Imperial College London

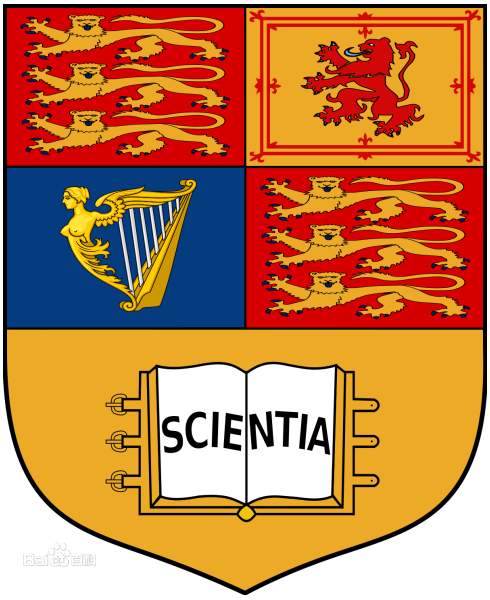
Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Forecasting Induced Seismicity in Oklahoma

Author:
Zhiyong Liu
zl1220@ic.ac.uk
GitHub: acse-liuzyon

Supervisor:
Dr. Stephen P. Hicks
s.hicks@imperial.ac.uk



August 2021

GitHub repository: https://github.com/acse-2020/acse2020-acse9-finalreport-liuzyon

Contents

Abstract

Human activities can cause minor earthquakes, which are called induced earthquakes and most have a low magnitude. In this project, a unique and rich dataset of industrial activities from regions combined with geological formation in Oklahoma are used and correlations between these factors and seismicity are confirmed. With existing high-quality earthquake catalogues in these regions, statistically significant factors were selected by stepwise regression approach for models input. A logistic regression model and a neural network model are generated to retrospectively forecast the seismicity. The performances about them are compared in various measures which denote neural network model has a better predicted ability and can be used in practice. The depth to basement is founded mostly contributed to the seismicity.

Key Words: induced earthquakes, forecasting model, main factors for seismicity, machine learning methods.

1 Introduction

It is well known that humans can cause earthquakes through fluid injection and extraction. (Ellsworth 2013) stated the understanding of the causes and mechanics of human-induced earthquakes. It includes wastewater injection, emerging oil and gas recovery technologies, and other indirect induced activities. Such cases of induced seismicity have been recorded and proven in Oklahoma, where seismicity has been increased dramatically since 2010. In cases like Oklahoma, because the rate of earthquakes was very low before high-rate wastewater injection, it is relatively easy to determine the fundamental causes of induced seismicity. However, in tectonically-active areas, it is more challenging to distinguish between natural and triggered seismicity. In tectonically active regions in the US, such as California, although oil and gas extraction has taken place for many decades, only a handful of studies have been published with a focus on these areas (Hough et al. 2017). Large-scale big-data and statistical approaches is one method to predict the locations where human triggered seismicity may be expected (Hincks et al. 2018).

To move onto an area with complex tectonic factors like California, we will use a less complicated testing environment to develop our big-data statistical approaches, like Oklahoma. In this project, several unique and rich datasets of candidate factors from regions in Oklahoma are used to statistically evaluate and retrospectively forecast possible signatures of human-induced seismicity from existing high-quality earthquake in these regions. These datasets contain geological formation, injections, hydraulic fracturing activities and wells, each of which have millions of items. In previous studies about induced earthquakes in Oklahoma, for instance, (Norbeck & Rubinstein 2018) associated the past injection trends with the seismicity patterns based on fluid flow and seismic physics to generate physical prediction models. This project originally designed and implemented a feedforward neural network (FNN or NN) for induced seismicity forecasting, which effectively improves the performance of model fitting and prediction compared with logistic regression model (LR) also common-used in previous studies. The superiority of our NN model was proven and their results about major factors to seismicity by two models were compared.

The detail introduction of geological visualization, feature selection and NN model will be delivered in Section 2. Section 3 will describe metadata of code, including data files, code repository and dependencies. Implementation about this project will be demonstrated in Section 4. The results of models (LR and NN) and comparison as well as feature attribution will be stated in Section 5. Finally, a conclusion of the project and future works will be presented in Section 6.

2 Software Description

2.1 Feature Selection: Stepwise Regression

Stepwise Regression method is widely used to find which factors are important and which are not. P-value is often used in stepwise regression to see if the patterns they measured were statistically significant. If the p-value of a statistical test is small enough, we are able to reject the null hypothesis of the test and the pattern measured is statistically significant. The most common threshold for p is 0.05 (Ioannidis 2018). For the variable whose p-value is greater than 0.05, we can assume that it is not statistically significant as shown in Appendix Fig 11.

