# Keystroke Dynamics Identification Based on Triboelectric Nanogenerator for Intelligent Keyboard Using Deep Learning Method

*Guangquan Zhao, Jin Yang, Jun Chen, Guang Zhu, Zedong Jiang, Xiaoyong Liu, Guangxing Niu, Zhong Lin Wang,* and Bin Zhang**

**Due to the heavy reliance on computers and networks, security issues have become a major concern for individuals, companies, and nations. Traditional security measures such as personal identification numbers, tokens, or passwords only provide limited protection. With the development of intelligent keyboard (IKB), this paper proposes a deep-learning-based keystroke dynamics identification method for increased security. The IKB is a kind of self-powered, nonmechanical-punching keyboard, which converts mechanical stimuli applied to the keyboard into local electronic signals. Multilayer deep belief network (DBN) is established to mine the useful information from raw electronic signals and output the keystroke dynamics identification result. The contributions include development of a novel solution that does not rely on manual feature extraction, and provides promising recognition accuracy on large amount of typing samples. One significant advantage of the proposed method is that it extracts features adaptively from the raw current signals and automatically recognizes the typing pattern, which simplifies the design of verification and identification system. The experimental results on 104 typing samples demonstrate the effectiveness of the proposed method. The proposed method has extensive applications in keyboard-based information security.**

Computers and networks are essential for our modern lives and have been utilized indispensably in research, industry, education, military, financial, etc. Keyboard, one of the most significant human–machine interface devices, is critical to the security of information input and control for lots of systems, such as financial managepment, bill payments, internet access, personal communications, and so forth.[1,2] Unauthorized access to networks or illegal manipulation of computers may lead to dangerous consequences to individuals, companies, and nations.[3–5] Due to the heavy reliance on computers and networks, security issues have emerged as a major concern in human society.

To prevent unauthorized access, security measures such as passwords or tokens are extensively applied nowadays. However, a critical weakness of a great number of prevalent authentication systems is that any identity thief could be recognized as the genuine owner with stolen personal identification information (e.g., passwords).[3–6] Therefore, it is necessary to develop new advanced identifiers that can defend the security for genuine owners. In mid-1970s, the behavioral biometric of keystroke dynamics was considered as a means of keystroke pattern recognition unprecedentedly. However, since biometric identifiers are intrinsic and more difficult to be separated or mimicked from the original owner, it was employed by placing an additional layer of security on existing systems. In 1980, some preliminary study was presented on keystroke dynamic-based authentication using the *T*-test on digraph features.[7] Soon after that, keystroke dynamics for verification and identification was enormously studied, benefiting from their inexpensiveness, little intrusiveness, and easy implementation on top of the available authentication systems with few modification.[8,9] Even so, the development of keystroke dynamics is rather slow and is still at its very early stage, partially owing to the fact that almost all of the proposed studies rely on dimensional keystroke timing vectors as typing patterns, which can only communicate with the keystroke timing characteristics, thus limiting the evolvement of biometric-based measures in universality, uniqueness, permanence, accuracy, and acceptability.[10–13] To sum up, an accuracy and unique and

Prof. G. Q. Zhao, Z. D. Jiang, X. Y. Liu
Department of Automatic Test and Control
Harbin Institute of Technology
Harbin 150001, China

Prof G. Q. Zhao, G. X. Niu, Prof. B. Zhang
Department of Electrical Engineering
University of South Carolina
Columbia, SC 29208, USA
E-mail: zhangbin@cec.sc.edu

Dr. J. Yang, Dr. J. Chen, Dr. G. Zhu, Prof. Z. L. Wang
School of Materials Science and Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
E-mail: zlwang@gatech.edu

Dr. J. Yang
Chongqing University
Chongqing 400044, China

Dr. G. Zhu, Prof. Z. L. Wang
Beijing Institute of Nanoenergy and Nanosystems
Chinese Academy of Sciences
Beijing 100083, China

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/admt.201800167.

**DOI: 10.1002/admt.201800167**

keystroke dynamics identification that can be used for practical applications is highly desired.

In this study, starting from the method presented by Wang and co-workers,[1,2] a novel keystroke dynamics identification method is developed for intelligent keyboard using deep learning technologies. Different from the traditional keyboard, the intelligent keyboard is a self-powered, nonmechanical-punching keyboard enabled by contact electrification between human fingers and keys,[1] which converts mechanical stimuli applied to the keyboard into local electronic signals without applying an external power. These electric signals can not only characterize the property of keystroke timing, but also quantitatively record the concrete dynamic changes in the course of typing motions, and thus they provide an unprecedentedly accurate, unique, and permanent typing pattern for further verification and identification purposes. The traditional keystroke pattern recognition generally includes preprocessing, feature extraction, feature selection, and pattern recognition. In these traditional methods, feature extraction and selection play a critical role in pattern recognition performance. A variety of feature extraction methods for keystroke dynamics were developed including time-domain features, wavelet features, and so on.[6–13] Existing methods show some limitations: (1) most features are manually extracted and selected, which requires complex signal processing and extensive expert involvement; (2) feature extraction and selection is ad hoc and time-consuming; and (3) the classifiers adopted in traditional methods have shallow architectures, which limits the capacity to learn the complex nonlinear relationships between keystroke dynamics and typists.

To address these limitations of traditional methods, this paper develops a deep learning based keystroke dynamics identification method for intelligent keyboard (IKB). Because of its excellent performance in adaptively extracting features from raw data and describing nonlinear dynamics, deep learning techniques have made great achievements in image processing, speech recognition, fault diagnosis, and so on.[14,15] To our knowledge, study on deep learning in keystroke dynamics identification remains an open problem yet to be studied. Deep belief network (DBN),[16] as a key framework of deep learning techniques, has significant potentials in keystroke dynamics identification mainly because of two major characteristics. First, its multilayer structure and its means of training enable DBN to adaptively extract features. As a result, raw electronic signals can be directly used in training and practical identification, which minimizes the needs and requirements of signal processing and domain knowledge. Second, DBN has unique advantages in handling high-dimensional and nonlinear data, which enables DBN to characterize complex mapping relationships between electric signals and typing patterns. This makes DBN a very powerful tool for keystroke dynamics identification.

