

MLP, Recurrent, Convolutional and LSTM Neural Networks Detect Seismo-TEC Anomalies Potentially Related to the Iran Sarpol-e Zahab ($M_w=7.3$) Earthquake of 12 November 2017

Akhoondzadeh, M.^{1*}, Hosseiny, B.² and Ghasemian, N.²

1. Associate Professor, Department of Photogrammetry and Remote Sensing, School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran

2. Ph.D. Student, Department of Photogrammetry and Remote Sensing, School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran

(Received: 27 June 2021, Accepted: 20 Sep 2021)

Abstract

A strong earthquake ($M_w = 7.3$) (34.911° N, 45.959° E, ~ 19 km depth) occurred on November 12, 2017, at 18:18:17 UTC (LT=UTC+03:30) in Sarpol-e Zahab, Iran. Six different Neural Network (NN) algorithms including Multi-Layer Perceptron (MLP), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN), Long-Short Term Memory (LSTM) and CNN-LSTM were implemented to survey the four months of GPS Total Electron Content (TEC) measurements during the period of August 01 to November 30, 2017 around the epicenter of the mentioned earthquake. By considering the quiet solar-geomagnetic conditions, every six methods detect anomalous TEC variations nine days prior to the earthquake. Since time-series of TEC variations follow a nonlinear and complex behavior, intelligent algorithms such as NN can be considered as an appropriate tool for modelling and prediction of TEC time-series. Moreover, multi-methods analyses beside the multi precursor's analyses decrease uncertainty and false alarms and consequently lead to confident anomalies.

Keywords: Earthquake Precursor; Anomaly; Ionosphere; GPS-TEC; Neural Network.

1. Introduction

Although there are many scientific papers indicating the pre-seismic Lithospheric-Atmospheric-Ionospheric (LAI) anomalous variations, it should be noted that the statistical analysis is an undeniable part of earthquake precursor's analysis. Since multi-precursors and multi-predictors analysis are appropriate ways to increase the confidence of the detected anomalies, intelligent algorithms such as Neural Networks (NN) enable to model the complexity and non-linearity of precursor's time series.

The ionospheric anomalies may be observed in the D, E and F layers, about 1 to 10 days before the strong earthquake and may be continued a few days after the event (Parrot, 1995; Liu et al., 2004; Hayakawa and Molchanov, 2002; Pulinets and Boyarchuk, 2004; Freund, 2009; Pulinets and Ouzounov, 2011; Sorokin and Pokhotelov, 2014; Akhoondzadeh et al., 2010; 2018; 2019).

Currently, thousands of GPS receivers are used to monitor the Earth's surface deformations. Total Electron Content (TEC) data retrieved from GPS measurements have made a considerable contribution to the

understanding of seismo-ionospheric variations. TEC is the integrated number of the electrons within the block between the satellite and receiver or between two satellites. The GPS satellites transmit two frequencies of signals ($f_1=1575.42$ MHz and $f_2=1227.60$ MHz). The received signals in ground stations contain many effects such as ionosphere, troposphere, hardware and random errors. Ionosphere on the contrary of the troposphere is a dispersive medium and its effects can be evaluated with measurement of the modulations on the carrier phases recorded by dual-frequency receivers. To study TEC variations, data of GIM (Global Ionospheric Map) provided by NASA Jet Propulsion Laboratory (JPL) were used. The GIM is constructed into $5^\circ \times 2.5^\circ$ (Longitude, Latitude) grid with a time resolution of two hours. GIM data are generated on a daily basis using data from about 150 GPS sites of the IGS (International Gns Service) and other institutions. The Vertical Total Electron Content (VTEC) is modeled in a solar-geomagnetic reference frame using a spherical harmonics expansion

*Corresponding author:

makhonz@ut.ac.ir

up to degree and order of 15. Instrumental biases, so-called differential P1-P2 code biases (DCB), for all GPS satellites and ground stations are estimated as constant values for each day (Mannucci et al., 1998). To convert line-of-sight TEC into vertical TEC, a modified single-layer model mapping function approximating the JPL extended block model mapping function is adopted (<http://www.gsfc.nasa.gov/>). In this research, TEC data based on the date and geographic location of each earthquake from about six weeks before to one week after the main event were processed.

Liu et al. (2004) statistically described the temporal parameters of the seismo-ionospheric precursors detected during 1-5 days prior to the earthquakes using TEC data for 20 major earthquakes in Taiwan. Akhoondzadeh (2012) applied four methods including mean, median, wavelet transform and Kalman filter to detect anomalous TEC variations concerning Tohoku earthquake.

In the border region between Iran and Iraq in the vicinity of the Sarpol-e Zahab town (34.911° N, 45.959° E, 19.00 km depth) a strong earthquake of $M_w=7.3$ happened at 18:18:17 UTC (LT=UTC+03:30) on November 12, 2017. TEC is the integrated number of the electrons within the block between the satellite and receiver or between two satellites.

The purpose of this study is to determine whether the deep learning neural network algorithms used are effective in modeling TEC non-linear variations and anomalies detection.

2. Methodology

In this paper, six different NN methods based on feedforward and recurrent architectures were implemented to observe potentially anomalous Total Electron Content (TEC) variations around the time and location of Sarpol-e Zahab earthquake.

2-1. Multi-Layer Perceptron (MLP) Neural Network

Artificial Neural Networks are a class of intelligent systems that can discover patterns with a few a priori assumptions and learn any complex functional relationship from the data to model a phenomenon. An ANN is made up by simple processing units, the neurons,

which are connected in a network by a large number of weighted links where the acquired knowledge is stored.

An input x_j is transmitted through a connection, which multiplies its strength by a weight w_{ij} to give a product $x_j w_{ij}$. This product is an argument to a transfer function f , which yields an output represented as: $y_i = f(x_j w_{ij})$ where i is an index of neurons in the hidden layer and j is an index of an input to the neural network.

The most popular and successful model is the feed forward Multi-Layer Perceptron (MLP) network. In a MLP, neurons are grouped in layers, and only forward connections exist. In order to detect anomaly, total available data are split into a training set and a test set. The training set is used for construction of the neural network, whereas the test set is used for measuring the predictive error of the model. The training process is used essentially to find the connection weights of the networks. If the prediction error exceeds the pre-defined threshold, the measured value could be considered as anomaly.

This type of Neural Network was discussed in details in paper of Akhoondzadeh (2013) to detect seismo-ionospheric anomalous variations induced by the powerful Tohoku earthquake of March 11, 2011.

2-2. Convolutional Neural Network (CNN)

Convolutional neural networks can be considered as one type of the neural networks applied on grid-like topology data such as 1-D time-series taking samples at evenly spaced time steps intervals. The name "convolutional" comes from their architecture using one kind of linear mathematical operation called convolution. These kinds of neural networks use convolution rather than matrix multiplication used in ordinary neural networks.

