



Contents lists available at ScienceDirect

Future Generation Computer Systems

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

A kernel machine-based secure data sensing and fusion scheme in wireless sensor networks for the cyber-physical systems[☆]

Xiong Luo^{a,b,*}, Dandan Zhang^{a,b}, Laurence T. Yang^c, Ji Liu^{a,b}, Xiaohui Chang^{a,b}, Huansheng Ning^{a,b}

^a School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing 100083, China

^b Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China

^c Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada

HIGHLIGHTS

- A novel data sensing and fusion scheme GM-KRLS is proposed in WSNs for the CPSs.
- GM-KRLS develops a prediction mechanism to reduce redundant transmissions in WSN.
- GM-KRLS improves the prediction accuracy with a kernel machine learning algorithm.
- Blowfish algorithm is employed to guarantee the confidentiality in our scheme.

ARTICLE INFO

Article history:

Received 15 June 2015

Received in revised form

8 September 2015

Accepted 30 October 2015

Available online xxxx

Keywords:

Secure data sensing and fusion

Wireless sensor networks

Kernel recursive least squares

Cyber-physical systems

ABSTRACT

Wireless sensor networks (WSNs) as one of the key technologies for delivering sensor-related data drive the progress of cyber-physical systems (CPSs) in bridging the gap between the cyber world and the physical world. It is thus desirable to explore how to utilize intelligence properly by developing the effective scheme in WSN to support data sensing and fusion of CPS. This paper intends to serve this purpose by proposing a prediction-based data sensing and fusion scheme to reduce the data transmission and maintain the required coverage level of sensors in WSN while guaranteeing the data confidentiality. The proposed scheme is called GM-KRLS, which is featured through the use of grey model (GM), kernel recursive least squares (KRLS), and Blowfish algorithm (BA). During the data sensing and fusion process, GM is responsible for initially predicting the data of next period with a small number of data items, while KRLS is used to make the initial predicted value approximate its true value with high accuracy. The KRLS as an improved kernel machine learning algorithm can adaptively adjust the coefficients with every input, while making the predicted value more close to actual value. And BA is used for data encoding and decoding during the transmission process due to its successful applications across a wide range of domains. Then, the proposed secure data sensing and fusion scheme GM-KRLS can provide high prediction accuracy, low communication, good scalability, and confidentiality. In order to verify the effectiveness and reasonableness of our proposed approach, we conduct simulations on actual data sets that are collected from sensors in the Intel Berkeley research lab. The simulation results have shown that the proposed scheme can significantly reduce redundant transmissions with high prediction accuracy.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

A cyber-physical system (CPS) as an integration of sensors networks with cyber resources responds intelligently to dynamic changes in physical world, where the wireless sensor networks (WSNs) as one of the key components collect sensor data from physical environment [1]. With the increasing presence and adoption of WSNs on the deployment of CPS, there has been a growing demand in data sensing and data fusion to utilize intelligence

[☆] This research is funded by the National Natural Science Foundation of China under Grants 61174103 and 61272357, the National Key Technologies R&D Program of China under Grant 2015BAK38B01, the Aerospace Science Foundation of China under Grant 2014ZA74001, and the Fundamental Research Funds for the Central Universities.

* Corresponding author at: School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing 100083, China. Tel.: +86 10 6233 2873.

E-mail address: xluo@ustb.edu.cn (X. Luo).

<http://dx.doi.org/10.1016/j.future.2015.10.022>

0167-739X/© 2015 Elsevier B.V. All rights reserved.

growing memory structure embedded in the weight coefficients, it naturally creates a growing radial-basis function network, while learning the network topology and adapting the free parameters directly from data at the same time. Compared with kernel least mean square (KLMS) algorithm as a typical kernel method, KRLS as an improved kernel machine learning algorithm has its unique features where the learning rule is a beautiful combination of the error-correction and memory-based learning [27,28]. Although KRLS and KLMS work in a similar way under error-correction learning scheme, the former aims at minimizing the sum of squared estimation errors and the latter aims at minimizing the instantaneous value of the squared estimation. The convergence rate of KRLS is therefore relatively faster than that of KLMS. In consideration of the above reasons, we choose KRLS for data prediction in our scheme, and GM-KRLS may achieve a high-accuracy data sensing and fusion with fast computing speed.

The rest of this paper is organized as follows. The related works are analyzed in Section 2. Section 3 describes the detailed secure data sensing and fusion scheme GM-KRLS. The simulation results and discussions are provided in Section 4. The conclusion is summarized in Section 5.

2. Related works

2.1. Grey model- (GM-) based prediction method

The grey system represents a system in which the information about it is poor, incomplete, or uncertain. Under the system analysis scheme using GM, it can use only a few data to estimate an unknown system [29]. Meanwhile, the GM is featured by a first-order differential equation used to characterize the system behavior. Since the storage ability of a sensor node is limited, it is not easy to provide complete information for the whole WSN. It therefore can be treated as a grey system with uncertain or incomplete information in the process of data sensing and fusion.

As a single variable first-order model, GM(1,1) is the most commonly used grey model. In this paper, it is employed to conduct the initial prediction for those nodes in WSN. The prediction procedure of GM(1,1) can be summarized as follows [30,31]:

- (1) Define the original positive data sequence as follows:

$$\mathbf{x}^{(0)} = [x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(k), \dots, x^{(0)}(n)], \quad (1)$$

where $x^{(0)}(k)$ is the time series data at time k , and n represents the length of the data sequence.

- (2) Generate a new sequence $\mathbf{x}^{(1)}$ by the accumulated generating operation (AGO) for the initial sequence $\mathbf{x}^{(0)}$:

$$\mathbf{x}^{(1)} = [x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(k), \dots, x^{(1)}(n)], \quad (2)$$

where

$$x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i), \quad k = 1, 2, \dots, n.$$

- (3) Form the first-order differential equation for $x^{(1)}(k)$ from $\mathbf{x}^{(1)}$:

$$\frac{\partial x^{(1)}(k)}{\partial k} + \omega x^{(1)}(k) = \vartheta, \quad (3)$$

where ω is the development coefficient, and ϑ denotes the grey input.

- (4) Use the ordinary least squares (OLS) method to estimate the grey parameters ω and ϑ in (3):

$$\begin{bmatrix} \hat{\omega} \\ \hat{\vartheta} \end{bmatrix} = (\mathbf{v}^T \mathbf{v})^{-1} \mathbf{v}^T \mathbf{q}, \quad (4)$$

where $\hat{\omega}$ and $\hat{\vartheta}$ are the estimated grey parameters, respectively.

$$\text{Moreover, } \mathbf{v} = \begin{bmatrix} -\frac{1}{2} [x^{(1)}(2) + x^{(1)}(1)] & 1 \\ -\frac{1}{2} [x^{(1)}(3) + x^{(1)}(2)] & 1 \\ \vdots & \vdots \\ \frac{1}{2} [x^{(1)}(n) + x^{(1)}(n-1)] & 1 \end{bmatrix}, \text{ and}$$

$$\mathbf{q} = [x^{(0)}(2), x^{(0)}(3), \dots, x^{(0)}(n)]^T.$$

- (5) Obtain the predictive function by solving (3) and using the estimated parameters in (4):

$$\hat{x}^{(1)}(k) = \left(x^{(0)}(1) - \frac{\hat{\vartheta}}{\hat{\omega}} \right) e^{-\hat{\omega}(k-1)} + \frac{\hat{\vartheta}}{\hat{\omega}}, \quad k = 1, 2, \dots \quad (5)$$

where $\hat{x}^{(1)}(k)$ is the predicted value of $x^{(1)}(k)$ at time k . Then, the predicted value $\hat{x}^{(0)}(k)$ at time k is:

$$\hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \hat{x}^{(1)}(k-1) \quad (6)$$

where $\hat{x}^{(1)}(0)$ is set to 0.

Recently, some extension and optimization for GM(1,1) have been conducted and new grey prediction models were developed, such as whitenization-based model GM(1,1,Whi) [32], interval grey prediction model considering uncertain information [33].

2.2. Kernel recursive least square (KRLS) learning algorithm

Adaptive algorithms can adjust their coefficients dynamically to adapt to the signal statistics in accordance with optimization algorithms. Consider an adaptive algorithm with M adjustable coefficients

$$y(i) = \mathbf{w}(i)^T \mathbf{u}(i) \quad (7)$$

where $\mathbf{u}(i) = [u(i-D), u(i-D-1), \dots, u(i-D-M+1)]^T$ is the input vector, $\mathbf{w}(i) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T$ is the coefficient vector, $y(i)$ denotes the output value, and D is the prediction delay ($D \geq 1$).