The contributions of this paper are twofold: (1) It proposes a novel and simple keystroke dynamics identification method for intelligent keyboard using DBN, which integrates feature extraction and pattern recognition. In this method, features are adaptively extracted from raw electric current signals layer-by-layer. Softmax classifier is used to automatically distinguish the typing patterns based on the extracted features. The proposed method makes the design of user authentication system much easier as it requires very little human involvement. (2) Thorough analysis and studies on 104 typing samples are conducted to verify the proposed method. Performance of the proposed method in terms of dimension of input layer, number of DBN layers, and number of typing patterns is discussed with experimental analysis. The results show that, the proposed method offers an unprecedentedly accurate, and stable typing pattern recognition. The proposed method will have extensive applications in the fields of artificial intelligence, cyber security, and computer or network access control.

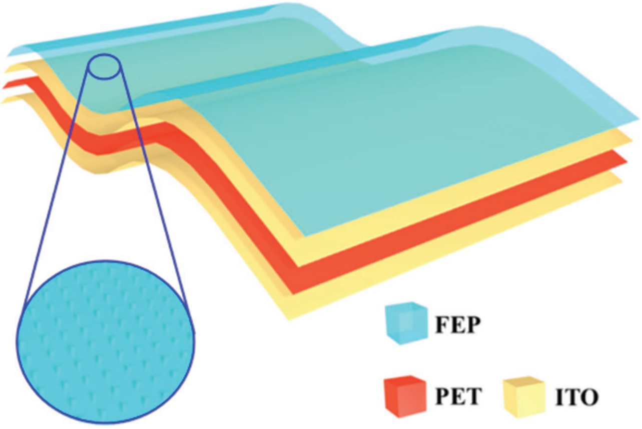

**Figure 1.** Structural design of the KFE for intelligent keyboard (adopted from ref. [1]).

The rest of this paper is organized as follows: Firstly, the principle of intelligent keyboard is briefly introduced. Secondly, DBN-based keystroke dynamics identification method is described. Then experiment results are presented to verify the effectiveness of the proposed method. Finally, the conclusions are provided.

The IKB to be used for this study is based on the one previously reported in[1]. **Figure 1** shows the structural design of the key functional element (KFE) for intelligent keyboard. The KFE of the IKB is composed of vertically stacked transparent thin film materials as indicated. A layer of polyethylene terephthalate (PET) sits between two layers of indium tin oxide (ITO) that are the bottom and the top electrodes. Then, a layer of fluorinated ethylene propylene (FEP) is applied onto the ITO surface as an electrification layer that generates triboelectric charges upon contact with human fingers.

**Figure 2** shows the operating principle of the intelligent keyboard. The basic working principle of the IKB is based on the principle of triboelectric nanogenerators (TENG) by using the coupling between contact electrification and electrostatic induction rather than the traditional mechanical switching.[17–20] When a human finger is brought into contact with FEP, charge transfer at the contact interface occurs. Once a keystroke is initiated, the positively charged human finger approaches the keyboard, the induced positive charges on the top electrode are expelled, resulting in a flow of free electrons from the bottom electrode to top electrode until the finger and the key are in contact (Figure 2a). When the finger separates, free electrons flow backward from the top electrode to the bottom electrode. Consequently, consecutive keystrokes result in a

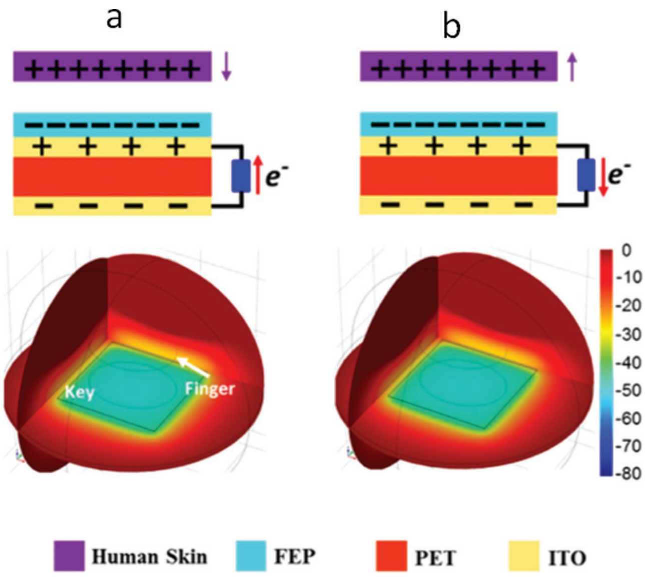

**Figure 2.** Operating principle of the intelligent keyboard. a) When a keystroke is initiated, the approach of positively charged human finger results in free electrons flowing from bottom ITO electron to top electrode. b)When the finger is up and a separation occurs, it produces another current in the external circuit flowing from the top electrode to bottom electrode (adopted from ref. [1]).

periodical-changing electric field that drives reciprocating flows of electrons between electrodes (Figure 2b).

To obtain the typing dataset, a customized multichannel data acquisition system was designed for the IKB to individually address the electric signal from each key (**Figure 3**a). Consequently, the real-time tracing and recording during typing can be realized (Figure 3b). Every channel was electrically but independently connected to a key in the keyboard as a functional unit. Each set of typing pattern corresponds to two subsets of characteristic signals (voltage and current), which are time-series data. These electric signals correlate to a variety of information, including the manner and rhythm of the keystroke, typing habit, finger size, individual bioelectricity, and applied typing force. Thus, they can not only characterize the keystroke timing, but also quantitatively record the concrete dynamic changes in the course of typing. In this paper, the current signals are selected as the input of DBN, because it represents the speed/force at which the key is stroke. More information about the IKB can be found in ref. [1].

This study proposes a keystroke dynamics identification method, which is featured by adaptive feature extraction from raw current signals and automatic pattern identification. Procedure of the proposed keystroke dynamics identification method is shown in **Figure 4**. First the raw current signals of all typists from IKB are obtained. Then the data sampled from raw current signals are segmented to construct dataset. The dataset is then normalized and divided into training and testing sets. The training set is used to train the DBN model in the training stage. Finally, the testing set is input to the trained DBN model to evaluate the identification performance.

