Can We Forecast And Detect Earthquakes From Heterogeneous Multivariate Time Series Data?

Luke Cullen[*][1] **Asadullah Galib**[*†][2] **Andrew Smith**[*][3] **Debvrat Varshney**[*][4]

Edward Brown[1] **Peter Chi**[5] **Xiangning Chu**[6] **Filip Svoboda**[1]

[1] University of Cambridge [2] Michigan State University
[3] Northumbria University [4] University of Maryland Baltimore County
[5] University of California Los Angeles [6] University of Colorado Boulder

[*]Equal contribution
[†]Correspondence to galibasa@msu.edu

Abstract

Earthquake forecasting is a topic of utmost societal importance, yet has represented one of the greatest challenges to date. Case studies from the past show that seismic activity may lead to changes in the local geomagnetic and ionospheric field, which may operate as potential precursors and postcursors to large-magnitude earthquakes. However, detailed and data-driven research has yet to support the existence of precursors and postcursors. This work makes an attempt to build data-driven deep learning networks that can learn the temporal changes in geophysical phenomena before and after large magnitude earthquake events. First, we do numerous experiments using various machine learning and deep learning models, but none of them are sufficiently generalizable to forecast earthquakes from potential precursors. Our negative findings may make sense as there is not any conclusive and comprehensive evidence yet supporting the existence of earthquake precursors. We, therefore consider detecting earthquakes from postcursors data to spot potential pitfalls and outline the scope of possibility. Our tests indicate that while detecting earthquakes from postcursor data might be promising, it would fall short. Poor performance could be brought on by a lack of data and extremely complex relationships. However, we are leaving room for future research with deeper networks and data augmentation.

1 Introduction

Forecasting earthquakes has been a longstanding and as yet unsuccessful scientific quest. Recent circumstantial evidence emerging from case studies suggests the occurrence of detectable precursors as well as postcursors in less obvious data sources, such as the variations in the geomagnetic field and ionosphere. While not unambiguously confirmed - or accepted by the community - if such signatures exist, then there is the exciting possibility of forecasting and detecting earthquakes using models that monitor the ionosphere and geomagnetic field. The goal of this study, therefore, is to test whether a model can be created to skillfully forecast and detect earthquakes based on the full range of data available. Such a model, if skillful, would be of immense societal value.

In this work, we perform time series modeling to capture the relationship between surrounding perturbations in the geomagnetic and ionospheric fields with the occurrence of earthquakes. As there

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

is no established scientific relationship between these physical phenomena and seismic activity, we take a data-driven approach to learn the effect of seismic activity on physical phenomena; observing the changes which can in turn help us forecast and detect earthquakes. This problem is further complicated by the overbearing effect of an extra-terrestrial phenomenon, principally the interaction of the solar wind with the ionosphere. We incorporate solar wind data as well in our networks to carry out multiple experiments with varying hyperparameters.

There are two aspects to our study. The heterogeneous multivariate time series data are first integrated and aligned. Next, we concentrate on the primary challenge of forecasting and detecting earthquakes. Our efforts with earthquake detection are somewhat more promising, despite the failure of earthquake forecasting models. For our deep networks to learn from, we think that larger datasets with longer time series would be helpful.

2 Problem Statement

Let the input time series of ionospheric data, solar wind data, geomagnetic data be $X^{tec}, X^{solar}, X^{geomag}$, respectively, and the target be y, which denotes whether an earthquake happens or not. The input time series can be multivariate. The spatial and temporal resolutions of the input time series are different from each other. Our first task is to integrate and align these large-scale, regionally, and temporally heterogeneous time series. After the alignment, the input and target variables for the forecasting problem can be defined as:

$$\begin{aligned} \mathrm{X}^{tec} &= [\, \mathrm{x}^{tec}_{t-k} \;\; \mathrm{x}^{tec}_{t-k+1} \ldots \mathrm{x}^{tec}_{t-2} \;\; \mathrm{x}^{tec}_{t-1} \,] \\ \mathrm{X}^{solar} &= [\, \mathrm{x}^{solar}_{t-k} \;\; \mathrm{x}^{solar}_{t-k+1} \ldots \mathrm{x}^{solar}_{t-2} \;\; \mathrm{x}^{solar}_{t-1} \,] \\ \mathrm{X}^{geomag} &= [\, \mathrm{x}^{geomag}_{t-k} \;\; \mathrm{x}^{geomag}_{t-k+1} \ldots \mathrm{x}^{geomag}_{t-2} \;\; \mathrm{x}^{geomag}_{t-1} \,] \\ \mathrm{y} &= \begin{cases} 1, & \text{if earthquake happens at time } t \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

