Spatiotemporal Applications with Deep Learning:
Earthquake Pattern Mining and Prediction in Southern California

Bo Feng
Department of Intelligent Systems Engineering
Indiana University - Bloomington
fengbo@iu.edu

May 29, 2021

Abstract

This report is a research survey of deep learning models for spatiotemporal analysis for Earthquake pattern mining and prediction in Southern California. Geoscience and seismology have utilized the most advanced technologies and equipment to monitor seismic events globally from the past few decades. With the enormous amount of data, modern GPU-powered deep learning presents a promising approach to analyze data and discover patterns. In recent years, there are plenty of successful deep learning models for picking seismic waves. However, forecasting extreme earthquakes, which can cause disasters, is still an underdeveloped topic in history. Relevant research in spatiotemporal dynamics mining and forecasting has revealed some successful predictions, a crucial topic in many scientific research fields. Most studies of them have many successful applications of using deep neural networks. In Geology and Earth science studies, earthquake prediction is one of the world's most challenging problems, about which cutting-edge deep learning technologies may help discover some valuable patterns. In this report, we illustrate two deep learning modeling approaches, namely EQNet and EQPred, which utilize data feature fusion and model fusion to mine spatiotemporal patterns from data to nowcast extreme earthquakes by discovering visual dynamics in regional coarse-grained spatial grids over time. In these modeling approach, we use synthetic deep learning neural networks with domain knowledge in geoscience and seismology to exploit earthquake patterns for prediction using convolutional long short-term memory neural networks and other spatiotemporal applicable networks. Our experiments show preliminary but promising corelation between location prediction and magnitude prediction for earthquakes in Southern California. Ablation studies and visualization validate the effectiveness of the proposed modeling method.

Contents

1 Introduction

Spatial and temporal attributes have played an essential role in addressing scientific issues mathematically and statistically with large volumes of data in real problems. A worldwide team of scientists studied the published datasets from The WorldPop project (www.worldpop.org) for discovering the spatiotemporal pattern of population in China from 1990 to 2010 [1]. For modern Geoscience, spatiotemporal modeling has been studied for a long time. Authors of this book [2] summarized some initial efforts by utilizing spatiotemporal features for scientific interpretation and prediction. However, the model complexity and size of datasets were very limited.

Tradition machine learning algorithms like the support vector machine (SVM) and decision trees perform well on small datasets. Optimization methods such as stochastic gradient descent (SGD) enable the deep learning algorithms can be trained in small batches for extensive data without sacrificing model performance. Over the past few decades, large volumes of data have been collected by the seismological community. This drives high demand for seismology data processing and analysis, providing opportunities to predict future dynamics from history. Spatiotemporal forecasting is an important topic in many scientific research fields, in which there are a plethora of successful applications. Recent studies using deep neural networks have shown various successful applications, including car traffic forecasting [3], ride-hailing forecasting [4], rain/weather forecasting [5].

Over the past few decades, large volumes of data have been collected by the seismological community. This drives high demand for seismology data processing and analysis, which also provides opportunities to predict future dynamics from history. Spatiotemporal forecasting is an important topic in many scientific research fields, in which there are a plethora of successful applications. Recent studies using deep neural networks have shown various successful applications, including car traffic forecasting [3], ride-hailing forecasting [4], rain/weather forecasting [6], etc.

Earthquake forecasting is a worldwide challenging problem due to stochastic physics processing. Scientists around the world have built an enormous number of detectors for picking up earthquake signals. It is a general belief that earthquakes are predictable under some assumption that quakes are formed underneath the Earth are accumulated stresses in a gradual process over a long time. In this case, it would be possible to predict earthquake shocks for future activities of quakes by learning patterns from historical seismic events.

Conventionally, earthquakes are located through a process of detecting signals, picking up arrival time, and estimating epicenters of events using a velocity model. Efforts have been made to filter P-waves and S-waves from the original waveform signals of earthquakes and seismic noise [7]. In this project, our goal is to utilize the preprocessed seismic signals forming epicenters (location labels) to forecast the probabilities of the next earthquakes in an area. Earthquake forecasting consist of three major tasks in machine learning. The first task is to predict when the next seismic event will happen in a specific region. The second task is to predict whether or not the next seismic event will come. The third

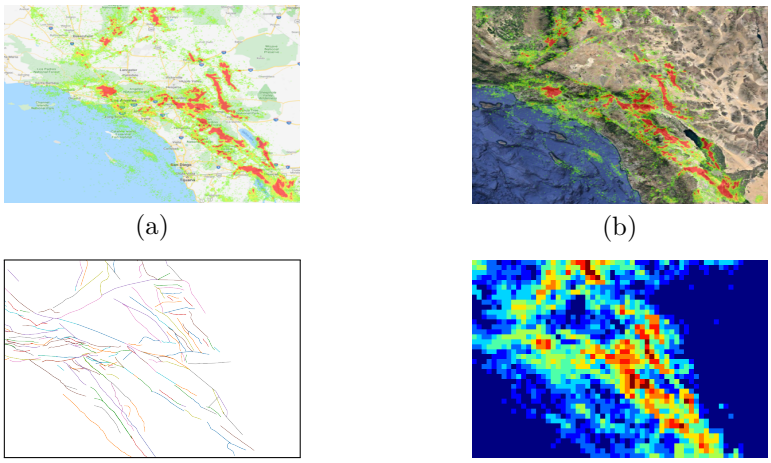


Figure 1: Dataset overview of earthquakes in Southern California. (a) Earthquake events mapped on Maps. (b) Earthquake events mapped on satellite images. (c) Fault lines plotted on Earth surface. (d) Heat map of events in a grid view.

task is to predict the level of magnitude of the upcoming seismic events so that a major shock can be predicted.

