Spatiotemporal Pattern Mining for Nowcasting Extreme Earthquakes in Southern California

Bo Feng
Intelligent Systems Engineering
Indiana University Bloomington
Bloomington, Indiana, USA
Email: fengbo@iu.edu

Geoffrey C. Fox
Intelligent Systems Engineering
Indiana University Bloomington
Bloomington, Indiana, USA
Email: gcf@indiana.edu

***Abstract*—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 project, we propose a deep learning modeling approach, namely EQPRED, to mine spatiotemporal patterns from data to nowcast extreme earthquakes by discovering visual dynamics in regional coarse-grained spatial grids over time. In this 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. Our experiments show a strong correlation between location prediction and magnitude prediction for earthquakes in Southern California. Ablation studies and visualization validate the effectiveness of the proposed modeling method.**

***Index Terms*—Spatiotemporal, Convolution, Recurrent Neural Network, LSTM, Temporal Convolution, Nowcasting**

I. 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. In this book [2], authors summarized some initial efforts by utilizing spatiotemporal features for scientific interpretation and prediction.

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], etc.

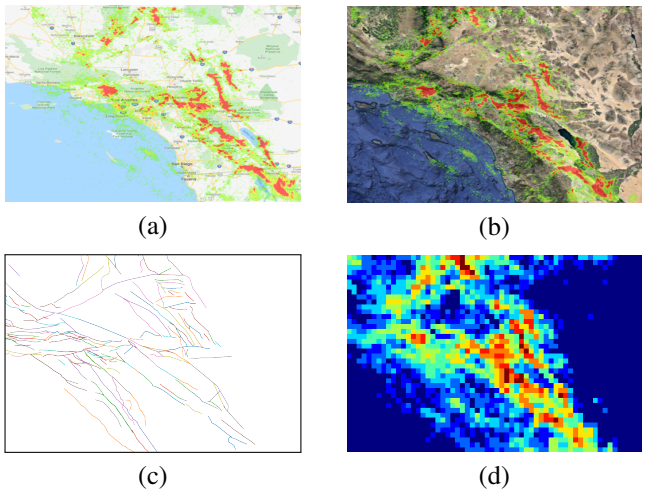


Fig. 1. Dataset overview of earthquakes in Southern California. (a) Earthquake events mapped on Maps. (b) Earthquake events mapped on satellite images.

Recent studies in deep neural networks have many successful applications of using deep learning for spatiotemporal 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 spatial dependencies and temporal dependencies. However, this is a nontrivial task due to the high dimension features of time series sequences and building models that can work well for some specific problems can also be very vague.

Earthquakes are caused by the sudden release of energy from seismic waves [6], [7]. However, this involves the movement of ground plates via a stochastic process, which makes Earthquake

forecasting is a worldwide challenging problem. 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 subsequent earthquakes in an area.

Earthquake forecasting consists 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 task is to predict the level of magnitude of the upcoming seismic events to predict major shocks.

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 this project, we propose joint modeling of using a self supervised autoencoder (AE) and temporal convolutional neural networks (TCN) [9] for earthquake prediction by modeling spatiotemporal dependencies in Southern California. Additionally, 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 nowcasting events in contrast to 11 baseline models.

This paper is organized as following: Section II reviews related work in four aspects. In Section III, we propose a spatiotemporal modeling approach, namely EQPRED, and illustrate the how spatial and temporal dynamics are modeled theoretically. In Section IV, we show the detailed implementation of EQPRED, run analysis on the dataset, and evaluate the effectiveness. Finally, we conclude in Section V with the discussion of the limitations and future research directions.

II. Related Work

Four subjects are related to this project: 1) predicting epicenters only with advanced machine learning techniques; 2) spatiotemporal modeling in a broad range of applications; 3) advanced dynamic pattern mining and prediction in visual applications; 4) extreme event prediction in other areas of research.

A. Convolutional Methods in Predicting Epicenters

Estimating and predicting the epicenters of earthquakes has a long history. Scientists from Geophysics, Geology, and Seismology have developed various tools and analytical functions to predict epicenters from datasets. In 1997, Bakun and Wentworth suggested using Modified Mercalli intensity datasets for Southern California earthquakes to bound the epicenter regions and magnitudes [6]. In 1998, Pulinets proposed predicting epicenters of strong earthquakes with the help of satellite-sounding systems. Scientists from Greece had illustrated a successful project which predicted the prominent aspects of earthquakes using seismic electric signals [10]. Recently, Guangmeng *et al.* attempted to predict earthquakes with satellite cloud images and revealed some possibilities of predicting earthquakes using geophysics data [11]. Zakaria *et al.* presented their work of predicting epicenters by monitoring precursors, such as crustal deformation anomalies and thermal anomalies, with remote sensing techniques [12]. These studies either used only too little data or too simple analytical models.

