# An Interaction Prediction Model of Monitoring Node Based on Observational Learning

Haiyan Chen, Bo Sun, and Jiandong Wang

***Abstract*—Data acquisition anomalies often occur in remote monitoring. In this paper, a software solution is presented to discover the abnormal monitoring node and predict the monitored data, rather than using hardware maintenance. Firstly, by analyzing the distribution characteristics of the monitoring data from each node, the highly correlated nodes of the abnormal node are selected. Then, an integrated BP neural network is applied to build an observational learning model, which can give interactive predictions for the abnormal node. To solve the under-fitting problem caused by small samples and improve the generalization performance of the model, we propose a new observational learning algorithm, in which the weights are calculated using the mean squared error (MSE) of learners on test set. Experiments conducted on the airport noise data set show that the proposed model has satisfying predictive ability, and the improved observational learning algorithm is more stable and effective than the traditional observational learning algorithm.**

***Index Terms*—Interaction prediction, observational learning, integrated BP neural network, noise monitoring.**

## I. Introduction

Internet of Things (IOT) technology is widely used in environmental monitoring. For example, we install noise monitoring nodes in an airport and its surrounding area to capture the noise produced by aircrafts, and then we analyze the data to provide a decision-making basis for the relevant sector to control the airport noise pollution problem. This noise monitoring system is generally distributed in a large area, and the nodes are arranged at important positions that needed to be monitored. However, due to the component damage, equipment aging or other problems, data acquisition exceptions, e.g., not sending data or sending abnormal data, often occur to the nodes. Monitoring data from an abnormal node can not reflect the real noise situation of the monitored area. Therefore, how to predict the noise situation of an area accurately and timely when its monitoring node fails to give correct data becomes a problem deserving of study. Compared to the equipment maintenance method, the software method implemented by analyzing the historical data and learning a prediction model can solve the problem more conveniently, and can correct or make up the monitoring data before the node is fixed or replaced. Up to now, few researches have studied this issue. In references [1], [2], neural network is used to simulate and predict the airport noise monitored by monitoring nodes and is proved to have good performance.

After the preliminary analysis and comparison of the historical monitoring data from each node, we can conclude that there is some correlation between the nodes, i.e., the monitoring data from two adjacent nodes is often close and has the similar trend. Thus, we can give the estimates of the monitoring data perceived by the abnormal node by utilizing its highly correlated nodes, which is a kind of interaction prediction.

Interaction prediction is a paradigm that predicts a state by using the correlation of all parts of a system. This prediction paradigm is first proposed by Takahara in 1965 to solve the dynamic optimization problem of complex systems [3], and then, it is widely used in automatic control field to give hierarchical prediction and control of the production process of large-lot producers [4]. Now, interaction prediction has been applied in more fields, i.e., intrusion detection [5], protein structure prediction [6], and deformation prediction of buildings [7], etc.

Based on observational learning theory[8], we present an interaction prediction model for monitoring nodes, which is constructed by an integrated BP neural network, to avoid the over fitting problem on small data sets and the influence of the initial weights of neural network nodes. During the training of the model, an improved observational learning algorithm is designed to enhance the generalization and prediction accuracy of the model.

## II. Observational Learning Algorithm

Observational learning algorithm (OLA) is proposed by Jang Min in 1999 [8], [9]. It is first applied in artificial neural networks as an ensemble learning technique. The training process of OLA is shown in Fig. 1.

OLA generally includes two steps: the training step (T-step) and the observation step (O-step). In T-step, many BP neural networks, i.e., the learners are trained. In O-step, each learner observes the learning results of other learners and generates virtual data set for the next training. The *'-i'* neural networks in the O-step means the set of all BP neural networks excluding the *i*th one $f_i$. That is, each learner should learn from the other learners. These two steps are executed alternately until the predefined maximum training times is reached. OLA believes that if a learner trained by the direct

Manuscript received June 15, 2014; revised October 10, 2014. This work was supported in part by the Natural Science Foundation of China under Project No. 61139002, the High Technology Research, and the Open Foundation of Information Technology Research Base of Civil Aviation Administration of China under Project No. CAAC-ITRB-201203, the Fundamental Research Funds for the Central Universities (NO.NZ2013306, 3122013P013) and the Postdoctoral Funding Scheme of Jiangsu Province (NO.1301013A).

