788. A new density estimation neural network to detect abnormal condition in streaming data

Qing Zhang[1], **Jing Wang**[2], **Guanghua Xu**[3]
School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China
E-mail: [1]*zhangq@mail.xjtu.edu.cn*, [2]*wangpele@gmail.com*, [3]*xugh@mail.xjtu.edu.cn*

(Received 17 February 2012; accepted 14 May 2012)

Abstract. Along with the development of monitoring technologies, numerous measured data pour into monitoring system and form the high-volume and open-ended data stream. Usually, abnormal condition of monitored system can be characterized by the density variation of measured data stream. However, traditional density estimation methods can not dynamically track density variation of data stream due to the limitation of processing time and computation memory. In this paper, we propose a new density estimation neural network to continuously estimate the density of streaming data in a time-based sliding window. The network has a feedforward structure composed of discretization, input and summation layer. In the discretization layer, value range of data stream is discretized to network nodes with equal intervals. Measured data in the predefined time window are pushed into input layer and updated with the window sliding. In summation layer, the activation results between input neurons and discretization neurons are summed up and multiplied by a weight factor. The network outputs the kernel density estimators of sliding segment in data stream and achieves a one-pass estimation algorithm consuming constant computation memory. By subnet separation and local activation, computation load of the network is significantly reduced to catch up the pace of data stream. The nonlinear statistics, quantile and entropy, which can be consecutively figured out with the density estimators output by the density estimation neural network, are calculated as condition indictors to track the density variation of data stream. The proposed method is evaluated by a simulated data stream consisting of two mixing distribution data sets and a pressure data stream measured from a centrifugal compressor respectively. Results show that the underlying anomalies are successfully detected.

Keywords: kernel density estimation, data streams, abnormal condition detection, neural network.

1. Introduction

The condition monitoring is of great importance in terms of system maintenance and process automation. To evaluate the overall performance of installed equipment or current health of key components, the data indicating running state is continuously measured by sensors. Recent advances of monitoring hardware and software have enabled the capture of real-time data in a wide range of fields. As a result, numerous measured data, which are fast, continuous, mutable and ordered, pour into monitoring system. When processing data stream, a data item can only be accessed at its arriving time or kept for a short period of time. If warning alarms or safety actions are triggered only by comparing the current data with the preset thresholds, there is no difficulty to deal with the stream. However, in most cases, the measured data are contaminated by the unexpected noises generated from various interfering source [1] and one measured value out of context cannot provide valuable information of abnormal condition. To extract the hidden condition characteristic and identify anomaly quickly, an appropriate processing technique for measured data stream is essential.

Due to the existence of noise, the measured data always exhibit random fluctuation whatever state the monitored object is in, i.e. normal or abnormal. The current data item is incompetent to recognize system condition, but statistical characteristic of measured data over a period of time can reveal the underlying system behavior. As a common and useful statistical analysis technique, density estimation gets density function of the given data set. From the density function, we can distinguish the dense or sparse areas in the data set, judge whether one or more data are drawn from the certain distribution and calculate statistics such as quantile and entropy. Recently, a new viewpoint is that knowledge of the data density would allow us to solve whatever problem can be solved on the basis of the data [2]. Thus, our goal is to detect abnormal condition by monitoring the density of measured data stream.

Kernel density estimation (KDE), which describes the distribution pattern of the data set without prior knowledge, is one of the most widely used density estimation methods [3]. Owning to its request for linear memory consumptions and kernel function calculations with data size, the original kernel density estimation can only deal with the static data sets [4-5]. However, once the monitoring systems start, the measured data inflow continuously and no one knows the exact ending time. It is impractical to store each data or process measured data after a long-term accumulation.

To deal with the high-volume and open-ended data stream, some improved kernel density estimation methods have been provided, such as merging the kernels with the minimum accuracy loss to keep the scheduled kernel calculation counts [6] or building a clustering feature structure called CF-tree to replace the kernel function on each data point [7]. These methods can keep up with the changes of the streaming data, but they estimate the overall density distribution in which all data are of equal importance. Unfortunately, the variation of overall density distribution is not sensitive to the system abnormal conditions [8], this means that it can not detect the system exceptions effectively. Our opinion is that density variation generated by latest data should get special concerns to detect abnormal states timely.

In this paper, we present a new Density Estimation Neural Network (DENN) to detect abnormal condition in streaming data. It estimates the kernel density of data stream in a time-based sliding window. Based on the output estimators, statistical parameters indicating density variation are calculated to detect abnormal condition. The highlights of the paper can be summarized as:

1) We construct a feedforward neural network composing of discretization layer, input layer and summation layer to estimate the density of data stream. It is a one-pass algorithm consuming constant computation memory. With subnet separation and local activation, the approximate recursive estimation is achieved in real-time.

2) The nonlinear statistics, quantile and entropy, are chosen as condition indictors. We discuss how to expediently figure out them from the density estimators output by DENN. Simulated and experimental results verify their ability of tracking density variation and identifying system anomaly.