Deep learning neural networks have presented a widely successful approach to capture spatial-temporal dependencies of problems to achieve accurate forecasting results. Convolutional neural networks have achieved convinced success in computer vision, image object recognition, etc [8]. Here we test the hypothesis that earthquake patterns can be perceived by learning historical seismic events. However, epicenters' prediction is learned from annotated seismograms. Due to the uncertainties of earthquakes, even the ground truth labels that are annotated by domain experts may be biased. Locations and magnitude of epicenters are maybe adjusted after the seismic event happened a long while. In summary, we cover two major endeavors in predicting earthquakes patterns for a selected region in Southern California.

- Both projects transform the single contiguous time series tablet dataset into sequences of images over time in different level of frequencies.
- EQNet fuses data features by building high dimensional properties of input to train a convolutional recurrent neural network model for future prediction.
- EQPred builds a predictive architecture by fusing two neural network models, in which one model learns the latent variables from normal quakes and another model learns to predict extreme large quakes from these latent variables.

2 EQNet: Feature Fusion for Nowcasting

2.1 Introduction

Spatial temporal forecasting is an important topic in many scientific research fields. Recent studies of using deep neural networks for spatiotemporal forecasting have been taking advances in many domain science areas, such as neuroscience, rain fall and transportation. Li et al [3] and Ma et al [9] have modeled spatio and temporal dependencies with convolutional neural networks for traffic forecasting.

The goal of spatiotemporal forecasting is to predict what and when the next event will happen. This is a task that includes two orthogonal sub-tasks: forecasting its spatio dependencies and temporal dependencies. However, this is a nontrivial task due t the high dimension features of time series sequences and building models that can work well for some scientific problems can also be very vague [10].

In this write-up, we summarize and discuss some critical techniques in Deep Learning and how they are applied to model spatial temporal problems. These techniques include a series of neural networks such as convolutional networks, recurrent neural networks and encoder decoder architectures. In this project, we propose to model the earthquake prediction in terms of spatiotemporal dependences in Southern California. In summary:

- We study the earthquake dataset for the Southern California and reconstruct the time series events into a sequence of 2D images.
- We model the spatiotemporal dependencies of earthquakes in Southern California with Convolutional Long Short Term Memory deep neural networks and show some preliminary results.

2.2 Earthquake Feature Fusion for Prediction

In this project, we convert the catalog dataset into a time series of images with the size 60x40 by enriching the properties of each pixel in images. We train and evaluate a model that can predict various of properties.

2.2.1 Dataset Statistics

All the following experiments are conducted using the same training data and test data from 1950 to 2019 in the Southern California, where the longitude is from -120 to -140, and the latitude is from 32 to 36. We divide this area into a grid with 60x40 cells, so that each cell can represent multiple properties. The total number of valid days in this dataset is 25567.

2.2.2 Features and Task Settings

We set the predictive goal of the training tasks as multiple variables including the properties of magnitudes in a year, 5 years, etc. So we enrich both the training and testing data to match our task settings in the model. The feature

Table 1: Basic statistics of the input and output features

Properties	**Count (NaN)**	**Min**	**Max**	**Mean**	**STD**
❶Magnitude	317736	0.000	6.400000	0.007577	0.114197
❷Log-Energy	317747	0.000	7.500379	0.008219	0.124390
❸Fourth-root of Energy	318109	0.000	649.594092	0.025375	0.524219
❻Energy-weighted Depth	317501	-2.122	105.600000	0.041278	0.668506
❹Multiplicity	318109	0.000	956.000000	0.009926	0.478297

properties are listed in Table 1. The selected area is split into a 60x40 grid. In each cell of the grid, ❶Magnitude means average magnitude within this cell and ❷Log-Energy means the log energy define as $(10^M)^{3/2}$. ❸Fourth-root of Energymeans the fourth root of the sum energy. ❹Multiplicity represents the number of shocks happened in this cell. And ❺Average Depth is the average of depth over all shocks in this cell. ❻Energy-weighted Depth is the energy weighed depth.

2.3 Method: Neural Network Models

In this section, we firstly formalize the earthquake prediction as a spatiotemporal forecasting problem in terms of its spatial and temporal dependencies. And then, we describe how the spatial and temporal dependencies can be modeled by using CNN-LSTM neural networks.

Earthquake hypocenter locations are a series of events that happened as a time sequence when different body wave phases. It is well known that these data are nonlinear functional of the compressional or shear wave velocity structure and the coordinates of the source in space-time [11]. Figure 2 shows all hypocenters are plotted in maps.