Then, with the assumption that earthquakes have nonlinear relationships with the input features, we can model the following:

$$\begin{aligned} \mathrm{y} = \lfloor \sigma \; (\textit{nonlinear}(\mathrm{W}^{tec}(\mathrm{X}^{tec})^T) + \textit{nonlinear}(\mathrm{W}^{solar}(\mathrm{X}^{solar})^T) + \\ \textit{nonlinear}(\mathrm{W}^{geomag}(\mathrm{X}^{geomag})^T)) \rceil \end{aligned} \tag{2}$$

Our final task is to estimate y using the input variables ($X^{tec}, X^{solar}, X^{geomag}$).

Later, we apply the same problem formulation and underlying assumptions to the detection of earthquakes, but the time frame of the input variables is derived from subsequent time steps:

$$\begin{aligned} \mathrm{X}^{tec} &= [\, \mathrm{x}^{tec}_{t+1} \;\; \mathrm{x}^{tec}_{t+2} \ldots \mathrm{x}^{tec}_{t+k-1} \;\; \mathrm{x}^{tec}_{t+k} \,] \\ \mathrm{X}^{solar} &= [\, \mathrm{x}^{solar}_{t+1} \;\; \mathrm{x}^{solar}_{t+2} \ldots \mathrm{x}^{solar}_{t+k-1} \;\; \mathrm{x}^{solar}_{t+k} \,] \\ \mathrm{X}^{geomag} &= [\, \mathrm{x}^{geomag}_{t+1} \;\; \mathrm{x}^{geomag}_{t+2} \ldots \mathrm{x}^{geomag}_{t+k} \;\; \mathrm{x}^{geomag}_{t+k} \,] \end{aligned} \tag{3}$$

So, we use input variables from pre-earthquake time steps $(t-1, \cdots, t-k)$ to forecast earthquakes at time t and we use input variables from post-earthquake time steps $(t+1, \cdots, t+k)$ to detect earthquakes at time t.

3 Methodology

This section presents the methodology for forecasting and detecting earthquakes from heterogeneous multivariate time series data. Our method can be divided into two parts. First, we integrate and align the heterogeneous time series data. Then, we focus on the main task of forecasting or detecting earthquakes. The underlying architectures for earthquakes detection and forecasting are the same, the only difference is the time frame of the input data: pre-earthquake time steps $(t-1, \cdots t-k)$ to forecast earthquakes and post-earthquake time steps $(t+1, \cdots t+k)$ to detect earthquakes.

Integrating and Aligning Heterogeneous Time Series Data

We are considering seismic ([1]), geomagnetic (SUPERMAG [2]), solar wind (OMNI [3]), and ionospheric (TEC [4]) time series data as input variables. The spatial and temporal resolutions of the data sets are different from each other. For instance, TEC data has a 5-minute interval compared to a 1-minute interval for solar wind data. The spatial resolution of TEC data is also considerably larger than that of geomagnetic data. Another consideration is the volume of data available including over 110 million data points of TEC data per month. Integrating these large-scale, regionally and temporally heterogeneous datasets is the first significant challenge. Each dataset comprises time series from individual stations that require prepossessing to compensate for missing values.

We analyze the heterogeneous time series data based on their relative time to earthquakes, relative distance to earthquakes, and different interpolation considerations. We experiment with these three major parameters to integrate and align the input time series. The parameters that we change for the input data are as follows:

- Relative Time Bounds (τ) - We look at physical phenomena (geomagnetic and ionospheric readings) that occur in a time window of ±1 hour to ±10 hours around the earthquake occurrence time.
- Spatial Bounds (δ) - The spatial grid around the earthquake epicenter where we are looking for data. This parameter is varied between [3°, 4°, 5°] (i.e. $\delta \in [3, 5]$). So, in essence, for an earthquake epicenter at (x,y), we collate all data sources present at locations (x±δ, y±δ). The data sources are grouped together as per the following parameter.
- Group Type (g) - The input data sources which are filtered as above (with respect to τ and δ) are then grouped together to a single reading with the τ time series. We perform two types of these group-by operations: “median” and “range”.

After aligning the datasets we have machine learning-ready data that can be fed to any deep or traditional network. Varying the parameters of τ, δ, and g, gives us a lot of room to learn the spatio-temporal dependencies between seismic activity and surrounding geophysical phenomena.