The paper is organized as follows. We briefly describe the fundamental theory and application limitation of kernel density estimation in the next section. In section 3, the structure of DENN is given and the approximate recursive algorithm for streaming data density estimation is proposed. Section 4 and 5 includes the calculation of condition indictors and simulation verification. Section 6 applies the DENN to process pressure signal of centrifugal compressor and detect the surge in its initial stage. Finally, the conclusions are given in section 7.

2. Kernel density estimation

Essentially, each continuous random variable x has a unique probability density function $f(x)$, which is a nonnegative function and integrates to one. It describes the probability of each possible value of x. In real-world applications, neither x nor $f(x)$ is known. We only have n observations of x in form of a data set $x_1, x_2, \cdots, x_n$. The density estimation problem is to construct an estimator $\hat{f}(x)$ of $f(x)$ based on the data set.

© VIBROENGINEERING. JOURNAL OF VIBROENGINEERING. JUNE 2012. VOLUME 14, ISSUE 2. ISSN 1392-8716

As a widely studied nonparametric density estimation method, KDE can approximate arbitrary distribution in probabilistic terms [9]. The estimating function is defined as

$$\hat{f}(x) = \frac{1}{nh}\sum_{j=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1}$$

where h is called the bandwidth and $K(x)$ is kernel function. Figure 1 provides a grasp of the estimating function. A "bump" is placed on each data point in data set, and the sum of all bumps is the density estimator reflecting the overall distribution of all data points. The shape of the bump is determined by the kernel function $K(x)$, which is a unimodal, symmetric and nonnegative function that centers at zero and integrates to one, such as the Gaussian kernel. The width or scope of each bump is determined by the bandwidth h. A large h provides a smooth density estimate but loses detailed distribution information, while a small h generates more accurate estimators but maybe produce some unnecessarily probability components. To minimize the mean integrated squared error of $\hat{f}(x)$, h should converge to zero as n increases but at a slower rate than n, i.e.

$$\lim_{n\to\infty} h = 0, \ \lim_{n\to\infty} nh = \infty \tag{2}$$

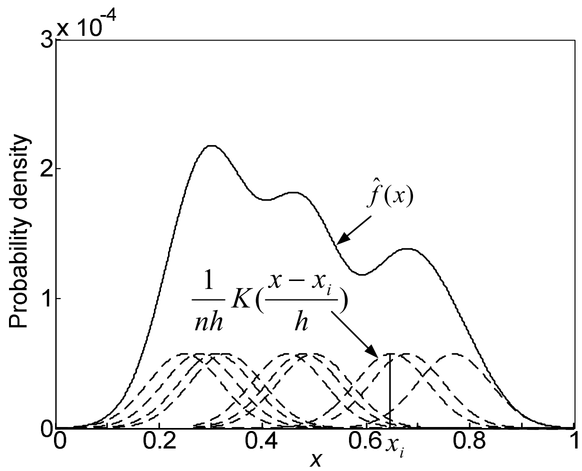

Fig. 1. Kernel density estimation

Kernel density estimation has several advantages that include: no need to assume any pre-specified functional forms for $f(x)$, the estimation is asymptotically unbiased and consistent; inheritance of the kernel function's continuity, differentiability and integrability properties. For the data set with constant probability distribution and medium data sizes, it is a good and practical choice. However, if the original kernel density estimation is applied to data stream, which has a variable distribution and endless data items, it becomes computationally expensive and memory insufficient. Supposed that there are n data points in the data set, we need n distinct bumps i.e. kernel functions to get $\hat{f}(x)$. As a result, all data points should be stored in the memory, and then for a specific value of x, whole data set is scanned to calculate its probability density. Obviously, the memory cost and computational complexity are both in linear with the data size. Though the large volume of data is conducive to approximate the true density, the endlessness of data stream makes it impossible to get all data in advance and keep in the memory. Moreover, if the previous data and later data have different probability distribution, $\hat{f}(x)$ only provides the gathering density information of all data, in which the density components appeared in different moment can not be distinguished.

3. Density estimation neural network

The bumps of kernel density estimation have the same effects with the activation operations of artificial neural networks. Therefore, many questions about density estimation have been solved by the neural networks, such as Probability Neural Networks (PNN) [10], which implements Bayesian classification strategy by a 4-layer feedforward neural network structure. It has been verified that the neural networks have excellent structure and expressiveness in probability density estimation.