The implementation of the proposed method can be further elaborated as follows

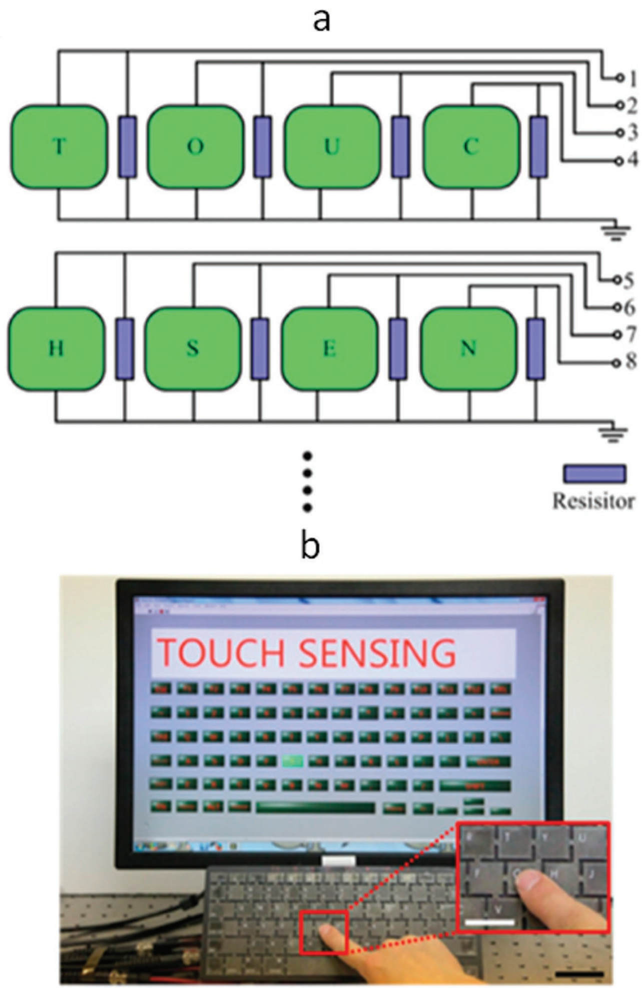

**Figure 3.** Keystroke tracing and real-time recording system. a) Multichannel data acquisition system for keystroke tracing and real-time recording. b) A photograph demonstrated the IKB for the real-time keystroke tracing and recording (adopted from ref. [1]).

*Procedure 1*: Define the keystroke dynamics identification problem and determine the potential number of typists for the IKB. It is worth mentioning that the number of typists is relatively small in most cases in reality, such as in the case of personal computer access. To fully validate the performance of the proposed method, the number of typists is set to 2–104 in this study.

*Procedure 2*: Obtain raw current signals from IKB for each typist. The raw current signals are sampled with 2 KHz sampling rate. Taking a typist with index of 008 as an example, **Figure 5** illustrates the current signal when he types the word "touch" for six times on the IKB. It can be seen that the total sampling number is 16001, and the total sampling time is 8 s.

*Procedure 3*: Construct dataset by equal interval segmentation on raw current signals. The segmentation method is in the form of multiple windows as illustrated in **Figure 6**.

Figure 6 shows the segmentation from 2000 to 4000 sampling points. Each segment contains 100 points and thus 20 instances named S1–S20 are obtained. The segmentation rule is as follows: if the number of total sampling points in

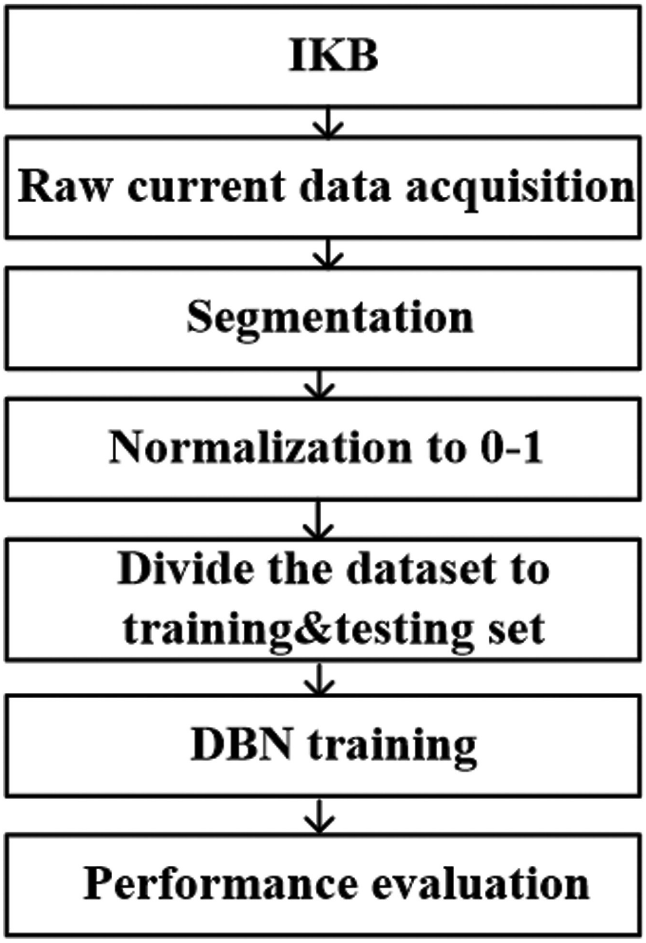
IKB
Raw current data acquisition
Segmentation
Normalization to 0-1
Divide the dataset to training&testing set
DBN training
Performance evaluation

**Figure 4.** Procedure of the proposed keystroke dynamics identification method.

a current signal for certain typing pattern is $t$, and the sampling points in each segment contains $d$ points, we have $n = \frac{t}{d}$ instances. Here $n$ is defined as the number of instances for certain typing pattern, and $d$ is the dimension of each instance, which is equal to the dimension of input layer for DBN. Taking typist 008 as an example, the raw current signal contains 16001 sampling points, if the number of sampling points in each segment is set as 100, then we can obtain 160 instances and the dimension of each instance is 100. In this example, the last sampling point is ignored.