The authors are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, CO 210016 China (e-mail: chenhaiyan@nuaa.edu.cn, sunbo87@126.com, aics@nuaa.edu.cn).

experience gained from the training set can not have good performance, it can get indirect experience by observing other learners. With the repetition of training and observation, the learners and their integrated model will gradually get better prediction accuracy [10]. Based on the work of Jang M, scholars have carried out further researches on OLA to verify its effectiveness [11]-[14].

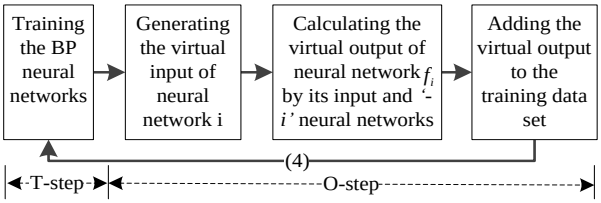



Fig. 1. OLA Working mechanism.

## III. Distribution Based Selection of Correlated Nodes

There are many ways to measure the correlation of nodes. The most intuitive way is to measure the correlation using the distance between nodes. Generally, the smaller the distance between two nodes, the closer the monitoring data that they will capture and the higher degree of correlation they will have. However, the distance based method is only suitable for the nodes-intensive monitoring systems, and it is not applicable for those systems with nodes in sparse distribution or with building interference. A more reliable way is to measure the correlation between two nodes by analyzing their historical monitoring data. For example, we can get two kinds of noise data from the noise monitoring system of Beijing Capital International Airport. One is the daily aviation noise evaluation data, which is the average value of the noise generated in one day, and the other is the time-series noise data, which is the real time noise collected every 0.5 second. Since the daily noise data is much more stable and has more significant statistical characteristics than the real time noise data, we choose the daily noise data from all the 16 nodes in the airport noise monitoring system for study. For each node, there are 730 daily aircraft noise data items in two years, and we fit each node using a normal distribution. The projection of the fitting result of one node is shown in Fig. 2.

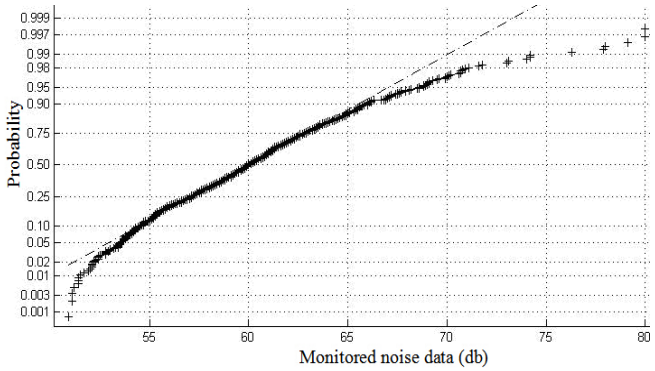



Fig. 2. Projection of normal distribution fitting of the monitored noise data.

It can be seen from Fig. 2 that the vast majority of the samples follow some kind of normal distribution, and a few samples fall outside. This means that these samples do not follow a single distribution, or they may follow a mixture distribution. Therefore, we use Kolmogorov-Smirnov Test to test the mixed normal distribution hypothesis of the samples, and the results are shown in Table I, where $h$=0 means the hypothesis is supported by these samples [15].

TABLE I: Kolmogorov-Smirnov Test Result

| Order | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $h$ | 1 | 0 | 1 | 1 |

The results in Table I show that the 2-order mixed normal distribution can fit these samples best. The fitting result is shown in Fig. 3. The two solid lines present two normal distributions and the dashed line presents their mixture distribution. Most of the samples obey one normal distribution and few obey the other one.

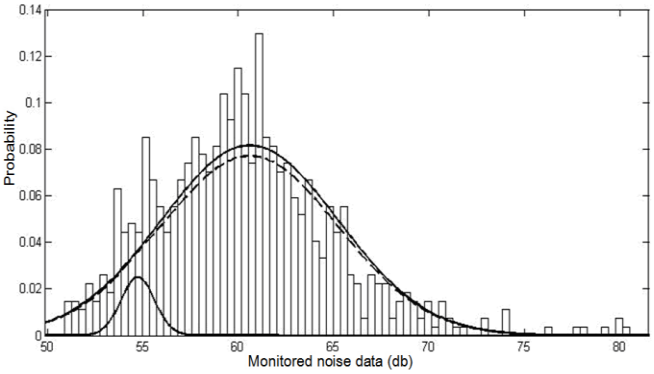



Fig. 3. 2-order mixture normal distribution fitting of the monitored noise data.