3. 1 Structure of density estimation neural network

To track the density variation of data streams, a new neural network architecture called DENN is constructed, as shown in Fig. 2. The DENN consists of three feedforward layers: discretization layer, input layer and summation layer. In discretization layer, the value range of data stream is discretized to network nodes $\{y_1, y_2, \cdots, y_l\}$ with equal intervals. y_1 and y_l are the minimum and maximum possible value in data stream or lower and upper limit of our concerned value range. l controls the amount of density value output by the network. Compared with PNN, the most noticeable feature of DENN is that the input layer is located in the middle of networks instead of first layer. The neurons of input layer included m data points in the streams, which are acquired by a sliding time window with length of m. As the time window slides, new data are inserted into the beginning of the input layer and a corresponding number of old data are removed from the end of the input layer. Therefore, the input layer is a continuously renewed data pool. The activation function of input neurons is just the kernel function. If the Gaussian kernel is chosen and the input neurons in kth sliding step is denoted as $\{x_1^k, x_2^k, \cdots, x_m^k\}$, the activation function is

$$K(y_i, x_j^k) = K(\frac{y_i - x_j^k}{h}) = \frac{1}{\sqrt{2\pi}} \exp - \frac{(y_i - x_j^k)^2}{2h^2} \quad i = 1,2,\cdots,l \quad j = 1,2,\cdots,m \tag{3}$$

Obviously, the activation level is controlled by the distance between the discretization neuron y_i and input neuron x_j^k. It becomes prominent when the distance gets closer and vice versa. In the summation layer, the results of activation function are summarized and then multiplied by a weight factor $1/mh$. As a result, the network produces outputs of

$$\hat{f}(y_i) = \frac{1}{mh} \sum_{j=1}^{m} K(y_i, x_j^k) = \frac{1}{\sqrt{2\pi} mh} \sum_{j=1}^{m} \exp - \frac{(y_i - x_j^k)^2}{2h^2} \quad i = 1,2,\cdots,l \tag{4}$$

Comparing equations (1) and (4), it can easily be observed that the outputs of the network are discrete kernel density estimators.

While the data move into and out of the DENN in sequence, the density of m data points in current time window is output continuously. In this way, the computation memory of density estimation is confined to m in spite of open-ended data volume. Furthermore, a constant

© VIBROENGINEERING. JOURNAL OF VIBROENGINEERING. JUNE 2012. VOLUME 14, ISSUE 2. ISSN 1392-8716

discretization layer is in favor of discovering the underlying density variation in the data stream. As long as the time window length m is small relative to the rate of density variation, the network assures availability of the discrete estimators describing the current distribution. If m is too small, however, this may result in insufficient examples to satisfactorily approximate the real density. Moreover, the computational cost of updating estimators with time window sliding should be economical, especially if data arrive at a rapid rate and the density variations quickly. Therefore, we give an optimization algorithm of the density estimation neural network in the next section.

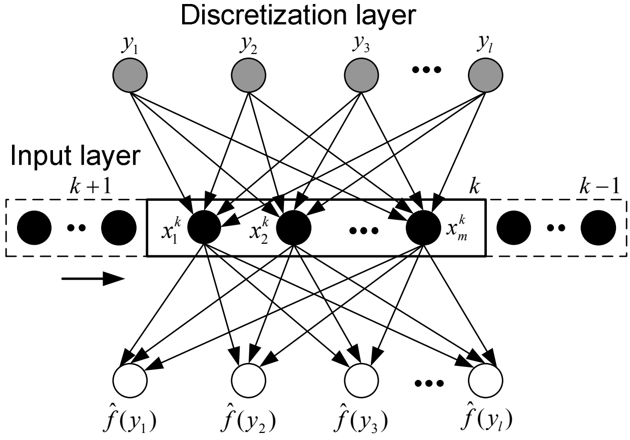

Fig. 2. Structure of DENN

3. 2 Optimization algorithm of density estimation neural network

To get density estimators by DENN, there are $m \times l$ activation operations, or rather kernel operations in each sliding step of time window. In most cases, it is incapable to keep pace with data streaming. And that the decrease of l or m , or both can not be accepted due to the accuracy requirement. In this work, we divide the DENN into partial subnets centered by input neurons. The density of data stream is recursively estimated by just replacing the relative subnets instead of updating the whole network. To process the data with higher streaming speed, an approximate algorithm is applied to reduce computation load further.

While the time window is sliding, the neurons in discretization layer are invariant, whereas the oldest neuron in input layer are replaced by a new neuron and the output density estimators in summation layer are updated along with the input layer. For highlighting the network change, the DENN is divided into m subnets, each of which is composed of one input neuron and all discretization and summation neurons connected with it, as shown in Fig. 3. If one input neuron is pushed out the network, its contribution to network output can be eliminated by deleting the subnet centered by it. Meanwhile, the new input neuron makes its contribution to density estimators by appended a new subnet. In this way, the output estimators in $(k+1)th$ sliding step can be recursively calculated as

$$\hat{f}^{k+1}(y_i) = \hat{f}^{k}(y_i) - K(y_i, x_m^k) + K(y_i, x_1^{k+1}) \quad i = 1,2,\cdots,l \tag{5}$$

With the recursive algorithm, activation operations decrease from $m \times l$ to $2 \times l$ per sliding step of time window.