Convolutional Neural Networks (CNN) have multiple building blocks that form the CNN layers (Khan et al., 2018): 1- pre-processing layer; 2- convolutional layers; 3- non-linearity (detector stage); 4- pooling layers; 5- fully connected layers.

2-2-1. Pre-processing

Before importing the input data to the network, it is essential to make the data zero-centered and normalized. Zero centering is

accomplished by subtracting the mean of the whole data (both train and test) from each data point and normalization is achieved by dividing the zero-centered data to the variance of each dimension. We can show the zero centering and normalization equations as follows (Khan et al., 2018):

$$x' = x - \hat{x} \quad \hat{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (1)$$

$$x'' = \frac{x'}{\sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x})^2}{N-1}}} \quad (2)$$

2-2-2. Convolutional layer

Generally, convolution can be defined as an operation applied to two functions with real-valued arguments. The convolution operation is usually defined by an asterisk (Equation 3).

$$s(t) = (x * w)(t) = \int x(a)w(t - a)da \quad (3)$$

The first argument, the function x , is referred to as the input and the second, w , is called the kernel in a convolutional neural network.

2-2-3. Convolutional Network Architecture for Time-Series

Time-series data can be considered as a 1-D array of time steps. Each string of time series data can be viewed as a $1 \times m$ vector, in which m is the number of time steps. We are supposed to perform the convolution on the time series data using a 1×3 filter. If we do not zero-pad the array, the output of the convolution operation would be 2 pixels smaller than the original input. We move the filter with a stride of a one-time unit in the horizontal direction to obtain the *feature map*. Applying multiple 1-D convolutions and pooling layers gives a CNN the ability to model intricate patterns existing in a complex and non-linear time series.

2-2-4. Non-linearity

The output of a convolutional or fully connected layer is fed into a non-linear or piece-wise linear function. This allows the network to learn non-linear mappings. If we eliminate this non-linearity, only modeling linear functions will be possible. Non-linearity also controls the degree of response of the neuron to a particular input. Non-linearity must be differentiable according to the backpropagation learning rule.

2-2-5. Pooling layer

Pooling operation is applied to the output of the non-linearity layer. It represents a statistical summary of the data and removes the distortions and disturbances in the primary feature map and also lessens the computational cost of the data. For example, max pooling chooses the maximum unit in a rectangular neighborhood. Some other pooling operations include average pooling, L2 norm in a rectangular neighborhood or weighted average in a rectangular neighborhood with weights based on distance. Pooling layers ensure that the learned function is invariant to the small changes in the input data and thus improve the generalization ability of the network.

2-2-6. Fully connected layers

Fully connected layers are usually placed at the end of a CNN. These layers are the same as the weight layers of a Multi-Layer Perceptron and can be considered as a convolutional layer with a filter size of 1×1 . The input data are multiplied by a weight matrix and are added to a bias vector and after passing the activation function (Equation (4)) the output vector is obtained.

$$y = f(W^T x + b) \quad (4)$$

Our implemented CNN contained two 1-D convolutional layers with 100 1×2 filters and one 1×2 1-D Max Pooling layer. In addition, a fully connected layer was also devised in the last layer. Each row of the state matrix was imported to the CNN. Four time-steps were considered ($[t_1, t_2, t_3, t_4]$) and our aim was to predict the TEC value at one time-step ahead.

2-3. Recurrent Neural Network (RNN)

RNN is a branch of artificial neural networks. In contrast with feedforward neural network, RNN contains a loop that connects every input data to former output of the network. Therefore, unlike a feedforward network, the recurrent network can process a sequential data, such as time-series, by having a recurrent hidden state (Williams and Zipser, 1989).

By considering $x = (x_1, x_2, \dots, x_T)$ as input sequence in period of T , where x_i is the data at i^{th} time step, an RNN updates its recurrent hidden state h_t at time step t by

$$h_t = H(W_{hh}h_{t-1} + W_{hx}x_t + b_h); \quad t > 0 \quad (5)$$

where the W terms denote weight matrices. W_{hh} is the weight matrix connecting the hidden state from the previous step to the current step, and W_{hx} is the weight matrix connecting the input data in the current step to the hidden state. Also, b_h is bias term and $H(\cdot)$ is the nonlinear activation function. *Tanh* and sometimes *ReLU* are common choices for $H(\cdot)$ in RNNs. Every recurrent block can also be estimated the output \hat{y}_t that is calculated by:

$$\hat{y}_t = g(W_{yh}h_t + b_y); \quad t > 0 \quad (6)$$

where $g(\cdot)$ is the nonlinear activation function depending on the form of the output can be logistic sigmoid or Softmax (Graves, 2013).

2-4. Long-Short Term Memory (LSTM)

Long-Short Term Memory networks are a more advanced version of Recurrent Neural Networks (RNN). One of the main problems of RNNs is that they cannot access information from a long range of sequences. Also, the exploding and vanishing gradient problems due to small or large values of the gradient as it cycles around the recurrent connections are addressed in LSTMs.

LSTM architecture contains multiple recurrently connected memory blocks. These memory blocks have one or more self-connected cells and three units called input gate, output gate and forget gate. The gates allow the memory blocks to store and access information over long periods of time (Graves, 2012).

Each cell in an LSTM is similar to a vanilla recurrent network but has more parameters and a set of gating units that control the flow of information (Goodfellow et al., 2016). The forward equations in an LSTM are presented below. At first, the state unit's weight is computed using a sigmoid function (h). b^f , U^f and W^f are the forget gates' bias, recurrent weights, and input weights respectively. x_t and h_t are the input and the hidden layer vectors at time t .

$$h_{t,i}^f = h(b_i^f + \sum_j U_{i,j}^f x_{t,j} + \sum_j W_{i,j}^f h_{t,j}) \quad (7)$$

The LSTM cell state is updated as follows:

$$s_{t+1,i} = h_{t,i}^f s_{t,i} h_{t,i}^e \sigma(b_i + \sum_j U_{ij} x_{t,j} + \sum_j W_{ij} h_{t,j}) \quad (8)$$

In Equation (8), σ denotes the non-linearity (tanh or logistic sigmoid). Parameters b , U , W are the bias, recurrent weights, and input weights to the cell. $h_{t,i}^e$ is the output of the input gate and is computed using a sigmoid function (h).

$$h_{t,i}^e = h(b_i^e + \sum_j U_{i,j}^e x_{t,j} + \sum_j W_{i,j}^e h_{t,j}) \quad (9)$$

We have control over how much of the output can be passed out of the cell using a sigmoid function. b_i^o , U_{ij}^o , W_{ij}^o are the bias, recurrent weights and input weights of the output gate. Parameter $h_{t,i}^o$ can prevent computed output pass the cell ($h_{t,i}^o=0$).

$$h_{t+1,i} = g(s_{t+1,i}) h_{t,i}^o \quad (10)$$

$$h_{t,i}^o = h(b_i^o + \sum_j U_{ij}^o x_{t,j} + \sum_j W_{ij}^o h_{t,j}) \quad (11)$$

LSTM has the ability to preserve the gradient information within time. As long as the forget gate is open and the input gate is closed, the network remembers the corresponding input. The sensitivity of the output layer can be switched on and off by the output gate. Figure 1 shows the state of input, output and forget gate as open ('o') or closed ('_') below, up and to the left of the hidden layer respectively.