Assume the daily noise monitoring data of node $Y$ in the last $n$ days is denoted as $Y=(y_1,y_2,...,y_n)$. Its 2-order normal mixture distribution can be expressed as follows.

$$Y \sim P(\mu,\sigma^2)=\alpha_1\times N_1(\mu_1,{\sigma_1}^2)+\alpha_2\times N_2(\mu_2,{\sigma_2}^2) \quad (1)$$

where $N_1(\mu_1,{\sigma_1}^2)$ and $N_2(\mu_2,{\sigma_2}^2)$ are two normal distributions, $\alpha_1$ and $\alpha_2$ are their weights. The mixture distribution $P(\mu,\sigma^2)$ can be calculated by the following equations.

$$Y \sim P(\mu,\sigma^2): \begin{cases} \mu=\alpha_1\times\mu_1+\alpha_2\times\mu_2 \\ \sigma=\alpha_1\times\sigma_1+\alpha_2\times\sigma_2 \end{cases} \quad (2)$$

By setting $\Delta\mu$ and $\Delta\sigma$, we can get the selection range $P_\Delta(\mu\pm\Delta\mu,(\sigma\pm\Delta\sigma)^2)$. Nodes within the range will be selected as the correlated nodes of node $Y$.

## IV. Interaction Prediction Model Based on MSE-OLA

### A. Construction of the BP Neural Networks

In this paper, we use integrated BP neural network to construct the observational learning model to give interaction prediction. After the selection of correlated nodes, we can get

the initial training set $D=\{(\vec{x}_k, y_k)\,|\,k=1\ldots n\}$ and test set $T=\{(\vec{x}_k, y_k)\,|\,k=1\ldots m\}$. $\vec{x}_k=(x_k^1, x_k^2, \ldots, x_k^{r'})$ is the daily noise data set of the $r'$ correlated nodes, and $y_k$ is the daily noise data of node $Y$.

The $L$ BP neural networks used in this paper have 3 layers: $r'$ nodes in the input layer, 1 node in the output layer, $2\times r'$ nodes in the hidden layer, and they construct an integrated model $F=[f_1, f_2, ..., f_L]$. Bootstrap is used to generate the training sets $D_F=[D_1, D_2, ..., D_L]$ for all the neural network learners.

### B. MSE-OLA

In order to better take advantage of observational learning, and to enhance the generalization ability of the model when dealing with small data set, we propose a MSE-OLA, in which the weights of learners are calculated using their mean square errors (MSE) on the test set. The process of MSE-OLA is as follows:

**Input:** *initialize BP neural networks* $F=[f_1, f_2, ..., f_L]$;
*initialize the training set* $D_F=[D_1, D_2, ..., D_L]$;
*the maximum training times is* $G$;
*the virtual data set* $V_{-i}^0$ *is empty.*

**Output:** *the interaction prediction model.*

*MSE-OLA( )*
*{*
*For* $(t=0; t\le G; t++)$
*{*
*Train neural networks (learners) on the training set* $D_i \cup V_{-i}^t$;
*Generate virtual input data set* $v_i^t$ *for each learner;*
*Calculate MSE of each learner on test set* $T$;
*Calculate the weight* $\beta_{j_i}^t$ *of each learner used in '-i' integrated neural network*
*Generate the virtual training set* $V_{-i}^{t+1}$ *for each learner based on* $\beta_{j_i}^t$.
*}*
*After* $G$ *times training, for each learner* $f_i^G$
*{*
*Calculate MSE of each learner on the test set* $T$;
*Calculate weight* $\alpha_i$ *of each learner used in the final integrated model.*
*}*
*Output the final integrated decision model* $\overline{f}^G=\sum_{i=1}^{L}\alpha_i f_i^G$.
*}*

How to generate the virtual input data set and calculate the weights in *'-i'* integrated model and the weights in final integrated model are discussed and clearly defined in the following three subsections.

### C. Generation of the Virtual Input Data Set

Generation of the virtual input data set is an important step of OLA. The data sets directly affect the performance of the learners. The training result of OLA is not sensitive to the white noise of the data set. It can always get good training effects if the white noise is not extremely large or small. Therefore, we can set the variance of the white noise of virtual input data set according to experience [9]. Here, we use the following equation to generate the virtual input data set for each BP neural network.

$$v_i^t=\{\vec{x}_k+\vec{z}_k\,|\,\vec{x}_k\in D_i,\ \vec{z}_k\sim N(0, 1/n), k=1\ldots n\} \tag{3}$$

where, $\vec{x}_k$ belongs to the initial training set $D_i$ of neural network $f_i$. $\vec{z}_k$ is the white noise with its mean equals to 0 and variance equals to $1/n$, $n$ is the size of $D_i$. If $n$ is small, the variance is large, and it can generate data set with high diversity to avoid over-fitting. If $n$ is large, the variance becomes small, and it can control the fitting deviation.