2.3.1 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) draw many successful applications in recent years and it becomes the state-of-the-art in image related approach in deep learning. In 2015, Berkeley [12] revealed that fully convolutional networks for semantic segmentation adopting AlexNet [13], VGGNet [14], and Google Net [15] can improve training and simplify the state-of-art training and inference for image classification.

CNN has also enabled its capability to efficiently deal with spatially-correlated problems via locally-connected convolution layers [16] Geo locations of hypocenters in the earthquake dataset can be represented in graph models, in which the distance can be measured as an Euclidean distance based on longitudes and latitudes of hypocenters in 2D maps.

2.3.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory Neural Networks (LSTM) is a type of recurrent neural networks (RNN) in which prior knowledge can be reused [17, 18, 19]. LSTM is capable of learning long term dependencies by remembering information for long periods of time. We propose to use LSTM to model temporal dependence of earthquakes in time-series sequence to sequence prediction.

2.3.3 Convolutional LSTM (ConvLSTM)

Convolutional LSTM is an extension of fully-connected LSTM (FC-LSTM) [10], which has convolutional LSTM and can be used to build trainable models for forecasting problems. It has been successfully adopted to predict actions affect object in a real world environment. [20] Compare to FC-LSTM, CNN-LSTM is able to handle spatiotemporal sequences. Internally, spatiotemporal sequence can be transformed from a sequence of 2D matrices, which is a 3D tensor in computing. CNN-LSTM learns the temporal state to state transitions while keeping the spatial information via convolutions. The following equations show how the CNN-LSTM cell does the computation:

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\
C_t &= f_t \circ C_{t-1} + i_t \circ tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\
H_t &= o_t \circ tanh(C_t)
\end{aligned}
\tag{1}
$$

, in which $*$ represents the convolution product, $\circ$ represents the element-wise product, X is the input, C is the cell output, H represents hidden states of cells, and i, f, o are gates within cells.

2.4 Experiments and Evaluation

2.4.1 Model Implemenation and Dataset Preprocessing

The earthquake dataset is a tablet formatted dataset in which each record is an earthquake hypocenter with a timestamp, a GEO location, a magnitude, and depth. In this study, we focus on the prediction of what time and location the next big earthquake will happen and the depth is ignored in this case. The dataset contains all earthquake events in Southern California ranging from 1990 to 2019 year. Figure 1 shows all events plotted in 2D maps, in which hot spots are areas where earthquakes frequently happened or big earthquakes happened in history.

We divide the Southern California (Longitude: -120~-140, Latitude: 32~36) into a grid with 40×60 cells, each of which has 1 degree of longitude and latitude. Firstly, it is easy to group events into years. Then, let x, y denote the longitude and latitude location of an event. All events are accumulated in corresponding cell where x, y fall into. The value of each cell is the mean of magnitudes of all

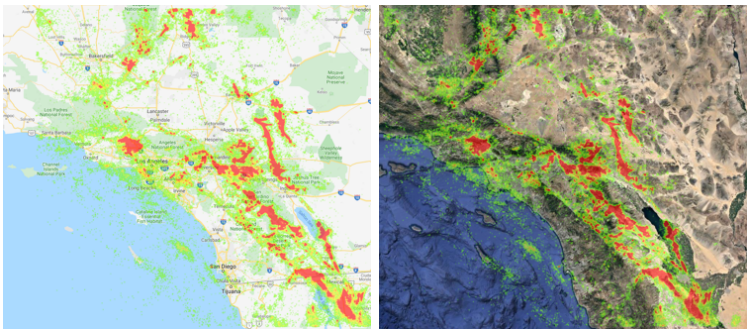


Figure 2: Dataset overview of earthquakes in Southern California

events within the cell. As a result, each year is represented by a 2D image-like matrix (40×60).

2.4.2 Empirical Study

We build a neural network model with CNN-LSTM layers as shown in Figure 3a, in which there are 3 layers of CNN-LSTM connecting with 3 layers of Batch Normalization, and 3D convolution layer being used as the final layer.

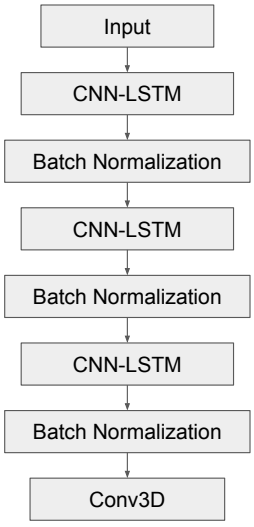
There are two aspects in the consideration of this model:

- In a period of time sequence, a sequence of k 2D matrices are the input: $t_1, t_2, \ldots, t_k$, and the output is another sequence:$t_2, t_3, \ldots, t_{k+1}$. In this way, the t_{k+1} is the predicted result.
- In Southern California, the model can be trained and predict the roughly location of the next hypocenter of earthquake with magnitude in the predicted 2D matrix. For example, if the input is X_t at t time, the output from the model is X_{t+1} at $t+1$ time. The predicted magnitude of GEO location at x, y in Southern California can be found in $X_{t+1}[y][x]$.