To predict the TEC values, we designed one LSTM layer including 200 neurons and sigmoid as the activation function. Also, two fully connected layers, one with 100 and the other with 1 neuron, were prepared in the last two layers.

2-5. Gated Recurrent Unit (GRU)

GRU is one of the solutions to vanishing gradient problem of simple recurrent network (according to backpropagation process) (Chung et al., 2014). Therefore, it helps for better capturing of long-range connections in sequential data (Goodfellow et al., 2016).

GRU contains update gates that determine how much of hidden states need to be updated. These logistic gates are computed by:

$$u_t = \sigma(W_{uh}h_{t-1} + W_{ux}x_t + b_u) \quad (12)$$

where u_t denotes update gate and $\sigma(\cdot)$ denotes logistic sigmoid function that nonlinearly determines update proportion. Therefore, hidden state at each step of GRU network can be updated by:

$$h_t = (1 - u_t)h_{t-1} + u_t\tilde{h}_t \quad (13)$$

where h_{t-1} is the hidden state at previous step and \tilde{h}_t is the candidate hidden state at current step that is computed similar to the simple RNN hidden state by:

$$\tilde{h}_t = H(W_{hh}(r_t \odot h_{t-1} + W_{hx}x_t + b_h); t > 0) \quad (14)$$

where r_t denotes reset gates that can be computed, similar to update gate, by:

$$r_t = \sigma(W_{rh}h_{t-1} + W_{rx}x_t) + b_r \quad (15)$$

2-6. CNN-LSTM model

Figure 1 shows the architecture of the proposed CNN-LSTM model. Each row of the state matrix containing 1×8 time sequences of TEC values, $[t_1, t_2, t_3, t_4, t_5, \dots, t_8]$, was divided into two 1×4 subsequences and imported to the proposed network. We used two 1-D convolutional layers with 100 1×2 filters (Huang and Kuo, 2018). Rectified Linear Unit (ReLU) was considered as the activation. First, the mentioned 1-D convolutional layers and a 1-D 1×2 MaxPooling layer extracted the relevant features. Then the extracted feature maps fed into an LSTM layer with 200 memory units and a ReLU as the activation function to predict the TEC value at one time-step ahead. All methods mentioned in this section were implemented using the Keras library of the Python programming language (<https://keras.io/>). Also, it should be noted that 65% of the initial data were considered as the train and the rest as the test.

3. Observations

In this study, four time-steps were considered ($[t_{i-4}, t_{i-3}, t_{i-2}, t_{i-1}]$) as network input vector, and our aim is to predict the TEC value at one time-step ahead (t_i). All the networks are trained in a supervised manner by 65% of the time-series. The list of the implemented neural networks in this study and their corresponding architecture are gathered in Table 1. MLP NN contains two hidden layers with 100 neurons at each layers and ReLU activation function. Convolutional network contains two 1-D convolutional layers with 100 1×2 filters and one 1×2 1-D Max Pooling layer. In addition, a fully connected layer was also devised in the last layer. Our proposed RNN and GRU networks, both, contain one layer with 100 memory units and fully-connected neurons with *tanh* activation function. Also, in all networks, the output layer contains a single neuron that is predicted value at $t+1$. The designed LSTM network contains one layer including 200 memory units and sigmoid as the activation function. Also, two fully connected layers one with 100 and the other with 1 neuron were prepared in the last two layers. We used two 1-D convolutional layers with 100 1×2 filters (Huang and Kuo, 2018). Rectified Linear Unit (ReLU) was considered as the activation. First, the mentioned 1-D convolutional layers and a 1-D 1×2 MaxPooling layer extracted the relevant features. Then, the extracted feature maps fed into an LSTM layer with 200 memory units and a ReLU as activation function to predict the TEC value at one time-step ahead.

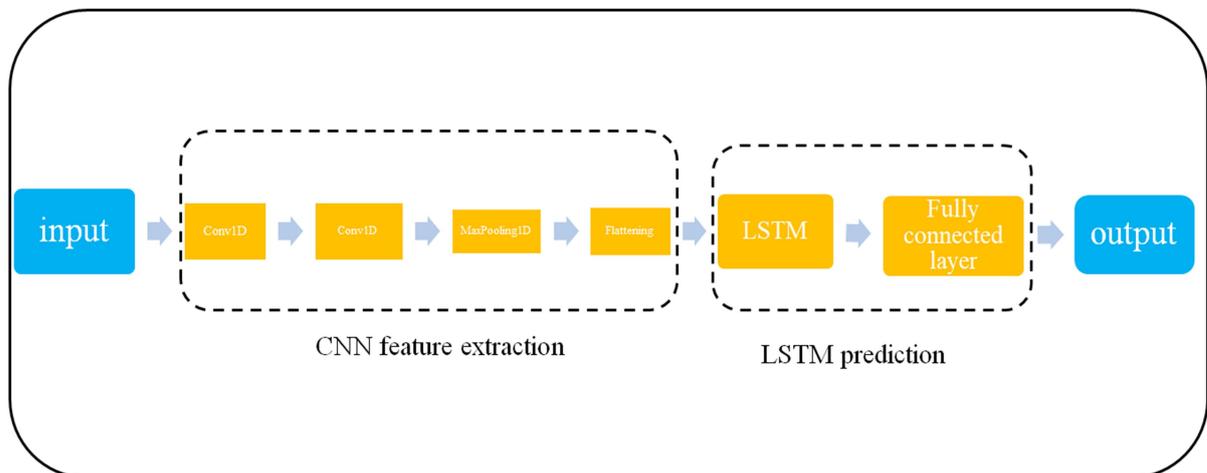


Figure 1. The proposed CNN-LSTM neural network.

Table 1. The list of the proposed neural networks and their corresponding architecture.

Proposed Neural Network	Architecture
MLP	2 hidden layers with 100 neurons at each layer and ReLU activation function
CNN	2 one-dimensional conv. layers with 100 1×2 filters and ReLU activation function and a Maxpooling unit
RNN	1 layer with 100 RNN memory units and <i>Tanh</i> activation function
LSTM	1 layer with 200 LSTM memory units and <i>sigmoid</i> activation function
GRU	1 layer with 100 memory units and <i>Tanh</i> activation function
CNN+LSTM	2 one-dimensional conv. layers with 100 1×2 filters and ReLU activation function and a Maxpooling unit, integrated with 1 layer of the LSTM network with 100 memory units and ReLU activation function

3-1. Solar and geomagnetic data

To discriminate the seismo-ionospheric perturbations from solar geomagnetic disturbances, the indices of K_p , A_p , D_{st} , and $F10.7$ were checked. The ionospheric effect of a geomagnetic storm has a global impact being observed all over the world, while the seismogenic effect is observed only by stations with distance less than 2000 km from the potential epicenter. It should be noted that an ionospheric storm usually lasts 8 – 48 h while the seismoionospheric disturbances have a duration of 3 – 4 h, a few days before the earthquake. The K_p index monitors the planetary activity on a worldwide scale while the D_{st} index records the equatorial ring current variations (Pulinets and Boyarchuk, 2004).