In the construction of regression model, there usually exists a problem of high multicollinearity. A little multicollinearity is not necessarily a serious problem that can be tolerated, but severe multicollinearity is a problem. Because it increases the variance of the regression coefficients which makes them unstable and difficult to interpret themselves. A VIF between 5 and 10 means problematic and if the VIF is larger than 10, it is likely to poorly estimated the regression coefficients.

Considering both p-value and VIF, we used a forward stepwise logistic regression combined with VIF checking algorithm introduced in Algorithm 1.

Algorithm 1 Stepwise logistic regression with VIF checking

procedure SLR($candidate\_features$)
  $P\_THRESHOLD \leftarrow 0.05$
  $VIF\_THRESHOLD \leftarrow 5$
  $X \leftarrow candidate\_features$
  $result \leftarrow [\,]$
  for $feature$ `in` X **do**
    $result.append(feature)$
    generate model with $result$
    while there exist some feature N which $N.p\_value$ is greater than $P\_THRESHOLD$ **do**
      eliminate feature with max p_value in $result$
      generate model with $result$
    end while
    while there exist some feature N which $N.VIF$ is greater than $VIF\_THRESHOLD$ **do**
      eliminate feature with max VIF in $result$
      generate model with $result$
    end while
  end for
end procedure

2.2 Neural Network Construction: PyTorch

PyTorch (PyTorch 2021) is a machine learning library for neural network in Python. It is open-source used for applications in natural language processing (NLP) and computer vision. To define a neural network in PyTorch, it is pretty convenient to create a class that inherits from *nn.Modules* within several rows of code. In the created class, user can define the layers of the network in the *__init__* function and indicates how data will pass through the network in the *forward* function.

2.3 Feature Attribution: Captum

Feature attribution is the technique to investigate how much each feature in the model contributes to the prediction for given dataset fitting. Because models in machine learning are more complicated and lack of transparency, the interpretability approaches for these models have becoming increasingly important. Captum is an interpretability and understanding library for models in PyTorch. It provides a lot of advanced algorithms, including Integrated gradients (Sundararajan et al. 2017), DeepLIFT (Shrikumar et al. 2017), Feature Ablation and Gradient SHAP (Lundberg & Lee 2017) used in this project. These Captum provide users a convenient way to know about the contributions of features to a models' output through these algorithms.

2.4 Model Evaluation Measures

The measures we used for model evaluation will be introduced in this section. These measures contain accuracy, confusion matrix, precision, recall, F1 score, ROC curve and AUC value.

2.4.1 Accuracy, Precision, Recall and F1 Score

In confusion matrix, it has four cells which represents TN, FN, FP, TP. The meanings of these four types are listed below:

- TN: Prediction is 0 and true is 0.
- FN: Prediction 0 but true is 1.
- FP: Prediction is 1 but true is 0.
- TP: Prediction is 1 and true is 1.

The accuracy, precision, recall, F1 score are calculated based on the four types in confusion matrix. According to the confusion matrix, they can be calculating by Equation (1)(2)(3)(4).

$$Accuracy ::= \frac{TN + TP}{TN + FN + FP + TP} \tag{1}$$

$$Precision ::= \frac{TP}{TP + FP} \tag{2}$$

$$Recall ::= \frac{TP}{TP + FN} \tag{3}$$

$$F1 ::= 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

The accuracy is a common measure in model evaluation, and it is an intuitive representation of whether the model is capable of making accurate predictions. However, it is easily affected by the imbalance of the samples. For example, there are a total of 100 samples and 90 of them are positive samples. If the model predicts all samples to positive sample, the accuracy can reach up to 90%, but such a model is meaningless obviously.

Precision states from the perspective of prediction results, it describes how many of the positive predictions predicted by the binary classifier are accurate. Recall states from the perspective of real results, it describes how many real positive samples are recalled by the binary classifier. Precision and Recall are contradictory with each other. Generally, recall tends to be low when precision is high. For earthquake prediction, we focus more on seismic samples rather non seismic samples. Therefore, we have two expectations:

- we want a high precision because it means more predictions in seismic predictions are correct.
- we want a high recall because it means more actual seismic samples are selected out by the model.

  Note: Seismic prediction here refers to the prediction whose result is a seismic area.

In order to take these two expectations into consideration, we introduced F1 score, which is the harmonic mean (Chhikara 1988) of precision and recall. If the F1 score is high, it means model highly meet both expectations.