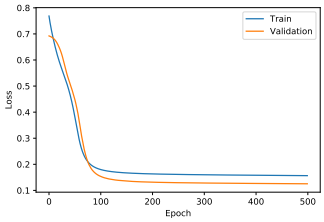
2.4.3 Performance Evaluation

We train the model for 500 epochs. The training loss history is shown in Figure 3b. The ground truth data and predicted data for 2017 and 2018 are compared in Figure 4, in which magnitudes are normalized within range of 0 to 1 and darker cells represent higher magnitude.

To quantify the prediction, we simply calculate the coefficient of determination (R^2) scores between between ground truth and prediction for 2017 and 2018. Table 2 shows the R^2 scores of the comparison between ground truth and prediction under different thresholds. Here the R^2 is defined as $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, $SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$,$SS_{\text{reg}} = \sum_i (f_i - \bar{y})^2$,$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$, in which y_i is the ground truth and f_i is the predicted data. Thresholds are used because we focus on events with big magnitudes and filter events with small magnitudes, which



(a) Model architecture used in these experiments



(b) Training and validation loss history of the model training

Figure 3: Experimental neural network model applied for earthquake prediction in Southern California

are tend to be just noise. According our experiments, the R^2 scores are 0.69, 0.71 for 2017, 2018 year respectively when the threshold is 0.25. The R^2 scores are 0.91, 0.93 for 2017, 2018 year respectively when the threshold is 0.3.

2.5 Summary

In this project, we discuss how we can model spatial temporal forecasting problems using deep neural networks and we propose a model to address this problem. In experiments, we demonstrate some preliminary results of using CNN,LSTM, and ConvLSTM to predict earthquakes in Southern California. According our experiments, we show some promising results when proper thresholds are chosen to filter out noisy. Future works could include tuning parameters of this model and compare the performance with other models.

3 EQPred: Model Fusion for Extreme Event

3.1 Introduction

In this project, we propose joint modeling of using a self supervised autoencoder and temporal convolutional (TCN) neural networks for earthquake prediction by modeling spatiotemporal dependencies in Southern California. Additionally,

Table 2: R^2 score evaluation for predictions

Year	Threshold	R^2 score
2017	0.25	0.69
2017	0.3	0.91
2018	0.25	0.71
2018	0.3	0.93

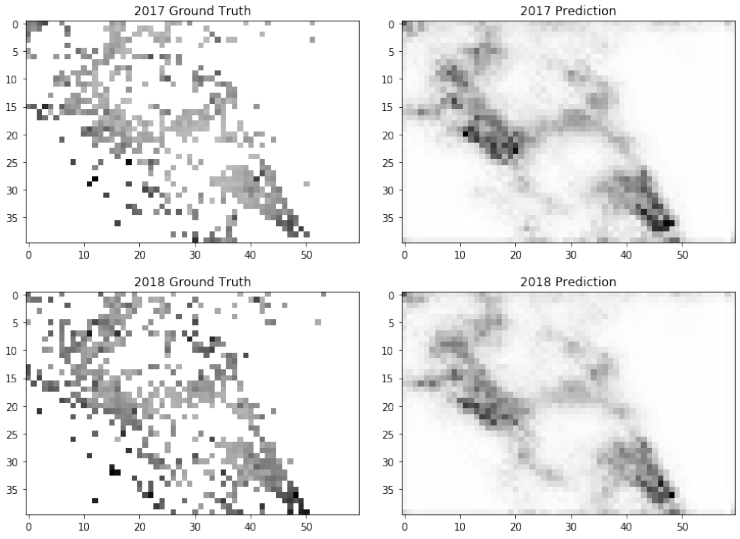


Figure 4: Ground truth vs. prediction of earthquakes in Southern California

EQPred comprehensively improves the autoencoder and TCN by incorporating skip connections and local temporal attention mechanisms. Compared to conventional recurrent neural networks or a single model, our joint modeling presents some advantages in predicting major shocks in the area of study. In summary:

- We study the earthquake dataset for Southern California and reconstruct the time series events into a sequence of 2D images.
- We model the spatiotemporal dependencies of earthquakes in Southern California with an improved autoencoder and TCN neural networks and show some preliminary but promising results for forecasting events.

3.2 Modeling Approach

The proposed prediction model consists of two major components, an autoencoder which learns the latent space distribution from the image like view of the earthquakes and a prediction network which learns to predict the likelihood of the next main shock happening within the same area.

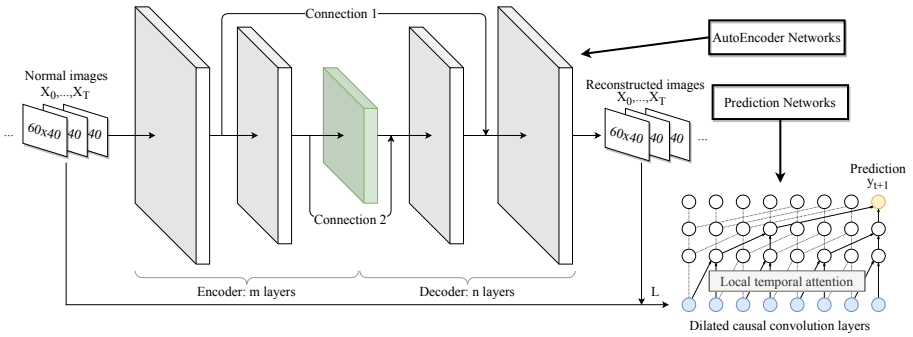


Figure 5: EQPred: overview of earthquake prediction networks.

