2019 IEEE International Conference on Data Mining (ICDM)

Aftershock Detection with Multi-Scale Description based Neural Network

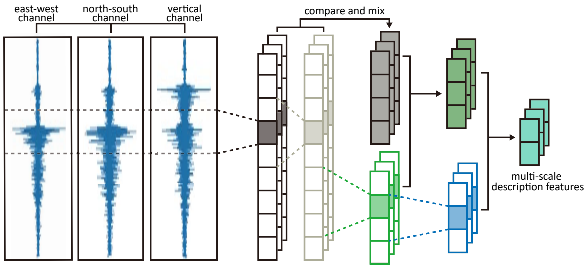
Qi Zhang[1,2], Tong Xu[1,*], Hengshu Zhu[2,*], Lifu Zhang[1], Hui Xiong[1,2,3], Enhong Chen[1], Qi Liu[1]
[1]Anhui Province Key Lab of Big Data Analysis and Application, University of Science and Technology of China
[2]Baidu Talent Intelligence Center, Baidu Inc, [3]Business Intelligence Lab, Baidu Research
{zq26, zlf123}@mail.ustc.edu.cn,{tongxu, cheneh, qiliuql}@ustc.edu.cn,
zhuhengshu@baidu.com, xionghui@gmail.com

***Abstract*—Aftershocks refer to the smaller earthquakes that occur following large earthquakes, in the same area of the main shock. The task of aftershocks detection, as a crucial and challenging issue in disaster monitoring, has attracted wide research attention in relevant fields. Compared with the traditional detection methods like STA/LTA algorithms or heuristic matching, neural network techniques are regarded as an advanced choice with better pattern recognition ability. However, current neural network-based solutions mainly formulate the seismic wave as ordinary time series, where existing techniques are directly deployed without adaption, and thus fail to obtain competitive performance on the intensive and highly-noise waveforms of aftershocks. To that end, in this paper, we propose a novel framework named Multi-Scale Description based Neural Network (MSDNN) for enhancing aftershock detection. Specifically, MSDNN contains a delicately-designed network structure for capturing both short-term scale and long-term scale seismic features. Therefore, the unique characteristics of seismic waveforms can be fully-exploited for aftershock detection. Furthermore, a multi-task learning strategy is introduced to model the seismic waveforms of multiple monitoring stations simultaneously, which can not only refine the detection performance but also provide additionally quantitative clues for discovering homologous earthquakes. Finally, comprehensive experiments on the data set from aftershocks of the Wenchuan M8.0 Earthquake have clearly validated the effectiveness of our framework compared with several state-of-the-art baselines.**

***Index Terms*—Multi-Scale Description, Multi-Task Learning, Aftershock Detection**

I. Introduction

Nature always teaches human beings via disasters. Earthquake is one kind of worst nature disasters which may cause injury and loss of life and collapse of buildings. The solutions for automatic earthquake detection are regarded as crucial to support a variety of emergence actions, and have attracted wide attention by seismologist in the past decades. Along this line, a critical challenge is how to effectively detect aftershocks, which refer to the smaller earthquakes that occur following a large earthquake, in the same area of the main shock. Different from the main shock, aftershocks usually have intensive waveforms with highly-noise and weak signal, which limit the performance of traditional detection methods [1]. Therefore, aftershock detection task requires an effective method which can describe the seismic waveforms according to the characteristics of aftershocks.



(a) original waveforms (b) multi-scale features extracting

Fig. 1: The diagrammatic sketch of extracting the multi-scale description of seismic waves.

In the literature, prior arts for automatic earthquake detection are mainly based on classic STA/LTA algorithms [2]–[4] or heuristic template matching [5], [6]. Traditionally, the STA/LTA algorithms, which are also known as the energy ratio criteria based methods, will define a characteristic function of waveforms for the first step, and then measure the difference between the short-term average (STA) and the long-term average (LTA) of this characteristic function. At the same time, the heuristic template matching assumes that similar waveforms may indicate similar seismic mechanism or even "repeating earthquakes". However, these methods are usually either noise-sensitive, or computational extensive for large-scale detection tasks [6]. Recently, with the rapid development of deep learning technology, neural networks are applied for earthquake detection task as a powerful method for modeling waveforms [7], [8]. For instance, [9] proved that neural networks are resistant to noise with better flexibility and generalization ability. However, current neural network based solutions mainly formulate the seismic waves as ordinary time series, where existing neural network structures are directly deployed without adaption for fully utilizing seismic features. Therefore, they may still fail to satisfy the aftershocks detection task, which usually have frequent and insignificant fluctuation of waveforms compared with large earthquakes.

To that end, in this paper we propose a novel framework named Multi-Scale Description based Neural Network (MSDNN) for aftershock detection task. Specifically, MSDNN attempts to extract the multi-scale description of seismic waves for fully revealing the latent characteristics in seismic waves of aftershocks, which inspired by the classic STA/LTA methods

[*] Corresponding Author

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00099

mentioned above. According to the STA/LTA methods, the long-term scale features could reflect the relatively stable background of geological characteristics, while the short-term scale features may reflect the real-time seismic status. Thus, the comparison between features with different scales could be significant clues to indicate the potential geological incidents.

The multi-scale description structure is summarized in Figure 1, where the left part shows original waveforms with three channels. Also, the right part shows that different scales of features are extracted from the waveforms, and then "compared and mixed" to generate new mixed features. The new mixed features will be "compared and mixed" with new scales features, either, which leads to the iteration process until waves within a pre-defined time window are resolved, and the final mixed features (i.e., the multi-scale description features) could better reveal the differences between aftershocks and noises. Meanwhile, the neural network structure of MSDNN is inspired by the (1×1) convolutional layer of Inception Net [10], [11] and memory unit of LSTM network [12]. Moreover, one earthquake usually can be captured by several monitoring stations and record as several different waveforms, which leads to homologous earthquake waveforms. For better utilizing the relationship between waveforms in different monitoring stations, we regard homology detection as a sub-task and develop a multi-task learning strategy [13], [14] for refining the detection performance. To be specific, the contributions of this paper can be summarized as follows:

- We propose a novel neural network based solution, i.e., MSDNN, for the aftershock detection task, which adapts the traditional STA/LTA methods with integrating the advanced neural networks, and fully exploits the seismic features with multi-scale description.
- We design a multi-task learning strategy to better describe the relationship between seismic waveforms recorded by different monitoring stations, which further refines the performance of aftershock detection.
- We evaluate our framework with extensive experiments on the data set from aftershocks of the Wenchuan M8.0 Earthquake. The experimental results clearly validate the effectiveness of MSDNN, compared with several state-of-the-art baselines.