2.4.2 ROC Curve and AUC Value

$$FPR ::= \frac{FP}{N} = \frac{FP}{FP + TN} \tag{5}$$

$$TPR ::= \frac{TP}{P} = \frac{TP}{TP + FN} \tag{6}$$

We also used the Receiver Operating Characteristic Curve (ROC) to evaluate LR model and NN model. The ROC curve is drawn by a series of different dichotomies (dividing values or determining thresholds). The abscissa represents the false positive rate FPR (1-specificity) shown in Equation (5) and the ordinate represents the true positive rate TPR (sensitivity) shown in Equation (6).

ROC has the superiority of independence of class distribution, which is suitable for evaluation with unbalanced datasets. We can see why it has this advantage from their equations. TRP used TP and FN, which both belongs to second row in confusion matrix. It only concentrates on positive samples (P, which was represented by the second row in confusion matrix). FRP used FP and TN, which both belongs to first row in confusion matrix. It only concentrates on negative samples (N, which was represented by the first row in confusion matrix). Therefore, even if the number of P or N changes, they do not affect each other. That is to say, even if the ratio of positive to negative samples changes greatly, the ROC curve does not change greatly. However, this is not a feature of the previous measures in Table 1. The performance of the classification model can also be represented intuitively by the area under the ROC curve (AUC). AUC ranges in value from 0 to 1. A model has an AUC of 0.0 if its predictions are 100% wrong; In reverse, if it is 100% accurate, it has an AUC of 1.0.

3 Code Metadata

This project was built under macOS Catalina environment with Jupyter Notebook 6.0.3 and Python 3.7.6. The information of libraries and their usages are listed in Table 2.

In the GitHub repository, source code files (*.ipynb*) are stored in the *src* directory, including two main parts, which are feature extraction and prediction model. An additional file *visualization.ipynb* is used for some figures in this report. The dataset files (*.csv*) generated from feature extraction part are located at *data* directory. The models (*.pt*) generated from prediction part are located at the *model* directory.

4 Implementation

4.1 Data Processing and Spatial visualization

In the initial phase of project, we imported all the available datasets (earthquake data, geological data, hydraulic fracturing data, injection data, well data). The metadata of earthquake dataset was exported from *Oklahoma Geological Survey Earthquake Catalog Download Tool* (Walter et al. 2020). The metadata of injection dataset was exported from *Oklahoma Corporation Commission* (OklahomaCorporationCommission 2021). The metadata of hydraulic fracturing dataset was exported from *FracFocus* (Registry 2021).

We then preprocessed the imported datasets, which includes missing values processing and data selecting by date. The date of injection dataset we imported was in the range of 2011 to 2020. Because the data files >2019 supported by the *Oklahoma Corporation Commission* are still under