B. Spatiotemporal Dynamics and Generative Models

Most recently, it is a prevailing method to make predictions by modeling the spatiotemporal dynamics for domain science problems. This is because large volumes of data are increasingly collected in the vast majority of domains including, social science, epidemiology, transportation, and geoscience. Cui *et al.* proposed to use graph convolutional long short-term memory neural networks to predict traffic via capturing spatial dynamics from the car traffic patterns [13]. Li *et al.* utilized a seq2seq neural network architecture to capture spatial and temporal dependencies for traffic forecasting by incorporating a diffusion filter in convolutional recurrent layers [3]. FUNNEL was a project proposed by Matsubara [14]. It was designed to use an analytical model and a fitting algorithm for discovering spatial-temporal patterns of epidemiological data.

C. Visual Pattern Prediction

Lotter *et al.* presented a model to predict video frames with deep predictive coding networks [15], which was based on the ConvLSTM2D network module with specific top-down states updating algorithm. [16] is another example in predicting video frames. The authors of this work presented the effectiveness of modeling object motion via predicting future object pixels.

For example, a ball moves and a block falls. These models are successful for predicting contiguous and dense image frames, whereases the earthquake data are very sparse, the extreme shocks are very rare in terms of probability.

D. Extreme rare event prediction

Laptev *et al.* [17] proposed their modeling to predict rare trip demands for ride-hailing service. In that paper, they built an end-to-end architecture using joint modeling by combining LSTM autoencoder and LSTM predictor networks. They showed their forecasting capability on a Uber's public dataset. Geng *et al.* [18] proposed another model to forecast the ride-hailing demand using graph-based recurrent neural networks, in which graphs are defined by road networks with Euclidean and non-Euclidean distances. We compared this approach with our EQPRED, a detailed discussion of which is in Section IV-C.

III. EQPRED MODELING

Earthquakes come with an epicenter which is a point at the surface on Earth, and the mainshocks of quakes are regarded as the main contribution to significant disasters. To model the spatiotemporal patterns of earthquake shocks, we firstly consider the data attributes of specialty, which are discussed in Subsection III-A, then we pre-process the data in Subsection III-B, and then the spatial and temporal dynamics are modeled in Subsection III-C, III-D respectively.

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 that learns to predict the likelihood of the next main shock happening within the same area.

A. Energy-based Data Models

Geographical data are coordinates related. Intuitively, those shocks that happened in different Geo terrain may take effect to future shocks unevenly. The earthquake dataset used in this project is a tablet formatted catalog containing information on shocks in terms of geo-coordinates and magnitudes. In this project, we focus on the time and geolocation of shocks, other attributes like types of quakes are not included. This dataset covers all earthquake events in Southern California from the year 1990 to 2019. The full dataset is used in all the following experiments. 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. Figure 3 shows scatter plots of events with magnitudes equal and greater than 0, 2.5, 3.5, 4.5, respectively, where extreme cases are order-of-magnitude large quakes pre-defined by a picked threshold in this project according to the domain knowledge.

Seismometers record seismic events by calibrating the vibrations of waves. Magnitude in the dataset represents measured amplitude as a measured seismogram. While they are discrete data points, accumulating 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 defines as:

$$\mathbf{E} = (10^{\mathbf{Mag}})^{3/2} \tag{1}$$

in which the magnitude $0 \leq \mathbf{Mag} \in \mathbb{R} \leq 10$. Earthquake magnitude value can be even negative for tiny events that are negligible. This scale is also open-ended, but events with magnitude values greater than 10 are clipped to 10.

B. 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 natural to reorganize the vector-valued 1D time-series dataset into a 2D sequence dataset by dividing a map region into small boxes according to longitudes and latitudes and aggregating the released energy within a small box per specific time-frequency.

Long short-term memory (LSTM) [19] is an advanced model of recurrent neural networks suitable for modeling series-like vector-valued data. Compared with the exiting LSTM approach such as [20], location-aware weaving gives finer-grained geolocations. Besides, 2D convolutional operations can put a strong prior on locations than the recurrent matrix multiplication for vector-valued observations in LSTM cells. Furthermore, for earthquakes, the underlying intuition is that 2D convolution may capture location-based plate movements, which is considered as the direct cause of earthquakes.