Overview. The rest of this paper is organized as follows. In Section II, we briefly introduce the related works of this paper. In Section III, we introduce the characteristics of the data and the motivations of our approach. Afterwards, the details of our MSDNN framework and multi-task learning strategy will be introduced in Section IV. We comprehensively evaluate the performances of our framework for aftershock detection, and then conduct some further discussion in Section V. Finally, in Section VI, we conclude the paper.

II. Related Work

In the literature, solutions for aftershock detection task could be roughly divided as two types, namely the *traditional solutions* based on heuristic methods, as well as the *machine-learning-based solutions* mainly based on neural networks with different structures.

Traditional Solutions. Guided by practical experiences, seismologists usually designed solutions following statistical analysis or heuristic template matching. For instance, the classic *STA/LTA* [2] algorithms, as well as its variations, e.g., *Allen Picker* [15], *BK87* [3] and so on [16]–[18], are comprehensively summarized and compared in [19], and then enhanced in [4] with the *FilterPicker* algorithm. This solution applies several filters to summarize the characteristic function of seismic waves, thus is applicable to the real-time seismic monitoring with adequate performance. However, these solutions are easily disturbed by the noises, e.g., artificial explosions, which severely limit the performance.

At the same time, some other researchers attempted to reveal earthquakes via heuristic template matching, since similar waves may indicate similar earthquake mechanism, or even "repeating earthquakes". These attempts, e.g., [5], [6], [20]–[22], have been proven as sensitive and discriminative solutions for finding a "repeating earthquakes" from seismograms, but their computational burden could be extremely expensive and the generalization ability could be poor. Thus, it will be truly difficult to ensure the efficiency and robustness. Meanwhile, some prior arts target at improving the efficiency via interdisciplinary technique, e.g., [23], [24] applied search engine techniques to find similar templates, so that the parameters of earthquake could be inferred within even a second. However, these solutions only can detect earthquake waveforms which are similar to the waveforms in seismic observed data.

Machine-learning-based Solutions. Due to the powerful fitting and generalization ability, machine-learning-based methods are widely used for a variety of sequential problems [25]–[28]. Since firstly proposed in [29] and [30] which applied the fully-connected neural networks, the machine-learning-based solutions are widely studied as a competitive choice for earthquake detection task with better flexibility and generalization ability, especially for the detection task with much noise [9]. Compared with the prior arts like [31]–[33] which mainly rely on the classic neural network structure, recently, the convolutional neural network structure has been treated as an accurate and efficient [34], [35] solution. Along this line, [7] adapted the convolutional neural network to achieve better performance, without storing the perpetually growing waveform templates. Though great achievements have been made, these solutions mainly formulate the seismic waves as ordinary time series, while seismic structure characteristics and the relationship between different monitoring stations are not fully utilized.

Compared with the main shocks of large earthquakes, aftershocks usually have much higher frequency and insignificant fluctuation of waveforms. Therefore, both of the above solutions often fail to achieve satisfied performance in terms of aftershock detection. In order to effectively distinguish aftershocks from noises, we propose a novel neural network based solution, named MSDNN.

III. Preliminaries

In this section, we briefly describe the real-world aftershock data set used in our study, and then clarify the motivations of our framework in detail, which is inspired by the characteristics of the seismic data.

A. Data Set Description

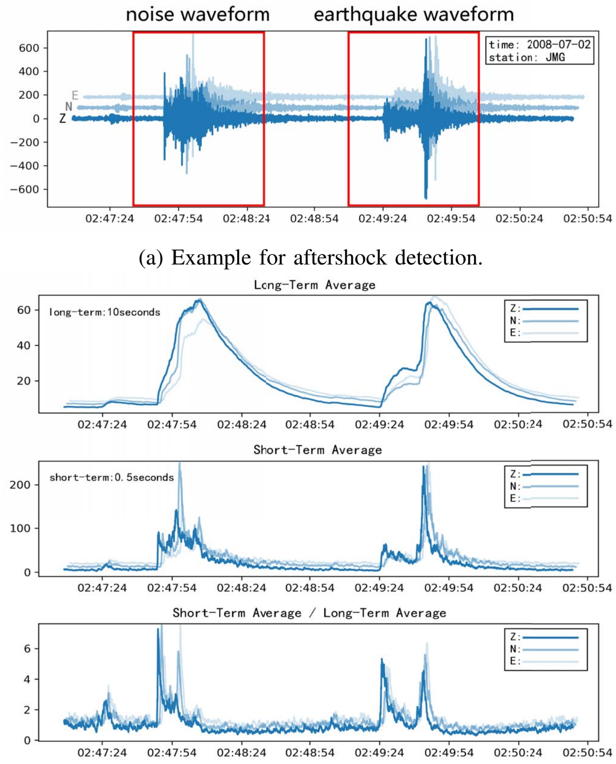
The seismic data set used in our paper is the monitoring signal from aftershocks of the Wenchuan M8.0 Earthquake. To be specific, there are 2,833 aftershocks, corresponding to 9,891 pieces of seismic waveforms in short time window. And all signals monitored by stations were recorded in three spatial dimensions (i.e., Z for the vertical channel, N for the north-south channel, and E for the east-west channel) by 15 monitoring stations. The frequency of the signals is 100Hz, and we preprocessed the signals by Bessel filter with 2-10Hz bandwidth for removing unexpected disturbing. Figure 2a shows an example of our seismic data during a short time period. Actually, the seismic signals always contain noise waveforms which are extremely difficult to be distinguished from aftershock waveforms.

B. Characteristics of the Data

Before introducing the technical details of our approach to aftershock detection, here we discuss some important characteristics, which significantly motivate the design of our MSDNN model.

First, the seismic signal data contain three signals in three spatial dimensions, which is different from general signal data that only contains single signal. However, traditional earthquake detection approaches usually analyze these signals separately, instead of modeling them in a holistic manner. In order to maximize the utilization of seismic information, we need to consider three signals simultaneously. Meanwhile, convolutional neural network can capture multi-signal as multi-channel, which is consistent with the characteristics of seismic data. Moreover, the powerful feature extraction capability of the convolutional neural networks can effectively capture the unique characteristics of seismic waveforms.