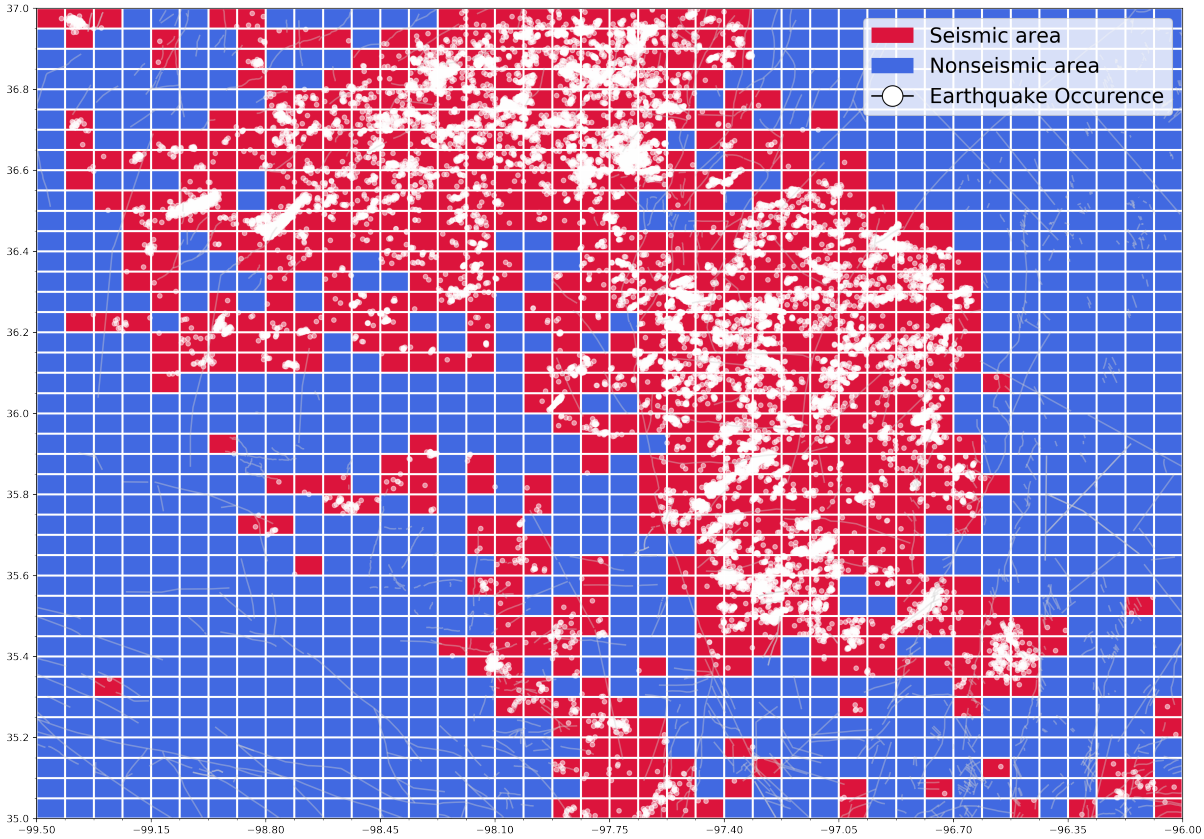


Figure 1: Target variable of earthquake occurrence within the interest area in Oklahoma (The white lines display the gridding regions, with a spacing in the x direction of 8 km and the y direction of 5.5 km. The white points denote the earthquake occurrences. The red region denotes it has seismicity in this region and blue region denotes it does not have seismicity in this region).

review and corrections are being made as warranted, those files are currently incomplete and subject to change. Therefore, we only used the data between 2011 and 2018. Due to date limitation of injection data, we set the interest time in the period of 2011 to 2018 for all datasets. We set the interest area to the area whose longitude ranges in (-99.5W, -96W) and latitude ranges in (35.0N, 37.0N) shown as red box in Figure 10, which is an available spatial scope provided by all datasets. We visualized all the data by *matplotlib* and *geopandas* module in the interest area and date. The geodetic coordinate system we used for mapping in this project is 'EPSG:4326' (Howard et al. 2007), which is the most popular coordinate system. The geological formation (depth to basement) along with various activities (seismicity, injections, wells, hydraulic fracturing) are shown in Fig 4.

4.2 Spatial Griding and Features Extraction

For the interest area (longitude ranges in (-99.5W, -96W) and latitude ranges in (35.0N, 37.0N)), we divided it into 40×40 grids. The size (40×40) of griding depended on the adequacy of generated dataset and the suitable size of the unit region. In this case, it generated 1600 data items for model construction and each of them has the distance of 8km in x-direction and 6 km in y-direction.

Then, we counted the values of all features and target for each grid by the various data from Section 4.1 to generate the dataset. This dataset files were generated in *.csv* and saved in *data* directory. The features and target of this dataset is shown in Table 3.

After this statistical process, we obtained a dataset in terms of spatial distribution. Each of item

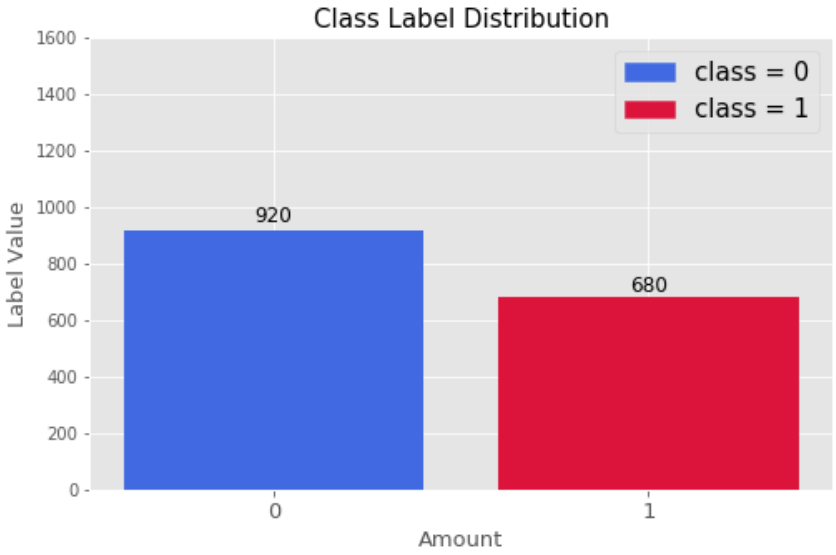


Figure 2: Class label distribution in whole dataset (0: nonseismic region; 1: seismic region).

represents a subarea of our interest area and contains the values of various candidate features. The target variable is set by seismicity, which represents the seismicity in spatial distribution. We made each grid region a value of 1 if there was at least one (magnitude>2) earthquake occurred during 2011 to 2018. Otherwise, if no earthquake occurred in this grid region, we assigned it a value of 0. The griding result of the earthquake occurrence (target) from 2011 to 2018 is shown in Figure 1. The class distribution with whole dataset is shown in Figure 2. There are 920 nonseismic grid regions with label '0' and 680 seismic grid regions with label '1'.