We denote $\mathbf{Mag_k^t}$ the value at location k and time t of a spatially and temporally continuous phenomenon of interest. So each element of the sequence becomes a summation of all energy released at the grid (i, j), given $i \in [0, M)$ and $j \in [0, N)$. Then the total energy for each grid element is defined in Eq 2 , which means X^t has a shape $M \times N$ for M boxes along the latitude and N boxes along the longitude.

$$X_{i,j}^t = \sum_{k \in grid[i][j]} ((10^{\mathbf{Mag_k^t}})^{3/2}) \tag{2}$$

Considering this area as a 2D mesh grid, this equation sums up all energy erupted in every grid for each time interval. For example, we can sum up how much energy released within a box region of Longitude from -120 to -119 and Latitude from 32 to 33 everyday.

C. Convolutional 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 mainshocks, we first model the spatial patterns within the southern California area.

We use an autoencoder to mine the spatial pattern changes under normal circumstances and abnormal circumstances. As shown in Figure 2, the autoencoder consists of three major

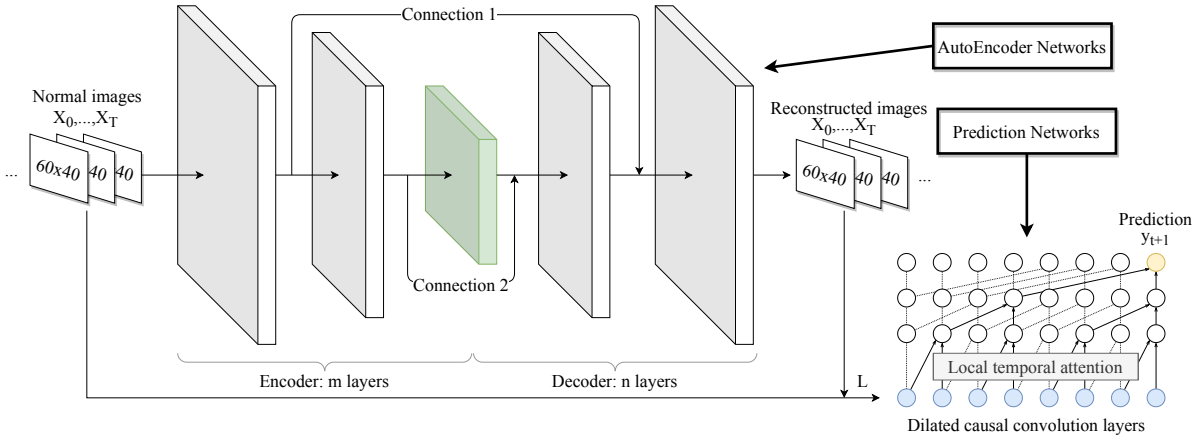


Fig. 2. EQPRED: overview of earthquake prediction networks.

components: 1) a bunch of convolutional layers encodes the input to 2) the bottleneck layer in green color, and 3) the layers in decoder up-sample the latent variables from the bottleneck layer to the output. Compared to the variational autoencoder (VAE), we do not assume Gaussian distribution or any other distributions for the latent space. In addition, the reconstructed results from VAE are tended to be noisier. We also make some experiments for complete comparison in Section IV. Spatial modeling is a semi-supervised process to train a model that learns the representation of earthquake images. We train this model by minimizing the following equation.

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

, 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. This setup enforces the same input and output so that the bottleneck layer can obtain the most critical latent variables from the dataset. Detailed modeling methods used in AE are covered in the following sub-sections.

1) Spatial modeling: The encoder networks are comprised of convolutional layers followed by Batch Normalization with the ReLU activation function. The decoder networks are comprised of de-convolutional layers with the ReLU activation function. After the seismic events are parsed and transformed to 2D image-like sequences in Section III-A, we can utilize the spatial dependencies between pixels. Convolutional operations are common image feature extraction means. Pixel relationships can be easily mapped to geology locations of events. The encoder component of this model is used to extract the spatial features from images which contribute to convolution layers in a downsampling manner. The downsampled feature maps enable the network to collect contextual information, which could be surface terrains on Earth. In convolution-base layers with a kernel K, the process takes the input X with the following form:

$$S_{i,j} = \sum_m \sum_n X(i+m, j+n)K(m,n) \qquad (4)$$

The decoder component of this model is used for upsampling variables from the bottleneck layer by inverse 2D convolutional operations. The final reconstructed image denoted as $\hat{X}$ is produced by the decoder.