Then the error sequence $e(i)$ can be formed as below:

$$e(i) = d(i) - y(i) \quad (8)$$

where $d(i)$ is a desired output and $e(i)$ can be used in optimization algorithms for updating the coefficients.

As the adaptive algorithms in kernel spaces with improved performance, kernel adaptive algorithms have been proposed in recent years. The kernel version of the recursive least square (RLS) is given as below.

Let $\kappa(\mathbf{u}, \mathbf{v})$ be the kernel function. And the Gaussian kernel is defined as:

$$\kappa(\mathbf{u}, \mathbf{v}) = e^{-\zeta \|\mathbf{u} - \mathbf{v}\|^2} \quad (9)$$

where \mathbf{u} and \mathbf{v} are input vectors for kernel function, and ζ is a kernel parameter.

To derive RLS in reproducing kernel Hilbert space (RKHS), we utilize the Mercer theorem to transform the data $\mathbf{u}(i)$ into the feature space \mathbb{F} as $\varphi(\mathbf{u}(i))$ (denoted as $\varphi(i)$) [27]. We formulate the RLS algorithm on the example sequence $\{d(1), d(2), \dots\}$ and $\{\varphi(1), \varphi(2), \dots\}$. At each iteration, the weight vector $\mathbf{w}(i)$ is the optimization solution of

$$\min_{\mathbf{w}} = \sum_{j=1}^i \beta^{i-j} |d(j) - \mathbf{w}^T \varphi(j)|^2 + \beta^i \lambda \|\mathbf{w}\|^2 \quad (10)$$

where λ is the regulation parameter, and β is the forgetting factor.

Algorithm 1: KRLS**Input:** The original input data set $\mathbf{u}(i)$ The desired output data set $\mathbf{d}(i)$ **Output:** The predicted data set $\mathbf{y}(i)$

```

1  $\chi(1) = (\lambda\beta + \kappa(\mathbf{u}(1), \mathbf{u}(1)))^{-1}$ ,  $\mathbf{a}(1) = \chi(1)\mathbf{d}(1)$ ,  $i = 1$ 
2 While  $\mathbf{u}(i)$  is available do
3   Calculate the output:
    $\mathbf{h}(i) = [\kappa(\mathbf{u}(i), \mathbf{u}(1)), \dots, \kappa(\mathbf{u}(i), \mathbf{u}(i-1))]^T$ ,
    $\mathbf{y}(i) = (\mathbf{h}(i))^T \mathbf{a}(i-1)$ 
4   Calculate the error:  $e(i) = \mathbf{d}(i) - \mathbf{h}(i)^T \mathbf{a}(i-1)$ 
5   Update the coefficients:  $\mathbf{z}(i) = \chi(i-1)\mathbf{h}(i)$ ,
    $r(i) = \beta^i \lambda + \kappa(\mathbf{u}(i), \mathbf{u}(i)) - (\mathbf{z}(i))^T \mathbf{h}(i)$ ,
    $\chi(i) = r(i)^{-1} \begin{bmatrix} \chi(i-1)r(i) + \mathbf{z}(i)(\mathbf{z}(i))^T & -\mathbf{z}(i) \\ -(\mathbf{z}(i))^T & 1 \end{bmatrix}$ ,
    $\mathbf{a}(i) = \begin{bmatrix} \mathbf{a}(i-1) - \mathbf{z}(i)(r(i))^{-1}e(i) \\ (r(i))^{-1}e(i) \end{bmatrix}$ 
6  $i = i + 1$ 
7 End while

```

By introducing

$$\mathbf{d}(i) = [d(1), \dots, d(i)]^T \quad (11)$$

$$\boldsymbol{\phi}(i) = [\phi(1), \dots, \phi(i)]. \quad (12)$$

We have

$$\mathbf{w}(i) = [\beta^i \lambda \mathbf{I} + \boldsymbol{\phi}(i) \boldsymbol{\zeta}(i) (\boldsymbol{\phi}(i))^T]^{-1} \boldsymbol{\phi}(i) \boldsymbol{\zeta}(i) \mathbf{d}(i) \quad (13)$$

where $\boldsymbol{\zeta}(i) = \text{diag}\{\beta^{i-1}, \beta^{i-2}, \dots, 1\}$.

Furthermore, by using the matrix inversion lemma, (13) can be rewritten as

$$\mathbf{w}(i) = \boldsymbol{\phi}(i) [\beta^i \lambda (\boldsymbol{\zeta}(i))^{-1} + (\boldsymbol{\phi}(i))^T \boldsymbol{\phi}(i)]^{-1} \mathbf{d}(i). \quad (14)$$

The weight is explicitly expressed as a linear combination of the input data:

$$\mathbf{w}(i) = \boldsymbol{\phi}(i) \mathbf{a}(i)$$

with

$$\mathbf{a}(i) = [\beta^i \lambda (\boldsymbol{\zeta}(i))^{-1} + (\boldsymbol{\phi}(i))^T \boldsymbol{\phi}(i)]^{-1} \mathbf{d}(i). \quad (15)$$

Denote

$$\chi(i) = [\beta^i \lambda (\boldsymbol{\zeta}(i))^{-1} + (\boldsymbol{\phi}(i))^T \boldsymbol{\phi}(i)]^{-1} \quad (16)$$

 $\chi(i)$ can be expressed as follows:

$$\chi(i) = (r(i))^{-1} \begin{bmatrix} \chi(i-1)r(i) + \mathbf{z}(i)(\mathbf{z}(i))^T & -\mathbf{z}(i) \\ -(\mathbf{z}(i))^T & 1 \end{bmatrix} \quad (17)$$

where

$$\mathbf{z}(i) = \chi(i-1)\mathbf{h}(i) \quad (18)$$

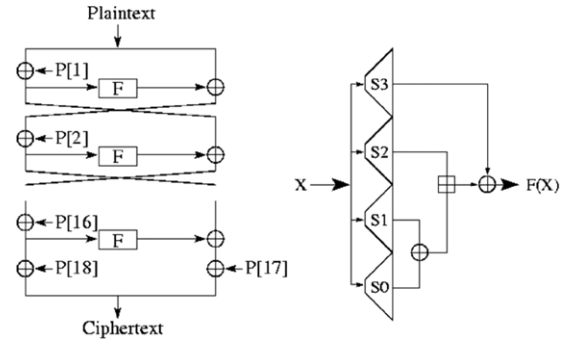
$$r(i) = \beta^i \lambda + \kappa(\mathbf{u}(i), \mathbf{u}(i)) - (\mathbf{z}(i))^T \mathbf{h}(i) \quad (19)$$

and $\mathbf{a}(i)$ can be calculated as follows:

$$\mathbf{a}(i) = \begin{bmatrix} \mathbf{a}(i-1) - \mathbf{z}(i)(r(i))^{-1}e(i) \\ (r(i))^{-1}e(i) \end{bmatrix}. \quad (20)$$

KRLS algorithm can be summarized in Algorithm 1. As defined in this algorithm, \mathbf{a} is a set of coefficients of the kernel expansion.

Furthermore, some improvements on the basis of such basic KRLS have been achieved. For instances, by incorporating an online vector quantization method [28] or a forgetting technique in Bayesian inspired framework [34], the learning performance of KRLS can be improved. Then, those KRLS algorithms are particularly applicable to cases in which data arrives sequentially. From this point of view, this learning algorithm may have great potential to address our discussed issue in this paper.

**Fig. 1.** Schematic diagram of Blowfish algorithm.**2.3. Blowfish algorithm (BA)**

Every sensor node has a secret key which differs from other nodes. The sink node will generate a session key at the beginning and broadcast it to all sensor nodes. The sensor node will calculate Needham–Schroeder symmetric key (NSSK) with its secret key and the session key for data encoding and decoding of this sensor node. Since the sink node knows all the secret keys of sensor nodes, it could calculate NSSKs to decode the data. Data transmissions between sink node and sensor nodes employ BA for encryption [35]. One run of data sensing, fusion, and transmission is listed as below:

- (1) The sensor node senses data and employs the data sensing and fusion scheme to data prediction and fusion.
- (2) If the prediction error is below the threshold, both the sink node and sensor nodes consider the predicted data as actual data, and transmission is canceled. Otherwise, sensor node sends the actual data to sink node.
- (3) For the data that need to be sent, the sensor node employs NSSK for encoding.
- (4) The sink node calculates NSSKs of sensor nodes and decodes data.