4.3 Stepwise Feature Selection Approach

As some activities were strongly overlapped in spatial distribution, they might have a high degree of multicollinearity. We calculated the feature similarity on all features based on Pearson (Benesty et al. 2009) correlation coefficients and generated a heatmap as Appendix Figure 12. In general, we consider a high degree of collinearity between two features with a coefficient greater than 0.9, like 'active_well_number' and 'well_under_basement_number'. To select statistically significant features and deal with high multicollinearity problem, we used the forward stepwise approach with VIF checking algorithm introduced in Section 2.1. After this process, we eliminated statistically insignificant features and also solved the problem of high multicollinearity between features.

4.4 Logistic Regression Model

With the selected features in Section 4.3, we constructed a multiple logistic regression model in this part, which used the features selected as independent variables and earthquake occurrence as the target variable. We standardized and normalized all input features by z-score normalization (Patro & Sahu 2015) to reduce data skewness and bring all features on the same scale. The normalization makes it feasible to interpret the relative differences of LR model coefficients.

We split the dataset into train dataset and test dataset with the ratio of 4:1, which respectively have 1280 and 320 samples (Each sample is associated with one gridding region in the interest area). In the build process of LR model, we used *sklearn.linear_model.LogisticRegression* to construct and fitted our LR model with training data by *fit* function.

4.5 Neural Network Model

4.5.1 Architecture

In this project, we used a neural network model which has its input layer with 7 neurons, two hidden layers of 8 neurons per layer and an output layer with 2 neurons as shown in Figure 3. In general, 1 to 5 hidden layers and same number of neurons for all hidden layers will be enough for most problems. Therefore, we chose to use two hidden layers and each layer has 8 neurons since we did not have too many input features. The input features used were also based on the feature selecting result in Section 2.1, so we had seven input features for our training model and it was taken for granted that we set the number of input neurons to 7, which represented each input feature. Because the earthquake prediction problem is a binary classification, we used 2 neurons in output layer. Each output neuron represents one class. The output represents the probability of the classes (earthquake or no earthquake). In the end, the SoftMax activation function (Dunne & Campbell 1997) are performed on the output layer, which make the final probabilities of classes sum to 1 (SoftMax we implemented in training function rather than in definition of NN class).

We used PyTorch to conveniently define the neural network model corresponding to architecture in Figure 3 by the code in Appendix Section 6.3.1. It implemented three linear transformations, which represented input layer $\rightarrow 1^{st}$ hidden layer, 1^{st} hidden layer $\rightarrow 2^{nd}$ hidden layer and 2^{nd} hidden layer $\rightarrow$ output layer. Each transformation performs a linear transformation to input data in the form of $y = xA^T + b$. *x* represents the vector of neuron values before transformation and *y* represents a vector of neuron values after transformation. *A* represents the weights multiplied in linear transformation (like coefficients in regression) and *b* represents the vector of bias. Our input layer corresponding to input features, so we set the *n_input* variable in code to 7. Our output layer corresponding to output target class, so we set the *n_output* variable in code to 2. The *__init__* function defines the architecture of neural network. The *forward* function in the code represents the process of propagation.

4.5.2 Configuration

Because this prediction problem is binary classification problem, we used the cross-entropy (De Boer et al. 2005) as the loss function of model. By lots of training attempts, we finally used a set of hyperparameters which make the model performance in a best level.

Generally, the best performance was reached by mini-batch sizes in the range of 2 to 32. In training attempts, we found that the model performed best when the batch size was set to 8. For the epoch size, we started from 400 epochs which is a large selection for epoch size and stopped early to halt training when performance stopped improving. The selection of the learning rate is also significant, we started with a pretty low value (10^{-6}) and slowly multiply it by 10 until it reaches 1 to determine which rate served well for our earthquake prediction problem. We found 10^{-3} is best and retrained our model using this optimal learning rate. For the optimizer used in our model, we decided to use Adam optimizer, which tends to excuse bad learning late and other non-optimal hyperparameters compared with most common used SGD optimizer.