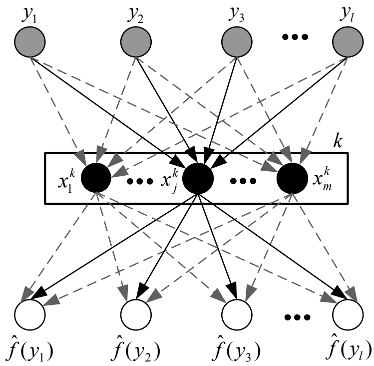

Fig. 3. Subnet of DENN

Because the choice of $K(x)$ is not critical to the accuracy of estimation and the Gaussian kernel is the most popular kernel function [3], we choose the Gaussian function of zero mean and unit variance, which has been written as Eq. (3), as activation function in the following discussion. According to the property of Gaussian function, it is obvious that activation level with the distance between x_j^k and y_i exceeding $3h$ is less 1.11 % than that when the distance equals to zero. That is to say, most activation levels of discretization neurons in subnet are too small to be taken into account except the discretization neurons in the range of $[x_j^k - 3h, x_j^k + 3h]$. If the summation layer just accepts the activation outputs in this range, it only requires about $\lceil 2\times 6h/(y_{i+1} - y_i)\rceil$ activation operations per sliding step of time window and the sum of evaluated errors keeps below 0.26 % relative to activating all discretization neurons, where $\lceil\ \rceil$ is the top integral function. Therefore, we propose an approximate recursive algorithm described as follows. While omitting the old input neuron x_m^k, the discretization neurons in $[x_m^k - 3h, x_m^k + 3h]$ are found out and denoted as $\{y_{a_1}, y_{a_1+1}, \cdots, y_{b_1}\}$, where $b_1 - a_1 = \lceil 6h/(y_{i+1} - y_i)\rceil - 1$, then the density estimators are updated as

$$\begin{aligned} &\hat{f}_1^{k+1}(y_i) = \hat{f}^k(y_i) - K(y_i, x_m^k) \quad i = a_1, a_1+1, \cdots, b_1 \\ &\hat{f}_1^{k+1}(y_i) = \hat{f}^k(y_i) \quad i = 1, \cdots a_1 - 1 \ \text{ and } \ b_1 + 1, \cdots l \end{aligned} \tag{6}$$

Similarly, while adding a new input neuron x_1^{k+1}, the density estimators are updated as

$$\begin{aligned} &\hat{f}_2^{k+1}(y_i) = \hat{f}_1^{k+1}(y_i) + K(y_i, x_1^{k+1}) \quad i = a_2, a_2+1, \cdots, b_2 \\ &\hat{f}_2^{k+1}(y_i) = \hat{f}_1^{k+1}(y_i) \quad i = 1, \cdots a_2 - 1 \ \text{ and } \ b_2 + 1, \cdots l \end{aligned} \tag{7}$$

where $\{y_{a_2}, y_{a_2+1}, \cdots, y_{b_2}\}$ are the discretization neurons in the range of $[x_1^{k+1} - 3h, x_1^{k+1} + 3h]$. By ignoring the discretization neurons in low activation level, the subnet is pruned as one center input neuron and $\lceil 6h/(y_{i+1} - y_i)\rceil$ discretization and summation neurons close to it. Since the discretization neurons are equal intervals and in ascending sort, the operations of locating a_1, b_1, a_2 and b_2 can be easily executed, e.g. $a_1 = \lfloor (x_m^k - 3h - y_1)/(y_{i+1} - y_i)\rfloor$. Generally, there exists $\lceil 6h/(y_{i+1} - y_i)\rceil << l$, therefore the optimizing effect to the density estimation neural network is notable.

With the approximate recursive algorithm, the distribution density of data stream can be estimated in real-time. Moreover, the computation memory remains constant to avoid memory overflow and the computation load decreases to $2\times\lceil 6h/(y_{i+1}-y_i)\rceil$ activation operations in one sliding window. In this way, detecting early abnormal state, which is hardly achieved with traditional KDE, becomes a possibility.

4. Condition indictors

Since the density variation of measured data stream reflect the variation of the underlying system condition, the characteristic parameters reflecting the density behavior can be used to detect the abnormal condition. Quantile and entropy are nonlinear statistics associated with density. Quantile indicates the boundary of cumulative probability and entropy is a measure of the average information content of data set. Compared with the linear statistics such as mean, quantile and entropy are less sensitive to outliers and provide more valuable information when the data are irregularly distributed. However, primitive nonlinear statistical operations are inapplicable for streaming data [11], because probability distributions of endless data are difficult to estimate. In this paper, DENN overcomes the obstacle by the time-based sliding window and the approximate recursive algorithm. With the sequences of density estimators, the quantile and entropy of measured data stream can consecutively be calculated to indicate the condition evolution.