2) Skip connections: We incorporate skip connections in the AutoEncoder architecture. Skip connections are forward shortcuts between layers in networks. Skip connections can help recover the full spatial resolution at the network output to avoid the gradient vanishing, making fully convolutional methods suitable for modeling segments on maps. They symmetrically connect layers from the encoder and decoder, as shown in Figure 2. 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]. The combination of low-level features and high-level features improves the training performance due to large gradients and improves accuracy due to complementary information summarized from different levels. Both long and short connections are used in the model as shown "Connection 1" and "Connection 2" in Figure 2.

The AE includes the bottleneck layer and other basic blocks. The design benefits of those are introduced in [22], [21]. For instance [23] includes short skip connections for detecting salient objects. Skipping some blocks with minimal modification encourages the information to pass through the non-linear functions to learn a residual representation from the direct input information. Similar to the short skip connections in Residual Networks [24], we sum the features from the encoder layers on the expanding path of decoder layers with long skip connections symmetrically.

3) Bottleneck layer: Hidden latent variables captured in the bottleneck layer have shown effectiveness in many applications, e.g. [3], [25]. The bottleneck layer in the autoencoder is deliberately set to a small vector of a size k feature map. This layer creates restrictions in the network to enforce the information pertaining to low dimensional space. Firstly, it regularizes the model from overfitting all samples. Secondly, a small feature map can better differentiate abnormal cases from normal cases. We set this k as a hyperparameter in our model.

D. Temporal Convolutional Model for Effective Temporal Modeling

In this work, the goal of nowcasting earthquakes is to predict the future probability of the next major shock in Southern California. This can be done in a prediction network, which is fed in the information gained from the AutoEncoder. Typically a long short-term memory (LSTM) model can predict well on this task. However, in EQPRED we incorporate an enhanced TCN (Figure 2), 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 [26]. We further analyze these features in the following sub-sections.

1) Conditional Temporal Convolution: Temporal convolutional 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 [9], [27]. A typical configuration of temporal convolutional layers is set the dilation rate corresponding to the i-th of layers, for example, 2^i.

$$p(y|\theta) = \prod_{t=1}^{T} p(y_{t+1}|y_1, \ldots, y_t, \theta) \tag{5}$$

2) Local Temporal Attention: A localized attention process to enhance temporal information passing is inspired by self-attention structure from Transformer [28], and Hao *et al.* work for sequence modeling [29]. 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) Smooth Joint Nash–Sutcliffe Efficiency: 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 [30]. 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.

$$NSE = 1 - \frac{\sum_{t=0}^{T}(\hat{y_t} - y_t)}{\sum_{t=0}^{T}(y_t - \bar{y})} \tag{6}$$

The goal of this metric is to force the predicted results to have a strong correlation between the distribution of predicted results and the distribution of expected results [31].

E. Joint Probability Analysis

Finally, we can combine the autoencoder and temporal convolutional networks together. After our model can learn the spatial dynamics via Eq 4 and temporal dynamics via Eq 5, the synthetic probability captured by the model can be defined as follows:

$$\begin{aligned} Loss_{AE} &= MAE(X - \hat{X}) \\ Loss_{TCN} &= NSE(y_t - \hat{y_t}) \end{aligned} \tag{7}$$

, which define the training targets.

$$\begin{aligned} y_t &= encode(X_t) \\ \hat{y_t} &= TCN(y_i, y_{i+1}, \ldots, y_{t-1}) \end{aligned} \tag{8}$$

, where $\hat{y_t}$ is directly related to the probability of the extreme events. This can be accessed by the defined threshold in AE.

IV. Experiments and Evaluation

The dataset is downloaded and parsed via the USGS website. It holds all earthquakes within the studied area in Southern California from 1990 to 2019. All the following experiments are conducted with the full dataset.

A. Implementation Details and Experimental Setup

Our EQPRED model and other baseline models are implemented with TensorFlow 2 [32] in Python. We conduct experiments on a computer equipped with an Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz, 128GB memory and 8 NVidia K80 GPUs. All models, including EQPRED and baseline models, are trained using Adam or SGD optimizers with a fine-tuned learning rate and mean squared error as 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.