The detected irregular variations of the ionosphere in quiet solar geomagnetic conditions ($K_p < 2.5$, $A_p < 25$, $D_{st} > -20$ nT, $D_{st} < 20$ nT and $F10.7 < 120$) may be associated with seismic activities.

Figure 2 illustrates the variations of K_p , A_p , D_{st} , and $F10.7$ indices, during the period of

September 01 to November 30, 2017 (<http://spider.ngdc.noaa.gov>). An asterisk indicates the earthquake time. The X-axis represents the days relative to the earthquake day. The Y-axis represents the universal time coordinate.

The high geomagnetic activities are clearly observed on September 08, 2017, when the K_p and A_p indices reach the maximum values of 8.3 and 236, respectively, between 13:00 and 15:00 UTC. The unusual variations of the K_p and A_p indices are also seen on four days before the earthquake between 13:00 and 15:00 UTC with the values of 5 and 48. The irregular D_{st} values are observed on four days before the event when this parameter exceeds the lower boundary value (i.e. -20 nT), reaching the value of -65 nT at 17:00 UTC. D_{st} value has a minimum value of -142 nT during the studied time period on September 08, 2017 at 02:00 UTC. The $F10.7$ value gradually increases from about September 01 and reaches the maximum value of 182.50 SFU on September 04, 2017 (69 days before the event).

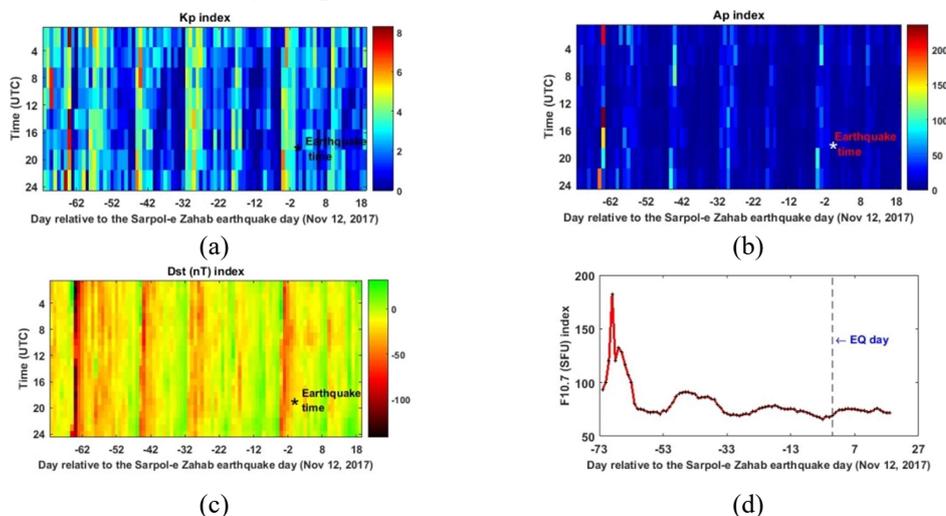


Figure 2. a), b), c) and d) show respectively, the variations of K_p , A_p , D_{st} and solar radio flux ($F10.7$) indices during the period of September 01 to November 30, 2017. An asterisk indicates the earthquake time. The X-axis represents the days relative to the Iran earthquake day.

3-2. GPS-TEC data

Figure 3 (a) shows TEC variations derived from GIM data and the closest node (35° N, 45° E) to the epicenter during the period of September 01 to November 30, 2017. The earthquake time is indicated by an asterisk. The x-axis represents the day relative to the earthquake day. The y-axis represents the time UTC (LT=UTC-5:00). By visual inspection and without performing any special analysis, unusual TEC values are clearly seen around September 8, but as mentioned before the geomagnetic indices show high activities on this time, and therefore, the observed unusual TEC variations during this period cannot be associated to a seismic event. Figure 3 (b) shows variations of the predicted TEC values using the MLP neural network algorithm. Figure 3 (c) illustrates the differences between the normalized and predicted TEC values using MLP during the period of September 01 to November 30, 2017. Figure 3 (d) indicates DTEC values where $DTEC = \frac{\Delta TEC - Median}{Interquartile}$, where ΔTEC is the difference between the observed and predicted TEC values using MLP method. Figure 3(e) shows detected TEC anomalies using the MLP method when $|DTEC| > 1.5$ and without considering the non-quiet conditions of solar and geomagnetic activities. To distinguish the likely seismo-ionospheric perturbations from the solar and geomagnetic activities, five conditions including $Kp < 2.5$, $Ap < 25$, $Dst > -20$ nt, $Dst < 20$ nt and $F10.7 < 120$ are joined to $|DTEC| > 1.5$ equation using AND operator. Figure 3(e) indicates striking TEC anomalies 9 and 10 days before the main shock at 09:00 and 24:00 UTC, when the DTEC reaches the values 2.16 and 1.95, respectively. Table 2 shows a list of the detected anomalies using different algorithms.

Different panels of Figures 4 and 5 are the same as Figure 13, but they were obtained using RNN and GRU

algorithms, respectively. Both of these algorithms detect clear anomalies 9 and 10 days prior to the event at 9:00 and 24:00 UTC (Table 1).

Figure 6 illustrates the time-series of observed (blue curve) and predicted TEC (red curve) values using CNN method at different times (UTC). The earthquake time is indicated by a vertical dotted line. The x-axis represents the day relative to the earthquake day. The y-axis represents the TEC values. It is seen that the predicted TEC values using CNN method follow the observed TEC values with a good approximation.

The differences between the observed and predicted TEC values using CNN method from September 1 to November 30, 2017 are seen in Figure 7. The x-axis represents the day relative to the earthquake day. The y-axis represents the time UTC (LT=UTC-5:00).

Figure 8 shows the DTEC values, where DTEC variations obtained by equation of $DTEC = \frac{\Delta TEC - Median}{Interquartile}$, where ΔTEC is the differences between the observed and predicted TEC values using CNN method. The x-axis represents the day relative to the earthquake day. Figure 9 shows the results of detected TEC anomalies using CNN method when $|DTEC| > 1.5$ for the Iran earthquake (November 12, 2017) from September 1 to November 30, 2017. The earthquake time and also clear anomalies on 9 and 10 days before earthquake after considering the quiet solar-geomagnetic conditions (Figure 2) are indicated by vertical dotted lines. Figures 10 and 11 are the same as Figure 9 but were obtained using LSTM and CNN-LSTM algorithms, respectively. By considering the quiet solar-geomagnetic conditions ($Kp < 2.5$, $Ap < 25$, $Dst > -20$ nt, $Dst < 20$ nt and $F10.7 < 120$) all the three methods show clear anomalies nine days preceding the earthquake at 9:00 UTC (Table 2).