Bruce Schneier designed the BA which can be available in the public domain [24]. Since BA was first introduced in 1993, it has not been cracked yet, where Blowfish is a variable length key with 64-bit block cipher. From the application side, BA has demonstrated many successful applications across a wide range of domains.

As shown in Fig. 1 [25], there are two parts of BA, i.e., the key expansion and the data encryption. The key expansion of BA begins with the P-array and S-box through the utilization of many sub-keys, while it converts a key of at most 448 bits into several sub-key arrays with 4168 bytes. Meanwhile the data encryption is implemented through a 16-round network, where a key-dependent permutation and a key-dependent and data-dependent substitution are conducted in each round. All of the operations include XOR and additions on 32-bit words. Here, the F-function of BA is probably the most complex part of this algorithm because it is the only part of utilizing the S-box. More recently, to simplify the precessing complexity, a novel F-function was designed to generate dynamic S-box and XOR operator [36], and a new method was also developed to generate S-box and P-array [37].

Considering those features of above surveyed algorithms, it may be an innovative case through the integration of those three methods in our proposed scheme. Specifically, compared with other existing data fusion methods in WSNs, the proposed method has some unique advantages. Our contributions could be summarized as follows. Firstly, we employ GM to reduce random errors while using KRLS to improve the prediction accuracy due to its powerful nonlinear mapping ability, and the combination of GM and KRLS is novel in kernel methods. Secondly, to guarantee data security, both the sink node and sensor node use the same

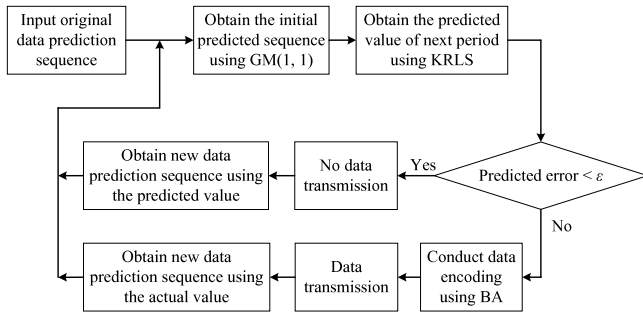


Fig. 2. Schematic diagram of the proposed data sensing and fusion scheme.

prediction mechanism to reduce redundant data transmission, meanwhile BA is introduced for data encoding and decoding. Thirdly, we use small number of recent data to predict the data of next period in data fusion scheme, thus, the increase of data set may not influence the prediction effect. In this case, the proposed method may be effective in addressing some big-scale data sets.

3. Secure data sensing and fusion scheme GM-KRLS

3.1. The scheme GM-KRLS

In consideration of the limited energy and storage as well as data security in sensor nodes, we present a prediction-based secure data sensing and fusion scheme GM-KRLS to reduce redundant data transmission, save energy with low computational cost, and keep the data confidentiality. To guarantee that the data series in the sink node and the sensor nodes are synchronous in every period while conducting the data sensing and fusion, both the sink node and the sensor nodes should employ the same data sequence and prediction mechanism [19]. Then the end-users can get the data of every sampling point in sensor nodes with low communication cost.

In the secure data sensing and fusion scheme, the data of the next period is predicted through our proposed method. The sensor node compares the sensed data with the predicted data. If the error between them is under the threshold ε , it is unnecessary for the sensor node to send the data to sink node, and the energy is saved, while achieving the goal of data fusion. Meanwhile, the sink node also employs the same prediction mechanism to predict the data of next period, and then considers the predicted data as the sensed data in current period. Furthermore, the sensor node should transmit the sensed data to sink node through the use of BA for encoding when the prediction error is beyond the threshold ε . It should be pointed out that ε is defined by end-users and it can be adjusted. Then the prediction accuracy will be influenced with different values of ε .

To improve the prediction accuracy through the use of a few sample data items, in this paper, after obtaining the initial predicted data sequence via GM, the proposed method employs KRLS to make this predicted sequence approximate its actual value. Thus the proposed secure data sensing and fusion scheme is presented in Fig. 2 and detailed description is listed as follows.

- (1) The sink node sends its acceptable prediction error threshold ε and the session key to all sensor nodes. In the first n periods, all sensor nodes transmit their sensed data to the sink node and calculate NSSKs with their keys and the session key. Then they construct the initial predicted data sequence.
- (2) Both the sink node and sensor node conduct prediction using the same data sequence and prediction mechanism. With the initial predicted data sequence of size n , a new data sequence of size $(n + 1)$ can be obtained via GM(1, 1).

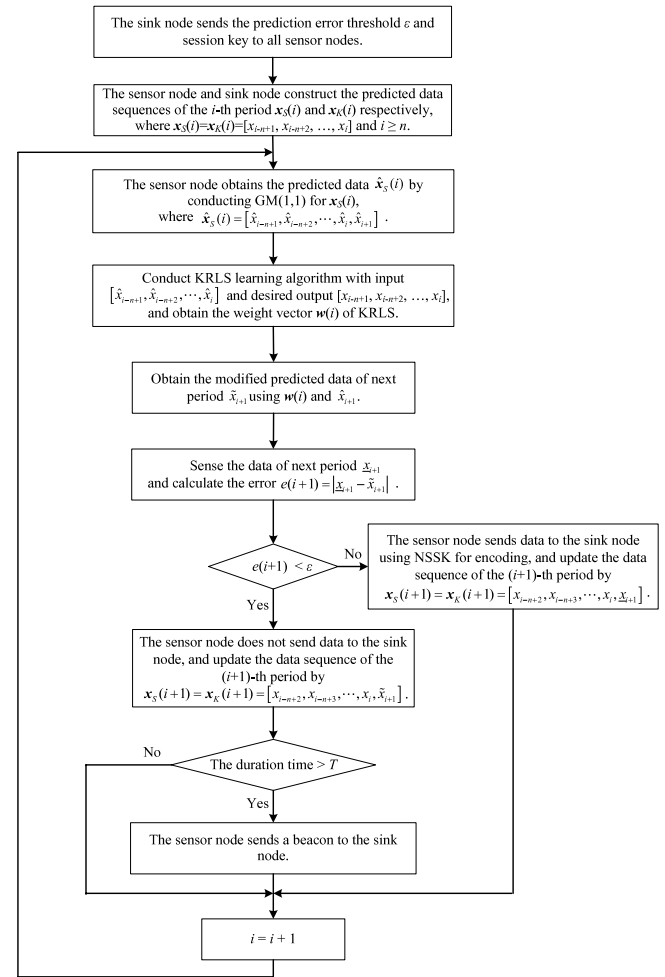


Fig. 3. The flowchart of scheme GM-KRLS.

- (3) Train KRLS model with the initial predicted data sequence and the first n values of new data sequence being its input and output, respectively. And then the hidden relationships between those data sequences, i.e., the weight vector of KRLS, can be obtained.
- (4) With the hidden relationships obtained above, the KRLS algorithm takes the $(n + 1)$ -th value of new data sequence generated by GM as input, and the predicted value of next period, i.e., the output of KRLS, could be obtained.
- (5) The sensor node senses the actual data of next period.
- (6) Calculate the error between the actual data and the predicted data of next period.
- (7) Compare the error with the threshold ε . If the error is less than ε , which means the error is acceptable, the sensor node does not send data to sink node. Meanwhile, both the sink node and sensor node think of the predicted value as the actual value. Otherwise, the sensor node sends the actual value to sink node using BA. Reconstruct new data prediction sequence.
- (8) Specifically, if the sensor node does not send the data to the sink node within a fixed time T , the sensor node should send a beacon to the sink node. Loop (2)–(8) with new data sequence.

On the whole, the proposed data sensing and fusion scheme can be summarized in Fig. 3.

3.2. Complexity analysis

We give an analysis on time complexity of our proposed secure data sensing and fusion method. Let q be the number of samples



in the proposed scheme. It is obvious that the main computation of our method is spent on GM(1,1) and KRLS learning. For every data sequence in GM, it will conduct calculations for n times to generate the new data sequence where n denotes the length of the original data sequence. And q data sequences will be generated in the whole prediction process. Thus, the computational complexity of GM(1,1) is $O(qn)$. Similarly, for every training process of KRLS, the number of times we perform calculations equals the number of input vectors (i.e., n). Each input vector is used for calculating the coefficients of KRLS which could identify the relationship between input and output. Meanwhile, KRLS will be trained q times. The learning complexity of KRLS is also $O(qn)$ [27,38]. Thus, we can conclude that the complexity of our proposed scheme GM-KRLS is $O(qn)$.