3.1 Forecasting and Detecting Earthquakes

The goal of this research is to develop data-driven deep learning networks that can recognize the temporal variations in geophysical phenomena before and after earthquakes of significant magnitude. To begin, we conduct a number of experiments utilizing various machine learning and deep learning models to forecast earthquakes from potential precursors. Later, to identify potential pitfalls and define the range of possibilities, we explore detecting earthquakes from postcursor data.

We try to learn the relationship between a big seismic event and the preceding (for forecasting) and succeeding (for detection) ionospheric, geomagnetic, and solar wind changes. As this relationship may be complex and non-linear, deep learning methods can help to learn the relationship. But we do not have enough data points due to the scarcity of earthquakes with a magnitude greater than 6. So, we experiment with both conventional machine learning models as well as deep learning models. In particular, we use Random Forest and AdaBoost as conventional machine learning models. In terms of deep learning models, we employ Fully Connected Networks and Long Short Term Memory (LSTM) [5] to learn the long time series. LSTMs, which are an upgraded form of recurrent neural networks (RNNs), are deep learning models with hidden states. These hidden states can temporarily store representations from a past event and use them to predict a future event.

For the conventional machine learning models (Random Forest and AdaBoost) and fully connected networks, we flatten the multivariate time series before feeding it into the model. We consider the following two data settings for each of the models: (ionospheric + solar wind data) and (ionospheric + solar wind + geomagnetic data).

For LSTMs, we primarily experiment with three architectures based on the data considered: Network 1 (ionospheric + solar wind data), Network 2 (ionospheric + solar wind + geomagnetic data), Network 3 (ionospheric + solar wind + geomagnetic data).

The basic idea behind these architectures is that seismic activity can result in changes in the physical phenomena of the ionosphere and the geomagnetic field. The ionosphere, on the other hand, can

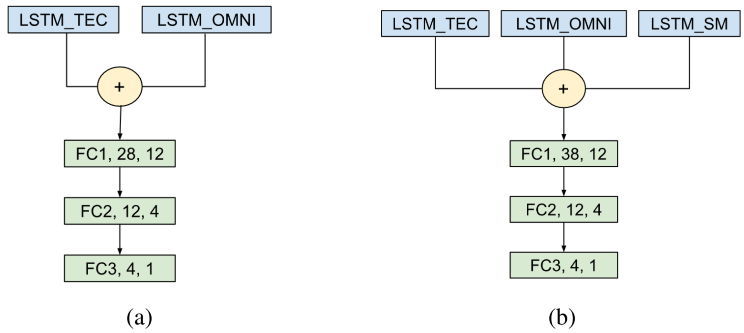

Figure 1: (a) Network 1: Built upon ionospheric and solar wind data (b) Network 2: Built upon all three data sources i.e. (TEC, OMNI, & SuperMAG)

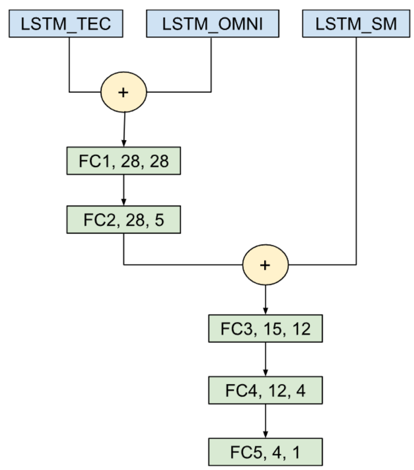

Figure 2: Network 3 - A modification on Network 2, where features from TEC and OMNI learn from each other.

further be affected by the solar wind emanating from the sun. To capture the dynamic changes in these physical phenomena, our deep learning models consist of an LSTM cell for each phenomenon. The models are made up of multiple linear layers, with a fully connected layer as the final layer to predict the magnitude of seismic activity.

In the Figure 1 and Figure 2, we show all the network architectures (Networks 1-3). Figure 1 shows the architectures of Network 1 and Network 2. Both networks are primarily made up of LSTM cells. Here, "FC" stands for a fully connected layer and the two numbers denote input and output nodes. Figure 2 shows the architecture of Network 3, which is a modification of Network 2 where hidden representations from TEC and OMNI data are concatenated first, and the output of which is later concatenated with the hidden representation of SuperMAG data. The reason is that TEC is heavily affected by the OMNI dataset, and a separate concatenation of these two LSTM cells should be able to capture and learn this relationship.

All networks have a ReLU activation [6] after each layer and a sigmoid layer at the end of the architecture as we are detecting earthquakes. They are trained using the Adam optimizer [7] with a binary cross entropy loss. All architectures use a batch size of 8 and are trained for 200 epochs.

4 Experimental Evaluation

This section presents data, experimental setup, and results analysis.