Second, inspiring by the most widely-used earthquake detection approach, i.e., STA/LTA [2]–[4], we extract the LTA and STA of the signals. For example, in Figure 2b, we set the time period of long-term and short-term as 10 second (1000-time steps) and 0.5 second (50-time steps), respectively. It can be seen that the long-term average reflects the trend information and the background level of signal, however, the information of local variation is insufficient. Correspondingly, the short-term average may reflect the real-time change of the signal without random fluctuations, while the trend information is not clear. Therefore, the description of different scales is required to reflect different level of features for signal. To that end, we introduce $\frac{STA}{LTA}$ to present the comparison, as well as mixture of different scale-aware descriptions, as shown in the bottom of Figure 2b. It can be seen that $\frac{STA}{LTA}$ is extremely sensitive to the change of signals and reflects the arrival time of waves. Indeed, the traditional approach mainly take advantage of these characteristics for aftershock detection. Based on these characteristics, we design a multi-scale description structure in our neural network to extract different scale-aware features, and then mix these features to obtain special features for improving the performance of aftershock detection.



(a) Example for aftershock detection.

(b) Example of classical multi-scale description.

Fig. 2: Some motivating examples of our multi-scale description based aftershock detection approach.

TABLE I: The number of aftershock waveforms.

Station	Number	Homology Number	Percent
JMG	1,208	1,208	100%
YZP	1,072	1,015	94.7%
QCH	894	890	99.6%
PUW	1,350	1,338	99.1%
WXT	839	838	99.9%
SPA	574	574	100%
XJI	614	612	99.7%
HSH	821	821	100%
YGD	166	166	100%
JJS	908	903	99.4%
MXI	1,215	1,196	98.4%
XCO	223	223	100%
WDT	6	6	100%
MIAX	1	1	100%
SUM	9,891	9,791	99.0%

Third, each earthquake could be usually captured by mul-

TABLE II: The mathematical notations.

Symbol	Description
D	The data set of waveform windows
S_i	The input memory status of i-th MSD-cell
F_i	The input feature status of i-th MSD-cell
S_{i+1}	The output memory status of i-th MSD-cell
F_{i+1}	The output feature status of i-th MSD-cell
F_i^c	The current scale feature of i-th MSD-cell
J_i	The joint feature of S_i and F_i^c
$J_{mix,i}$	The comparison and mixture feature
$W_{c,i}$	The parameter matrix of $(1 \times 3 \times 32/1)$ convolutions
$W_{m,i}$	The parameter matrix of $(1 \times 1 \times 32/1)$ convolutions
d	The element of D
y	The classification result of d
l	The label of d
$j_{n,i}$	The n-th channel of J
$j_k^{mix,i}$	The k-th channel of J_{mix}
$\alpha_{n,k,i}$	The $(n \times k)$-th kernel of W_m

tiple monitoring stations, which results in the homologous earthquake waveforms (i.e., the waveforms generated by the same earthquake). According to the summarization in Table I, we realize that almost all the earthquakes may generate homologous waveforms. Thus, we set homology detection as a sub-task, and propose a multi-task learning strategy [13], [14] for improving the performance of aftershock detection.

In summary, the neural network approach we proposed in this paper is based on the multi-scale description structure and the multi-task learning strategy. In follwing section, we will provide technical details regarding how we design the aftershock detection approach.

IV. Technical Framework

In this section, we first introduce the problem formulation of this paper, and then introduce the technical details of our MS-DNN framework. For better illustration, related mathematical notations used in this paper are summarized in Table II.

A. Problem Formulation

The problem studied in this paper is to use machine learning technologies for aftershock detection, which focuses on distinguishing the primary waves (P-waves) of aftershocks from other waveforms. Specifically, to formulate the problem, we use D to denote a data set of n equal-length waveforms windows, represented as $D = \{d_1, d_2, ..., d_n\}$. The waveform windows are sliced from real-time signals, and each of them contains 3 channels waveforms corresponding to three spatial dimensions (i.e., vertical, north-south, and east-west), denoted as $d_i = \{z_i, n_i, e_i\}$. Correspondingly, we have a label $l_i \in \{0, 1\}$ to indicate whether the d_i contains an aftershock P-wave arrival. Therefore, the problem of machine-learning-based aftershock detection can be defined as follow.

Definition 1. **Machine-learning-based Aftershock Detection.** Given a set of waveform windows D, where each $d_i \in D$ has a label l_i for indicating the existence of seismic P-wave, the objective is to learn a predictive model $\mathscr{M}$ for classifying waveform windows with respect to the label y_i.

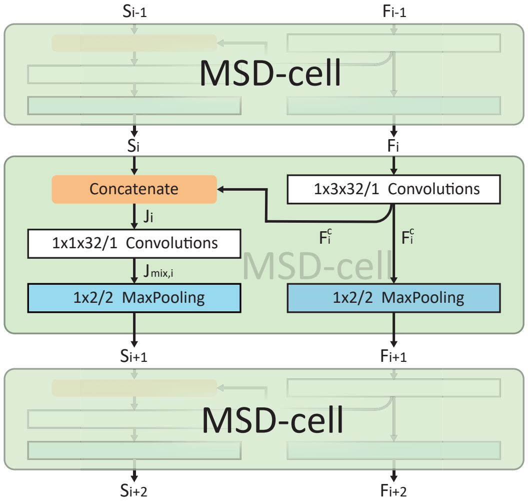


Fig. 3: The detailed structure of MSD-cell, which can be expanded easily.

B. MSD-cell: Generating Multi-Scale Description

As discussed in section III, the comparison between different scale features might be significant clues to indicate the potential geological activities. Therefore, inspired by the classic STA/LTA methods, we develop a unique module in MSDNN for extracting the multi-scale description of seismic waves, which is named MSD-cell. Along this line, the module needs to implement two key functions. The first function can remember prior features on different scales and add new scale feature, while the second function can compare and mix these two kinds of features. Thus, multi-scale description can be extracted in the process of constantly comparing and mixing new scale feature with prior features. Because of the similar ability to the first function, the memory unit of LSTM networks [13], [36], which can remember long time information and add new short time information, could be adapted as a reliable framework. Correspondingly, for the second function, the Inception Net [10], which combines different filters together in convolutional neural network, is chosen to compare and mix different features.

The detailed structure of MSD-cell will be introduced as follows. In each MSD-cell, as shown in Figure 3, there are two inputs, namely S_i and F_i, where S_i is a memory status and F_i is a feature status. Specifically, S_i is used to store multi-scale features, and F_i represents input scale feature. When F_i enters MSD-cell, a $(1 \times 3 \times 32/1)$ convolutional layer (i.e., convolutional layer with (1×3) kernel size, 32 channels and 1 strides) will be applied on it to get a feature on higher scale, named F_i^c, which is the current scale feature of this cell. Then, on the one hand, F_i^c will go through a $(1 \times 2/2)$ max-pooling layer, with the output as F_{i+1}. On the other hand, F_i^c will be jointed with S_i to prepare for comparison and mixture. The output of the joint operation is denoted as J_i, which will go through a $(1 \times 1 \times 32/2)$ convolution layer to compare and mix the multi-scale features S_i with the current scale feature F_i^c. Finally, it will be passed through a $(1 \times 2/2)$ max-pooling layer, with the output as S_{i+1}.