For any q, $0<q<1$, the q-quantile is the data value that satisfies the equation $P\{x\le x_q\}=q$, that is to say, if n data are sorted from small to large, q-quantile is the data item at rank $\lceil qn\rceil$. For $q=0.5, 0.25$ and $0.75,$ the quantiles are called the median, lower quartile and upper quartile. In this paper, the q-quantile of data stream are estimated based on the approximate equation that

$$\sum_{i=1}^{r^k}\hat{f}^k(y_i)\approx q\sum_{i=1}^{l}\hat{f}^k(y_i) \tag{8}$$

where r^k is the rank of q-quantile in discretization neurons. While the time window slides to $(k+1)$th step, $\sum_{i=1}^{l}\hat{f}^{k+1}(y_i)$ is calculated by subtracting the output of subnet centered by x_m^k from $\sum_{i=1}^{l}\hat{f}^k(y_i)$ and adding the output of subnet centered by x_1^{k+1}. Then add $\hat{f}^{k+1}(y_i)$ in order of i until the sum is equal or greater than $q\sum_{i=1}^{l}\hat{f}^{k+1}(y_i)$. The number of added density estimators is r^{k+1} and $y_{r^{k+1}}$ is the approximate q-quantile of input neurons in $(k+1)$th time window.

In the information theory, Shannon entropy quantifies the expected value of the information contained in a data set. For a given data source, it also indicates the uncertainty, or more precisely unpredictability of the next measured data. If the data set obeys the uniform distribution, i.e. all possible values are acquired with equal probabilities, then the entropy has a maximum, because it is the situation of maximum uncertainty. On the contrary, the data set with certain values has small entropy. In the extreme case all data are equal, the entropy is zero. For the data set in kth time window, the calculation of entropy can be written as

$$H^k=-\sum_{i=1}^{l}\hat{f}^k(y_i)\ln\hat{f}^k(y_i). \tag{9}$$

While the time window moves forward, there are only $2\times\lceil 6h/(y_{i+1}-y_i)\rceil$ density estimators which are updated according to Eq. (6) and (7). Then, H^{k+1} is acquired by

$$\begin{aligned}H^{k+1}=H^k&+\left[\sum_{i=a_1}^{b_1}\hat{f}^k(y_i)\ln\hat{f}^k(y_i)-\sum_{i=a_1}^{b_1}\hat{f}_1^{k+1}(y_i)\ln\hat{f}_1^{k+1}(y_i)\right]\\&+\left[\sum_{i=a_2}^{b_2}\hat{f}^k(y_i)\ln\hat{f}^k(y_i)-\sum_{i=a_2}^{b_2}\hat{f}_2^{k+1}(y_i)\ln\hat{f}_2^{k+1}(y_i)\right]\end{aligned} \quad (10)$$

5. Simulation verification

To test the performance of our method in condition identification, we construct two large synthetic data sets. The first data set is drawn randomly and independently from a mixing distribution with $N(30,5)$, $N(45,2)$ and $N(65,10)$, where $N(\mu,\sigma)$ is the normal distribution, μ and σ are mean and variance respectively. The second data set obeys the mixing distribution with $N(24,3)$, $N(55,12)$ and $N(80,5)$. These two data sets both have 10k sampling points and constitute a data stream as shown in Fig. 4a, which simulate two different underlying conditions. In Fig. 4a, we cannot acquire an explicit indication for condition transition.

Then, the data stream is processed by DENN with time window length $m=400$. Parameters in discretization layer are set as $l=500$, $y_1=15$ and $y_2=100$. According to reference [3], determination of bandwidth h should comply with Eq. 2. Here, we set h as

$$h=\frac{y_l-y_1}{2\sqrt{m}} \quad (11)$$

When $m\to\infty$, the empirical equation satisfies the requirement of asymptotic unbiased estimation.

The DENN algorithm is written in Matlab7 (MathWorks, Natick, MA) and runs on a PC with 3 GHz Intel core processor and 2 GB RAM. It spends 0.34 s to scan the entire simulated data stream and gets 19600 groups of density estimators in sliding time window, i.e. the average processing time of each group estimators is about $1.73\cdot10^{-5}$ s. For comparison, the original KDE is also implemented and the results are listed in Table 1. Each estimator of KDE is calculated with 20 k sampling points to get overall density estimation and the computation time is about 0.48 s. Compared with original KDE, the proposed DENN algorithm is much faster and the density estimation can be calculated in real time.

Table 1. Comparison of DENN and the original KDE

Density estimation method	Results	Computation memory (sample sizes)	Processing time of each group estimators (s)
Original KDE	Overall density estimators	20000	0.48
DENN	Density estimators in 19600 sliding time windows	400	$1.73\cdot10^{-5}$

The estimators, which are output by DENN in the sliding step of $k=100$ and 19500, are shown in Fig. 4c. Compared with the theoretical density shown as Fig. 4b, we can see that the density estimators match with the actual distribution except a few regions. These error regions are generated by the randomness of data points, the finiteness of m and non-optimal h. Since our goal is to detect system abnormal condition by tracking density variation rather than acquiring accurate density estimators, these errors will not affect the result.