### D. Weights of Learners in the '-i' Integrated Model

We use $\beta_{i_j}$ to represent the importance of neural network $f_j$ to the '-*i*' integrated BP neural network. In traditional OLA, $\beta_{i_j}=1/L-1$ which means that all the learners in '-*i*' neural network set have the same effect on the '-*i*' integrated model, or learner $f_i$ learns equally from other learners. Although it is an easy way to calculate the weights, it can not reflect the actual difference of the learners. In order to measure the importance of the learners more accurately, here, we use the MSEs of each learner on the test set as an evaluation criterion. Thus, to learner $f_j$, after its $t$ times training, its weight $\beta_{i_j}^t$ for '-*i*' integrated model can be calculated by the following equations:

$$\sigma_j^t=\frac{1}{m}\sum_{k=1}^{m}\left(y_k-f_j^t\left(\vec{x}_k\right)\right)^2, \left(\vec{x}_k, y_k\right)\in T \tag{4}$$

$$\beta_{i_j}^t=\frac{1/\sigma_j^t}{\sum_{u=1,u\neq i}^{L}1/\sigma_u^t} \tag{5}$$

where, $f_j^t$ is the learner obtained from the $t$ th training of BP neural network $f_j$, and $\sigma_j^t$ is its MSE on test set $T$. By calculating the weights of learners in all trainings, we can get a weights matrix as follows:

$$\beta^t=\begin{bmatrix} 0 & \beta_{1_2}^t & \ldots & \beta_{1_L}^t \\ \beta_{2_1}^t & 0 & \ldots & \beta_{2_L}^t \\ \ldots & \ldots & \ldots & \ldots \\ \beta_{L_1}^t & \beta_{L_2}^t & \ldots & 0 \end{bmatrix} \tag{6}$$

MSE-OLA can be understood as that each learner learns more from those learners with better performance, and learns less from the learners with lower ability. It is clearly more

reasonable than the traditional OLA.

Thus, for learner $f_i$, its output virtual data set can be generated by the following equation.

$$\overline{f}_{-i}^{t} = \sum_{j=1, j\neq i}^{L} \beta_{i_j}^{t} f_j^{t},\ j = 1, ..., L \tag{7}$$

So far, the virtual training set for the next training can be expressed as:

$$V_{-i}^{t+1} = \left\{ \left( v_i^t, \overline{f}_{-i}^{t}\left(v_i^t\right) \right) \right\} \tag{8}$$

### E. Weights of Learners in the Final Integrated Model

After $G$ times training, we can get the final integrated decision model, which is expressed as:

$$\overline{f}^{G} = \sum_{i=1}^{L} \alpha_i f_i^{G} \tag{9}$$

where, $f_i^G$ is the learner obtained from the $G$ th training of the BP neural network $f_i$, and $\alpha_i$ is its weight in the final integrated decision model. In traditional OLA, $\alpha_i = 1/L$, which means all the learners have the same importance to the final decision model [16]. Refer to the same problem discussed in last subsection, the importance of learners should be measured by their generalization ability, in this case, the stronger learner may play more important role in the final integrated model. Therefore, we use the same way to calculate $\alpha_i$ as calculating $\beta_{i_j}^t$. The expression is as follows:

$$\alpha_i = \frac{1/\sigma_i}{\sum_{u=1}^{L} 1/\sigma_u} \tag{10}$$

where, $\sigma_i$ is calculated by equation (4).

## V. Experiments and Analysis

The data used in the experiments is the monitoring data acquired by the 16 nodes in the noise monitoring system of Beijing Capital International Airport. We select a node as the abnormal one, the others as the candidate correlated nodes. The data set contains 730 samples of two years. Each sample is a vector of the daily noise monitored by the 16 nodes. We choose 200 samples as the test set and the rest 530 samples as the original training set.

### A. Selecting the Correlated Nodes

For all monitoring data of the 16 nodes, calculate their 2-order mixture normal distribution model, the results are listed in Table II.