Table 2. The list of the detected anomalies.

Method	Day	Date	Time (UTC)	DTEC
MLP	-9	03 Nov. 2017	1	-1.84
			2	-2.21
			3	-2.40
			8	-1.69
			9	-2.16
			15	-1.64
			20	-1.64
			21	-1.72
	22	-1.75		
	-10	02 Nov. 2017	24	-1.95
RNN	-9	03 Nov. 2017	1	-1.69
			2	-2.2
			3	-2.25
			8	-1.68
			9	-2.51
			10	-1.53
			20	-1.72
			21	-2.02
	22	-1.81		
	-10	02 Nov. 2017	24	-1.56
GRU	-9	03 Nov. 2017	1	-1.71
			2	-1.90
			3	-2.00
			8	-1.60
			9	-2.17
			12	-1.81
			20	-1.60
			21	-1.71
	22	-1.70		
	-10	02 Nov. 2017	24	-1.89
LSTM	-9	03 Nov. 2017	1	1.86
			2	1.76
			3	1.59
			8	1.24
			12	1.47
			19	1.22
			20	1.56
			21	1.91
			22	1.86
	23	1.60		
		-10	02 Nov. 2017	24
			24	1.31
CNN	-9	03 Nov. 2017	3	1.53
			9	2.27
			15	1.68
CNN-LSTM	-8	04 Nov. 2017	23	-3.80
	-9	03 Nov. 2017	2	2.60
			5	1.54
			9	1.74
			13	1.59
	16	1.73		
	-11	01 Nov. 2017	2	-1.60

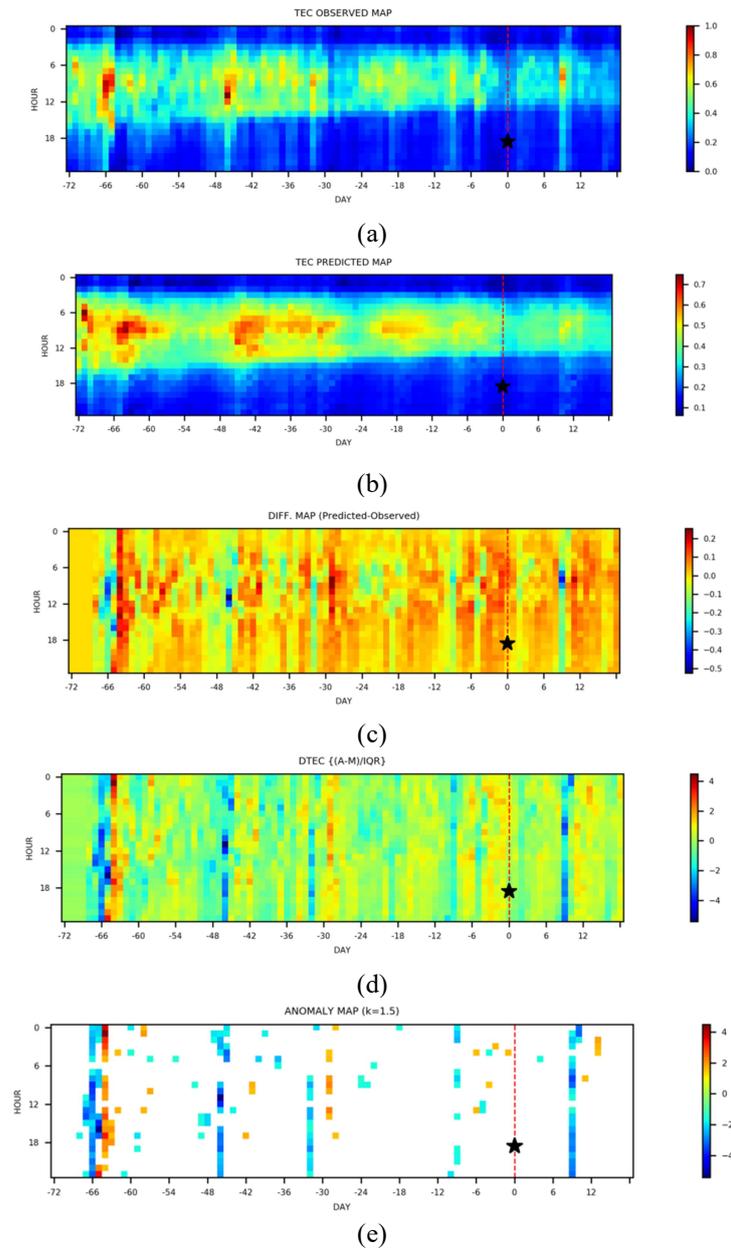


Figure 3. Results of TEC analysis using MLP Method for the Iran earthquake (November 12, 2017) from September 1 to November 30, 2017. a) Normalized TEC variations, b) Predicted TEC values using MLP algorithm, c) The differences between the normalized and predicted TEC values, d) DTEC variations, e) Detected TEC anomalies when $|DTEC| > 1.5$.

4. Discussion and Conclusions

It is true that the many studies have been done in the field of ionospheric precursors, but the complexity and unpredictability of ionosphere makes use of the intelligent methods such as NN to model and recognize the pattern of precursor's time-series. Therefore, in this study, six different NN algorithms including MLP, RNN, GRU, CNN, LSTM, and CNN-LSTM were implemented to observe seismo-TEC anomalies around the time and location of Sarpol-e Zahab earthquake in Iran.

Every six methods detect very striking irregular variations, nine days before the earthquake in quiet solar and geomagnetic conditions.

This study aims to provide substantial evidences on the efficiency of deep neural networks in non-linear time series modelling. As a neural network is non-parametric method and do not require any assumptions about the underlying model, it could be a powerful tools in modeling complex phenomena such as earthquake precursor time series that we may not know what the

underlying data generating process is. Although different hypotheses have been proposed about the mechanism of seismic LAI precursors based on the geophysical and geochemical processes, but none of them have been accepted among the majority of scientists (Freund, 2009; Pulinets and Ouzounov, 2011). To better understand the preparatory phase of the Iran's November 12, 2017 earthquake Akhoondzadeh et al. (2019) also investigated four atmospheric meteorological/climatological parameters to detect possible chemical/physical alteration of the atmosphere: skin temperature, total column water vapour, aerosol optical

thickness and sulphur dioxide. All investigated parameters present some anomalies distributed in different times, but most of all seem to indicate a final disturbance of the atmosphere that precedes by some days the ionospheric disturbance in TEC, Ne and magnetic data. This is a very interesting feature, as a lithospheric activity is expected to be disturbed before the atmosphere and then the ionosphere as it propagates upwards. It should be noted that statistical, multi-precursors and multi predictor's analyses are key factors to detect the real anomalies and to understand the earthquake precursor's mechanism.