Figure 4d-f give the median, upper quartile and entropy of the simulated data stream. The x-axis is the sum of sliding step k and time window length m, which is equal to the sequence number of x_1^k in data stream. From the figures, we can see that three characteristic parameters all have distinctly different values before and after density variation. They have the ability to extract different distribution features, such as the quantiles which give the sketch information of distribution and the entropy evaluates the whole uncertainty of distribution. Due to the randomness of data generating, the quantiles have more fluctuations in the distribution regions with big variance than small variance, such as the medians shown as Fig. 4d. Data set 2 has a more even distribution than data set 1 on the whole, therefore the entropy values of data set 2 are bigger than for the data set 1 shown as Fig. 4f.

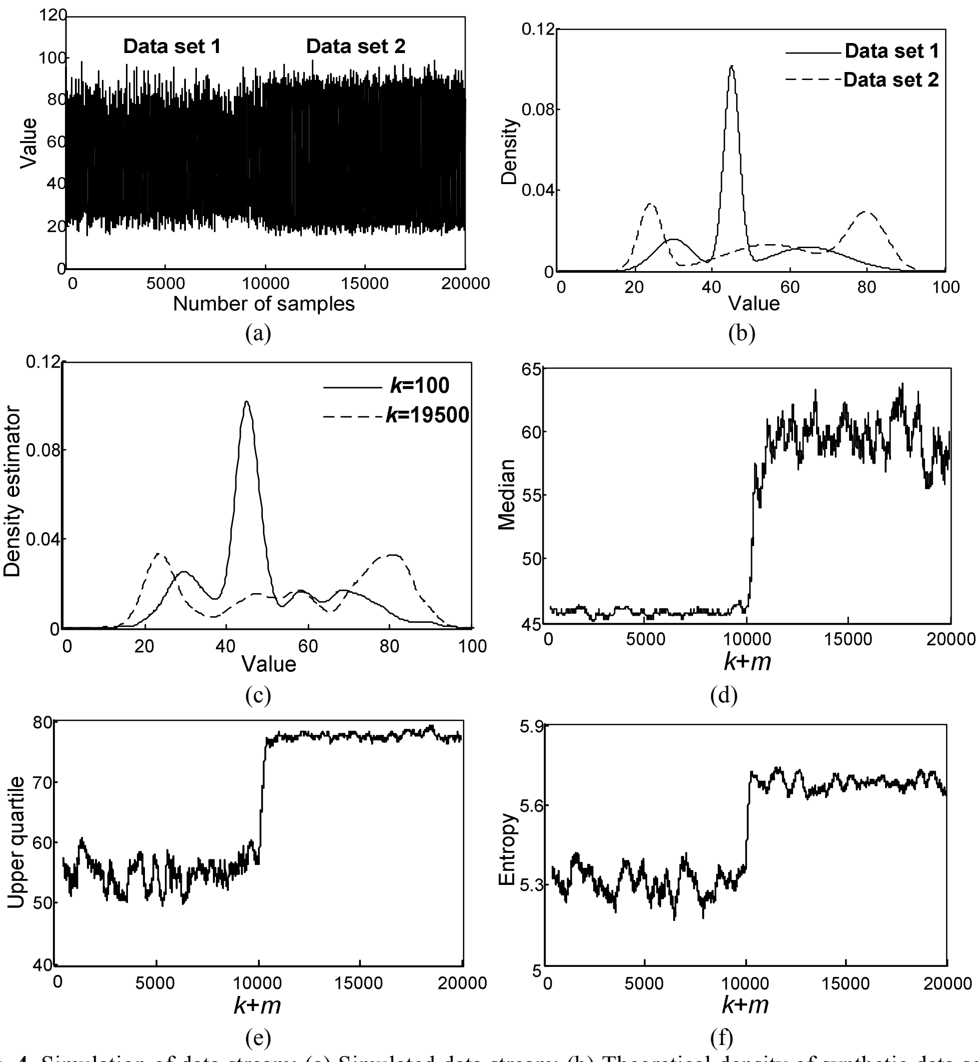

Fig. 4. Simulation of data stream: (a) Simulated data stream; (b) Theoretical density of synthetic data sets; (c) Density estimators output by DENN; (d) Median of data stream; (e) Upper quartile of data stream; (f) Entropy of data stream

From the simulation result, it can be verified that our method can estimate the density of data stream and effectively indicate density variation. So long as the system condition transition causes the density variation of measured data stream, quantile and entropy will reflect anomaly.