The configuration which is called hyperparameters in machine learning is summarized below:

- *learning rate*: 10^{-3}
- *batch size*: 8
- *epoch*: 200
- *optimizer*: Adam
- *loss function*: cross entropy

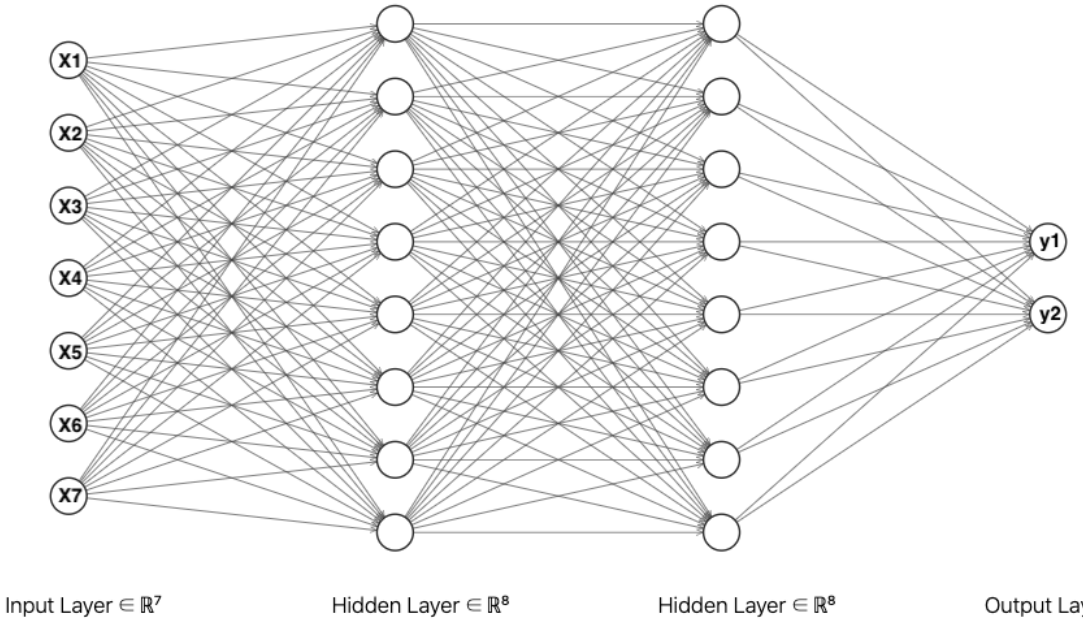


Figure 3: Constructed feedforward neural network model architecture.

4.5.3 Training

We initially divided the dataset into training dataset and testing dataset and normalized them as we done before fitting logistic regression model in 4.4. We trained the NN model with the optimized hyperparameters above. For each epoch, we trained using the whole training dataset and tested the accuracy of current model in testing dataset. We used *torch.utils.data.DataLoader* to load the training dataset and testing dataset. Every time we loaded a batch (8 samples) of training data and input into model for propagation, the outputs were compared with the labels. According to loss of current batch, we updated the weights in model by *optimizer.step()*. After training by each epoch, we calculate the accuracy of NN model on training dataset and testing dataset respectively and updated the loss of them in log plot. The training process of our NN model is shown in Appendix Figure 13.

5 Results and Discussion

This section will analyze the results from Section 4.

5.1 Visualization

Through these activities distribution in Fig 4, we can find a rough spatial correlation between some of them. We firstly compare the depth to basement with earthquake activities. In the Figure 4 (e) it shows lower left region has larger depth value and upper right region almost has 0 value, which means the lower left region of interest area has a thicker softer sedimentary rocks than upper right region. This corresponds to earthquake occurrences more in the upper right region than in the lower left region of interest area. It can be inferred that the induced seismicity may be related to depth to basement. Besides, injection activities are similar to earthquakes activities in spatial distribution. As the Figure 4 shows, three black rectangular regions (marked by 1 to 3) represent the seismicity and injections dense area respectively, which are highly similar in spatial distribution, even in small area