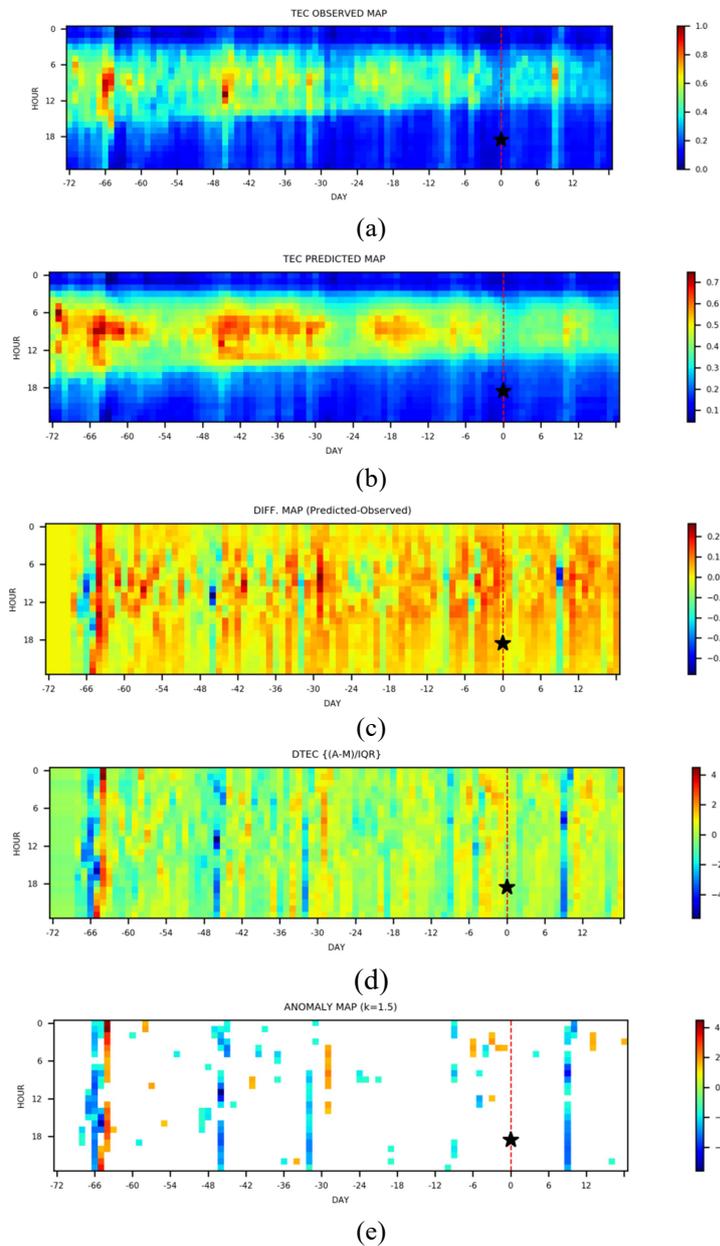


Figure 4. The same as Figure 3 but using RNN algorithm.

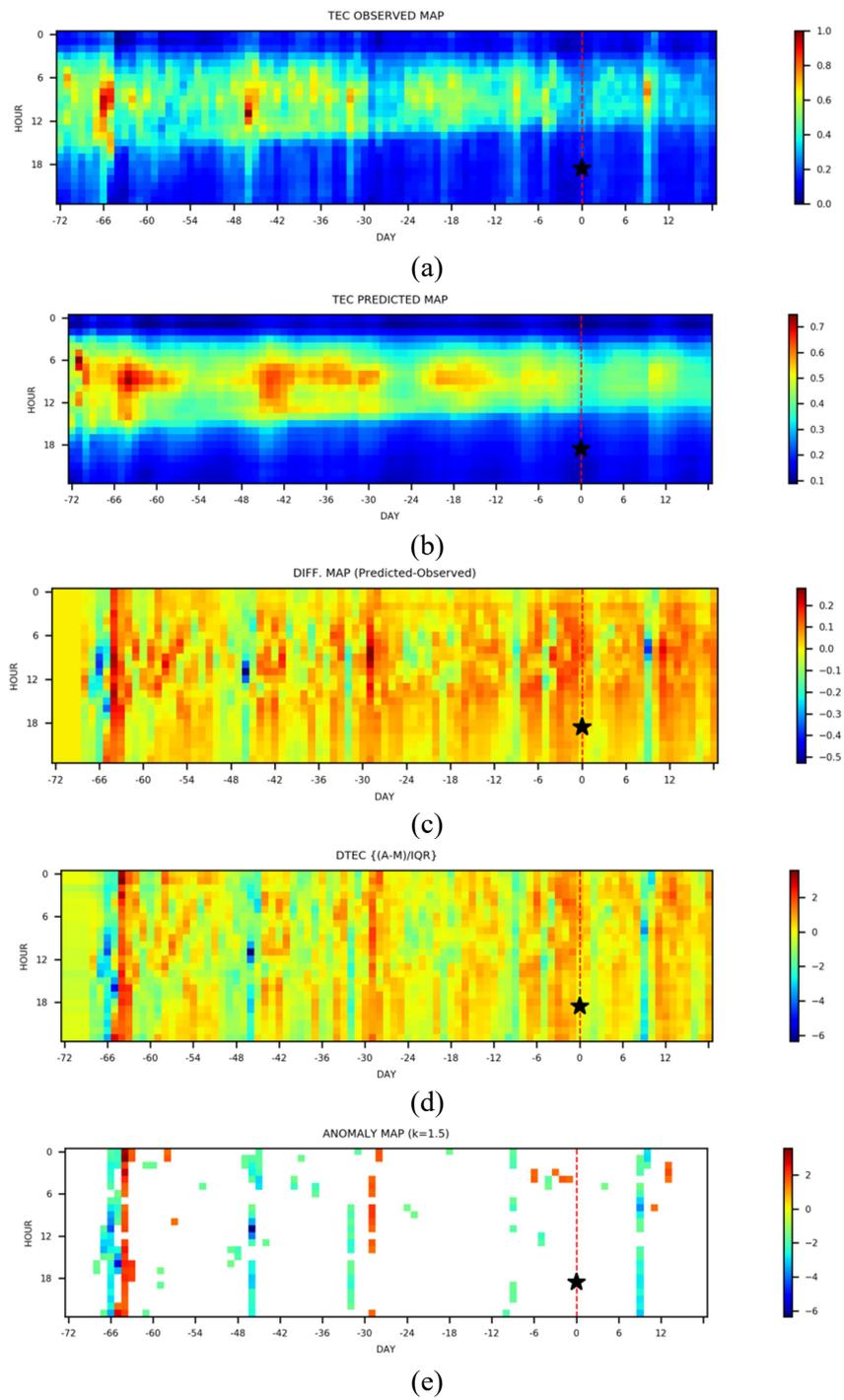


Figure 5. The same as Figure 3 but using GRU algorithm.