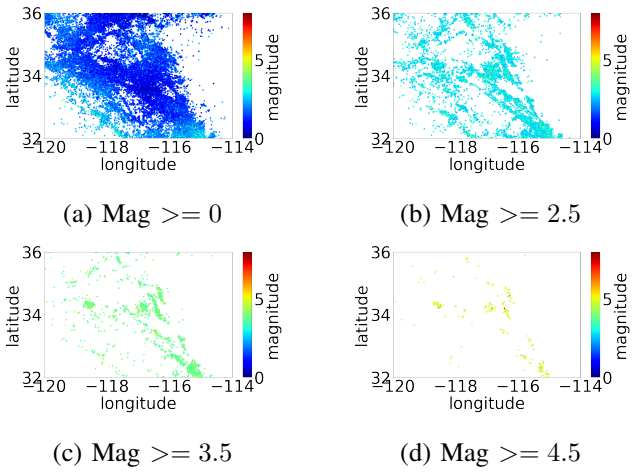
The encoder consists of 2D convolutional layers by varying the filter size from 4, 16, 32, 64. And the layers in the decoder vary the filter size symmetrically. The encoder and decoder are trained by minimizing the Mean Squared Error (MSE) loss between the input and its reconstruction. We use the Adam optimizer with a learning rate of 0.001 by default. We set an early stop in the training process when the validation loss has stopped improving for 20 epochs and the best model is restored from checkpoints. The training procedure iterates up to a maximum of 100 epochs. The batch size is set to 16, 64, and 128 respectively. In the temporal predictor, we test the time step size varying from 3 to 100 with appropriate filter sizes. The batch size is always set to one in order to make the model fully stateful. Due to the stochastic nature of shocks, the output series from the autoencoder is denoised by the LOESS smoothing method [33]. We describe two groups of modeling approaches below: models in the first group have only one neural network, while the modeling from the second group uses a joint of two neural networks. The modeling approaches compared in this project are listed as follows:

- ❶ **MLP** refers to a neural network model with only fully connected dense layers. In TensorFlow, these layers are implemented with the "keras.layers.Dense" class. The number of layers in the MLP model is set as a hyperparameter. The number of neurons in the input and output layer are set according to the dataset. The number of neurons in the hidden layers varies from 32, 64, to 128.

The following models labeled with 'MLP' also apply this strategy.

- ❷ **LSTM** refers to a typical model with LSTM layers, including necessary input and output conversion. The recurrent unit in LSTM layers varies from 32, 64, to 128. These layers are implemented by the "keras.layers.LSTM" class.
- ❸ **Conv2D-FC** is a composite model in which 2D convolutional layers are followed by fully connected dense layers. Conv2D is brought by the "keras.layers.Conv2D" class in TensorFlow, while the filter size varies from 3 to 40 and kernel size varies from 7x7, 5x5, to 3x3.
- ❹ **Conv2D-LSTM** consists of one 2D convolutional layer to encode the input and one LSTM to predict the output.
- ❺ **ConvLSTM2D-FC** represents a model consisting of one ConvLSTM2D layer followed by a fully connected dense layer as the output. We use the TensorFlow class "keras.layers.ConvLSTM2D" as the implementation of the ConvLSTM2D layer, the algorithm of which was introduced from [34]. The hyperparameters of ConvLSTM2D are similar to those used in Conv2D and LSTM. FC layer is set as above.
- ❻ **MLP+MLP** refers to joint training of two MLP models. The first MLP refereed as a dense AE consisting of 3 to 5 dense layers encoding and decoding the input. The second MLP represents a 3-layer fully connected neural network for prediction.
- ❼ **MLP+LSTM** refers to joint modeling with an MLP model and an LSTM model as mentioned above, where an MLP model acts as the AE and an LSTM model acts as the predictor.
- ❽ **MLP+Conv1D** is similar to the above case, except that one Conv1D-based model is used as the predictor, in which the filter size is set according to the task and the kernel size varies from 3 to 50 continuously.
- ❾ **Conv2D+MLP** combines a model with Conv2D layers as the AE and a simple MLP model as a predictor. Hyperparameters used in the two models are set similarly to those mentioned above.
- ❿ **Conv2D+LSTM** uses a Conv2D-based model and a LSTM-based model defined as above. Hyperparameters used in the experiments are also similar.
- ⓫ **Conv2D+Conv1D** combines a model with Conv2D layers as the AE and a Conv1D-based model as the predictor. Hyperparameters used in the experiments are also similar.