The detailed structure of MSD-cell is illustrated in Figure 3. Formally, for the output S_{i+1} and F_{i+1}, we have

$$\begin{aligned} F_i^c &= relu(conv(F_i, W_{c,i})), & (1)\\ J_i &= concat(S_i, F_i^c), & (2)\\ J_{mix,i} &= relu(conv(J_i, W_{m,i})), & (3)\\ S_{i+1} &= maxpool(J_{mix,i}), & (4)\\ F_{i+1} &= maxpool(F_i^c), & (5) \end{aligned}$$

where $W_{c,i}$ and $W_{m,i}$ are the parameter matrixes of $(1 \times 3 \times 32/1)$ convolutional layer and $(1 \times 1 \times 32/1)$ convolutional layer in one MSD-cell, respectively. At the same time, functions in these formulas are defined as follows:

- $relu()$: the non-linear activation function.
- $conv()$: the convolutional layer.
- $concat()$: the concatenation of two matrixes along the last dimension.
- $maxpool()$: the max-pooling layer with 2 strides.

According to these formulas, the feature status F_i increases the feature scale and evolves into F_{i+1}, the memory status S_i involves new scale feature F_i^c and evolves into S_{i+1}. Specially, the key of multi-scale comparison and mixture is Equation 3. For understanding the Equation 3 in detail, we have

$$j_{k,i}^{mix} = relu(\sum_{n=1}^{C_i} j_{n,i} \times \alpha_{n,k,i}), \quad (6)$$

where C_i is the channels number of input J_i, k means the channel of output $J_{mix,i}$, and α denotes the coefficients which need learning. The objective of this layer is to multiply each channel of the input, i.e, $j_{n,i}$, by the coefficients, and then sum them together. Different output channels will use different coefficient schemes. The half of the channels in J_i are multi-scale features that contain information of all prior different scale features S_i, the other half are the current scale features F_i^c. Therefore, after passing the convolutional layer with (1×1) kernel size, these two parts will be compared and mixed in various schemes, and the coefficients in schemes reflect the comparison way between different features. After that, the features of different scales are correlated, and the information of current scale feature is added to memory status S_{i+1}. In our experiment, similar with the heuristic solution in [7], the output channel numbers of convolutional layers are set to 32. The max-pooling layers are set at the end of MSD-cell to ensure the feature map size consistently when comparing and mixing, and to expand the scale of feature in next cell.

C. Multi-Scale Description based Neural Network

For the aftershock detection task, the input is a waveform group of three channels, corresponding to three spatial dimensions (i.e., vertical channel, north-south channel, and east-west channel). Therefore, the size of the input data is $1 \times m \times 3$, where m represents the length of the detection window. The data sampling frequency used in our experiment is 100Hz, and the length of detection window is set to 50s, namely $m = 5000$. In our MSDNN framework, the beginning of the neural network is a $(1 \times 3 \times 32/1)$ convolutional layer, which can initialize the data and prepare for the input to MSD-cells. Then, the output of the convolutional layer is fed into both two inputs of MSD-cell as the first scale feature, and two outputs of this cell are fed into next MSD-cell. The size of the feature map is reduced by half for each time passing an MSD-cell. There are 10 MSD-cells in total, where the output of the last MSD-cell is S_{11} and F_{11}, and the sizes of them are both $1 \times 5 \times 32$. We take the output S_{11} as the multi-scale description feature, which contains the information of all scale description features. After that, S_{10} is fed into two fully connected layers, with the output of the first layer being 128 and the output of the second layer being 2. Finally, we use softmax function to get the classification result of the aftershock detection.

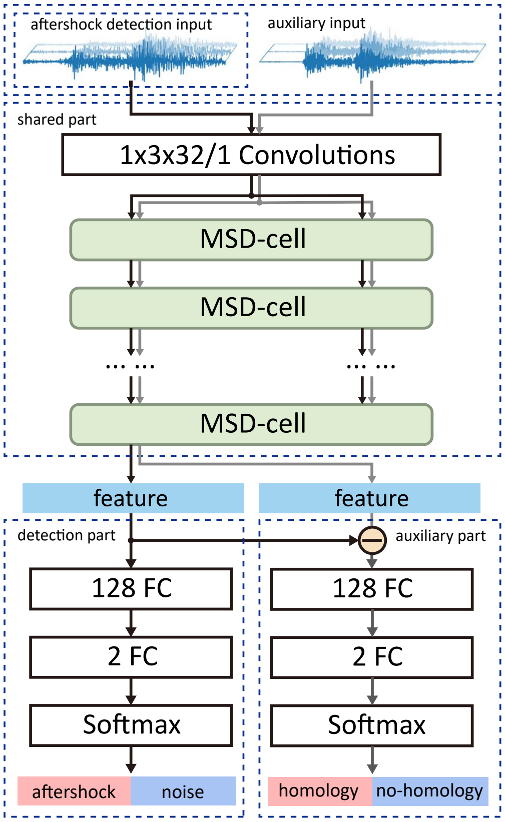


Fig. 4: The framework of MSDNN and multi-task learning, which is divided into shared, detection and auxiliary part.

The *shared part* and *detection part* in Figure 4 show the detailed structure of MSDNN, and the *shared part* will be shared in multi-task learning strategy. Since our network is relatively deep, for improving the efficiency and effectiveness of training process, as well as preventing the problem of overfitting and gradient disappearance in backpropagation, we add

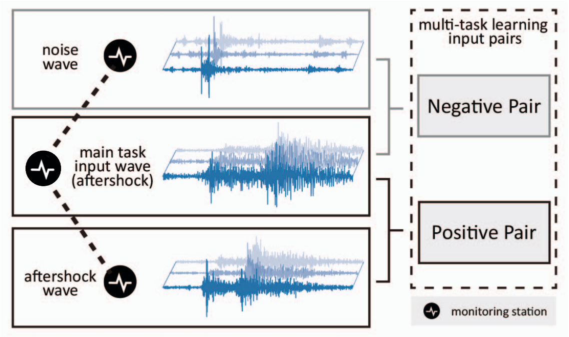


Fig. 5: Sampling pairs of multi-task learning.

batch normalization after each convolutional layer and fully connected layer [37], and add parameter regularization to the optimization goal. The main optimization goal in our network is the cross-entropy loss, which is widely used in classification problems. The loss function of MSDNN can be written by:

$$L_{main} = -\frac{1}{n}\sum_{i=1}^{n}\Big[l_i \log y_i + (1-l_i)\log(1-y_i)\Big] + \frac{\lambda}{2n}\sum_{w}||w||^2$$

where $y_i \in \{0,1\}$ represents the prediction of input ($y_i = 1$ means there exists an aftershock, and vice versa), $l_i \in \{0,1\}$ represents the label of input, and n is the number of the inputs batch. The second part is the $L2$ regularization term, and λ is the regularization coefficient, reflecting the degree of regularization constraints.