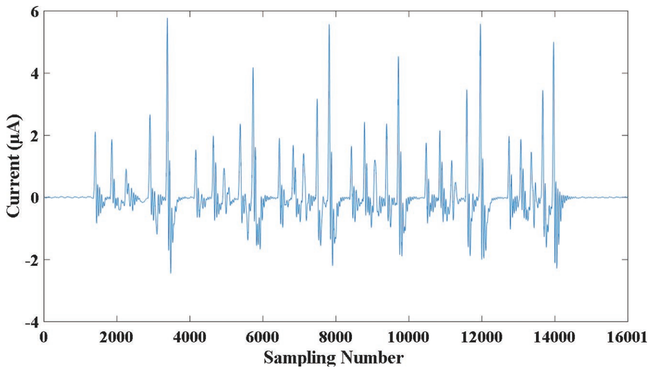

**Figure 5.** Raw current data of one typist (index 008).

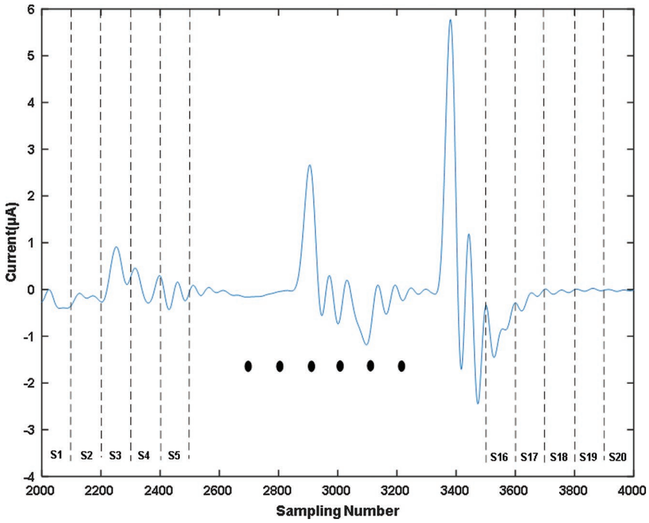

**Figure 6.** Schematic of segmentation on raw current signal.

*Procedure 4*: Normalize each real attribute of the constructed dataset to [0, 1] by $x^* = (x - x_{\min})/(x_{\max} - x_{\min})$, where $x^*$ is the normalized data, $x$ is real attribute value sampled from the raw current signal, $x_{\max}$ and $x_{\min}$ are the maximum and minimum of $x$, respectively. Taking typist 008 as an example, after the 160 instances are obtained in Procedure (3), the matrix formed by 160 instances is normalized. **Figure 7** shows the normalized current signal in Figure 6.

*Procedure 5*: Divide the normalized datasets into training set and testing set. If not specified, in this study, the first 80% of instances in the order of time for each current signal are used for training and the next 20% of instances are used for testing. Taking the signal from typist 008 as an example, the first 128 instances are used for training and the remaining 32 instances are used for testing.

*Procedure 6*: Use the training set to train a DBN model. The DBN model is inspired by the neural network structure of

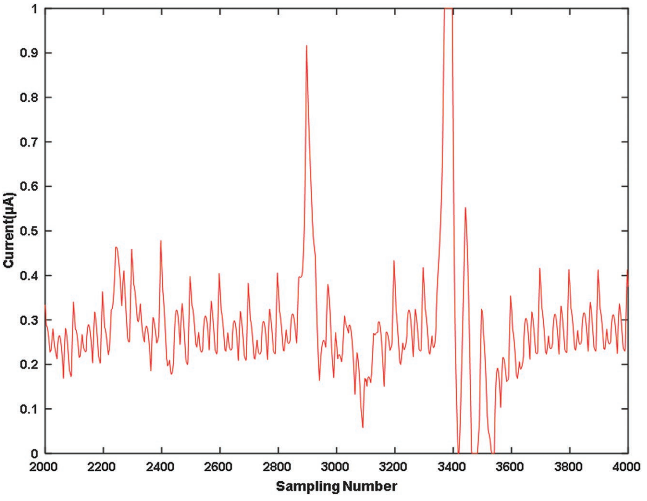

**Figure 7.** Normalized current signal from 2000 to 4000 sampling points.

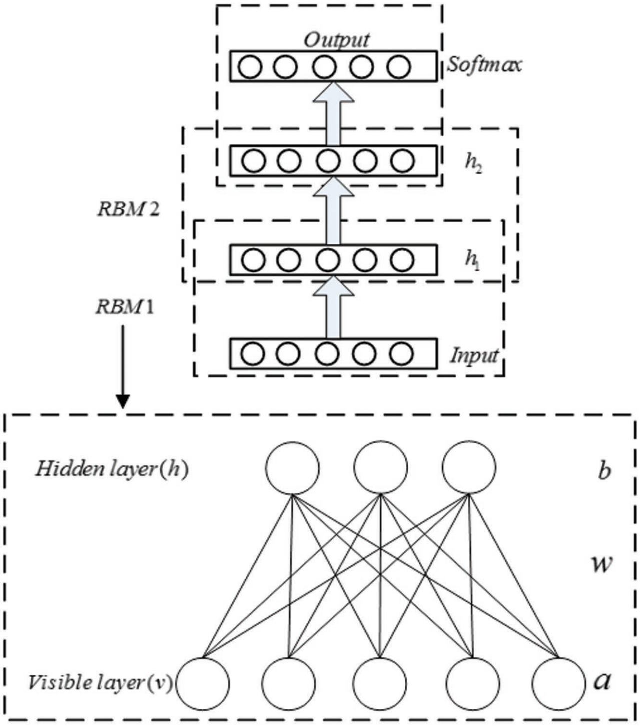

Figure 8. A four-layer structure of DBN and basic structure of RBM.

human brain. Through multiple nonlinear operational hidden layers, the natural features of data can be learned from low level to high level. A DBN model is composed of stacked restricted Boltzmann machine (RBM) and a classifier.[16] **Figure 8** shows an example of four-layer DBN structure, which consists of input, two stacked RBMs, and output layers. The two-layer network in the dashed curve is the structure of RBM. Each RBM contains a visible layer and a hidden layer. The units in the same layer are not connected and the units in two adjoining layers have directed symmetrical connections.[21,22] In this structure, layer 1 (input layer) and layer 2 (h1) forms RBM1, layer 2 (h1) and layer 3 (h2) forms RBM2. In Figure 8, the input layer of RBM1 takes the normalized vector v sampled from raw current signals. The two hidden layers extract features from lower layer to higher layer automatically. Features extracted at the top layer of RBM2 are used in pattern recognition, and the output layer uses softmax as the classifier to determine the typing pattern. For example, a DBN structure of 100-150-50-10 means the number of units of input layer in RBM1 is 100, and the numbers of units of the two hidden units are 150 and 50, respectively, and 10 is the number of typists to be identified.

**Figure 9** shows the procedure of training a DBN model. The training process of DBN can be divided into an unsupervised layer-by-layer pretraining stage of stacked RBMs and a global fine-tuning stage by back propagation algorithm.[22,23] The pretraining stage aims to fully extract features from low-level to high-level and, at the same time, avoid local optimum. The fine-tuning stage of network parameters is to further optimize the network capability.

In the pretraining stage, each layer of DBN is trained using the RBM learning rule with two phases, namely positive and negative phases. The positive learning phase transfers data from visible layer to hidden layer, and the negative phase transfers data from hidden layer to a reconstruction of the visible layer. The goal of pretraining is to make sure the trained RBM model describes the distributions of input data as close as possible.