4.1 Data

Synthetic Data

The relationship between seismic events and the preceding (for forecasting) and succeeding (for detection) ionospheric, geomagnetic, and solar wind changes can be very complex and non-linear. Also, we have limited data due to the frequency of earthquakes with a magnitude greater than 6. So, before working on the real-world data, we created a synthetic dataset to assess our model's ability to correctly forecast and detect earthquakes. It is typical to evaluate methods using synthetic data in this way [8, 9]. For the synthetic data, the input time series is generated as follows:

$$\mathrm{X}^{tec}, \mathrm{X}^{solar}, \mathrm{X}^{geomag} \sim random \tag{4}$$

$$\mathrm{W}^{tec}, \mathrm{W}^{solar}, \mathrm{W}^{geomag} \sim uniform \tag{5}$$

$$\tag{6}$$

Then, we calculate the target - y using the initial assumption as mentioned in Equation (2).

Real World Data

To forecast and detect earthquakes, we use the USGS earthquake catalog [1]. This catalog records critical features such as the timing, magnitude, location, and depth of earthquakes, and has been produced for a long time interval - albeit with increasing sensitivity as more stations and data processing methods have come online.

Three open source data sets are used to search for physical changes linked to earthquake precursors. First, to investigate ionospheric signals we use measurements of the total electron content (TEC) which describes the state of the ionosphere. The Madrigal database [4] provides global maps of vertical TEC, calculated from GNSS (Global Navigation Satellite Systems) data, and provided in a 1° by 1° grid at a cadence of 5 minutes. Data is incomplete, with typical global completeness of 25 [10]. Second, to probe seismic effects on the geomagnetic field, the SuperMAG collection of individual stations data is used [2]. Although 1-second resolution data is available in some instances, the 1-minute resolution is used to vastly increase spatial and temporal coverage. Third, to explicitly calibrate for the effects of space weather on the ionosphere and geomagnetic field, we use a series of heliospheric data known as OMNI [3], at 1-minute resolution. OMNI collates a range of heliospheric observations and derivatives describing the geomagnetic conditions near the Sun's surface and in near-Earth space.

4.2 Experimental Setup

We have tested conventional machine learning models (Random Forest and AdaBoost) and other deep learning models (Fully Connected Networks, LSTMs) for their capability of learning non-linear and complex relationships. For LSTM models, we have tested using three customized architectures as mentioned before. For all experiments, we use 8:1:1 for train, validation, test split. We performed hyper-parameter tuning on all the models and on the parameter space for large-scale data integration. We tune different learning rates in the range $[10^{-2}, 10^{-5}]$, batch size, model-specific hyperparameters, and data-specific parameters. Accuracy, precision, recall, and F-1 score are considered for model performance. Google Cloud high memory CPUs and GPUs were used for training.

4.3 Results Analysis

According to experiments for forecasting earthquakes, none of the models converge. The models/architectures themselves are not an issue since the models converged perfectly and performed excellently when tested on the synthetic data.

For the earthquake detection part, the LSTM models did not converge easily on the real dataset. Since the models converged smoothly and performed well when tested on the synthetic data (See Figure **??**), the models themselves are not a concern. The lack of seismic data (i.e., insufficient training events) would be the main factor contributing to the failure of the models on real-world data. However, conventional machine learning models such as Random Forest and AdaBoost converge on real data, as do other deep learning models (fully connected networks). Models' performance is better than random chance, but not exceptional (see Table 1). We also test our LSTM models on synthetic data to get an **accuracy of 0.963**, which suggests that either the signals are not uniquely identifiable in the data, or that there is insufficient data for the task.

5 Discussion and Limitation

All of the models employed in this study to forecast earthquakes from precursory signals fail on real-world data. Our negative findings on real data may make sense as there is not any comprehensive and convincing evidence to date in favor of earthquake precursors. There is circumstantial evidence from previous studies has indicated the presence of precursory changes in the ionosphere and geomagnetic field linked to individual earthquakes [11, 12, 13]. But, those studies are specific cases rather than comprehensive and are not unambiguously confirmed or accepted by the community. Also, a major limitation of case studies is the major contribution of external factors, primarily space weather, to fluctuations in ionospheric and geomagnetic properties, which can lead to misinterpretation of external signals as earthquake-related [14, 15]. Using all available events, our study is the first to employ a data-driven methodology that can account for all external influences, allowing us to demonstrate consistency within earthquake precursor signatures and present a comprehensive outlook.