We train the models ❶ to ❺ with the Adam optimizer and the NSE loss. For the above methods ❻ to ⓫ and our EQPRED, the autoencoder is trained with the Adam optimizer and an MSE loss. The prediction network is trained with the Adam optimizer and an NSE loss. While many hyperparameters are required in all these experiments, we tend to choose the best set of parameters for each set of models and prevent them from overfitting.



(a) Mag $>= 0$ (b) Mag $>= 2.5$

(c) Mag $>= 3.5$ (d) Mag $>= 4.5$

Fig. 3. 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. We can observer the number of larger earthquakes is order of magnitude less than smaller ones.

B. Dataset Preprocessing and Augmentation

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 pre-process the events from the catalog by accumulating the events and parsed as spatial grids according to the analysis in Section III-A. We implement the data feeding pipeline with the TensorFlow data streaming APIs. The prepared dataset is loaded from disks with multiple workers and then fed in the training model from CPU to GPU batch by batch so that the pipeline can boost the I/O parallelism and increase the CPU and GPU efficiency.

Figure 3 shows four scatter plots by filtering events with magnitudes greater than or equal to 0, 2.5, 3.5, 4.5 in (a), (b), (c), and (d) respective sub-figures. 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 the 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.

C. Performance Analysis

In these sets 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.

1) AutoEncoder: We use Mean Absolute Error (MAE) as a metric to test the AE performance, the results of which are

TABLE I

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. PLEASE NOTE SOME MODELS DO NOT HAVE MEANINGFUL MAE VALUES, WHICH ARE OMITTED IN THE TABLE. PLEASE REFER TO SECTION IV-C FOR MORE DETAILS.

Models	MAE	Precision	Recall	F-1	F-0.2	NSE
❶ MLP	-	0.2631	0.2845	0.2096	0.2494	-1.4739
❷ LSTM	-	0.4596	0.5186	0.3801	0.4058	-0.2059
❸ Conv2D-FC	-	0.4589	0.3963	0.4340	0.4394	-0.1867
❹ Conv2D-LSTM	-	0.4299	0.4069	0.4217	0.4243	-0.4022
❺ ConvLSTM2D-FC	-	0.4633	0.3289	0.3763	0.3801	-0.1714
❻ MLP+MLP	0.2570	0.7525	0.6338	0.6652	0.7113	0.6778
❼ MLP+LSTM	0.1637	0.8420	0.7085	0.7599	0.8021	0.7890
❽ MLP+Conv1D	0.1484	0.8571	0.9351	0.8029	0.8342	0.8133
❾ Conv2D+MLP	0.1484	0.8577	0.7944	0.7887	0.8098	0.8108
❿ Conv2D+LSTM	0.1410	0.8640	0.8776	0.8609	0.8683	0.8222
⓫ 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 II

EQPRED AUTOENCODER VS. VAE.

Model	MSE	Accuracy	Variance
EQPRED	0.148	0.968	1.432
VAE [35]	0.157	0.971	1.986

TABLE III

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

listed in the MAE column from Table I. Models from ❶ to ❻ do not contain AE, so the results of them are omitted in the table. From this table, the MAE score of our EQPRED is 0.0483 and the lowest score for other models is 0.0588. EQPRED outperforms all other modeling approaches from ❻ to ⓫, whereases Conv2D-based autoencoders can generate competitive performance. From another aspect, this means the convolutional layers actually work for mining spatial patterns.

We also compare the autoencoder used in EQPRED in Section III as opposed to a variational autoencoder (VAE) separately. We test some metrics of using EQPRED autoencoder and a common VAE. The performance results are summarized in Table II, from which our EQPRED outperforms a single VAE. Even though VAE can achieve almost the same performance in terms of accuracy, it has a higher mean squared error loss and variance for the final output. Higher MAE loss and variance affect the performance of the prediction network. Besides the benefits from applying skip connections may outweigh other aspects for this case.

To show the critical hyperparameter that affects the overall performance, we list the results varying the latent space dimension from 16 to 1024, as shown in Table III. Larger latent space tends to have better fitting to the dataset, however this value should be regularized as small as possible.

2) Prediction: We use four metrics to compare all temporal predictors from ❶ to ⓫ with EQPRED: Precision, Recall, F-1, F-0.2, and NSE, where precision and recall are used to evaluate the capability of predicting positive events and F-scores give overview of accuracy. Even though basic models from ❶ to ❺ can predict some events. Their NSE scores are very small, which means those models are not reliable. Modeling ❻ to ⓫ can have competitive scores. ⓫ can have a better Recall value than EQPRED. In contrast to modeling approaches from ❶ to ⓫, the prediction network in EQPRED can outperform all these approaches overall. We summarize the experimental results in Table I.