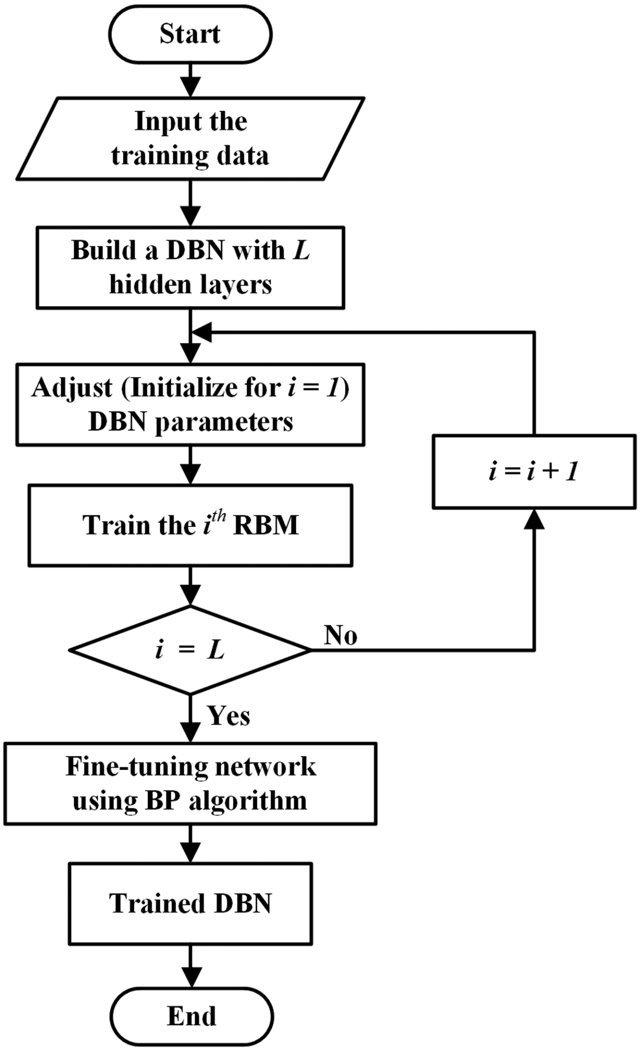
Start
Input the training data
Build a DBN with L hidden layers
Adjust (Initialize for i = 1) DBN parameters
Train the i^th RBM
i = L
No
i = i + 1
Yes
Fine-tuning network using BP algorithm
Trained DBN
End

Figure 9. Procedure of training a DBN model.

After the pretraining is completed, supervised training of DBN in the fine-tuning stage will further reduce the training error and improve the recognition accuracy. Back propagation algorithm is adopted to fine-tune the parameters using the labeled data. Different from the unsupervised pre-training, the supervised fine-tuning updates all parameters at the same time until the maximum number of iterations is reached. When both pretraining and fine-tuning stages are completed, the DBN can be used for practical keystroke dynamics identification.

In this step, parameters of DBN, such as the number of hidden layers, the number of units for each layer, and the value of learning rate, etc. need to be determined. In this study, the number of units for input layer is set to equal to the number of sampling points in each instance.

*Procedure 7*: After the training is completed, the performance of the trained DBN is evaluated on the testing set.

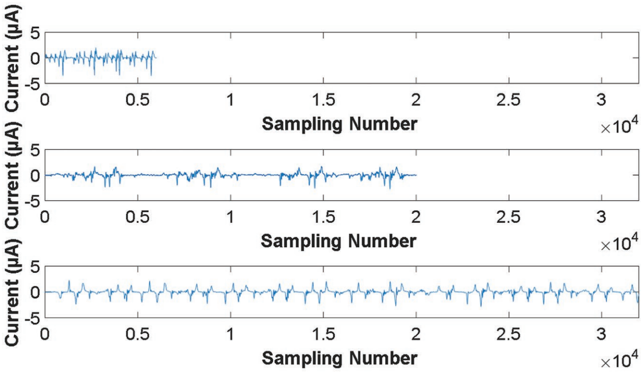

**Figure 10.** Current signals from three participants.

If the performance is acceptable, the trained DBN can be used for practical keystroke dynamics identification. Otherwise, DBN parameters are adjusted and Procedure (6) is repeated to improve the performance of DBN.

The highlight of the proposed method is that the proposed method integrates feature extraction and pattern recognition. In this method, DBN takes the raw current signals collected from IKB as input, it adaptively extracts features from raw current signals and automatically outputs the keystroke dynamics identification result. The new method does not rely on complex signal processing or expert involvement, which simplifies the design of identification system and makes it applicable to many identification systems directly and effectively.

To evaluate the effectiveness of the proposed method, 104 participants were invited to independently type the word "touch" for more than four times on the IKB in each personalized manner. To ensure the randomness and diversity, all the selected participants are from 14 to 69 years old, male and female people from different countries. As a result, a dataset of 104 individual typing patterns was collected.[1]

**Figure 10** shows three current signal examples from three participants. Due to the different typing speed, the three current signals contain different numbers of sampling points. It is clear that the keystroke dynamics looks quite different. These current signals correlate to a variety of information, including the manner and rhythm of the keystroke, applied typing force, finger size, typing habit, and individual bioelectricity. Thus, they can not only characterize the keystroke timing, but also quantitatively record the concrete dynamic changes in the course of typing. As a result, the IKB provides a superior method in creating accurate, unique, and permanent typing patterns for verification and identification purposes. To facilitate the segmentation and processing of current signals, the last section of sampling points that cannot form a complete instance are discarded. The obtained dataset is normalized and divided into training set and testing set.

The DBN algorithm is programmed by Matlab R2016a, and all the experiments are run on a computer with 2.0 GHz processor and 32 GB RAM. In the experiment, we found that DBN structure (number of layers, dimension of input layer, etc.) has significant influence on the experimental results. **Table 1** summarizes the average testing accuracy of ten runs under different dimension of DBN input layer. In this experiment, the structure of DBN is set as *D*-150-50-*N*. Note that the DBN in this experiment has a 4-layer structure and keeps unchanged. In this structure, the number of units in the input layer is *D* (*D* is from 30 to 150 in this study), which is equal to the number of sampling points in each instance. The numbers of units in the two hidden layers are 150 and 50, respectively. The number of units in the output layer is *N* (*N* is from 2 to 104 in this study), which is equal to the number of typing patterns. The pretraining iterations of each RBM is 500 and the fine-tuning iterations is set as 20000. The learning rate and the momentum are set as 0.11 and 0.9, respectively. Note that these parameters are selected based on trial-and-error. In this study, the first 80% of instances in the order of time for each typing pattern are used for training, and the next 20% of instances for each typing pattern are used for testing. The evaluation criterion is the recognition accuracy, which is given by the number of correct recognition instances divided by the number of all instances.