Assume node16 to be the abnormal node, the correlated nodes selection range is $P_{\Delta}(58.1042 \pm 1, (2.2642 \pm 1)^2)$. Thus, node1, node2, node6, node9 and node10 are selected as the correlated nodes of node16

TABLE II: 2-Order Mixed Normal Distribution Fitting to the Monitoring Data of 16 Nodes

| Node | Normal distribution1 | | | Normal distribution2 | | |
|---|---|---|---|---|---|---|
| | $\alpha_1$ | $\mu_1$ | $\sigma_1$ | $\alpha_2$ | $\mu_2$ | $\sigma_2$ |
| 1 | 0.9236 | 57.3748 | 1.9919 | 0.0764 | 63.7310 | 7.3519 |
| 2 | 0.9221 | 57.2634 | 1.7175 | 0.0779 | 60.1878 | 7.9745 |
| 3 | 0.9501 | 61.4379 | 1.9378 | 0.0410 | 66.0517 | 6.8793 |
| 4 | 0.9376 | 61.2931 | 2.2707 | 0.0624 | 66.3599 | 5.7458 |
| 5 | 0.9440 | 61.2472 | 1.9882 | 0.0560 | 66.4542 | 6.3427 |
| 6 | 0.9744 | 57.6206 | 2.3217 | 0.0256 | 67.5025 | 4.1839 |
| 7 | 0.9689 | 55.7642 | 2.7723 | 0.0311 | 64.5710 | 7.7502 |
| 8 | 0.9825 | 61.1283 | 3.4118 | 0.0175 | 51.1112 | 0.8571 |
| 9 | 0.9494 | 58.2326 | 2.7703 | 0.0506 | 55.2089 | 0.2699 |
| 10 | 0.9190 | 58.7978 | 2.2431 | 0.0810 | 65.6002 | 7.1396 |
| 11 | 0.9310 | 56.4443 | 1.8136 | 0.0690 | 63.8573 | 6.8186 |
| 12 | 0.9920 | 60.8327 | 1.6223 | 0.0080 | 72.4153 | 2.0886 |
| 13 | 0.9567 | 59.1362 | 1.5623 | 0.0433 | 66.2375 | 7.7116 |
| 14 | 0.9214 | 62.3949 | 1.3880 | 0.0785 | 67.2220 | 7.8497 |
| 15 | 0.9389 | 56.6205 | 1.6376 | 0.0611 | 62.8002 | 7.8177 |
| 16 | 0.9342 | 57.7525 | 1.9257 | 0.0657 | 63.1047 | 7.0765 |

### B. Training the Interaction Prediction Model

Since 5 correlated nodes are selected, the input dimension of the training samples is 5, and the output dimension is 1. Firstly, we generate 5 data sets by Bootstrapping on the original data set to train 5 BP neural networks, i.e., 5 learners. Then, 21 times training and observational learning are carried out according to the MSE-OAL and the parameters calculation methods discussed in Section IV-B. The first training is on the original training set and the rest 20 times on virtual training set. Table III shows the MSEs of the 5 learners and the integrated model of their 21 trainings.

TABLE III: MSEs of MSE-OLA

| Times | BP1 | BP2 | BP3 | BP4 | BP5 | Integrated Model |
|---|---|---|---|---|---|---|
| 0 | 0.1030 | 0.1380 | 0.0952 | 0.0869 | 0.0869 | 0.0814 |
| 1 | 0.1030 | 0.0853 | 0.0568 | 0.0869 | 0.1000 | 0.0775 |
| 2 | 0.0719 | 0.0853 | 0.0741 | 0.0678 | 0.0693 | 0.0701 |
| 3 | 0.0719 | 0.0853 | 0.0723 | 0.0678 | 0.0693 | 0.0681 |
| 4 | 0.0762 | 0.0816 | 0.0723 | 0.0678 | 0.0693 | 0.0689 |
| 5 | 0.0762 | 0.0615 | 0.0675 | 0.0678 | 0.0693 | 0.0668 |
| 6 | 0.0747 | 0.0615 | 0.0675 | 0.0678 | 0.0693 | 0.0652 |
| 7 | 0.0747 | 0.0615 | 0.0675 | 0.0678 | 0.0710 | 0.0657 |
| 8 | 0.0747 | 0.0615 | 0.0675 | 0.0678 | 0.0548 | 0.0623 |
| 9 | 0.0749 | 0.0868 | 0.0675 | 0.0678 | 0.0548 | 0.0627 |
| 10 | 0.0749 | 0.0868 | 0.0675 | 0.0678 | 0.0548 | 0.0620 |
| 11 | 0.0749 | 0.0868 | 0.0675 | 0.0678 | 0.0549 | 0.0621 |
| 12 | 0.0749 | 0.0858 | 0.0675 | 0.0553 | 0.0549 | 0.0597 |
| 13 | 0.0749 | 0.0858 | 0.0675 | 0.0553 | 0.0549 | 0.0589 |
| 14 | 0.0778 | 0.0858 | 0.0675 | 0.0553 | 0.0549 | 0.0592 |
| 15 | 0.0778 | 0.0858 | 0.0675 | 0.0553 | 0.0549 | 0.0591 |
| 16 | 0.0778 | 0.0858 | 0.0675 | 0.0553 | 0.0549 | 0.0591 |
| 17 | 0.0778 | 0.0858 | 0.0675 | 0.0550 | 0.0549 | 0.0581 |
| 18 | 0.0782 | 0.0858 | 0.0693 | 0.0550 | 0.0549 | 0.0581 |
| 19 | 0.0756 | 0.0826 | 0.0693 | 0.0550 | 0.0549 | 0.0579 |
| 20 | 0.0756 | 0.0826 | 0.0693 | 0.0550 | 0.0549 | 0.0580 |