6. Experimental verification

In this section, DENN is applied to detect abnormal performance in the pressure data stream measured from a centrifugal compressor. The test rig is a single-stage centrifugal compressor connected to a speed-increasing gearbox, which is driven by a 250 kW DC motor. It operates in an open circuit: air enters through a radial bell-mouth inlet duct, flows through the compressor, is fed to a cylindrical plenum by a connecting pipe, and is discharged into the atmosphere through a motorized throttle valve. Fig. 5 portrays the actual image of the test compressor. The inlet total pressure and outlet static pressure are measured by the transducers with the range 0~5 MPa and precision 0.002 MPa. The mounting positions of transducer are marked up in Fig. 5 and sampling frequency is 150 Hz.

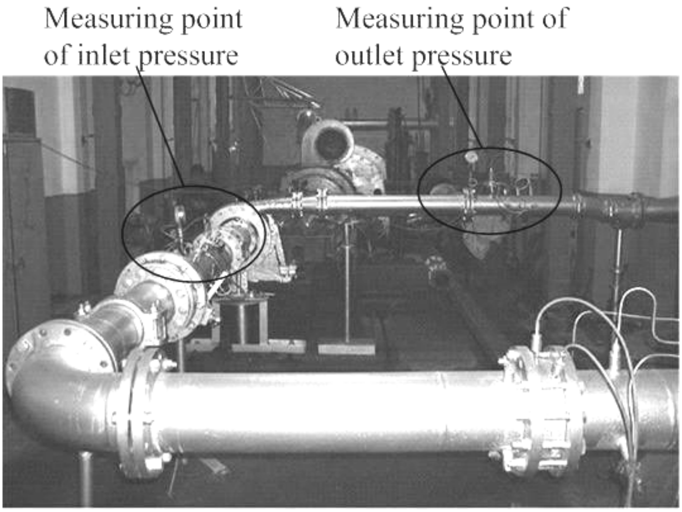

Fig. 5. Compressor test rig

When the surge occurs, it causes low frequency violent oscillation to inlet and outlet pressure and maybe severely damage the compressor. In our experiment, the throttle valve is adjusted to produce a surge process. The inlet total pressure signal is chosen as a test data stream and the segment from normal state to developed surge is shown in Fig. 6a. The data stream is input into the improved density estimation neural network with $m = 200$, $l = 200$, $y_1 = 380$ kPa and $y_2 = 400$ kPa. The output density estimators, which undergo the stages of normal and initial surge, are depicted in Fig. 6b. To facilitate observation, they are graphed with intervals of 10 sliding steps. Although we cannot be sure of anomaly in data stream, the obvious density variation has emerged. The entropy curve of test data stream is shown in Fig. 6c. If the entropy is chosen as monitoring parameter and alarm threshold is set to 4.35, surge precursor can be detected from inlet pressure data stream at time w, which is also marked in Fig. 6c. As a comparison, the surge's feature of low frequency oscillation is extracted by empirical mode decomposition (EMD) [12]. After three-level decomposition, the residual acquired by separating the intrinsic mode functions $c_1 \sim c_3$ from the pressure signals is shown in Fig. 6d. With the local time scale decomposition of EMD, the residual signal clearly exhibits the evolution process of surge oscillation. We notice that the time w lies in the initial stage of low frequency oscillation. Consequently, our method has a good ability of detecting compressor surge. More significantly, DENN is an online approach to detect anomaly in data stream instead of an ex-post analysis method like EMD.