Given the scarcity of real-world data and the complexity of the underlying relationships, the performance of the models for earthquake detection from postcursors data is fair. Our findings show that while trying to detect earthquakes using postcursor data may seem intriguing, it is not that convincing. It is possible that a lack of data and incredibly intricate relationships are to blame for poor performance. We are, however, leaving space for additional data augmentation and deeper network analysis in the future.

There may be problems in this data assimilation process, despite the fact that we integrate and align heterogeneous multivariate time series data based on recommendations and insights from literature and domain experts. Additionally, we did not take into account alternative deep learning models, such as convolutional neural networks, graph neural networks, etc. We ought to have tried out deeper and more complicated designs. We should have experimented with more complex and deeper architectures.

	M>7			M>6		
Model	Accuracy	Recall	F1	Accuracy	Recall	F1
Random Forest	0.66	0.83	0.71	0.55	0.61	0.57
AdaBoost	0.64	0.73	0.66	0.54	0.58	0.56
Fully Connected	0.63	0.71	0.65	0.59	0.63	0.60
LSTM (Network 1)	0.57	0.64	0.60	0.54	0.59	0.55

Table 1: Model performance on real-world data for earthquake detection. 'M' denotes earthquake magnitude.

6 Conclusion

Earthquakes can be hard to forecast and detect due to the unknown relationships between the associated geophysical phenomena. This work explores the power of machine learning algorithms in forecasting and detecting earthquake occurrence patterns. The networks capture different intuitive approaches and are able to learn randomly generated dummy data. Though earthquake forecasting models fail, our experiments with earthquake detection are somewhat more promising. We believe that bigger datasets with a longer time series might be useful for our deep networks to learn from.

Acknowledgments

This work has been enabled by the Frontier Development Lab (FDL) USA program. FDL USA is a collaboration between SETI Institute and Trillium Technologies Inc., in partnership with NASA, DOE, Google Cloud, Intel, IBM, Lockheed Martin, and NVIDIA. We are grateful to NASA's support for this work.

References

[1] USGS, "Search earthquake catalog," *Earthquake Hazards Program*, 2022.

[2] J. W. Gjerloev, "Supermag: Global magnetic field observations," 2022.

[3] NASA, "Omniweb," 2022.

[4] CEDAR, "Madrigal database," 2022.

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

[6] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological cybernetics*, vol. 20, no. 3, pp. 121–136, 1975.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[8] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowledge and information systems*, vol. 34, no. 3, pp. 483–519, 2013.

[9] A. H. Galib, A. McDonald, T. Wilson, L. Luo, and P. Tan, "Deepextrema: A deep learning approach for forecasting block maxima in time series data," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022* (L. D. Raedt, ed.), pp. 2980–2986, ijcai.org, 2022.

[10] H. Sun, Z. Hua, J. Ren, S. Zou, Y. Sun, and Y. Chen, "Matrix completion methods for the total electron content video reconstruction," *The Annals of Applied Statistics*, vol. 16, no. 3, pp. 1333–1358, 2022.

[11] M. Kamogawa and Y. Kakinami, "Is an ionospheric electron enhancement preceding the 2011 tohoku-oki earthquake a precursor?," *Journal of Geophysical Research: Space Physics*, vol. 118, no. 4, pp. 1751–1754, 2013.

[12] P. Xiong, C. Long, H. Zhou, R. Battiston, A. De Santis, D. Ouzounov, X. Zhang, and X. Shen, "Pre-earthquake ionospheric perturbation identification using cses data via transfer learning," *Frontiers in Environmental Science*, p. 514, 2021.

[13] L. Conti, P. Picozza, and A. Sotgiu, "A critical review of ground based observations of earthquake precursors," *Frontiers in Earth Science*, vol. 9, p. 676766, 2021.

[14] S. Pulinets and K. Boyarchuk, *Ionospheric precursors of earthquakes*. Springer Science & Business Media, 2004.

[15] I. A. Moldovan, A. O. Placinta, A. P. Constantin, A. S. Moldovan, and C. Ionescu, "Correlation of geomagnetic anomalies recorded at muntele rosu seismic observatory (romania) with earthquake occurrence and solar magnetic storms," *Annals of Geophysics*, vol. 55, no. 1, 2012.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 5 for limitations.
   (c) Did you discuss any potential negative societal impacts of your work? [No] No negative societal impact were apparent from improving earthquake forecasting and detection potential.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Citations for data and instructions needed to reproduce the main experimental results are included in section 3 and 4.1.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Google Cloud high memory CPUs and GPUs were used for processing.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Data sources are cited in section 4.1.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Open-source stated in section 4.1.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Impersonal geomagnetic data is being used.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]