**Table 1.** Testing accuracy under different dimension of input layer (%).

| Dimension of input layer (*D*) | Number of typing patterns (*N*) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 2 | 4 | 8 | 13 | 30 | 50 | 104 |
| 30 | 100 | 100 | 99.60 | 99.11 | 98.27 | 96.44 | 94.27 |
| 40 | 100 | 100 | 99.64 | 99.31 | 98.92 | 98.14 | 95.60 |
| 50 | 100 | 100 | 99.91 | 99.15 | 98.11 | 98.16 | 97.05 |
| **60** | **100** | **100** | **100** | **99.40** | **98.45** | **98.5** | **98.01** |
| 70 | 100 | 100 | 99.44 | 99.17 | 98.23 | 98.05 | 97.23 |
| 80 | 100 | 100 | 99.91 | 98.89 | 98.11 | 97.56 | 96.72 |
| 90 | 100 | 100 | 98.47 | 97.64 | 97.18 | 96.25 | 95.23 |
| 100 | 100 | 100 | 98.26 | 97.25 | 97.02 | 96.05 | 94.85 |
| 110 | 100 | 99.23 | 98.81 | 98.44 | 97.66 | 95.89 | 95.10 |
| 120 | 100 | 99.32 | 98.95 | 99.01 | 98.26 | 96.52 | 95.89 |
| 130 | 100 | 99.56 | 99.36 | 98.86 | 97.89 | 97.08 | 96.52 |
| 140 | 100 | 99.81 | 99.88 | 99.20 | 98.01 | 97.64 | 96.22 |
| 150 | 100 | 99.25 | 99.48 | 98.56 | 97.68 | 96.94 | 95.28 |

Table 1 summarizes the testing accuracy with increase of the dimension of input layer and the number of typing patterns. It can be seen that testing accuracy increases as the dimension of input layer increases from 30 to 60. It also shows that the testing accuracy reaches the highest when the dimension of input layer reaches 60, with the testing accuracy being 100% when the number of typing patterns is equal to 2, 4, 8, respectively. The testing accuracy declines slightly when the number of typing patterns increases. However, the testing accuracy still reaches 98.01% when the number of typing patterns is equal to 104. Table 1 also shows that, after the dimension of input layer reaches 60, further increase of dimension of input layer will not improve the performance. This is mainly because the number of instance will decrease with the increase of dimension for input layer. This analysis indicates that informative input vectors and sufficient instances are crucial for DBN to achieve high performance.

To further investigate the influence of number of DBN layers, testing accuracy under different number of DBN layers is studied and summarized in **Table 2**. In this experiment, the

**Table 2.** Testing accuracy under different number of DBN layers (%).

| Number of typing patterns (*N*) | Number of DBN layers | | | |
|---|---|---|---|---|
| | 3 | 4 | 5 | 6 |
| 2 | 89.56 | **100** | 100 | 100 |
| 4 | 73.22 | **100** | 100 | 100 |
| 8 | 69.54 | **100** | 99.47 | 99.17 |
| 13 | 64.58 | **99.40** | 98.66 | 97.99 |
| 30 | 50.56 | **98.45** | 97.77 | 96.06 |
| 50 | 44.83 | **98.5** | 97.64 | 95.86 |
| 104 | 40.25 | **98.01** | 95.77 | 95.43 |

dimension of input layer is a constant of 60, while the number of layers increases with the structure of DBN being set as 60-100-*N* (3-layer), 60-150-50-*N* (4-layer), 60-150-100-50-*N* (5-layer), and 60-150-120-80-50-*N* (6-layer), respectively. The other parameters of DBN are same as those of previous experiment.

It can be seen from Table 2 that the 3-layer DBN has the worst recognition results while the 4-layer DBN has the best results. This is due to the fact that the 3-layer DBN has only one hidden layer and cannot fully extract the features of input signal. Since the amount of current data collected in this research is not very large, the 4-layer DBN are the most suitable structure in this study, and the 5-layer and 6-layer DBNs have shown a slight overfitting. This analysis shows that choosing the appropriate network structure is important for obtaining good recognition results.

To explain the fundamental reason of DBN's good performance in pattern recognition and demonstrate the feature extraction performance of the proposed method, t-SNE technique is used to visualize the performance of feature extraction.[21] The t-SNE technique produces significantly better visualizations by reducing the tendency to put points together in the center of the map. Figures 11–13 visualize the raw data and the extracted features by two hidden layers of DBN. To make the figures clear, these figures only show the case of $N = 5$.

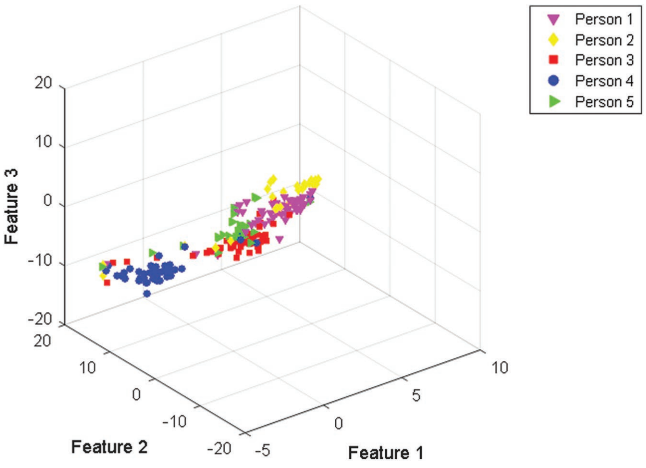

**Figure 11.** Visualization of raw data under 5 typing patterns.

**Figure 11** shows that the raw data under five typing patterns have varying degrees of overlapping. On the contrary, **Figures 12** and 13 illustrate that the features extracted by the hidden layers of DBN show a trend of separation, especially the features extracted by the second hidden layer, which are completely separated. From a pattern recognition point of view, a classifier with inputs from features shown in **Figure 13** will have a higher accuracy than the one with inputs shown in Figure 11. This result shows that DBN is very effective in feature extraction for keystroke dynamics identification under a small number of patterns. It also indicates that one hidden layer is not sufficient to extract effective features, while two hidden layers are able to extract effective features for keystroke dynamics identification in this case.

For verification and identification, the number of typing patterns must be big. **Figures 14–16** visualize the raw data and the extracted features by two hidden layers of DBN in the case of $N = 104$. Because one figure cannot show 104 patterns clearly, ten of them are randomly selected for display. Figure 14 shows that the raw data under 104 patterns has severe overlapping.

