the sink node. In the proposed protocol, a double-queue mechanism is designed to synchronize predicted data at both sensor node and sink node to avoid cumulative error in continuous predictions. Based on this novel data collection protocol, three prediction-based data aggregation approaches are proposed: GMDA, KFDA and CoGKDA. Experiments have been carried out based on a real data set collected from a temperature and humidity monitoring application in a grain repertory. The results demonstrate that the proposed approaches can reduce energy consumption caused by redundant communications with minimally increased overhead. Experiments also show CoGKDA achieves better performance compared to traditional prediction-based approaches (including SAF, PAQ and AR).

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant No. 60803161.

References

[1] J.L. Deng, Introduction to Grey system theory, Journal of Grey System 1 (1) (1989) 1–24.
[2] R. Rajagopalan, P.K. Varshney, Data aggregation techniques in sensor networks: a survey, IEEE Commununications Surveys Tutorials 8 (4) (2006) 48–63.
[3] S. Ozdemir, Y. Xiao, Secure data aggregation in wireless sensor networks: a comprehensive overview, Computer Networks 53 (2009) 2022–2037.
[4] G. Anastasi, M. Conti, M.D. Francesco, A. Passarella, Energy conservation in wireless sensor networks: a survey, Ad Hoc Networks 7 (2009) 537–568.
[5] A. Deshpande, C. Guestrin, S. Madden, J.M. Hellerstein, W. Hong, Model-driven data acquisition in sensor networks, in: Proceedings of the 30th International Conference on Very Large Data Bases, 2004, pp. 588–599.
[6] D. Chu, A. Deshpande, J.M. Hellerstein, W. Hong, Approximate data collection in sensor networks using probabilistic models, in: Proceedings of the 22nd International Conference on Data Engineering 2006, pp. 48–59.
[7] B. Kanagal, A. Deshpande, Online filtering, smoothing and probabilistic modeling of streaming data, in: Proceedings of the 24th International Conference on Data Engineering 2008, pp. 1160–1169.
[8] D. Tulone, S. Madden, PAQ: time series forecasting for approximate query answering in sensor networks, in: Proceedings of the Third European Conference on Wireless Sensor Networks, 2006, pp. 21–37.
[9] D. Tulone, S. Madden, An energy-efficient querying framework in sensor networks for detecting node similarities, in: Proceedings of the Ninth International ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2006, pp. 291–300.
[10] Y. Le-Borgne, S. Santini, G. Bontempi, Adaptive model selection for time series prediction in wireless sensor networks, Signal Process 87 (12) (2007) 3010–3020.
[11] S. Goel, A. Passarella, T. Imielinski, Using buddies to live longer in a boring world, in: Proceedings of 2006 IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing, 2006, pp. 342–346.
[12] Q. Han, S. Mehrotra, N. Venkatasubramanian, Energy efficient data collection in distributed sensor environments, in: Proceedings of the 24th IEEE International Conference on Distributed Computing Systems, 2004, pp. 590–597.
[13] R.E. Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME Journal of Basic Engineering 82 (Series D) (1960) 35–45.
[14] B. Pasztor, M. Musolesi, C. Mascolo, Opportunistic mobile sensor data collection with SCAR, in: Proceedings of MASS 2007, 2007, pp. 1–12.

[15] M. Musolesi, S. Hailes, C. Mascolo, Adaptive routing for intermittently connected mobile ad hoc networks, in: Proceedings of 2005 IEEE WoWMoM, 2005, pp. 183–189.

[16] M.J. Goris, D.A. Gray, I.M.Y. Mareels, Reducing the computational load of a Kalman filter, Electronics Letters 33 (18) (1997) 1539–1541.

[17] R. Olfati-Saber, Distributed Kalman filtering for sensor networks, in: The 46th IEEE Conference on Decision and Control, 2007, pp. 5492–5498.

[18] W. Yu, G. Chen, Z. Wang, W. Yang, Distributed consensus filtering in sensor networks, IEEE Transactions on Systems, Man, and CyberneticsłPart B: Cybernetics 39 (6) (2009) 1568–1577.

[19] R. Kay, F. Mattern, The design space of wireless sensor networks, IEEE Wireless Communications 11 (6) (2004) 54–61.

[20] Y. Yao, B.B. Giannakis, Energy-efficient scheduling for wireless sensor networks, IEEE Transactions on Communications 53 (8) (2005) 54–61.

[21] Q. Xin, L. Gasieniec, C. Su, P. Wong, Routing via single-source and multiple-source queries in static sensor networks, in: Proceedings of IPDPS 2005, 2005, pp. 183–189.