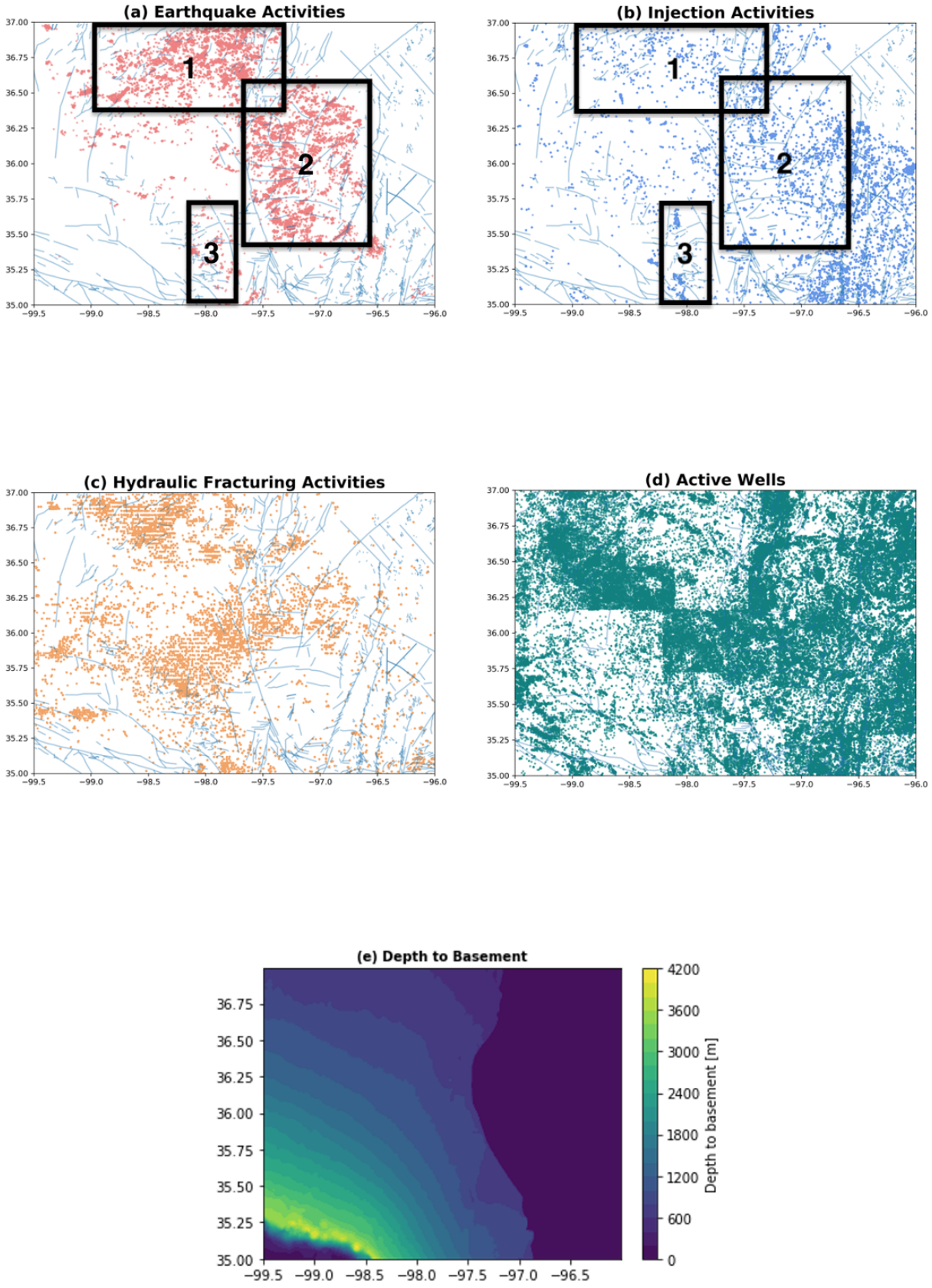


Figure 4: Activities during 2011 to 2018 in interest area from Oklahoma (The three black boxes in (a)(b) denote similar dense distribution between seismicity and injections).

as rectangular area 3. By the high similarity, we can conclude that the induced seismicity has high correlation with injection activities. However, in terms of the hydraulic fracturing activities and well spatial data, we cannot find some relation between induced earthquake and them and we need to explore and interpret through the model generated.

5.2 Features Selection

The process of features selection with stepwise regression approach is shown in Appendix Section 6.3.2 and the result of feature selecting along with their p-value and VIF is shown in Figure 5.

```
Feature selected:  injection_vol_sum
Optimization terminated successfully.
         Current function value: 0.590683
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:                  class   No. Observations:                 1600
Model:                          Logit   Df Residuals:                     1593
Method:                           MLE   Df Model:                            6
Date:                Sat, 07 Aug 2021   Pseudo R-squ.:                  0.1337
Time:                        16:10:17   Log-Likelihood:                -945.09
converged:                       True   LL-Null:                       -1091.0
Covariance Type:            nonrobust   LLR p-value:                 4.787e-60
=================================================================================================
                                    coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
hf_base_water_volume_sum         -0.5612      0.114     -4.908      0.000      -0.785      -0.337
hf_number                         0.9908      0.152      6.517      0.000       0.693       1.289
depth_to_basement_avg            -1.4814      0.177     -8.372      0.000      -1.828      -1.135
well_depth_avg                    0.9690      0.166      5.827      0.000       0.643       1.295
injection_under_basement_number  -0.6736      0.119     -5.660      0.000      -0.907      -0.440
injection_depth_avg               0.4854      0.060      8.037      0.000       0.367       0.604
injection_vol_sum                 0.2731      0.105      2.612      0.009       0.068       0.478
=================================================================================================
                          feature       VIF
0         hf_base_water_volume_sum  2.631406
1                        hf_number  2.814120
2            depth_to_basement_avg  3.822348
3                    well_depth_avg  3.851949
4  injection_under_basement_number  1.999341
5               injection_depth_avg  1.143023
6                 injection_vol_sum  2.161789
```

Figure 5: Result of feature selection using stepwise regression with VIF checking.