Figures 15 and 16 show that the features extracted by DBN separate most typing patterns and all typing patterns are in a state of spreading outward. The features obtained by the second

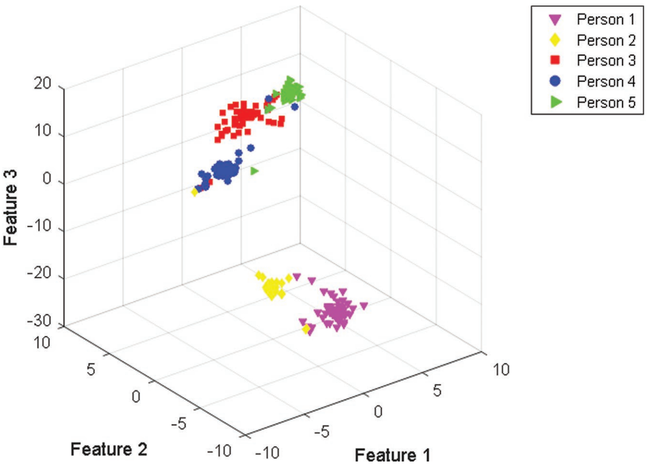

**Figure 12.** Visualization of features by the 1st hidden layer with 5 typing patterns.

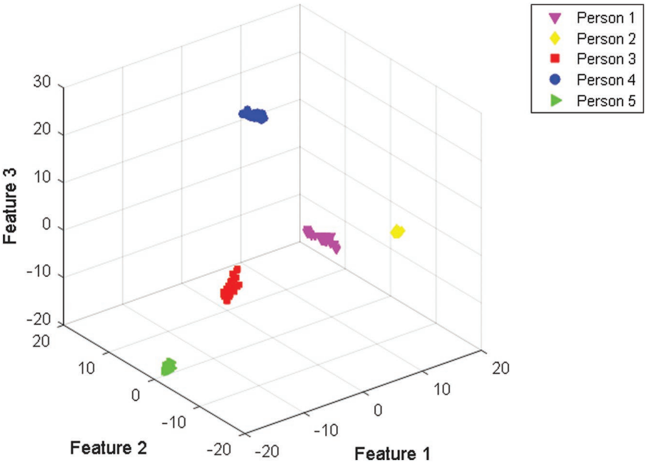

**Figure 13.** Visualization of features by the 2nd hidden layer with 5 typing patterns.

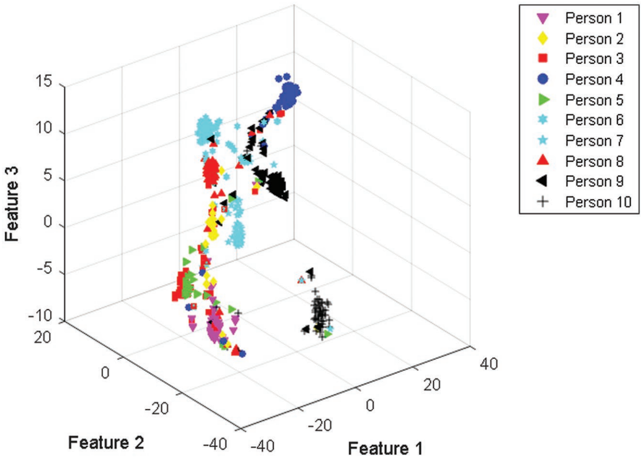

Figure 14. Visualization of raw data under 104 typing patterns.

hidden layer are almost completely separated. Obviously, a classifier with inputs from features shown in Figure 16 will have a higher accuracy than the one with inputs shown in Figure 14. In conclusion, DBN shows excellent performance in feature extraction for keystroke dynamics identification under a big number of patterns.

**Tables 3** summarizes and compares the experiment results under three conditions. Here, the testing accuracy is obtained under the following condition: the first 80% of instances in the order of time for each current signal are used for training and the remaining 20% of instances are used for testing. The validation accuracy is obtained under the following condition: the dataset is divided into training dataset (60%), testing dataset (20%), validation dataset (20%), and the validation dataset is separated from training dataset and testing dataset. The cross-validation accuracy is obtained using fivefold cross-validation experiment. In these experiments, the structure of DBN is set same as 60-150-50-*N*. It can be seen from Table 3 that the proposed method achieves 100% recognition accuracy on all training data. More importantly, the validation accuracy and cross-validation accuracy are in good agreement with the testing accuracy. In addition, the validation accuracy is slightly lower than that of testing accuracy because its training rate is lower. Since cross-validation accuracy is the average result of 5 models, its accuracy is also slightly lower than that of test accuracy. In conclusion, these experiment results futher demonstrate the effectiveness of the proposed method.

**Table 3.** Keystroke dynamics identification results using DBN.

| Number of patterns (*N*) | Training accuracy | Testing accuracy | Validation accuracy | Cross-validation accuracy |
|---|---|---|---|---|
| 2 | 100% | 100% | 100% | 100% |
| 4 | 100% | 100% | 100% | 100% |
| 8 | 100% | 100% | 100% | 99.95% |
| 13 | 100% | 99.40% | 99.12% | 98.90% |
| 30 | 100% | 98.75% | 98.57% | 98.32% |
| 50 | 100% | 98.50% | 98.14% | 98.05% |
| 104 | 100% | 98.01% | 97.75% | 97.58% |

Take the fivefold cross-validation results as an example. When the number of typing patterns increases from 2 to 4, the cross-validation accuracy remains at 100%. When the number of typing patterns increases from 4 to 50, the cross-validation accuracy decreases slightly from 100% to 98.05%. When the number of typing patterns reaches 104, the proposed method still achieves 97.58% cross-validation accuracy. From the experiment results, it is clear that, with the increase of number of patterns, it becomes more complicated and difficult to identify typing patterns. In most verification and identification application fields, such as financial management, bill payment, and personal computer access, the number of those who have access to the system is small, which means the number of typing patterns is small. By combining typing patterns with personal identification number, the IKB and the proposed method will significantly increase the system security.