3.2.1 Data Considerations

The earthquake catalog is a tablet formatted dataset. In this project, we focus on time and geo location shocks. The dataset contains all earthquake events in Southern California ranging from the year 1990 to 2019. Figure 1 shows all events plotted in 2D maps, in which hot spots are areas where earthquakes frequently happened or big earthquakes happened in history.

3.2.2 Energy-based data models

Seismometers record seismic events from calibrating vibrations of waves. Magnitude in the dataset represents measured amplitude as measured seismogram. While they are discrete data points, accumulating all magnitudes by summing them up by averaging makes the temporal information loss, and deemphasizes large earthquakes. In contrast to magnitude, earthquakes release energy can help mitigate this issue by two folds: 1) accumulated energy value in a region can represent the energy released by the stress of Earth over time; 2) energy data model naturally highlights large events since the energy of large events can be an order of magnitude higher than that of small events. The formula of converting earthquake magnitude to energy is defined as

$$\mathbf{E} = (10^{\mathbf{M}})^{3/2} \tag{2}$$

in which the magnitude $0 \leq \mathbf{M} \in \mathbb{R} \leq 10$. [1]

3.2.3 Location-aware data weaving

As a time-series prediction task, the earthquake catalog contains locations and magnitudes, which could be used as target properties. However, it could be more nature to reorganize the 1D time-series dataset into a 2D sequence dataset by dividing a map region into small boxes according to longitudes and latitudes and

[1] Earthquake magnitude can be even negative for very small events that are negligible. This scale is also open-ended, but events with magnitudes greater than 10 are clipped to 10.

aggregating the released energy within a small box per specific time frequency. So each element of the sequence becomes a summation of all energy released at the location (i, j): $X^t_{i,j} = \sum((10^{\mathbf{M}})^{3/2}), i \in [0, M)$ and $j \in [0, N)$, which means X^t has a shape $M \times N$ for M boxes along the latitude and N boxes along the longitude.

3.2.4 AutoEncoder for Effective Spatial Modeling

Main shocks with large magnitudes are rare in terms of statistics and nature physics. In addition, earthquakes are full of stochastic processing, resulting in seismic signals are very noisy. To predict the future main shocks, we first model the spatial patterns within the southern California area.

We use an autoencoder to recognize the spatial pattern changes under normal circumstances and abnormal circumstances. Compared to variational autoencoders (VAE), we do not assume Gaussian distribution or any other kinds of distributions for the latent space. In addition, the reconstructed results from VAE are tended to be more noisy. We also make some experiments for full comparison in Section 3.3. This is a semi-supervised process of pre-training a model that learns the representation of earthquake images. We train this model by using the following equation.

$$L(\mathbf{X_{normal}}, g(f(\mathbf{X_{normal}})) + \Omega(h, \mathbf{X_{normal}}) \tag{3}$$

, where $\mathbf{X_{normal}}$ are images of earthquakes with magnitudes less than a threshold, f is an encoder function, g is an decoder function, and Ω is a function that regularizes or penalizes the cost.

3.2.5 Spatial modeling

After the seismic events are parsed and transformed to image-like sequences in Section 3.2.1, we can utilize the spatial dependencies between pixels. Convolutional operations are common image feature extraction means. Pixel relationship can be easily mapped to geology locations of events.

3.2.6 Skip connections

We incorporate skip connections in the AutoEncoder architecture. Skip connections are forward shortcuts in networks. They symmetrically connect layers from the encoder and decoder as shown in Figure 5. This strategy allows long skip connections to pass features from the encoder path to the decoder path directly, which can recover spatial information lost due to downsampling, according to [21].

3.2.7 Bottleneck layer

The bottleneck layer in the AutoEncoder is deliberately set to a small vector of a size k feature map. This design is effective for two reasons. Firstly, it regularize

the model from overfitting all samples. Secondly, a small feature map can better differentiate the abnormal cases from normal cases.

3.2.8 TCN Model for Effective Temporal Modeling

In this work, the goal of forecasting earthquakes is to predict the future probability of a major shock happening in Southern California. This can be done in a prediction network, which is fed in the information gained from the AutoEncoder. A long short-term memory (LSTM) model can predict well on this task. However, in EQPred we incorporate an enhanced TCN (Figure 5), which can outperform LSTM. This situation is similar in predicting other physics related fields of study. For example, TCN is used to predict climate changes [22]. This is further analyzed in the following sub sections.

Conditional Temporal Convolution Temporal convolution neural networks are used to improve the temporal locality prediction over time. Temporal convolutional layers are layers containing causal convolution with varied dilation rate in 1D convolutional layers [23, 24]. A typical configuration of temporal convolution layers is set the dilation rate corresponding to the i-th of layers, for example 2^i.