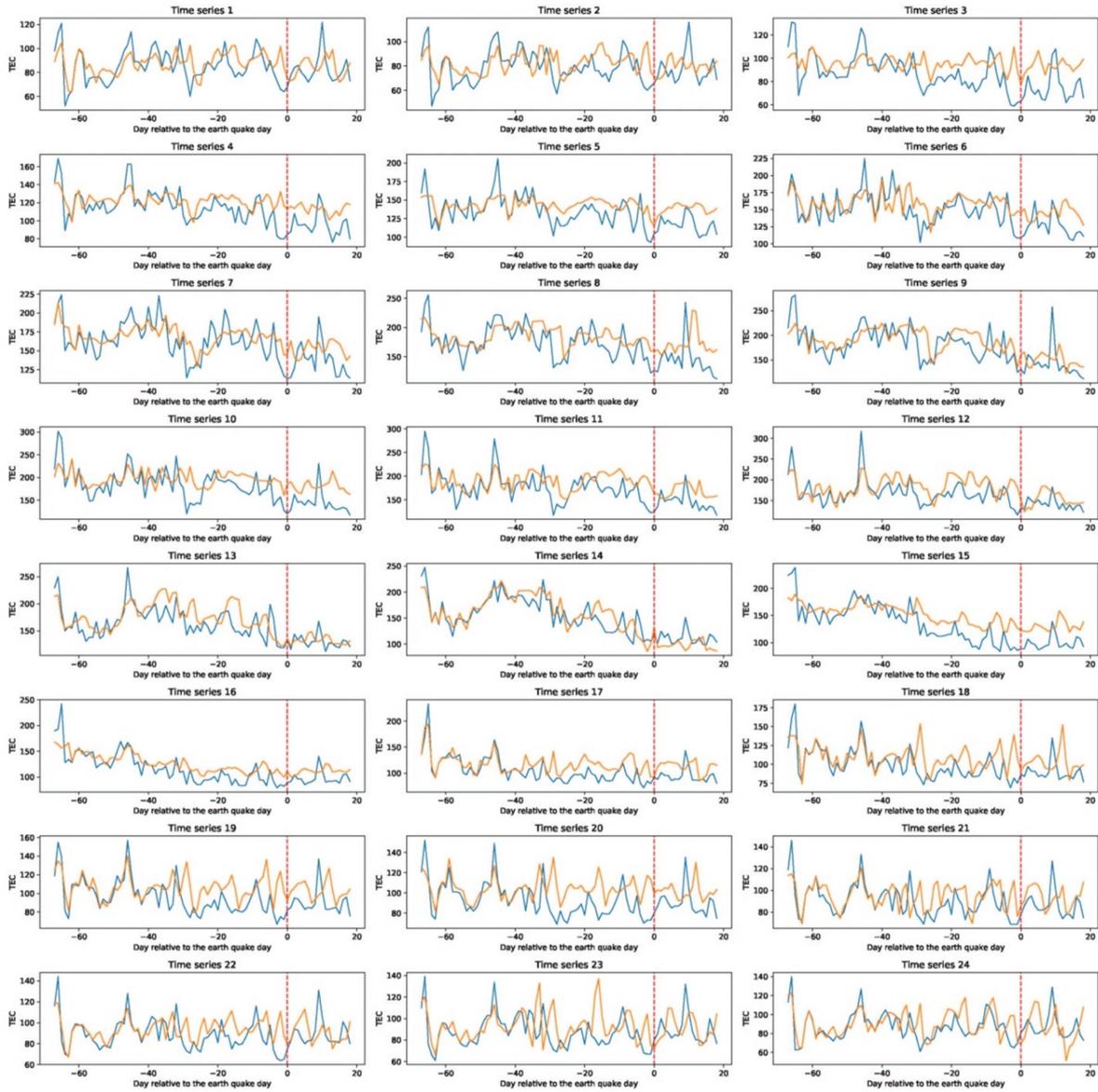


Figure 6. Time series of TEC (blue curve) and predicted TEC (red curve) values using CNN method at different times (UTC). The earthquake time is indicated by a vertical dotted line. The x-axis represents the day relative to the earthquake day. The y-axis represents the TEC values.

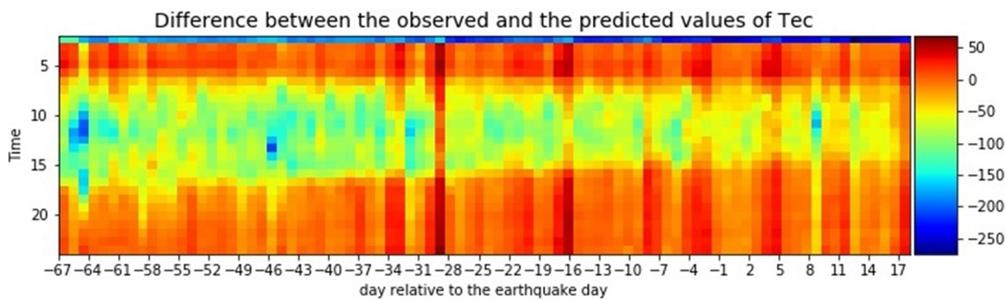


Figure 7. Results of TEC analysis indicating the differences between the observed and predicted TEC values using CNN method for the earthquake (November 12, 2017) from 1st September 1 to November 30, 2017.

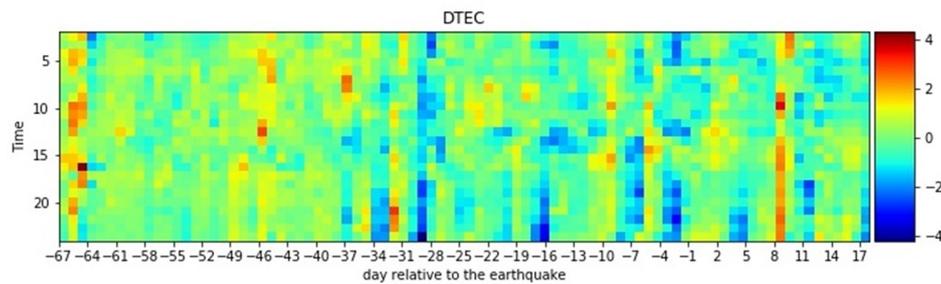


Figure 8. Results of DTEC variations.

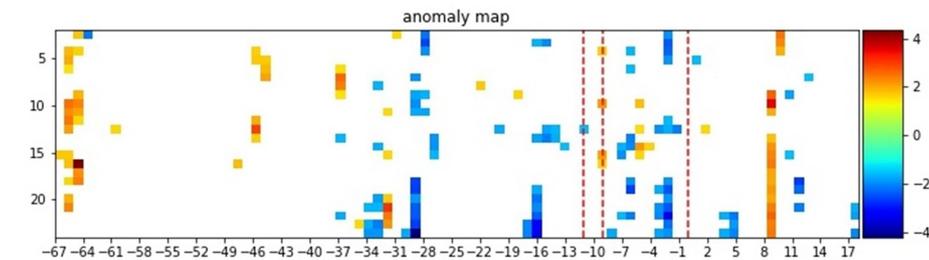


Figure 9. Results of detected TEC anomalies using CNN method when $|DTEC| > 1.5$ for the earthquake (November 12, 2017) from September 1 to November 30, 2017.