In order to verify the effectiveness of the proposed data sensing and fusion scheme in WSN, three actual data sets are imported for simulation. We evaluate the performance of scheme through prediction results and the successful prediction rate. GM is the basis of GM-LSSVM, GM-OP-ELM, and GM-KRLS. GM-LSSVM and GM-OP-ELM seem to perform better than GM, however, they all have their own defects. GM-LSSVM spends much time in computation, and GM-OP-ELM may have a worse prediction accuracy in some situations although its computational speed is quite fast. Therefore, in order to evaluate the computational efficiency and prediction accuracy of those schemes, we conduct simulations of our method compared with GM, GM-LSSVM, and GM-OP-ELM. All the simulations are conducted in MATLAB computing environment running in an Intel(R)Core(TM)i5-2410M, 2.30 GHz CPU. Here the algorithm LS-SVM designed in GM-LSSVM scheme is implemented by using a MATLAB Toolbox LS-SVM [39,40].

temperature, humidity, and light data streams. It includes 2000 continuous data items used in the following simulations. In our simulations, we choose the first 40 sampling points of each data set to construct the initial predicted data sequence. It means that n defined in Section 3 is set to 40, and we conduct predictions for the following 1960 sampling points. In the proposed scheme, ε represents the requirements of end-users on data accuracy. Therefore, in our simulations, we test the successful prediction rate and the communication overhead of sensor node under different threshold ε .

The parameters of KRLS are set as follows: kernel parameter $\zeta = 0.01$, regularization parameter $\lambda = 0.0008$, and forgetting factor $\beta = 0.8$. The prediction results of the temperature data sequence are shown in Fig. 5 when $\varepsilon = 0.17$. This figure shows that the prediction values by using GM-LSSVM, GM-OP-ELM, and GM-KRLS schemes with the actual sensed data. It can be observed that the prediction values of these three methods are in good agreement with the actual values, which means they could achieve a good prediction effect. However, in some sampling points, the predicted values of GM-LSSVM and GM-OP-ELM vary hugely with a worse prediction effect. Thus it is clear that the performance of GM-KRLS is better than that of others.

In addition, it is optional for choosing the threshold ε . In Fig. 5 we choose the threshold as $\varepsilon = 0.17$, and it is nearly in the middle of the range we have set. Actually, we could change the value of ε , then the related simulation is also conducted when $\varepsilon = 0.33$ and the similar results are obtained in Fig. 6. It can be found that the prediction performance with $\varepsilon = 0.33$ is worse than that with $\varepsilon = 0.17$. Generally speaking, a smaller threshold means a high-quality prediction process with a smaller predicted error, and it also means a smaller successful prediction rate with more data transmission in WSN. After performing many runs for our method, we find in our simulation, 0.17 is a proper choice for the threshold considering the practical requirement of trade-off between the prediction effectiveness and the amount of transmitted data.

Fig. 7 shows the successful prediction rate as the threshold ϵ changes. From this figure, it can be observed obviously that the bigger the value of the threshold is, the higher the successful prediction is. Furthermore, as the threshold changes, the successful prediction rate of GM-KRLS is always higher than that of other methods, which means that GM-KRLS has the best prediction effect. Meanwhile, it is known that the higher the successful prediction rate is, the less communication the overhead ϵ produce. That is the goal of data sensing and fusion. Thus, it is obvious that the GM-OP-ELM and GM-LSSVM schemes outperform

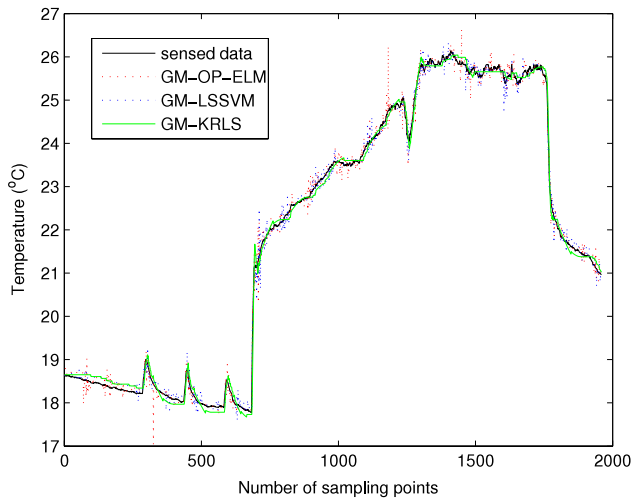


Fig. 5. Prediction results of GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for temperature items ($\varepsilon = 0.17$).

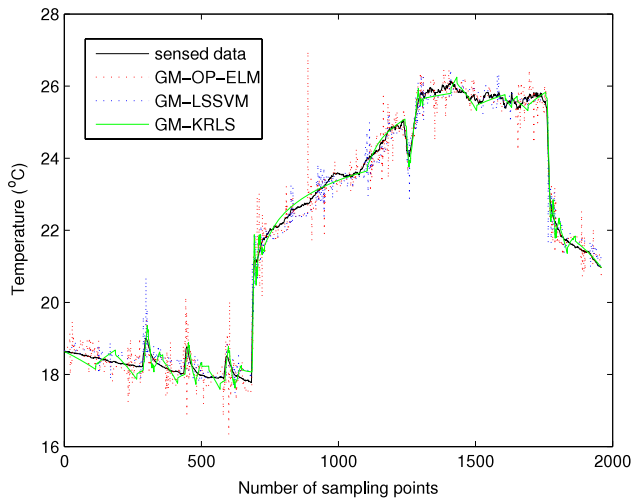


Fig. 6. Prediction results of GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for temperature items ($\varepsilon = 0.33$).

GM-based scheme. It means that the GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes can save the energy by improving the prediction accuracy, and extend lifetime of the whole WSN simultaneously. Moreover, GM-KRLS seems to perform better in dealing with this issue.

Fig. 8 shows the corresponding average error of these three schemes when the threshold ε changes. The average error can be used to evaluate the performance of method, and a smaller average error indicates a better prediction effect. It can be found that the average errors of GM-KRLS are minimum and the errors of GM-OP-ELM are maximum. Thus the overall prediction effect of GM-KRLS is best. Specifically, Fig. 9 shows the predicted error of GM-KRLS at every sampling point when $\varepsilon = 0.17$. We can find that these errors are constrained within relatively tight bounds.

4.2. Results of humidity data set

The parameters of KRLS are set as follows: kernel parameter $\zeta = 0.03$, regularization parameter $\lambda = 0.0008$, and forgetting factor $\beta = 0.8$. Fig. 10 depicts the actual humidity data of every sampling point and the prediction results using GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes when $\varepsilon = 0.17$. It is clear that the prediction values of three schemes almost follow the actual sensed

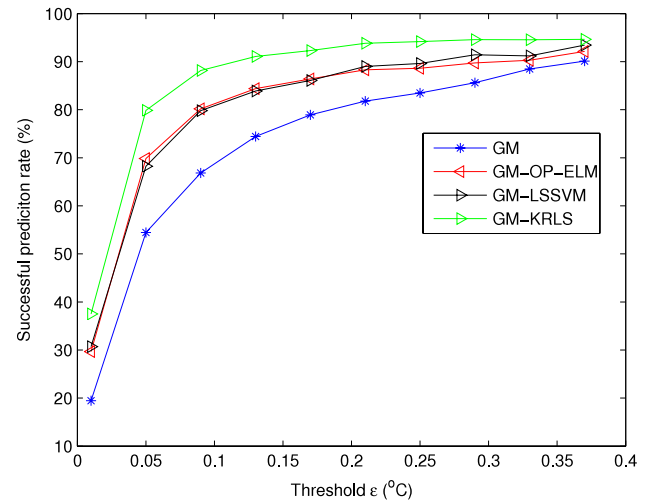


Fig. 7. Successful prediction rate with different threshold of GM, GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for temperature item.