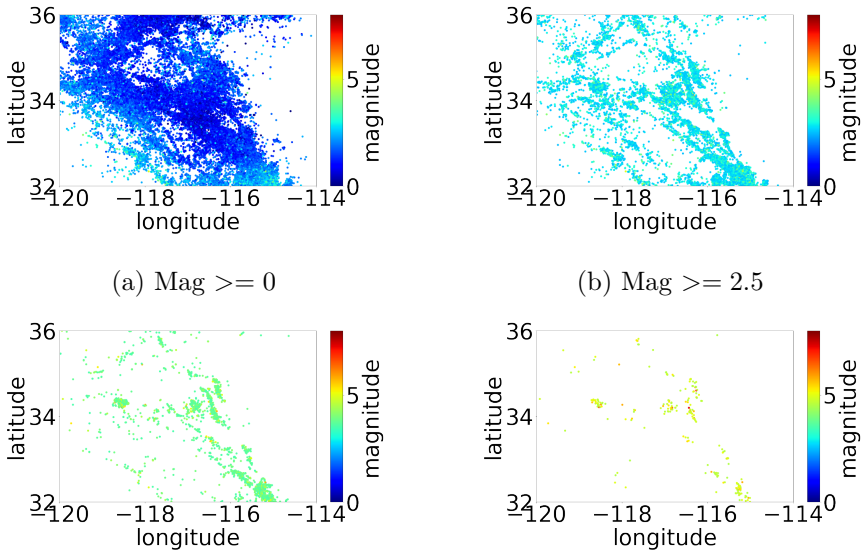
$$p(y|\theta) = \prod_{t=1}^{T} p(y_{t+1}|y_1, \ldots, y_t, \theta) \tag{4}$$

Local Temporal Attention A localized attention process to enhance temporal information passing is inspired by self-attention structure from Transformer [25], and Hao *et al.* work for sequence modeling [26]. The process incorporates functions f, g, and h to calculate d dimensional vector of keys $\mathcal{K}$, queries $\mathcal{Q}$, and values $\mathcal{V}$ respectively. Then, we calculate the weight matrix by $W = \frac{\mathcal{K} \cdot \mathcal{Q}}{\sqrt{d}}$. Finally, we apply a softmax function to the lower triangle of W to get a normalized attention weight $W_{attention} = softmax(W)$ and the final out of this layer can be calculated via this attention weighted summary: $\sum_{t=1}^{T} W_{attention} \cdot y_t$.

3.2.9 Smooth joint Nash–Sutcliffe efficiency: NSE

Nash–Sutcliffe model efficiency coefficient (NSE) is a commonly used metric to evaluate a predictive model. NSE is widely used to evaluate predictive skills in scientific studies, such as hydrology [27]. The value range of NSE is $(-\infty, 1)$. NSE can become negative when the mean error in the predictive model is larger than one standard deviation of the variability. Its equation is defined as follows.

$$NSE = 1 - \frac{\sum_{t=0}^{T} (\hat{y}_t - y_t)}{\sum_{t=0}^{T} (y_t - \bar{y})} \tag{5}$$



(a) Mag >= 0

(b) Mag >= 2.5

(c) Mag >= 3.5

(d) Mag >= 4.5

Figure 6: Dataset overview: (a) 444, 589 events with magnitude ≥ 0.0, (b) 24, 822 events with magnitude ≥ 2.5, (c) 2, 489 events with magnitude ≥ 3.5, (d) 237 events with magnitude ≥ 4.5

3.3 Experiments and Evaluation

3.3.1 Dataset augmentation and preprocessing

The earthquake dataset is a tablet formatted dataset in which each record is an earthquake epicenter with a timestamp, a GEO location, a magnitude, and depth. We preprocess the catalog according to the analysis in Section 3.2.1.

We divide the Southern California (Longitude: -120~-140, Latitude: 32~36) into a grid with 60×40 cells, each of which has 0.1 degree of longitude and latitude for about 11.1km (1 degree in kilometers is about 111km). Firstly, it is easy to group events into daily intervals. Then, let x, y denote the longitude and latitude location of an event. All events are accumulated in corresponding cell where x, y fall into. The value of each cell is the mean of magnitudes of all events within the cell. As a result, each day is represented by a 2D image-like 60×40 matrix.

3.3.2 Experimental setup

Our EQPred model and other baseline models are implemented with TensorFlow in Python. All experiments are conducted on a machine with 8 NVidia K80 GPUs. All models, include EQPred and baseline models are trained using Adam or SGD optimizers with a fine-tuned learning rate and mean squared error as

Table 3: EQPred AutoEncoder vs. VAE.

Model	MSE	Accuracy	Variance
EQPred	0.148	0.968	1.432
VAE [28]	0.157	0.971	1.986

Table 4: Varying the latent space dimension.

Latent space dimension	MSE	Accuracy
16	0.148	0.968
64	0.140	0.968
128	0.138	0.972
1024	0.137	0.984

training loss. All model weights are check-pointed and we select the best model weights for testing. Events with magnitudes ≥ 4.5 are labeled as extreme major shocks.

3.3.3 Experimental Results

In these set of experiments, we aim to demonstrate the performance of EQPred compared to a series of baseline models. Firstly, we show the performance differences between autoencoder in EQPred and a VAE. Then, we compare the prediction network with a LSTM. Finally, we illustrate the comprehensive results from using EQPred comparing with a series of methods.