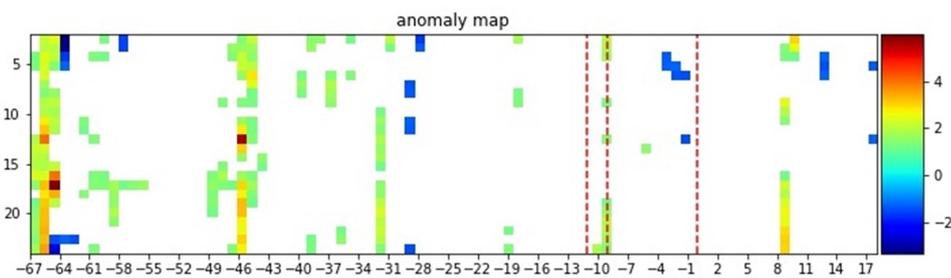


Figure 10. The same as Figure 10 but for LSTM method.

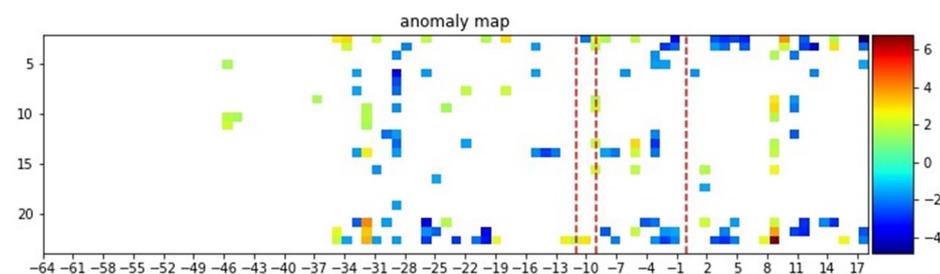


Figure 11. The same as Figure 10 but for CNN-LSTM method.

Acknowledgement

The authors would like to acknowledge the NASA Jet Propulsion Laboratory for the TEC data (<ftp://cddis.gsfc.nasa.gov/>) and solar and geomagnetic indices (<ftp://ftp.ngdc.noaa.gov/>).

References

Akhoondzadeh, M., Parrot, M. and Saradjian, M. R., 2010a, Electron and ion density

variations before strong earthquakes ($M > 6.0$) using DEMETER and GPS data. Nat. Hazards Earth Syst. Sci., 10, 7–18. doi: 10.5194/nhess-10-7-2010.

Akhoondzadeh, M., 2012, Anomalous TEC variations associated with the powerful Tohoku earthquake of 11 March 2011. Nat. Hazards Earth Syst. Sci., 12, 1453–1462, doi:10.5194/nhess-12-1453-2012.

Akhoondzadeh, M., 2013a, A MLP neural

- network as an investigator of TEC time series to detect seismo-ionospheric anomalies. *Advances in Space Research*, 51, 2048-2057.
- Akhoondzadeh, M., De Santis, A., Marchetti, D., Piscini, A. and Cianchini, G., 2018, Multi precursors analysis associated with the powerful Ecuador ($M_w=7.8$) earthquake of 16 April 2016 using Swarm satellites data in conjunction with other multi-platform satellite and ground data. *Advances in Space Research*, 61, 248-263. <https://doi.org/10.1016/j.asr.2017.07.014>.
- Akhoondzadeh, M., De Santis, A., Marchetti, D., Piscini, A. and Jin, S., 2019, Anomalous seismo-LAI variations potentially associated with the 2017 $M_w = 7.3$ Sarpol-e Zahab (Iran) earthquake from Swarm satellites, GPS-TEC and climatological data. *Advances in Space Research*, 64, 143–158. <https://doi.org/10.1016/j.asr.2019.03.020>
- Chung, J., Gulcehre, C., Cho, K. and Bengio, Y., 2014, Empirical evaluation of gated recurrent neural networks on sequence modeling. arxiv prepr. arxiv1412.3555.
- Freund, F., 2009, Stress-activated positive hole charge carriers in rocks and the generation of pre-earthquake signals. in *electromagnetic phenomena associated with earthquakes*, ed. by m. Hayakawa, Transworld research network, Trivandrum, 41-96.
- Goodfellow, I., Bengio, Y. and Courville, A., 2016, *Deep learning*, Mit press.
- Graves, A., 2012, In supervised sequence labelling with recurrent neural networks. *springer, berlin, Heidelberg*, 5-12.
- Graves, A., 2013, Generating sequences with recurrent neural networks. Arxiv Prepr. Arxiv1308.0850.
- Hayakawa, M. and Molchanov, O. A., 2002, Seismo- electromagnetics: lithosphere-atmosphere-ionosphere coupling. *terra scientific publishing co. Tokyo*, 477.
- Huang, C. J. and Kuo, P. H., 2018, A deep CNN-LSTM model for particulate matter (pm_{2.5}) forecasting in smart cities. *Sensors*, 18 (7), 2220-2232.
- Khan, S., Rahmani, H., Shah, S.A.A. and Bennamoun, M., 2018, A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, 8(1), 1-207.
- Liu, J.Y., Chuo, Y.J., Shan, S.J., Tsai, Y.B., Pulinets, S.A. and Yu, S.B., 2004, Pre-earthquake-ionospheric anomalies registered by continuous GPS TEC. *Ann. Geophys.*, 22, 1585-1593.
- Mannucci, A. J., Wilson, B. D., Yuan, D. N., Ho, C. H., Lindqwister, U. J. and Runge, T. F., 1998, A global mapping technique for GPS-derived ionospheric total electron content measurements, *Radio Sci.*, 33, 565-582, doi: 10.1029/97RS02707.
- Parrot, M., 1995, Use of satellites to detect seismo-electromagnetic effects, main phenomenological features of ionospheric precursors of strong earthquakes. *Advances in Space Research*, 15 (11), 1337-1347.
- Pulinets, S. and Boyarchuk, K. A., 2004, *Ionospheric precursors of earthquakes*. Springer, berlin.
- Pulinets, S. and Ouzounov, D., 2011, Lithosphere - atmosphere - ionosphere coupling (LAIC) model - An unified concept for earthquake precursors validation. *Journal of Asian Earth sciences*, 41, 371-382.
- Sorokin, V. M. and Pokhotelov, O. A., 2014, Model for the vlf/lf radio signal anomalies formation associated with earthquakes. *Advances in Space Research*, 54 (12), 2532-2539.
- Williams, R. J. and Zipser, D., 1989, A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1 (2), 270–280.