D. Multi-Task Learning Strategy

Finally, we turn to introduce our novel multi-task learning strategy for improving the detection performance. This strategy aims at a unique characteristic of seismic data, i.e., when an earthquake occurs, it is usually detected by multiple monitoring stations, which results in the homologous seismic waves with the same seismic source mechanism. Therefore, we want to leverage the information from multi-stations for improving the performance of aftershock detection.

In this paper, we construct an auxiliary task of homologous earthquake detection to form a multi-task learning strategy together with the aftershock detection task. The objective of this auxiliary task is to determine whether the two seismic waveforms are homologous. Thus, we treat each pair of waveforms as the input for this auxiliary task, which is identical to the input of the aftershock detection network. Specifically, for each pair of waveforms detected by different stations, if the label of pair is $True$, both waveforms in the pair are mutually homologous earthquakes. On the contrary, the label as $False$ means that the pair of waveforms are not homologous. Figure 5 shows the inputs of multi-stations.

Moreover, multi-task learning requires that the networks of main task (i.e., aftershock detection task) and the auxiliary task need have some parameters shared, thus it can leverage the domain-specific information contained in the training signals of auxiliary tasks to improve the detection performance. To this end, we share all MSD-cells. For easily sharing the begin part of the network, the paired inputs of the auxiliary task can be seen as the parallel inputs, which is the same with the main task. After passing through the shared part, two features corresponding to the input can be obtained. Here, we hope that if the input pair are homologous earthquakes, the features of them are similar. If one of the pair is not an aftershock, that is, the pair are not homologous earthquakes, the features of them are difference. So that, the features obtained by shared part are more distinguishable for aftershock waveforms. In order to achieve this goal, in auxiliary part, we first subtract the two features obtained by shared part. Then, we send the output to two fully-connected layer. The output of the first layer is 128, and the output of the second layer is 2. Finally, we use the softmax function to get the classification result of the homologous earthquake detection. Figure 4 shows the detailed structure of multi-task learning strategy. For the auxiliary task, we still use the cross entropy as the main optimization goal. The final loss function of our MSDNN framework with multi-task learning strategy can be written as

$$\begin{aligned} L_{multi} &= L_{main} + \frac{\lambda}{2k}\sum_{w^h}||w^h||^2 \\ &\quad -\frac{1}{k}\sum_{i=1}^{k}\Big[l_i^h \log y_i^h + (1-l_i^h)\log(1-y_i^h)\Big], \end{aligned}$$

where the first part is the loss function of aftershock detection task. When a pair is subjected to homology detection, they also perform aftershock detection separately. The second part is the $L2$ regularization term. The third part is the cross entropy of auxiliary task, where y_i^h, $l_i^h \in \{0,1\}$. When $y_i^h = 1$, the waveforms will be predicted as homologous. On the contrary, $y_i^h = 0$ indicates the non-homologous ones. Similarly, l_i^h represents the label of input pair.

V. Experiments

To validate the performance of MSDNN framework, in this section, we conduct a series of experiments on a large-scale real-world data set from aftershocks of the Wenchuan M8.0 Earthquake [38]. Also, some empirical case studies and discussions will be presented.

A. Data Pre-processing

As introduced in section III, the real-world data set is the monitoring signal from aftershocks of the Wenchuan M8.0 Earthquake during July 1-31, 2008, which is provided by the China Earthquake Administration. We preprocessed the signals by Bessel filter with 2-10Hz bandwidth for removing unexpected disturbing.

Samples for Aftershock Detection. For labeling the samples, intuitively, 9,891 waveform windows which contains aftershocks were treated as positive samples. At the same time, considering that in most cases, the signals monitored by stations kept as stable as nearly a straight line, thus random sampling for negative samples would extremely ease the discrimination, as waveform for aftershock could be much more significant. To that end, we generated the negative samples by the *FilterPicker* [4] model, i.e., those "aftershocks" which were wrongly captured by *FilterPicker*, but not belong to the 9,891 waveform windows, were treated as negative samples. Totally, 109,719 waveform windows were captured as "negative". In this case, the aftershock detection task would be challenging enough.

However, the number of negative samples are much more than positive samples, which results in the imbalanced data set. As mentioned in [39], the distribution of training data would severely impact the performance, as a balanced training set could be optimal. Therefore, similar with the heuristic solution in [7], we generated additional "*aftershocks*" by perturbing existing ones with zero-mean Gaussian noise, whose signal-to-noise ratio was set between 20-80dB, thus the waveforms won't be violently influenced. Finally, the amount of positive samples was equal with the negative ones.

Samples for Multi-task Learning. Along this line, to build the specific data set for multi-task learning framework, we grouped the waveforms based on their distances and time differences. In detail, the distance between epicenter and monitoring station should be within the range as the time difference multiply the spread speed of seismic waves, namely the minimal spread speed is 3km/s. Correspondingly, if two monitoring stations captured aftershock waveforms by the same epicenter, they should be grouped together as *homologous earthquakes*. What should be noted is that, as mentioned above, we considered both "*true pairs*" (i.e., pairs of positive samples) and "*false pairs*" (i.e., at least one negative sample), as shown in Figure 5, where the similarities between "true pairs" should be minimized, and vice versa. Finally, 642,112 "true pairs" and 288,962 "false pairs" were captured in total.

B. Experimental Settings

Details of Implementation. Our MSDNN framework is implemented by the TensorFlow framework [42]. Specifically, the mini-batch size of *Stochastic Gradient Descent* (SGD) was set as 8 for the aftershock detection task (main task). Along this line, pairs including these 8 inputs in the multi-task learning training set were selected to form the mini-batch of multi-task learning task. During the training process, both main task and multi-task learning were trained in turn, so that the multi-task learning acted as a constraint to guide network structure developing in a beneficial direction.

All the tasks were trained using Momentum [43] with a decay rate as 0.8. The initial learning rate of main task was set as 0.02. However, to reduce the constraint effect of multi-task learning, the learning rate of multi-task learning was set as 0.02*0.1 initially. Besides, ReLU was performed right after each Batch Normalization, except the output layer.