3.3.4 AutoEncoder

As we mention an autoencoder is used in EQPred in Section 3.2 as opposed to a variational autoencoder, we compare the results of using EQPred autoencoder with a common VAE. The performance results are summarized in Table 3. Even though VAE can achieve almost the same performance in terms of accuracy, it has higher mean squared loss and variance for the final output. Higher MAE loss and variance affect the performance of the prediction network.

3.3.5 Prediction

We analyze the TCN in EQPred in Section 3.2 comparing with a LSTM model. For this time series forecasting, the prediction network in EQPred can outperform the LSTM network. Due to the stochastic nature of shocks, the output series from the autoencoder is denoised by the LOESS smoothing method [29]. We summarize the experimental results in Table 5.

Table 5: Results comparison between EQPred and baseline models. Some models adopt the same architecture of using an autoencoder and a prediction network. These models are named with a '+' sign.

Models	**MAE**	**Precision**	**Recall**	**F-1**	**F-0.2**	**NSE**
MLP	-	0.2631	0.2845	0.2096	0.2494	-1.4739
LSTM	-	0.4596	0.5186	0.3801	0.4058	-0.2059
Conv2D-FC	-	0.4589	0.3963	0.4340	0.4394	-0.1867
Conv2D-LSTM	-	0.4299	0.4069	0.4217	0.4243	-0.4022
ConvLSTM2D-FC	-	0.4633	0.3289	0.3763	0.3801	-0.1714
MLP+MLP	0.2570	0.7525	0.6338	0.6652	0.7113	0.6778
MLP+LSTM	0.1637	0.8420	0.7085	0.7599	0.8021	0.7890
MLP+Conv1D	0.1484	0.8571	0.9351	0.8029	0.8342	0.8133
Conv2d+MLP	0.1484	0.8577	0.7944	0.7887	0.8098	0.8108
Conv2D+LSTM	0.1410	0.8640	0.8776	0.8609	0.8683	0.8222
Conv2D+Conv1D	0.0588	0.9420	0.9115	0.8998	0.8688	0.9293
EQPred	**0.0483**	**0.9563**	0.9016	**0.9251**	**0.9341**	**0.9323**

Table 6: Ablation study by removing core components in EQPred.

Models	**F-1**	**NSE**
W/O skip connections	0.9001	0.9233
W/O local temporal attention	0.9247	0.9289
EQPred	**0.9251**	**0.9323**

3.3.6 Comprehensive Analysis

In this set of experiments, we list several commonly used models for predicting the future main shocks. The results are summarized in Table 5. In this table, MLP represents a three-layer of fully connected neural networks. LSTM represents a two-layer of stateful LSTM neural networks. Conv2D, Conv1D represent a neural network consisting of one 2D convolutional and one 1D convolutional layer, respectively. From this table, we illustrate EQPred can outperform a single model significantly and other combination of models for this task.

3.3.7 EQPred Ablation Study

In the following two sets of experiments, we demonstrate the two major techniques that can improve the autoencoder and the prediction network: skip connections and local temporal attention. In the first set, we remove the skip connections in the autoencoder and keep the remaining parts the same. In the second set, we remove the local temporal attention in the prediction network and use the same autoencoder as the EQPred. Table 6 shows the results of these two sets of experiments.

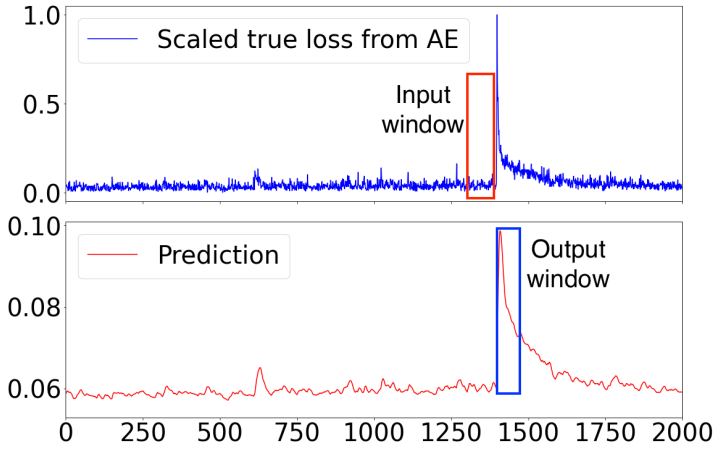


Figure 7: EQPred prediction.

3.3.8 Discussion and Empirical Study

We build joint models as shown in Figure 5, in which the autoencoder can learn the spatial pattern and the predictor can forecast future event. Figure 7 shows a prediction example. Given an input sequence window, the predictor can output a future sequence window, from which a major shock can be detected. There are two aspects in the consideration of this model: 1) During the training period, a sequence of T 2D matrices are the input: $X_{t_1}, X_{t_2}, \ldots, X_{t_T}$, and the output is another sequence:$y_{t_2}, y_{t_3}, \ldots, y_{t_{T+1}}$. In this way, the $y_{t_{T+1}}$ is the predicted result. This means that the model can be trained on rolling basis as the data stream in. 2) In Southern California, the model can be trained and predict a novelty score which represents the probability of the next major shock. For example, if the input is X_t at t time, the output from the model is X_{t+1} at $t+1$ time. The predicted probability of this area can be told from y_{t+1}.