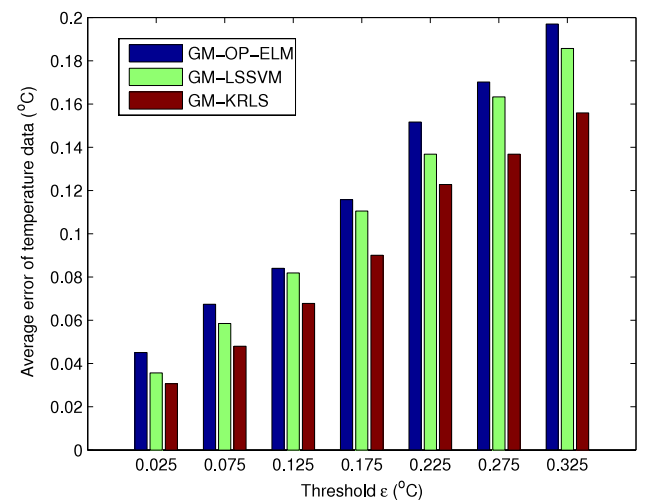


Fig. 8. Average predicted error with different threshold of GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for temperature item.

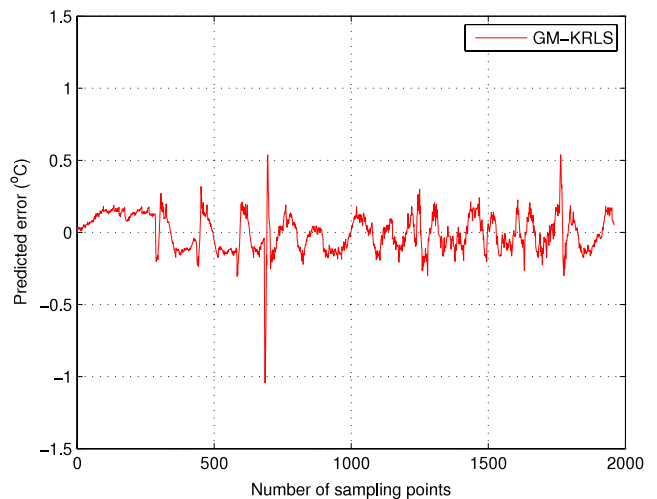


Fig. 9. The predicted error of GM-KRLS scheme for temperature item ($\varepsilon = 0.17$).

humidity data. Fig. 11 shows the successful prediction rate under different threshold ε . The successful prediction rate of GM-OP-ELM

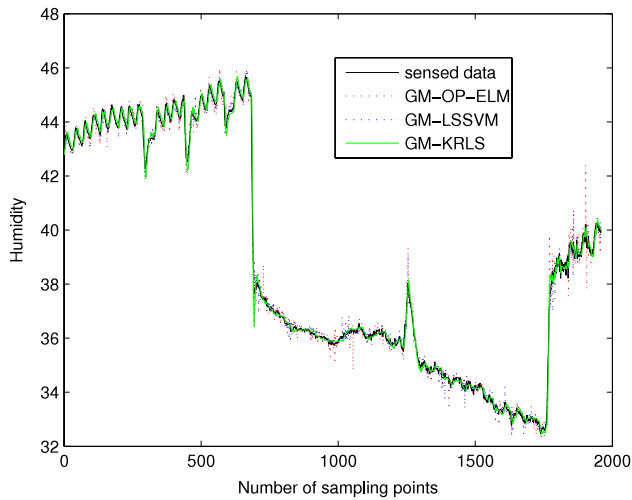


Fig. 10. Prediction value of GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for humidity items ($\varepsilon = 0.17$).

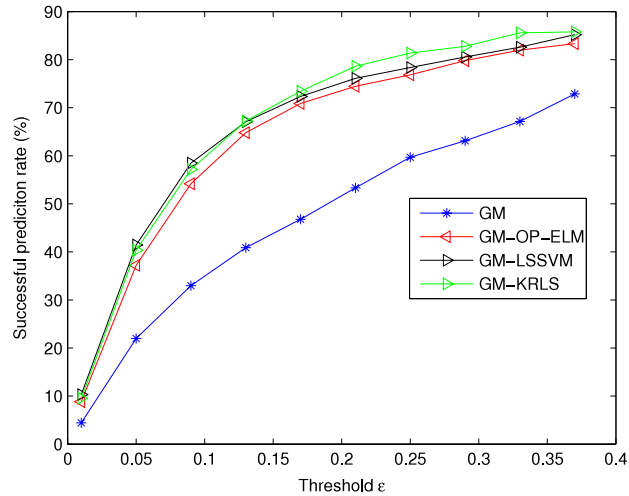


Fig. 11. Successful prediction rate with different threshold of GM, GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for humidity item.

is close to that of GM-LSSVM for the humidity data sequence, and GM-KRLS has a higher successful rate than other schemes.

Here Fig. 12 shows the corresponding average error of these three schemes when the threshold ε changes. Fig. 13 shows the predicted error of GM-KRLS at every sampling point when $\varepsilon = 0.17$.

4.3. Results of light data set

The parameters of KRLS are set as follows: kernel parameter $\zeta = 0.01$, regularization parameter $\lambda = 0.5$, and forgetting factor $\beta = 0.8$. Fig. 14 displays the actual light sensed data and prediction value using GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes at every sampling point when $\varepsilon = 0.17$. It can be seen that this data sequence is nonlinear. Fig. 15 shows the successful prediction rate under different threshold ε of GM, GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for light data sequence. As we can see from this figure, the GM-KRLS also slightly outperforms GM-LSSVM and GM-OP-ELM for these high nonlinear light data items. Here, we can see that from Fig. 14, the light data set is different from temperature and humidity data sets because its values vary within a relatively bigger. Then, as a high-nonlinear and fluctuant time series data sequence, this light sensed data set has some complex characteristics. It results in a lower successful prediction rate with

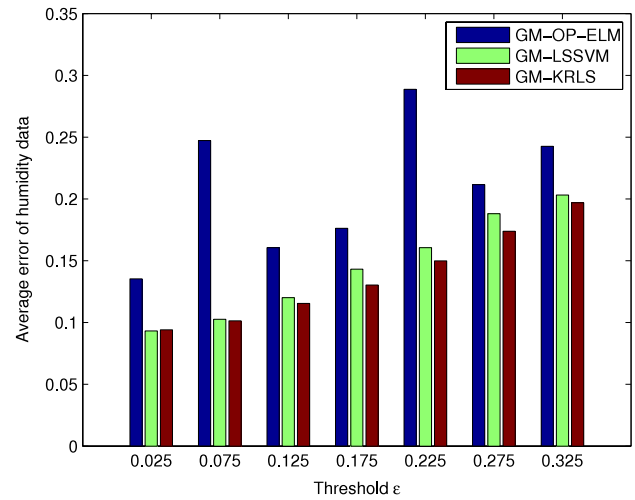


Fig. 12. Average predicted error with different threshold of GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for humidity item.

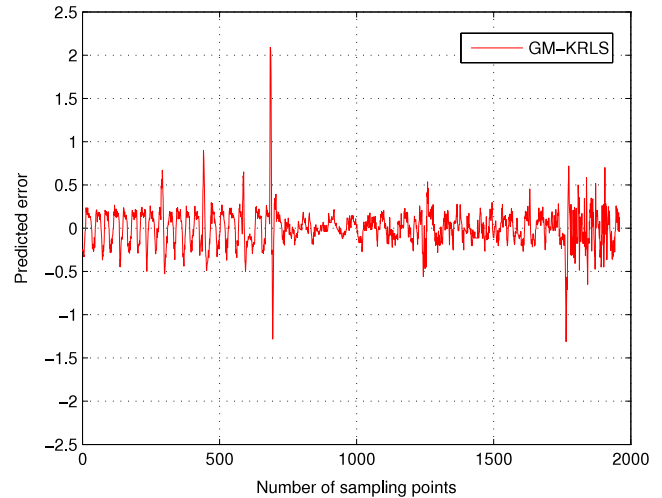


Fig. 13. The predicted error of GM-KRLS scheme for humidity item ($\varepsilon = 0.17$).

the threshold less than 0.4. However, our proposed method is still efficient since the successful prediction rate is higher than that of other two GM-based methods. Furthermore, in order to verify the effectiveness of our scheme, we assign a series of bigger values to the threshold in Fig. 15, where we can find that with the increase of threshold, GM-KRLS still performs better than other three methods.

In addition, Fig. 16 shows the corresponding average error of these three schemes when the threshold ε changes. Fig. 17 shows the predicted error of GM-KRLS at every sampling point when $\varepsilon = 0.17$.