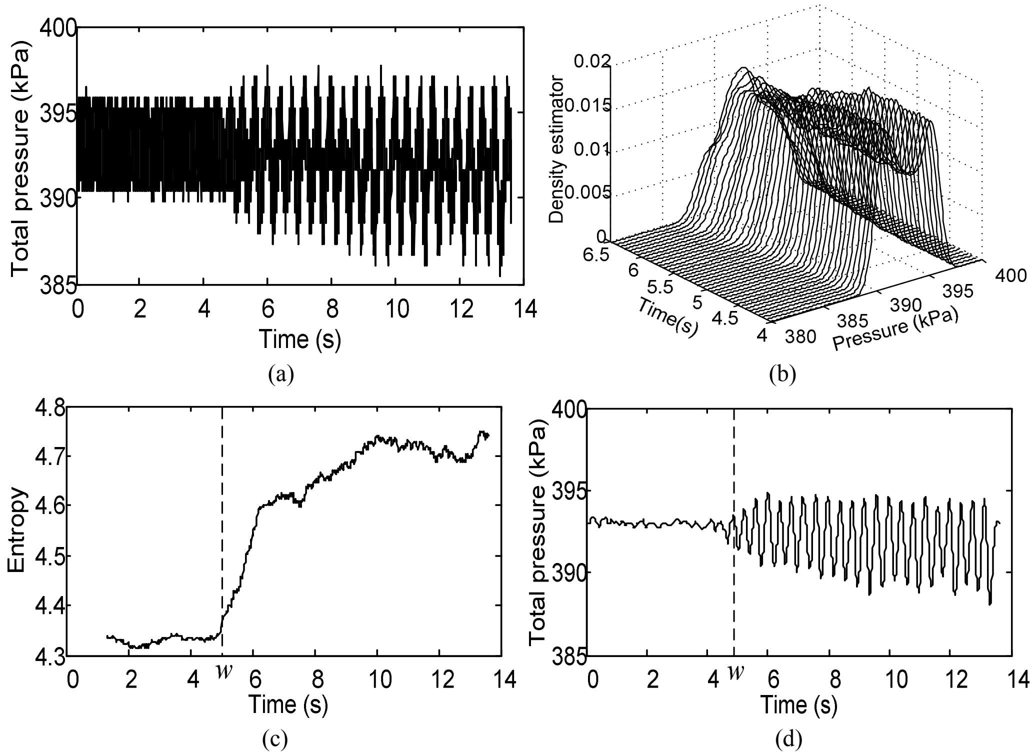

Fig. 6. Detection of compressor surge: (a) Inlet total pressure signal; (b) Density estimators of pressure data stream; (c) Entropy of pressure data stream; (d) Residue of empirical mode decomposition

7. Conclusions

Along with the appearance of more and more long-term monitoring data in industry fields, efficient anomaly detection in data streams becomes vital due to limitation of processing time and computation memory. This paper introduces a new density estimation neural network to consecutively estimate density of sliding segment in data stream. DENN outputs discrete density estimators with fixed computation memory. By subnet separation and local activation, the approximate recursive algorithm makes density estimation catch up the pace of data streaming. With the output of the network, the nonlinear statistics, quantile and entropy, can easily be calculated and be used to reveal the density variation. Simulated and experimental results demonstrate that our method can track the density variation of data stream and identify anomaly timely. Although it is an online data processing method, DENN can achieve the same detection effect to compressor surge precursor with EMD.

In DENN, the accuracy of density estimators significantly depends on the choice of kernel bandwidth h. However, a global bandwidth is not always sufficient to model all local distribution especially to the complex and inconstant data stream. If a middle layer is added into the networks to adaptively adjust bandwidth, the network may provide more precise and robust density estimators. These are what our research work will focus on next.

Acknowledgements

This work was supported by the National Natural Science Foundation of P. R. China (approval No. 51005174), Doctoral Fund of Ministry of Education of China (approval No. 20100201120003) and National High Technology Research and Development Program of China (approval No. 2012AA040913).

References

[1] **Jing Wang, Qing Zhang, Guanghua Xu** Genetic stochastic resonance: a new fault diagnosis method to detect weak signals in mechanical systems. Advanced Science Letters, Vol. 4, Issue 6, 2011, p. 2508 – 2512.
[2] **B. Scholkopf, J. C. Platt, J. S. Taylor, A. J. Smola** Estimating the support of a high-dimensional distribution. Neural Computation, Vol. 13, Issue 7, 2001, p. 1443 – 1471.
[3] **B. Silverman** Density Estimation for Statistics and Data Analysis. London: Chapman and Hall, 1986.
[4] **C. Heinz, B. Seeger** Towards kernel density estimation over streaming data. Proceedings of Management of Data, 2006, p. 14 – 16.
[5] **M. Ceci, A. Appice, C. Loglisci, C. Caruso, F. Fumarola, D. Malerba** Novelty detection from evolving complex data streams with time windows. Lecture Notes in Computer Science, Vol. 5722, 2009, p. 563 – 572.
[6] **A. Zhou, Z. Cai, L. Wei, W. Qian** M-Kernel merging: towards density estimation over data streams. Proceedings of the Eighth International Conference on Database Systems for Advanced Applications, 2003, p. 26 – 28.
[7] **T. Zhang, R. Ramakrishnan, M. Livny** Fast density estimation using CF-kernel for very large database. Proceedings of Knowledge Discovery and Data Mining, 1999, p. 15 – 18.
[8] **M. Davy, F. Desobry, A. Gretton, C. Doncarli** An online support vector machine for abnormal events detection. Signal Processing, Vol. 86, Issue 8, 2006, p. 2009 – 2025.
[9] **E. Parzen** On estimation of a probability density function and mode. Annals of Mathematical Statistics, Vol. 33, Issue 3, 1962, p. 1065 – 1076.
[10] **D. F. Specht** Probabilistic neural networks and the polynomial adaline as complementary techniques for classification. IEEE Transactions on Neural Networks, Vol. 1, Issue 1, 1990, p. 111 – 121.
[11] **Jian Xu, Xuemin Lin, Xiaofang Zhou** Space efficient quantile summary for constrained sliding windows on a data stream. Lecture Notes in Computer Science, Vol. 3192, 2004, p. 34 – 44.
[12] **Q. Gao, C. Duan, H. Fan, Q. Meng** Rotating machine fault diagnosis using empirical mode decomposition. Mechanical Systems and Signal Processing, Vol. 22, Issue 5, 2008, p. 1072 – 1081.