Baseline Methods To comprehensively validate the performance of MSDNN framework, we compared it with four types of baseline methods as follows:

- *ConvNetQuake* [7], which is the state-of-the-art method that firstly introduced the convolutional neural network to detect earthquakes. Currently, *ConvNetQuake* was treated as one of the most effective solutions for this task.
- *Inception Net* [11], which is one of state-of-the-art convolutional neural network methods for classification.
- *XGboost* [41] and *Random Forest* [40], which are representative methods for classification with ensemble learning, and perform well in real-world applications.
- *Support Vector Machine* and *Logistic Regression*, which are traditional solutions for classification task.

In order to compare each method in the same seismic information, the inputs of all the above methods and our method are the same.

Evaluation Metrics As a typical classification task, to measure the performance, we selected the *accuracy* metric to measure the overall effectiveness. Also, as an unbalanced classification task, we also selected the *precision* and *recall* metrics to measure the performance on positive samples, i.e., the aftershocks, which could be more important for our task. Finally, the F_1 metric was selected to measure the comprehensive effectiveness of *precision* and *recall.*

C. Overall Performance

First of all, we summarized the overall performance of validation. Specifically, considering the temporal correlation, we treated the former 5/6 of samples as training samples, while half of the rest 1/6 samples is the validation samples and another half is the test samples. Also, the length of time window for each sample was set as 50 seconds. The sensitiveness for these two parameters and the reason of temporal correlation data set split will be discussed in following subsection.

The results are illustrated in Table III. Unsurprisingly, we observed that our MSDNN methods consistently outperform all the baselines in terms of most of the metrics, especially for the F_1 score. Besides, the multi-task learning can improve performance in all metrics. Meanwhile, it can be observed that the Random Forest and XGboost achieve high performance of *recall*, but the performance of *precision* is noneffective, which means these methods have a high false positive rate and tend to classify waveforms as aftershocks. This is because that the models of these methods cannot learn the waveform features which can effectively distinguish between noise and aftershock waveforms. These low *precision* methods cannot be accept in real-world earthquake detection systems. When considering the performance of both *recall* and *precision*, the F_1 value can reflects the recognition ability of the model. The F_1 value of our method is 125% higher than Random Forest, 37.8% higher than XGboost, and 12.3% higher than the best baseline method, Inception Net. In other words, our methods can achieve a better balance between *recall* and *precision* for aftershock detection which is important for real-world earthquake detection systems.

D. Discussion with Experiment Settings

In this subsection, we turn to evaluate the experiment settings. In our MSDNN framework, three settings were con-

TABLE III: The overall performance.

Method	Accuracy	Recall	Precision	F_1
Logistic Regression	0.505	0.520	0.080	0.130
Support Vector Machine	0.515	0.520	0.080	0.130
Random Forest [40]	0.767	0.680	0.190	0.300
XGboost [41]	0.882	0.770	0.350	0.490
ConvNetQuake [7]	0.935	0.602	0.544	0.571
Inception Net [11]	0.941	0.637	0.582	0.608
Our Solutions				
MSDNN	0.952	0.638	0.678	0.658
MSDNN+Multi-task Learning	0.954	0.667	0.683	0.675

TABLE IV: Performance with different split strategy.

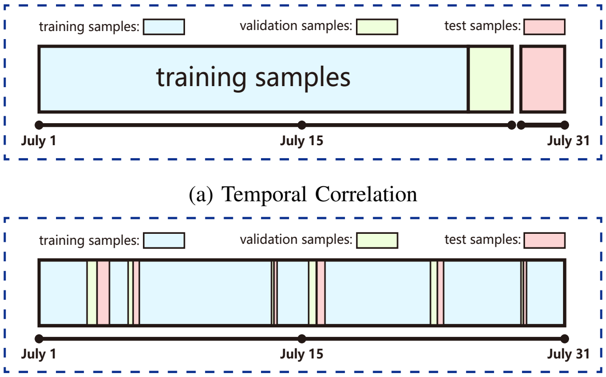
Strategy	Accuracy	Recall	Precision	F_1
Temporal Correlation	0.952	0.638	0.678	0.658
Random Mixing	0.963	0.651	0.859	0.740

TABLE V: Performance with ratio of training data.

Ratio	Accuracy	Recall	Precision	F_1
1/2	0.944	0.605	0.630	0.617
3/4	0.943	0.636	0.620	0.628
5/6	0.952	0.638	0.678	0.658

TABLE VI: Performance with different time window.

Length (s)	Accuracy	Recall	Precision	F_1
10	0.942	0.569	0.604	0.586
20	0.949	0.520	0.671	0.625
30	0.949	0.624	0.655	0.639
40	0.952	0.598	0.693	0.642
50	0.952	0.638	0.678	0.658



(a) Temporal Correlation

(b) Random Mixing

Fig. 6: Examples of our data set split strategies.

cerned, i.e., the strategy of data set split, the ratio of training samples, as well as the length of time window for each sample.

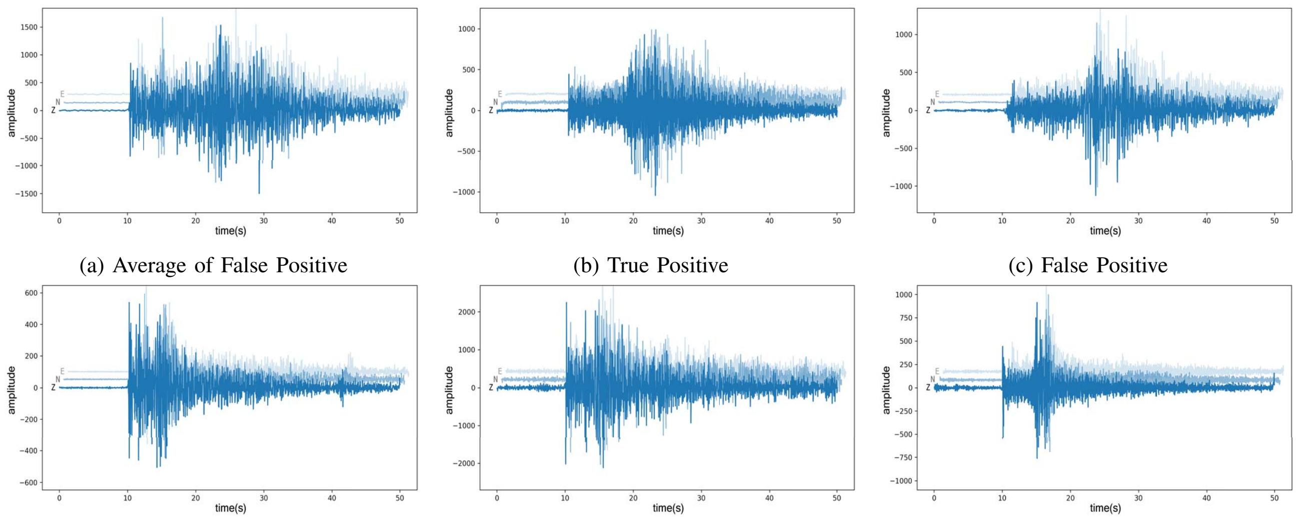
First, in order to discuss the temporal correlation of the data, we set two **strategies of data set split** . One strategy considered temporal correlation and treated the former 5/6 of samples as training samples, another strategy treated the random 5/6 of samples as training samples. Figure 6 shows these strategies in detail and Table IV shows the results of different data set split strategies. Obviously, the performance of random mixing split strategy is better than temporal correlation split strategy. It is because that the training samples with random mixing split strategy contained samples for the same period as the test samples [44], [45]. In real world experiments, the training samples cannot contain the samples of the test period, so we must split the data set according to the temporal correlation to conduct experiments. General cross-validation cannot be applied to this temporal correlation data set, thus, we used different ratio of training samples to evaluate the performance of our model. The experiments with different ratio of training samples will be discussed following.