3) 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 I. 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. Please note the classes are not balanced in this case, so that F-1 and F-0.2 scores may be higher than expected but we keep them as a reference here.

D. Ablation Studies

To verify the effectiveness of skip connections and local temporal attention applied in EQPRED, we test the models and compare the performance without these techniques. In the following two sets of experiments, we demonstrate the two primary 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 IV shows the results of these two sets of experiments.

TABLE IV
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 (with both)	**0.9251**	**0.9323**

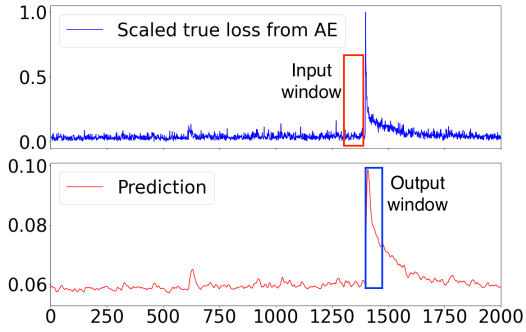


Fig. 4. EQPRED prediction.

E. Discussion and Empirical Study

We build joint models as shown in Figure 2, in which the autoencoder can learn the spatial pattern and the predictor can forecast future events. Figure 4 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 representing 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}.

This modeling approach and experimental results are still empirical. Throughout all experiments, according to Figure 3, we test pre-defined thresholds that filter extremely large quakes of magnitude at 4, 4.2, and 4.5. In all cases, this modeling approach can generalize to the same effective results. However, we have not expanded the dataset to cover a larger area in Southern California or a different area on the earth.

V. CONCLUSIONS AND FUTURE WORK

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. 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] A. E. Gaughan, F. R. Stevens, Z. Huang, J. J. Nieves, A. Sorichetta, S. Lai, X. Ye, C. Linard, G. M. Hornby, S. I. Hay, H. Yu, and A. J. Tatem, "Spatiotemporal patterns of population in mainland China, 1990 to 2010," *Scientific Data*, vol. 3, no. 1, p. 160005, Dec. 2016. [Online]. Available: http://www.nature.com/articles/sdata20165

[2] G. Christakos, *Modern Spatiotemporal Geostatistics*. Oxford University Press, Nov. 2000, google-Books-ID: 6CmJk7yPMbwC.

[3] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting," *arXiv:1707.01926 [cs, stat]*, Jul. 2017, arXiv: 1707.01926. [Online]. Available: http://arxiv.org/abs/1707.01926

[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)*, Nov. 2017, pp. 103–110, iSSN: 2375-9259.

[5] S. Wang, J. Cao, and P. Yu, "Deep Learning for Spatio-Temporal Data Mining: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020, conference Name: IEEE Transactions on Knowledge and Data Engineering.

[6] W. H. Bakun and C. M. Wentworth, "Estimating earthquake location and magnitude from seismic intensity data," *Bulletin of the Seismological Society of America*, vol. 87, no. 6, pp. 1502–1521, Dec. 1997, publisher: GeoScienceWorld. [Online]. Available: https://pubs.geoscienceworld.org/ssa/bssa/article/87/6/1502/120272/Estimating-earthquake-location-and-magnitude-from

[7] S. M. Mousavi, W. L. Ellsworth, W. Zhu, L. Y. Chuang, and G. C. Beroza, "Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking," *Nature Communications*, vol. 11, no. 1, p. 3952, Aug. 2020, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41467-020-17591-w

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: http://www.nature.com/articles/nature14539

[9] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *arXiv:1609.03499 [cs]*, Sep. 2016, arXiv: 1609.03499. [Online]. Available: http://arxiv.org/abs/1609.03499

[10] S. A. Pulinets, "Strong earthquake prediction possibility with the help of topside sounding from satellites," *Advances in Space Research*, vol. 21, no. 3, pp. 455–458, Jan. 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0273117797008806

[11] G. Guangmeng and Y. Jie, "Three attempts of earthquake prediction with satellite cloud images," *Natural Hazards and Earth System Sciences*, vol. 13, no. 1, pp. 91–95, Jan. 2013. [Online]. Available: https://nhess.copernicus.org/articles/13/91/2013/

[12] Z. Alizadeh Zakaria and F. Farnood Ahmadi, "Possibility of an earthquake prediction based on monitoring crustal deformation anomalies and thermal anomalies at the epicenter of earthquakes with oblique thrust faulting," *Acta Geophysica*, vol. 68, no. 1, pp. 51–73, Feb. 2020. [Online]. Available: https://doi.org/10.1007/s11600-019-00390-3

[13] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 2019, publisher: IEEE.