As can be seen from Table III, with the adding of the virtual training data and the repetition of the training, MSEs of the 5 learners fluctuate, however, the MSE of their integrated model keeps declining, which means the prediction accuracy of the model is improved. Besides, we

can see that the MSE of the integrated model decreases rapidly after the first few trainings and then becomes stable.

An extra experiment is carried out to validate the advantage of the proposed MSE-OLA when compared with the traditional OLA, and all the weights are calculated by the average method. Table IV shows the MSEs of the 5 learners and the integrated model of traditional OLA in their 21 trainings and Fig. 4 demonstrates the MSE trends of the two OLAs.

TABLE IV: MSEs OF TRADITIONAL OLA

| Times | BP1 | BP2 | BP3 | BP4 | BP5 | Integrated Model |
|---|---|---|---|---|---|---|
| 0 | 0.1030 | 0.1380 | 0.0952 | 0.0869 | 0.0869 | 0.0854 |
| 1 | 0.0974 | 0.0529 | 0.0922 | 0.1110 | 0.1101 | 0.0853 |
| 2 | 0.0974 | 0.0534 | 0.0747 | 0.1110 | 0.1101 | 0.0821 |
| 3 | 0.0873 | 0.0539 | 0.0747 | 0.1285 | 0.0836 | 0.0787 |
| 4 | 0.0873 | 0.0556 | 0.0689 | 0.1285 | 0.0836 | 0.0790 |
| 5 | 0.0873 | 0.0556 | 0.0689 | 0.0976 | 0.0850 | 0.0750 |
| 6 | 0.0873 | 0.0556 | 0.0689 | 0.0976 | 0.0850 | 0.0750 |
| 7 | 0.0708 | 0.0556 | 0.0689 | 0.0976 | 0.0850 | 0.0719 |
| 8 | 0.0693 | 0.1048 | 0.0689 | 0.0976 | 0.0850 | 0.0729 |
| 9 | 0.0693 | 0.1048 | 0.0689 | 0.0976 | 0.0850 | 0.0729 |
| 10 | 0.0693 | 0.1048 | 0.0689 | 0.0976 | 0.0850 | 0.0729 |
| 11 | 0.0693 | 0.1048 | 0.0689 | 0.0961 | 0.1381 | 0.0789 |
| 12 | 0.0693 | 0.0593 | 0.0689 | 0.0961 | 0.1305 | 0.0773 |
| 13 | 0.0693 | 0.0593 | 0.0689 | 0.0961 | 0.1305 | 0.0773 |
| 14 | 0.0693 | 0.0593 | 0.0689 | 0.0961 | 0.1359 | 0.0777 |
| 15 | 0.0658 | 0.0593 | 0.0689 | 0.0924 | 0.1359 | 0.0765 |
| 16 | 0.0658 | 0.0593 | 0.0689 | 0.0924 | 0.0959 | 0.0716 |
| 17 | 0.0658 | 0.0593 | 0.0689 | 0.0924 | 0.0959 | 0.0716 |
| 18 | 0.0658 | 0.0593 | 0.0683 | 0.0924 | 0.0959 | 0.0710 |
| 19 | 0.0658 | 0.0593 | 0.0683 | 0.0899 | 0.0959 | 0.0707 |
| 20 | 0.0658 | 0.0593 | 0.0721 | 0.0899 | 0.0959 | 0.0713 |