In addition to the above evaluation for prediction accuracy of those schemes, we also provide a comparison for the computational time of GM-KRLS, GM-OP-ELM, and GM-LSSVM schemes in every prediction period for three items. The results are listed in Fig. 18. It can be observed that the computational time of GM-OP-ELM is the least, and the computational time of GM-KRLS is almost the same with that of GM-OP-ELM. But when compared with GM-LSSVM, GM-KRLS is more efficient since the computational time of GM-LSSVM is almost 10 times as much as that of GM-KRLS.

In consideration of the prediction accuracy and the computational time simultaneously, our proposed scheme GM-KRLS may be a competitive choice. Finally, it should be pointed out that although GM-OP-ELM performs better than GM-KRLS in terms of

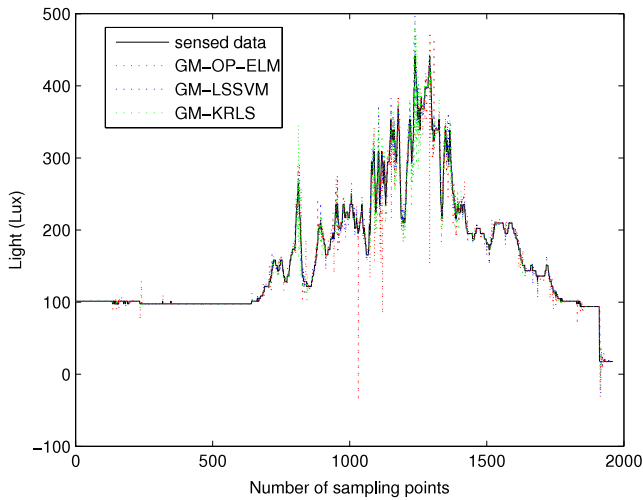


Fig. 14. Prediction value of GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for light items ($\varepsilon = 0.17$).

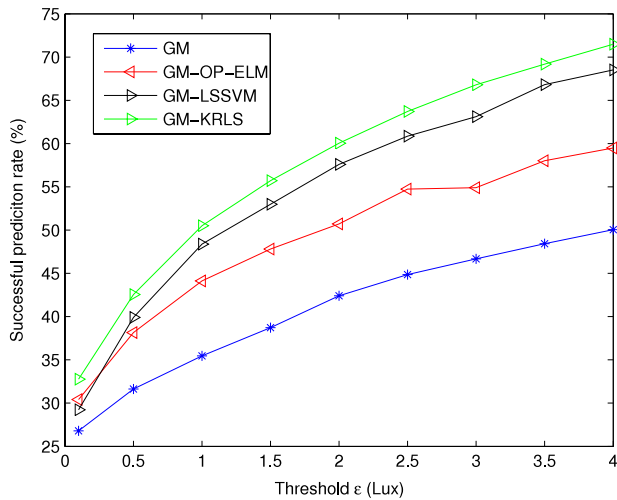


Fig. 15. Successful prediction rate with different threshold of GM, GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for light item.

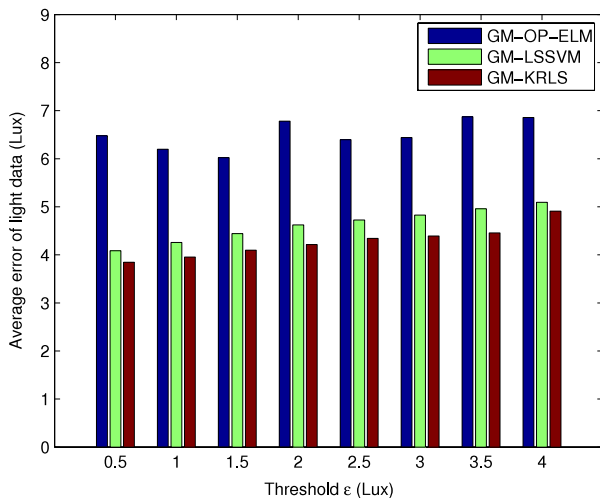


Fig. 16. Average predicted error with different threshold of GM-OP-ELM, GM-LSSVM, and GM-KRLS schemes for light item.

the computational time, the former may suffer from design choices during its implementation. Actually, ELM uses random projection

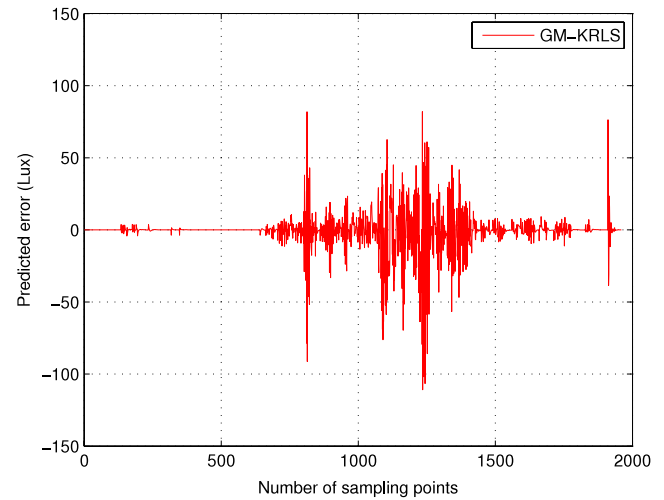


Fig. 17. The predicted error of GM-KRLS scheme for light item ($\varepsilon = 0.17$).

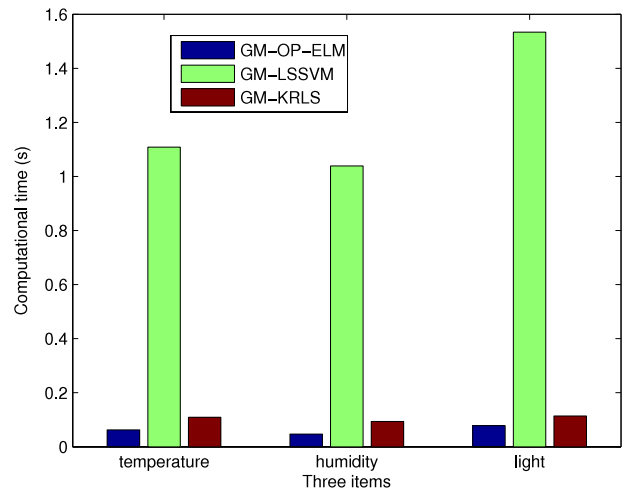


Fig. 18. Computational time of GM-KRLS, GM-OP-ELM, and GM-LSSVM schemes in every prediction period for three items.

spaces, while KRLS uses data centered functional bases. It is difficult to set the design parameters optimally under the mathematical framework of ELM, practically we need to conduct many trials and cross validation to select the number of hidden neurons and the nonlinear functions with the purpose of finding a good projection space. On the other hand, the KRLS learning algorithm can avoid such limitation by just mapping the data nonlinearly and deterministically to a Hilbert space, and adapting online the projection [38]. KRLS and its data centered basis functions are therefore able to concentrate bases on the part of the functional space where the input data exists. In view of it, the scheme GM-KRLS has its unique advantage under the current computational framework.

4.4. Results of other data set

From the descriptions of the proposed data sensing and fusion scheme GM-KRLS, it can be observed that a data sequence with fixed length is used for the data prediction of next period. In this case, the prediction effect is closely related to the length of data sequence (i.e., n defined in Section 3) and the increase of data set may not influence the algorithm effect. From this point of view, the proposed schema may be appropriate for other data sets with bigger size. Here we employ a larger temperature data set with 15,000 values to conduct simulation. When $\varepsilon = 0.17$, Figs. 19 and 20 show the prediction effect and the predicted error at every

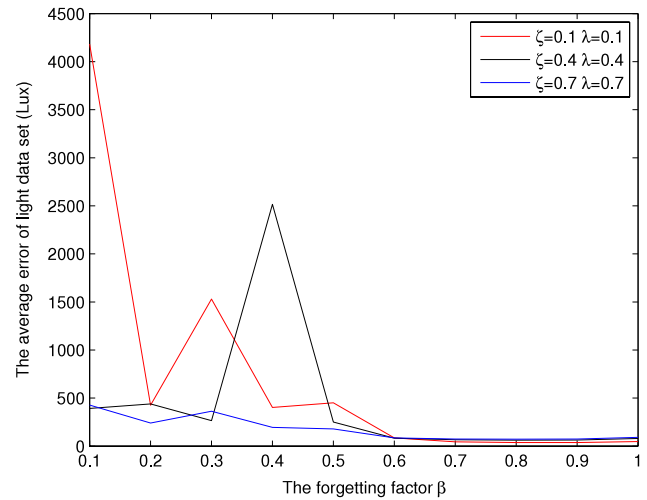


Fig. 21. Predicted error with different β .