For the **ratio of training samples**, we conducted experiments with three different ratios, namely 1/2, 3/4 and 5/6. All the samples were split by the temporal order so that their time dependence would not be destroyed. The results are shown in Table V. Obviously, the performance becomes better with increasing ratio of training samples, however, generally the improvements were not so significant, which indicates the robustness of our MSDNN framework, which keeps relatively stable with less training samples.

At the same time, for **the length of time window**, five different lengths were set, namely 10-50 seconds. According to the results in Table VI, as expected, generally most metrics becomes better with longer time window, especially for the F_1 metric, which is reasonable as richer information was collected. However, it is well known that a shorter time window leads to an earlier warning of upcoming aftershocks, so that more lives could be potentially saved. Thus, it is necessary to keep the balance between effectiveness and efficiency to ensure the capability.

E. Discussion with Case Studies

Finally, we turn to conduct some discussions to better understand the performance of our MSDNN framework. On the one hand, we would like to know in which position the MSDNN framework will fail to detect the aftershocks, so that further enhancement could be deployed. First, we picked up those "False Positive" samples, i.e., those negative samples that MSDNN detected as aftershocks. As shown in Figure 7, we can find that the average waveform (shown in Figure 7a), as well as correspondingly example (shown in Figure 7b) look quite similar with typical aftershocks (or True Positive sample,



(a) Average of False Positive

(b) True Positive

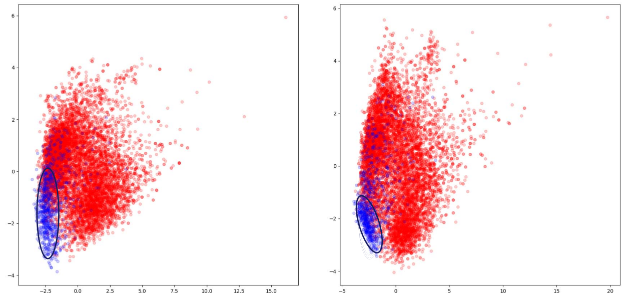
(c) False Positive

(d) Average of False Negative

(e) False Negative

(f) False Negative

Fig. 7: The average waveforms and samples.



(a) MSDNN

(b) MSDNN+Multi-task Learning

Fig. 8: The visualization of PCA results based on features learned by our approaches.

as shown in Figure 7c). Considering that the generation of ground truth (mainly by manual labeling) may miss some aftershocks, we asked several geophysical experts to re-check these 163 “False Positive” samples. Interestingly, 161 samples of them were labeled as “Positive”. To a certain degree, this phenomenon further proves the performance of MSDNN framework with making up the fault of manual checking.

Second, we picked up those “False Negative” samples, i.e., those positive samples that were missed by MSDNN. Based on Figure 7d, 7e, 7f, we realized that those waveforms with insignificant tail of second waves (S-wave), or the waveforms where primary waves (P-wave) and S-wave are close could be probably missed, which should be refined in the future.

On the other hand, we would like to check whether the multi-task learning framework indeed improve the effectiveness, not only based on the evaluating metrics. As mentioned above, the multi-task learning framework targets at refining the features of each sample, so that the samples in “true pairs” should be more similar, and vice versa. To that end, first, we counted the *Euclidean distances* between network features of positive and negative samples. The network features are the outputs of the shared part according to Figure 4. According to the results, without the multi-task learning framework, the average distance was 3.7. Then, this distance turned to 4.3 after deploying the multi-task learning framework, which proved that the differences between positive and negative samples were enlarged by the multi-task learning framework, and the discrimination task for aftershock detection is eased. Second, we applied Principal Component Analysis (PCA) to the network features for showing the effect of multi-task learning. As shown in Figure 8, blue dots are the positive samples and the red dots are negative samples. Compared the Figure 8a and Figure 8b, it can be seen that the positive blue dots are close together and easily separated from the red dots when multi-task learning is deployed. In summary, multi-task learning can optimizes the distribution of features and improves the performance of aftershock detection.

VI. Conclusion

In this paper, inspired by the classic STA/LTA method, we proposed a novel solution for the aftershock detection task, named Multi-Scale Description based Neural Network (MSDNN). Specifically, with considering both short-term scale and long-term scale seismic features and comparing them to each other, the MSDNN framework could fully describe the seismic waves. Along this line, we further deployed the multi-task learning framework to better analyze the seismic waves of multiple monitoring stations, so that additionally clues for discovering homologous earthquakes could be provided and the performance of aftershock detection were improved. Comprehensive experiments on the data set from aftershocks of the Wenchuan M8.0 Earthquake have clearly validated the effectiveness of our framework compared with several state-of-the-art baselines, which demonstrated the capability of our MSDNN framework in aftershock detection task.

Acknowledgment

This research was partially supported by grants from the National Natural Science Foundation of China (Grant No. 71531001, 61703386, U1605251).

References

[1] S. Stein and M. Liu, “Long aftershock sequences within continents and implications for earthquake hazard assessment,” *Nature*, vol. 462, no. 7269, p. 87, 2009.

[2] R. V. Allen, “Automatic earthquake recognition and timing from single traces,” *Bulletin of the Seismological Society of America*, vol. 68, no. 5, pp. 1521–1532, 1978.

[3] M. Baer and U. Kradolfer, “An automatic phase picker for local and teleseismic events,” *Bulletin of the Seismological Society of America*, vol. 77, no. 4, pp. 1437–1445, 1987.