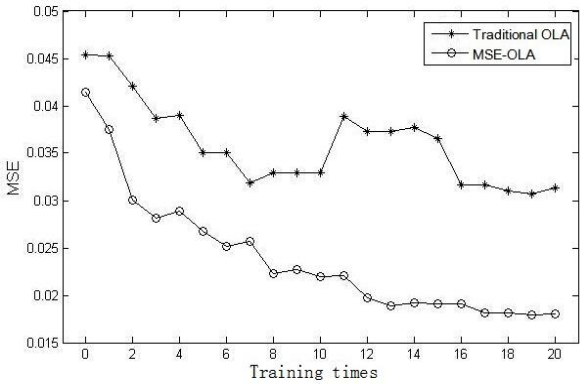



Fig. 4. MSE trends comparison of the two OLAs.

It can be seen from Table IV that the integrated model trained by traditional OLA also has better performance than the 5 single learners, but the descent speed and the final value of its MSE is inferior to the integrated model trained by MES-OLA, what is also illustrated by Fig. 4. Therefore, we can conclude that the proposed MSE-OLA can get a better integrated model in less training time, which is a great improvement over the traditional OLA.

### C. Performance Analysis of the Interaction Prediction

In Section V-B, we train the interaction prediction model for the abnormal node 16. The predictions on test set and its comparison with the monitored data are shown in Fig. 5.The percentage errors of the interaction prediction model are also calculated and illustrated in Fig.6.

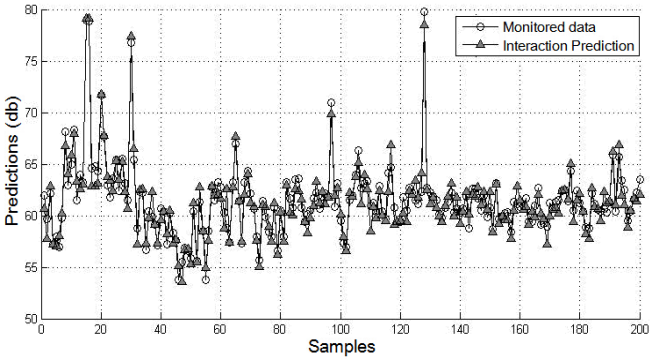



Fig. 5. Comparison of interaction predictions and monitored noise data.

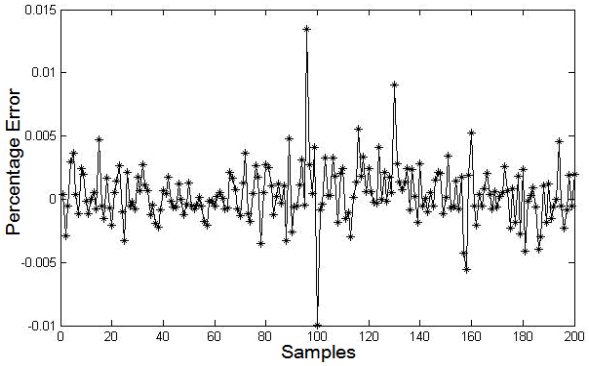



Fig. 6. Percentage error of interaction prediction.

As can be seen from Fig. 6, the percentage errors of the interaction prediction model of monitoring node based on MSE-OLA are between [-0.03, 0.03], and most of the errors fall in [-0.01, 0.01]. The prediction accuracy is satisfying.

## VI. CONCLUSIONS

In this paper, we present an interaction prediction model based on OLA for the monitoring nodes that may fail to give correct monitoring data. The normal distribution characteristics of the historical monitoring data from all the nodes are analyzed, and then the highly correlated nodes are selected to structure the original training set, on which the integrated BP neural network is trained by using MSE-OLA. Experiments conducted on the test set show that the proposed MSE-OLA is more efficient than the traditional OLA, and the interaction prediction model has very high prediction accuracy.

To the airport noise monitoring system, the work presented in this paper can help to make a daily noise interaction prediction for the abnormal nodes. However, to the time-series noise data monitored by the nodes, because of its continuous dynamic changes, it is inappropriate to select the correlated nodes based on distribution characteristics. The further works will be focused on the interaction prediction of time-series noise data.

## REFERENCES

[1] Y. Yang, C. Hinde, and D. Gillingwater, "Airport noise simulation using neural networks," *International Joint Conference on Neural Networks*, pp. 1917-1923, Hong Kong: IEEE Press, 2008.