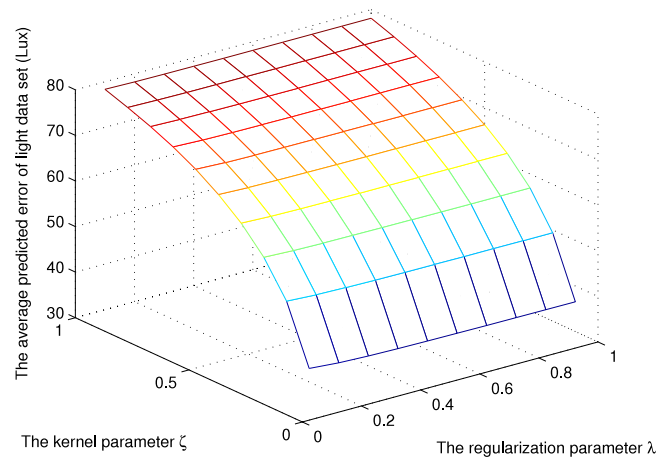


Fig. 22. Predicted error with different ζ and λ ($\beta = 0.8$).

error is shown in Fig. 22. Here, the average error decreases when the ζ value is decreased from 1.0 to 0.1, meanwhile, λ has rather little meaning for error. Therefore, we will select ζ first.

We vary the values of ζ from 0.005 to 0.1 with a step value of 0.005 where λ value is randomly set to 0.1. Fig. 23 shows the average predicted error with different ζ . Obviously, the ζ value of 0.01 achieves the best performance, and this is why we use $\zeta = 0.01$ as the default setting in this simulation. Now, we only have an unknown value of λ . Fig. 24 shows the average predicted error while changing the values of λ . According to this figure, we set $\lambda = 0.5$, then the minimum predicted error can be obtained. So far, three parameters are selected. Similarly, those parameters of GM-KRLS in processing temperature and humidity data sets could be selected.

5. Conclusion

In WSNs, the prediction-based secure data sensing and fusion are effective in reducing redundant data communications, saving the energy of sensor nodes, keeping data confidentiality, and enhancing the lifetime of network. Considering that the data sensed by the sensor nodes are of high temporal redundancy and the sensor nodes have limited energy, storage capacity, and data processing ability, a novel prediction method based on secure data sensing and fusion scheme GM-KRLS using GM, KRLS, and BA is proposed in this paper to deal with those issues mentioned above. In order to guarantee the data synchronization between

4.6. Parameter selections

In kernel methods, the parameter selections play an important role in the design of learning algorithm. Here, taking the processing for light data set as example, we explain how to choose proper parameters in simulations. As mentioned above, λ , ζ , and β are three important parameters of KRLS in GM-KRLS scheme. Firstly, we will choose an appropriate β after considering three different combinations of λ and ζ . Fig. 21 shows the average predicted error in all three cases. It can be observed that the error is minimum when $\beta = 0.8$, thus β is set to 0.8. Then, we vary the values of λ and ζ from 0.1 to 1.0 with a step value of 0.1. The average predicted

Table 1
Simulation for encoding and decoding.

NSSK	Raw data	After encoding	After decoding
nikiisexcellent	25.341	160–102–23110–25–13–693551	25.341
nikiisexcellent	66.5	14015–15–71–117–46–26–23–111	66.5
nikiisexcellent	128	130–6689–5099–66504–75	128
nikiisexcellent	0.3861	16069–80–2711–104–11–106–82	0.3861
youyouisperfect	25.341	16063–82–22–88–98–666234	25.341
youyouisperfect	66.5	140–41116107–108470–112–88	66.5
youyouisperfect	128	130–44–127848–10940–5389	128
youyouisperfect	0.3861	160–123–63–121–127–19–50111	0.3861

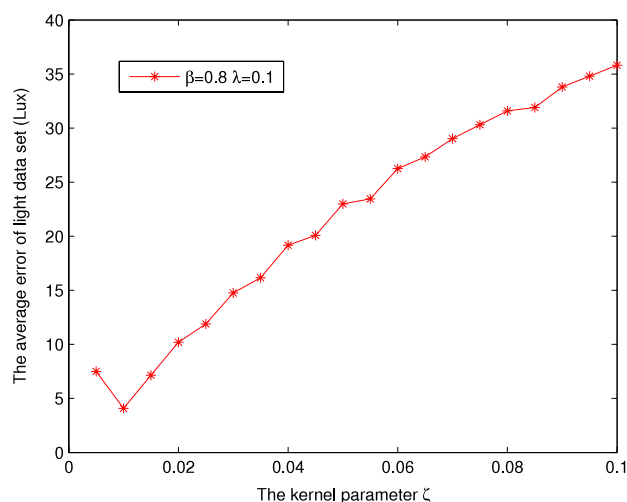


Fig. 23. Predicted error with different ζ .

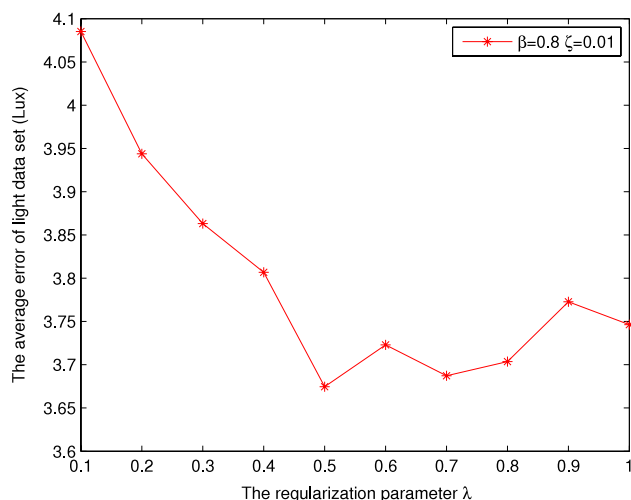


Fig. 24. Predicted errors with different λ .

sensor nodes and sink node, the proposed scheme develops a prediction mechanism. During the data sensing and fusion process, the GM–KRLS firstly uses GM to implement the initial prediction with a small number of data items, then employs KRLS learning algorithm to modify the predicted value with lower computational cost and higher successful prediction rate. Meanwhile, BA is used for data encoding and decoding during transmission process. It can be found from simulation results that the proposed scheme can improve the prediction accuracy and reduce the energy consumption caused by redundant transmission. GM–KRLS performs better than GM–LSSVM and GM–OP–ELM in terms of accuracy, moreover its computational speed is close to GM–OP–

ELM but much more faster than GM–LSSVM. For high nonlinear and fluctuant time series data sequence (such as data set of light in Section 4), GM–KRLS has significant advantages than other schemes. The confidentiality of data values is guaranteed by means of reducing transmission and encoding data. In view of those issues mentioned above, compared with other schemes, GM–KRLS may be a better choice in industry WSNs on the deployment of CPSSs.