[4] A. Lomax, C. Satriano, and M. Vassallo, “Automatic picker developments and optimization: Filterpicker—a robust, broadband picker for real-time seismic monitoring and earthquake early warning,” *Seismological Research Letters*, vol. 83, no. 3, pp. 531–540, 2012.

[5] S. J. Gibbons and F. Ringdal, “The detection of low magnitude seismic events using array-based waveform correlation,” *Geophysical Journal International*, vol. 165, no. 1, pp. 149–166, 2006.

[6] C. E. Yoon, O. O’Reilly, K. J. Bergen, and G. C. Beroza, “Earthquake detection through computationally efficient similarity search,” *Science advances*, vol. 1, no. 11, p. e1501057, 2015.

[7] T. Perol, M. Gharbi, and M. Denolle, “Convolutional neural network for earthquake detection and location,” *Science Advances*, vol. 4, no. 2, p. e1700578, 2018.

[8] S. Yuan, J. Liu, S. Wang, T. Wang, and P. Shi, “Seismic waveform classification and first-break picking using convolution neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 272–276, 2018.

[9] S. Gentili and A. Michelini, “Automatic picking of p and s phases using a neural tree,” *Journal of Seismology*, vol. 10, no. 1, pp. 39–63, 2006.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[11] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 5987–5995.

[12] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] S. Ruder, “An overview of multi-task learning in deep neural networks,” *arXiv*, 2017.

[14] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.

[15] R. Allen, “Automatic phase pickers: their present use and future prospects,” *Bulletin of the Seismological Society of America*, vol. 72, no. 6B, pp. S225–S242, 1982.

[16] P. S. Earle and P. M. Shearer, “Characterization of global seismograms using an automatic-picking algorithm,” *Bulletin of the Seismological Society of America*, vol. 84, no. 2, pp. 366–376, 1994.

[17] M. Joswig, “Pattern recognition for earthquake detection,” *Bulletin of the Seismological Society of America*, vol. 80, no. 1, pp. 170–186, 1990.

[18] B. Kennett and E. Engdahl, “Traveltimes for global earthquake location and phase identification,” *Geophysical Journal International*, vol. 105, no. 2, pp. 429–465, 1991.

[19] M. Withers, R. Aster, C. Young, J. Beiriger, M. Harris, S. Moore, and J. Trujillo, “A comparison of select trigger algorithms for automated global seismic phase and event detection,” *Bulletin of the Seismological Society of America*, vol. 88, no. 1, pp. 95–106, 1998.

[20] J. R. Brown, G. C. Beroza, and D. R. Shelly, “An autocorrelation method to detect low frequency earthquakes within tremor,” *Geophysical Research Letters*, vol. 35, no. 16, 2008.

[21] R. J. Skoumal, M. R. Brudzinski, B. S. Currie, and J. Levy, “Optimizing multi-station earthquake template matching through re-examination of the youngstown, ohio, sequence,” *Earth and Planetary Science Letters*, vol. 405, pp. 274–280, 2014.

[22] K. Plenkers, J. R. Ritter, and M. Schindler, “Low signal-to-noise event detection based on waveform stacking and cross-correlation: Application to a stimulation experiment,” *Journal of seismology*, vol. 17, no. 1, pp. 27–49, 2013.

[23] J. Zhang, H. Zhang, E. Chen, Y. Zheng, W. Kuang, and X. Zhang, “Real-time earthquake monitoring using a search engine method,” *Nature communications*, vol. 5, p. 5664, 2014.

[24] A. C. Aguiar and G. C. Beroza, “Pagerank for earthquakes,” *Seismological Research Letters*, vol. 85, no. 2, pp. 344–350, 2014.

[25] J. Sun, K. Xiao, C. Liu, W. Zhou, and H. Xiong, “Exploiting intra-day patterns for market shock prediction: A machine learning approach,” *Expert Systems with Applications*, vol. 127, pp. 272–281, 2019.

[26] K. Xiao, Q. Liu, C. Liu, and H. Xiong, “Price shock detection with an influence-based model of social attention,” *ACM Transactions on Management Information Systems (TMIS)*, vol. 9, no. 1, p. 2, 2018.

[27] X. Zhao, T. Xu, Y. Fu, E. Chen, and H. Guo, “Incorporating spatio-temporal smoothness for air quality inference,” in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 1177–1182.

[28] L. Zhang, H. Zhu, T. Xu, C. Zhu, C. Qin, H. Xiong, and E. Chen, “Large-scale talent flow forecast with dynamic latent factor model?” in *The World Wide Web Conference*. ACM, 2019, pp. 2312–2322.

[29] J. Wang and T.-L. Teng, “Artificial neural network-based seismic detector,” *Bulletin of the Seismological Society of America*, vol. 85, no. 1, pp. 308–319, 1995.

[30] J. Wang and T.-l. Teng, “Identification and picking of s phase using an artificial neural network,” *Bulletin of the Seismological Society of America*, vol. 87, no. 5, pp. 1140–1149, 1997.

[31] Y. Zhao and K. Takano, “An artificial neural network approach for broadband seismic phase picking,” *Bulletin of the Seismological Society of America*, vol. 89, no. 3, pp. 670–680, 1999.

[32] D. Maity, F. Aminzadeh, and M. Karrenbach, “Novel hybrid artificial neural network based autopicking workflow for passive seismic data,” *Geophysical Prospecting*, vol. 62, no. 4, pp. 834–847, 2014.

[33] P. M. DeVries, F. Vigas, M. Wattenberg, and B. J. Meade, “Deep learning of aftershock patterns following large earthquakes,” *Nature*, vol. 560, no. 7720, p. 632, 2018.

[34] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[35] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, “Convolutional neural networks for time series classification,” *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[36] R. Caruana, “Multitask learning,” *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[37] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv*, 2015.

[38] C. E. Administration, “Aftershock detection contest,” https://tianchi.aliyun.com/competition/introduction.htm?raceId=231606&_lang=en_US, 2017, 2017.

[39] P. Hensman and D. Masko, “The impact of imbalanced training data for convolutional neural networks,” *Degree Project in Computer Science, KTH Royal Institute of Technology*, 2015.

[40] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[41] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[42] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: a system for large-scale machine learning.” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, vol. 16, 2016, pp. 265–283.

[43] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in *International conference on machine learning*, 2013, pp. 1139–1147.

[44] C. Bergmeir and J. M. Benítez, “On the use of cross-validation for time series predictor evaluation,” *Information Sciences*, vol. 191, pp. 192–213, 2012.

[45] L. J. Tashman, “Out-of-sample tests of forecasting accuracy: an analysis and review,” *International journal of forecasting*, vol. 16, no. 4, pp. 437–450, 2000.