3.4 Summary

In this project, we propose EQPred, a joint modeling approach that mines the spatial and temporal dynamics from the dataset and predict extreme event by using learned latent variables. We dissect the problem settings for forecasting earthquakes, discuss how we model spatial temporal forecasting problems using deep neural networks. In contrast to 11 different approaches in the experiments, we demonstrate the effectiveness of EQPred to predict extreme cases in Southern California. According the metrics from our experiments, we show some promising when proper thresholds are chosen to filter out noisy.

4 Conclusions

In this report, we propose two modeling approaches for earthquakes analysis, namely EQNet and EQPred. EQNet augments data by aggregating multiple features to form a time series of images, and EQPred combines two models so that one mines the spatial and temporal dynamics from the dataset and another predicts extreme event by using learned latent variables. We dissect the problem settings for forecasting earthquakes, discuss how we model spatial temporal forecasting problems using deep neural networks.

Even though we have study a few modeling approach and find our the most effective one, the domain knowledge is still required from Geoscience experts. In future, we plan to verify this approach in wider areas and we also consider other physics quantities like seismicity, electric field, magnetic field, deformation which are highly possible correlated to earthquake events.

Code and data availability

The earthquake raw event dataset used in the paper is available to download from the USGS website at https://www.usgs.gov/. Model codes and parsed datasets used in the paper will be published upon acceptance of this manuscript.

References

[1] Andrea E. Gaughan, Forrest R. Stevens, Zhuojie Huang, Jeremiah J. Nieves, Alessandro Sorichetta, Shengjie Lai, Xinyue Ye, Catherine Linard, Graeme M. Hornby, Simon I. Hay, Hongjie Yu, and Andrew J. Tatem. Spatiotemporal patterns of population in mainland China, 1990 to 2010. *Scientific Data*, 3(1):160005, December 2016.

[2] George Christakos. *Modern Spatiotemporal Geostatistics*. Oxford University Press, November 2000.

[3] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. *arXiv:1707.01926 [cs, stat]*, July 2017.

[4] L. Zhu and N. Laptev. Deep and Confident Prediction for Time Series at Uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 103–110, November 2017.

[5] Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep Factors for Forecasting. In *International Conference on Machine Learning*, pages 6607–6617. PMLR, May 2019.

[6] Senzhang Wang, Jiannong Cao, and Philip Yu. Deep Learning for Spatio-Temporal Data Mining: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.

[7] S. Mostafa Mousavi, William L. Ellsworth, Weiqiang Zhu, Lindsay Y. Chuang, and Gregory C. Beroza. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, 11(1):3952, August 2020.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[9] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors*, 17(4):818, April 2017.

[10] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015.

[11] Gary L. Pavlis and John R. Booker. The mixed discrete-continuous inverse problem: Application to the simultaneous determination of earthquake hypocenters and velocity structure. *Journal of Geophysical Research: Solid Earth*, 85(B9):4801–4810, 1980.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[13] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *arXiv:1602.07360 [cs]*, November 2016.

[14] Andrew Lavin and Scott Gray. Fast Algorithms for Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4013–4021, 2016.

[15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Boston, MA, USA, June 2015. IEEE.

[16] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, January 2013.

[17] F.A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with LSTM. In *1999 Ninth International Conference on Artificial*

Neural Networks ICANN 99. (Conf. Publ. No. 470), volume 2, pages 850–855 vol.2, September 1999.

[18] Alex Graves. Generating Sequences With Recurrent Neural Networks. *arXiv:1308.0850 [cs]*, June 2014.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised Learning for Physical Interaction through Video Prediction. *arXiv:1605.07157 [cs]*, October 2016.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 630–645, Cham, 2016. Springer International Publishing.

[22] Jining Yan, Lin Mu, Lizhe Wang, Rajiv Ranjan, and Albert Y. Zomaya. Temporal Convolutional Networks for the Advance Prediction of ENSO. *Scientific Reports*, 10(1):8055, May 2020.

[23] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv:1609.03499 [cs]*, September 2016.

[24] Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. Conditional Time Series Forecasting with Convolutional Neural Networks. *arXiv:1703.04691 [stat]*, September 2018.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[26] Hongyan Hao, Yan Wang, Yudi Xia, Jian Zhao, and Furao Shen. Temporal Convolutional Attention-based Network For Sequence Modeling. *arXiv:2002.12530 [cs]*, March 2020.

[27] Daniel N. Moriasi, Jeffrey G. Arnold, Michael W. Van Liew, Ronald L. Bingner, R. Daren Harmel, and Tamie L. Veith. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, 50(3):885–900, 2007.

[28] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, December 2013.

[29] Jesús Rojo, Rosario Rivero, Jorge Romero-Morte, Federico Fernández-González, and Rosa Pérez-Badia. Modeling pollen time series using seasonal-trend decomposition procedure based on LOESS smoothing. *International journal of biometeorology*, 61(2):335–348, 2017.