References

- [1] L.A. Tang, J. Han, G. Jiang, Mining sensor data in cyber-physical systems, *Tsinghua Sci. Technol.* 19 (2014) 225–234.
- [2] F.J. Wu, Y.F. Kao, Y.C. Tseng, From wireless sensor networks towards cyber physical systems, *Pervasive Mob. Comput.* 7 (2011) 397–413.
- [3] H.S. Ning, H. Liu, J.H. Ma, L.T. Yang, R.H. Huang, Cybermatics: Cyber-physical-social-thinking hyperspace based science and technology, *Future Gener. Comput. Syst.* (2015) <http://dx.doi.org/10.1016/j.future.2015.07.012>.
- [4] C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio, L.T. Yang, A survey on communication and data management issues in mobile sensor networks, *Wirel. Commun. Mob. Comput.* 14 (2014) 19–36.
- [5] R. Tan, G. Xing, B. Liu, J. Wang, X. Jia, Exploiting data fusion to improve the coverage of wireless sensor networks, *IEEE/ACM Trans. Netw.* 20 (2012) 450–462.
- [6] P. Ji, C. Wu, Y. Zhang, F. Chen, A low-energy adaptive clustering routing protocol of wireless sensor networks, in: *Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China, 2011*, pp. 1–4.
- [7] M. Li, Z. Li, A.V. Vasilakos, A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues, *Proc. IEEE* 101 (2013) 2538–2557.
- [8] Y. Nam, S. Rho, B.G. Lee, Intelligent context-aware energy management using the incremental simultaneous method in future wireless sensor networks and computing systems, *EURASIP J. Wirel. Commun. Netw.* 2013 (2013) 10.
- [9] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy conservation in wireless sensor networks: A survey, *Ad Hoc Networks* 7 (2009) 537–568.
- [10] E.F. Nakamura, A.A.F. Loureiro, A.C. Frery, Information fusion for wireless sensor networks: methods, models, and classifications, *ACM Comput. Surv.* 39 (2007).
- [11] H. Li, K. Li, W. Qu, S. Ivan, Secure and energy-efficient data aggregation with malicious aggregator identification in wireless sensor networks, *Future Gener. Comput. Syst.* 37 (2014) 108–116.
- [12] R.H. David, G.S. Antonio-Javier, G.S. Felipe, G.H. Joan, On the improvement of wireless mesh sensor network performance under hidden terminal problems, *Future Gener. Comput. Syst.* 45 (2015) 95–113.
- [13] C.T. Cheng, H. Leung, P. Maupin, A delay-aware network structure for wireless sensor networks with in-network data fusion, *IEEE Sens. J.* 13 (2013) 1622–1631.
- [14] H. Luo, H.X. Tao, H.D. Ma, S.K. Das, Data fusion with desired reliability in wireless sensor networks, *IEEE Trans. Parallel Distrib. Syst.* 22 (2011) 501–513.
- [15] C.L. Chen, J. Yan, N. Lu, Y. Wang, X. Yang, X. Guan, Ubiquitous monitoring for industrial cyber-physical systems over relay assisted wireless sensor networks, *IEEE Trans. Emerging Top. Comput.* 3 (2015) 352–362.
- [16] A.R. Pinto, C. Montez, G. Arajo, F. Vasques, P. Portugal, An approach to implement data fusion techniques in wireless sensor networks using genetic machine learning algorithms, *Inf. Fusion* 15 (2014) 90–101.
- [17] O. Kreibich, J. Neuzil, R. Smid, Quality-based multiple-sensor fusion in an industrial wireless sensor network for MCM, *IEEE Trans. Ind. Electron.* 61 (2014) 4903–4911.
- [18] A. Hui, L. Cui, Forecast-based temporal data aggregation in wireless sensor networks, *Comput. Eng. Appl.* 43 (2007) 121–125.
- [19] J. Kang, L. Tang, X. Zuo, X. Zhang, H. Li, GMSVM-based prediction for temporal data aggregation in sensor networks, in: *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, China, 2009*, pp. 1–4.
- [20] N.M. Xie, T.X. Yao, S.F. Liu, Multi-variable grey dynamic forecasting model based on complex network, in: *Proceedings of International Conference on Management Science and Engineering, Moscow, Russia, 2009*, pp. 213–219.
- [21] R. Wang, J. Tang, D. Wu, Q. Sun, GM–LSSVM based data aggregation in WSN, *Comput. Eng. Des.* 33 (2012) 3371–3375.

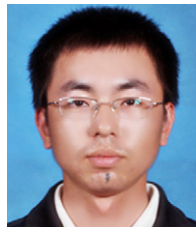
- [22] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally pruned extreme learning machine, *IEEE Trans. Neural Netw.* 21 (2010) 158–162.
- [23] X. Luo, X.H. Chang, A novel data fusion scheme using grey model and extreme learning machine in wireless sensor networks, *Int. J. Control Autom. Syst.* 13 (2015) 539–546.
- [24] B. Schneier, The Blowfish encryption algorithm. Available: <http://www.schneier.com/blowfish.html> (accessed June, 2015).
- [25] T.Y. Nie, T. Zhang, A study of DES and Blowfish encryption algorithm, in: *Proceedings of IEEE Region 10 Annual International Conference*, Singapore, 2009.
- [26] Y. Engel, S. Mannor, R. Meir, The kernel recursive least-squares algorithm, *IEEE Trans. Signal Process.* 52 (2004) 2275–2285.
- [27] W.F. Liu, J.C. Principe, S. Haykin, *Kernel Adaptive Filtering*, Wiley, New Jersey, 2011.
- [28] B.D. Chen, S.L. Zhao, P.P. Zhu, J.C. Principe, Quantized kernel recursive least squares algorithm, *IEEE Trans. Neural Netw. Learn. Syst.* 24 (2013) 1484–1491.
- [29] F.M. Tseng, H.C. Yu, G.H. Tzeng, Applied hybrid grey model to forecast seasonal time series, *Technol. Forecast. Soc. Change* 67 (2001) 291–302.
- [30] Y.H. Wang, Y.G. Dang, X.J. Pu, Improved unequal interval grey model and its applications, *J. Syst. Eng. Electron.* 22 (2011) 445–451.
- [31] B. Zeng, C. Li, G. Chen, X.J. Long, Equivalency and unbiasedness of grey prediction models, *J. Syst. Eng. Electron.* 26 (2015) 110–118.
- [32] S.F. Liu, Y. Lin, *Grey Systems Theory and Applications*, Springer-Verlag, Berlin, Heidelberg, 2010.
- [33] B. Zeng, G. Chen, S.F. Liu, A novel interval grey prediction model considering uncertain information, *J. Franklin Inst.* 350 (2013) 3400–3416.
- [34] S. Van Vaerenbergh, M. Lazaro-Gredilla, I. Santamaria, Kernel recursive least-squares tracker for time-varying regression, *IEEE Trans. Neural Netw. Learn. Syst.* 23 (2012) 1313–1326.
- [35] G. Kumar, M. Rai, G.S. Lee, Implementation of cipher block chaining in wireless sensor networks for security enhancement, *Int. J. Secur. Appl.* 6 (2012) 57–72.
- [36] L.M. Jawad, G. Sulong, Chaotic map-embedded Blowfish algorithm for security enhancement of colour image encryption, *Nonlinear Dynam.* 81 (2015) 2079–2093.
- [37] A. Abd El-Sadek, T.A. El-Garf, M.M. Fouad, Speech encryption applying a modified Blowfish algorithm, in: *Proceedings of the 2nd International Conference on Engineering and Technology*, Cairo, Egypt, 2014, pp. 1–6.
- [38] J.C. Principe, B. Chen, Universal approximation with convex optimization: gimmick or reality? *IEEE Comput. Intell. Mag.* 10 (2015) 68–77.
- [39] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002.
- [40] LS-SVMLab Toolbox. Available: <http://www.esat.kuleuven.be/sista/lssvmlab/> (accessed June, 2015).
- [41] S. Madden, Intel Berkeley research lab data. Available: <http://db.csail.mit.edu/labdata/labdata.html> (accessed June, 2015).



Dandan Zhang is currently working toward the Master's degree at the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. Her research interests include cyber-physical systems and machine learning.



Laurence T. Yang received the Ph.D. degree in computer science from the University of Victoria, Canada. He currently works as a Professor in the Department of Computer Science, St. Francis Xavier University, Canada. His current research interests include parallel and distributed computing, and embedded and ubiquitous/pervasive computing. He has published many papers in various refereed journals and conference proceedings in these areas. His research is supported by the National Sciences and Engineering Research Council and the Canada Foundation for Innovation. He has been involved actively in conferences and workshops as a program/general/steering conference chair and numerous conferences and workshops as a program committee member. In addition, he is the editor-in-chief of several international journals. He is serving as an editor for many international journals. He has been acting as an author/co-author or an editor/co-editor of many books from Kluwer, Springer, Nova Science, American Scientific Publishers and John Wiley & Sons.



Ji Liu is currently working toward the Master's degree at the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His research interests include machine learning and computational intelligence.



Xiaohui Chang is currently working toward the Master's degree at the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. Her research interests include machine learning and computational intelligence.



Huansheng Ning received the Ph.D. degree from Beihang University, China, in 2001. He currently works as a Professor in the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research interests include internet of things, cyber-physical systems, aviation security, electromagnetic sensing, and computing. He has published more than 50 papers in journals, international conferences/workshops.



Xiong Luo received the Ph.D. degree from Central South University, China, in 2004. From 2005 to 2006, he was with the Department of Computer Science and Technology, Tsinghua University, China, as a Postdoctoral Fellow. From 2012 to 2013, he was with the School of Electrical, Computer and Energy Engineering, Arizona State University, USA, as a Visiting Scholar. He currently works as an Associate Professor in the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research interests include machine learning, internet of things, cyber-physical systems, and computational intelligence.