[2] J. Du, Y. Zhang, and T. Xu, "Airport noise prediction model based on BP neural network," *Computer Engineering and Applications*, vol. 49, pp. 236-239, September 2013.

[3] M. Mesarovic, J. Pearson, and Y. Takahara, "A multilevel structure for a class of linear dynamic optimization problems," in *Proc. 1965, Joint Automatic Control Conf.*, 1965, pp. 93-99.
[4] G, Gu, "Improved interaction prediction of discrete dynamic system," *Journal of Xiamen University*, vol. 40, pp. 848-851, April 2001.
[5] B. Morin, L. M é and H. Debar *et al*., "M2D2: A formal data model for IDS alert correlation," in *Proc. Recent Advances in Intrusion Detection Conf.*, 2002, pp. 115-137.
[6] Y. Zhao and G. Karypis, "Prediction of contact maps using support vector machines," *International Journal on Artificial Intelligence Tools*, vol. 14, pp. 849-865, May 2005.
[7] X. Zhou, J. Tian, and C. Gu, "A gray interaction prediction and its application in deformation prediction," *Railway Investigation and Surveying*, vol. 35, pp. 28-30, January 2009.
[8] M. Jang and S. Cho, "Ensemble learning using observational learning theory," *International Joint Conference on Neural Networks*, pp. 1287-1292, vol. 6, USA: IEEE Press, 1999.
[9] M. Jang and S. Cho, "Observational learning algorithm for an ensemble of neural networks," *Pattern Analysis & Applications*, vol. 52, pp. 154-167, February 2002.
[10] S. Wang, "Class imbalance learning," Ph.D. dissertation, Dept. Comp. Sci., Birmingham Univ., Birmingham, 2009.
[11] F. Yu, L. Yang, and Z. Qin, "Classifiers combination with observational learning," *International Journal of Computer Science and Network Security*, vol. 6, pp. 53-56, May 2006.
[12] P. Wong, M. Jang, and S. Cho *et al*., "Multiple permeability predictions using an observational learning algorithm," *Computers & Geosciences*, vol. 26, pp. 907-913, August 2000.
[13] H. Shin, H. Lee, and S. Cho, "Observational learning with modular networks," in *Proc. International Conference on Intelligent Data Engineering and Automated Learning*, pp. 126-132, Hong Kong: Springer-Verlag Press, 2000.
[14] Z. Lu and T. Xu, "A new method to predict probability distribution based on heterogeneous ensemble learning," *International Journal of Advancements in Computing Technology*, vol. 4, pp. 17-25, July 2012.
[15] H. Chen, J. Wang, and T. Xu, "Modeling of flight delay state-space model based on genetic EM algorithm," *Transactions of Nanjing University of Aeronautics & Astronautics*, vol. 28, pp. 276-281, May 2011.
[16] M. Hagiwara, "Removal of hidden units and weights for back propagation networks," in *Proc. 1993 IJCNN Conf.*, 1993, pp. 351-354.



**Haiyan Chen** was born in Changzhou, P.R. China, on April 4, 1979. She received her B.S. degree in computer science and technology from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, P.R. China, in 2002, her M.S. degree in computer application from NUAA in 2005, and her doctor's degree in computer application from NUAA in 2012. Her major field of study is data mining.

From 2005 to 2006, she worked as a teaching assistant at College of Computer Science and Technology of NUAA and as a lecture since 2007. In Decemeber 2012, she entered the post doctoral research station of Software Engineering in NUAA worked as a postdoctor.

Dr. Chen's research interests are data mining algorithm and its applications in civil aviation informatization.



**Bo Sun** was born in Taian, P.R. China, on July 6, 1987. He received his B.S. degree in computer science and technology from the Liaocheng University, Liaocheng, P.R. China, in 2009, his M.S. degree in computer application from Jiangsu University, Zhenjiang, P.R. China, in 2012. He is now pursuing his Ph.D degree in college of computer sicence and technology, Nanjing University of Aeronautics and Astronautics. His main research interests include data mining and machine learning.



**Jiandong Wang** was born in Shuyang, P.R. China, in 1945. He received the M.S. degree in electrical engineering from Shanghai Jiao Tong University.

He was a visiting scholar at the University of Ottawa, Canada, from 1990 to 1991. He is currently a professor and doctoral students adviser in the College of Computer Science and Technology at NUAA. His main research interests include artificial intelligence, data mining and information security.