[14] Y. Matsubara, Y. Sakurai, W. G. van Panhuis, and C. Faloutsos, "FUNNEL: automatic mining of spatially coevolving epidemics," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '14. New York, NY, USA: Association for Computing Machinery, Aug. 2014, pp. 105–114. [Online]. Available: https://doi.org/10.1145/2623330.2623624

[15] W. Lotter, G. Kreiman, and D. Cox, "Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning," in *International Conference of Learning Representations (ICLR)*, Feb. 2017, arXiv: 1605.08104. [Online]. Available: http://arxiv.org/abs/1605.08104

[16] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised Learning for Physical Interaction through Video Prediction," *arXiv:1605.07157 [cs]*, Oct. 2016, arXiv: 1605.07157. [Online]. Available: http://arxiv.org/abs/1605.07157

[17] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series Extreme Event Forecasting with Neural Networks at Uber," p. 5.

[18] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3656–3663.

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

[20] Q. Wang, Y. Guo, L. Yu, and P. Li, "Earthquake Prediction Based on Spatio-Temporal Data Mining: An LSTM Network Approach," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 148–158, Jan. 2020, conference Name: IEEE Transactions on Emerging Topics in Computing.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *Computer Vision – ECCV 2016*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 630–645.

[22] ——, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778. [Online]. Available: http://ieeexplore.ieee.org/document/7780459/

[23] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply Supervised Salient Object Detection With Short Connections," 2017, pp. 3203–3212. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Hou_Deeply_Supervised_Salient_CVPR_2017_paper.html

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," 2015, pp. 3431–3440. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html

[25] L. Dong, Y. Gan, X. Mao, Y. Yang, and C. Shen, "Learning Deep Representations Using Convolutional Auto-Encoders with Symmetric Skip Connections," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 3006–3010, iSSN: 2379-190X.

[26] J. Yan, L. Mu, L. Wang, R. Ranjan, and A. Y. Zomaya, "Temporal Convolutional Networks for the Advance Prediction of ENSO," *Scientific Reports*, vol. 10, no. 1, p. 8055, May 2020, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41598-020-65070-5

[27] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional Time Series Forecasting with Convolutional Neural Networks," *arXiv:1703.04691 [stat]*, Sep. 2018, arXiv: 1703.04691. [Online]. Available: http://arxiv.org/abs/1703.04691

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, \. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[29] H. Hao, Y. Wang, Y. Xia, J. Zhao, and F. Shen, "Temporal Convolutional Attention-based Network For Sequence Modeling," *arXiv:2002.12530 [cs]*, Mar. 2020, arXiv: 2002.12530. [Online]. Available: http://arxiv.org/abs/2002.12530

[30] D. N. Moriasi, J. G. Arnold, M. W. Van Liew, R. L. Bingner, R. D. Harmel, and T. L. Veith, "Model evaluation guidelines for systematic quantification of accuracy in watershed simulations," *Transactions of the ASABE*, vol. 50, no. 3, pp. 885–900, 2007, publisher: American society of agricultural and biological engineers.

[31] R. H. McCuen, Z. Knight, and A. G. Cutter, "Evaluation of the Nash–Sutcliffe Efficiency Index," *Journal of Hydrologic Engineering*, vol. 11, no. 6, pp. 597–602, Nov. 2006, publisher: American Society of Civil Engineers. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%291084-0699%282006%2911%3A6%28597%29

[32] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A System for Large-Scale Machine Learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[33] J. Rojo, R. Rivero, J. Romero-Morte, F. Fernández-González, and R. Pérez-Badia, "Modeling pollen time series using seasonal-trend decomposition procedure based on LOESS smoothing," *International journal of biometeorology*, vol. 61, no. 2, pp. 335–348, 2017, publisher: Springer.

[34] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 802–810. [Online]. Available: http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcas pdf

[35] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114 [cs, stat]*, Dec. 2013, arXiv: 1312.6114. [Online]. Available: http://arxiv.org/abs/1312.6114