Considering that deep learning method benefits from large amount of training data, the performance of the proposed method has the potential for further improvement with more

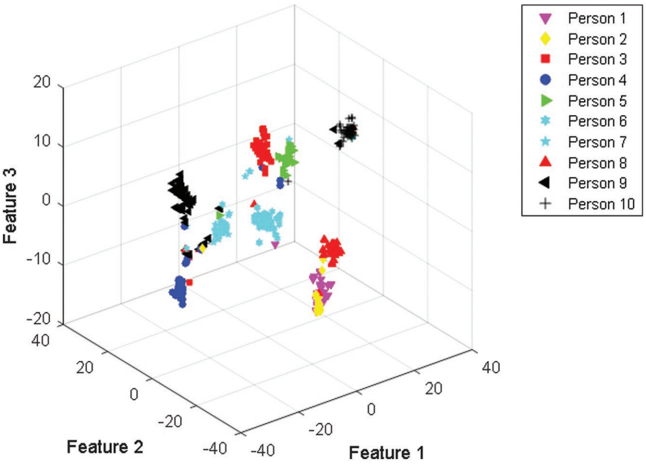

Figure 15. Visualization of features by the 1st hidden layer with 104 typing patterns.

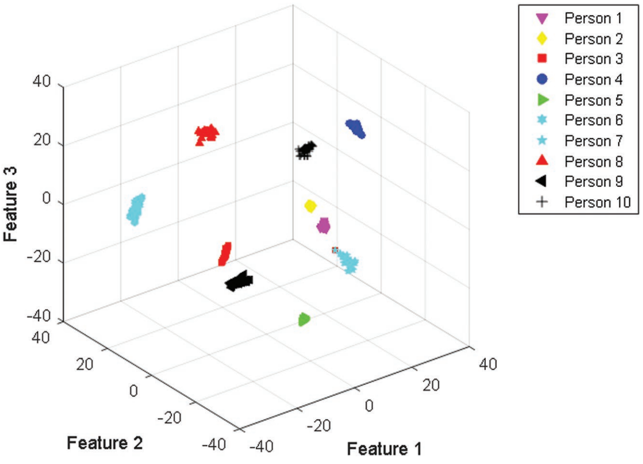

Figure 16. Visualization of features by the 2nd hidden layer with 104 typing patterns.

training data. Actually, it is very easy to obtain more training data by typing more words or more times. In addition, the proposed method is integrated with the intelligent keyboard, which is extremely cost-effective, less intrusive, and user friendly. It can be easily implemented as an additional layer of strong security on current authentication systems without major modification. Thus, the presented work is a practical step toward highly secured computer systems and networks. The proposed method has extensive applications in the fields of computer or network access control, cyber security, and person information management.

Based on the approach proposed by Wang and co-workers,[1,2] this paper presents a data analysis algorism of keystroke dynamics identification for intelligent keyboard using deep belief network. The design and experimental results of the proposed method are discussed in detail. One significant advantage of the proposed method is that it processes the current signals collected from IKB directly to automatically and adaptively extract features for typing pattern recognition. The proposed method does not rely on human knowledge and complex signal processing techniques. Experiments on 104 typing datasets are presented to demonstrate the effectiveness of the proposed method. The results show that the proposed method has high identification accuracy, stable, and reliable performance. The proposed method can be applied to computer or network access control, cyber security, and person information management. Our future work will focus on practical application of the proposed system.

## Acknowledgements

This work was supported by China Scholarship Fund, Hightower Chair foundation, and the ASPIRE grant program at the University of South Carolina (United States).

## Conflict of Interest

The authors declare no conflict of interest.

## Keywords

deep belief network, feature extraction, intelligent keyboard, keystroke dynamics identification

Received: May 18, 2018
Revised: July 1, 2018
Published online:

[1] J. Chen, G. Zhu, J. Yang, Q. Jing, P. Bai, W. Yang, X. Qi, Y. Su, Z. L. Wang, *ACS Nano* **2015**, *9*, 105.
[2] C. Wu, C. S. Ding, W. B. Liu, R. Y. Wang, J. Y. Wang, C. Aurelia, J. Wang, S. M. Li, Y. L. Zi, Z. L. Wang, *Mater. Today* **2018**, *21*, 216.
[3] Choo, K. K. R , *Sci. Rep.* **2011**, *4*, 719.
[4] S. Kraemer, P. Carayon, J. Clem, *Comput. Secur.* **2009**, *28*, 509.
[5] D. Besnard, B. Arief, *Comput. Secur.* **2004**, *23*, 253.
[6] M. Karnan, M. Akila, N. Krishnaraj, *Appl. Soft Comput.* **2011**, *11*, 1565.
[7] R. S. Gaines, W. Lisowski, S. Press, S. J., N. Shapiro, *RAND Corp.* **1980**, 1.
[8] Y. Sheng, V. V. Phoha, S. M. Rovnyak, *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **2005**, *35*, 826
[9] L. C. F. Araujo, M. G. Lizarraga, L. L. Ling, J. B. T. Yabu-Uti, *IEEE Trans. Signal Process* **2005**, *53*, 851.
[10] E. Yu, S. Cho, *Comput. Secur.* **2004**, *23*, 428.
[11] S. Hwang, S. Cho, S. Park, *Comput. Secur.* **2009**, *28*, 85.
[12] F. Monrose, A. D. Rubin, *Comp. Syst.* **2000**, *16*, 351.
[13] F. Bergadan, D. Gunetti, C. Picardi, *ACM T. Inform. Syst. Se* **2002**, *5*, 367.
[14] Y. LeCun, Y. Bengio, G. Hinton, *Nature* **2015**, *521*, 436.
[15] G. Q. Zhao, G. H. Zhang, Q. Q. Ge, X. Y. Liu, PHM Conf. Research advances in fault diagnosis and prognostic based on deep learning, Chengdu, China, October **2016**.
[16] G. E. Hinton, R. R. Salakhutdinov, *Science* **2006**, *313*, 504.
[17] Z. L. Wang, L. Lin, J. Chen, S. M. Niu, Y. L. Zi, *Triboelectric Nanogenerators*, Springer International Publishing, NY, USA **2016**.
[18] F. R. Fan, R. Feng, Z. Q. Tian, Z. L. Wang, *Nano Energy* **2012**, *1*, 328.
[19] S. M. Niu, X. F. Wang, F. Yi, Y. S. Zhou, Z. L. Wang, *Nat. Commun.* **2015**, *6*, 8975.
[20] a) G. Zhu, W. Q. Yang, T. Zhang, Q. Jing, J. Chen, Y. S. Zhou, P. Bai, Z. L. Wang, *Nano Lett.* **2014**, *14*, 3208; b) H. Shao, H. Jiang, X. Zhang, M. Niu, *Meas. Sci. Technol.* **2015**, *26*, 1.
[21] G. E. Hinton, S. Osindero, Y. W. Teh, *Neural Comput.* **2006**, *18*, 1527.
[22] G. E. Hinton, *Neural Comput.* **2002**, *14*, 1771.
[23] L. V. D. Maaten, G. Hinton, *J. Mach. Learn. Research* **2008**, *9*, 2579.