We can see all the selected features by stepwise regression approach meet the requirement of p-value <0.05 and VIF <5. This means we successfully eliminated the features which are not statistically significant and there is no high multicollinearity in all features selected.

5.3 Model Evaluation and Comparison

By the Section 4.4 and Section 4.5, we fitted the LR model and trained the configured NN model on the training dataset. Their prediction accuracy on testing dataset achieved at 72.18% and 80.3% respectively. It denotes that NN model performs better than LR model in terms of unseen dataset and have better ability of generalization. Besides, we finally used these models to test on the whole dataset, which contains all the grids point of interpret area. The results on whole dataset will be evaluated in this section. The confusion matrixes of results by these two models are shown on Figure 6. The measures we introduced in Section 2.4 to evaluate models are calculated based on confusion matrices and shown in Table 1.

The accuracy denotes the proportion of all correct predictions in the total predictions. We can see for all regions in interest area, the prediction accuracy of NN model (79.1%) is 9 percent higher than LR model (70.5%). Through the accuracy comparison for LR model and NN model in Table 1, it shows that for a task of predicting a class label of one region in interest area, NN model have

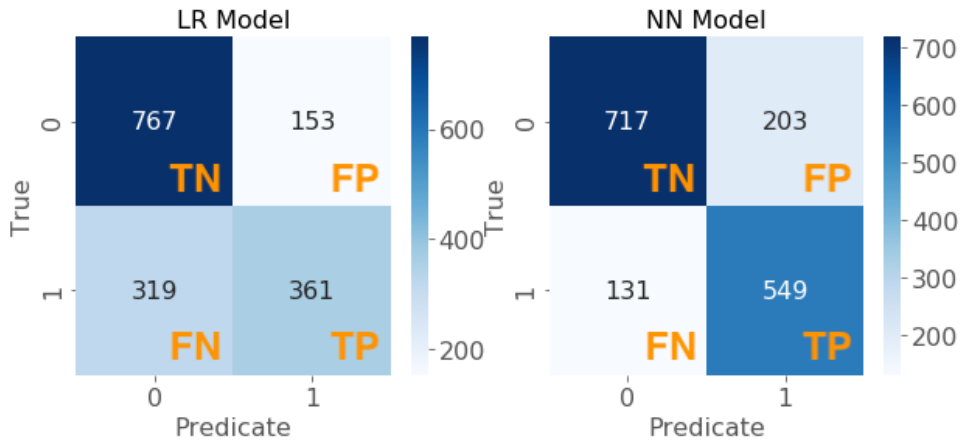


Figure 6: Confusion matrices of prediction results by LR model and NN model.

Table 1: Measures calculated using confusion matrices by LR model and NN model

Model\Indicator	Accuracy	Precision	Recall	F1-score
Logistic Regression Model	0.705	0.702	0.530	0.604
Neural Network Model	0.791	0.730	0.807	0.766

a higher prediction accuracy. Unfortunately, the samples in our dataset are not balanced in the ratio of 2:3 as shown in Figure 2. Therefore, using the accuracy to evaluate has great defects introduced in Section 2.4.1 in this earthquake prediction problem, which may have high accuracy but meaningless.

We also compare two models in precision and recall. According to the Table 1, the precisions of two models are similar. This denotes that for all seismic predictions, two models have the similar ability to predict correctly. However, we found the recall of NN model is much higher than that of LR model, which is 0.53 (LR) and 0.807 (NN). This means that for one real seismic region, the NN model has the probability of 80.7% but LR model only have a probability of 53% to predict correctly. The LR model shows a terrible performance for binary classification problem concentrating on real seismic regions. Furthermore, in order to meet our two expectations mentioned simultaneously in Section 2.4.1, we also introduced the F1 score. As Table 1 shows, LR model has the F1-score of 0.604 and NN model has the F1-score of 0.766. It shows that NN model indeed satisfied expectations that it can both make more accurate predictions and select out more actual seismic samples.

In ROC curve, the diagonal line (black line in Figure 7) represents random guessing. Therefore, a model with ROC curve below diagonal has the performance inferior to random guessing. When we evaluate model on ROC, we hope that the TP rate to be high when the FP rate is low because we want no false positive and no false negative. This state is just the ideal perfect state for a binary classification problem (FPR $= 0$, TPR $= 1$, which is (0, 1) in left top corner in Figure 7), where all the predictions are correct. In other words, the steeper the ROC curve or the closer to left top corner, the more accurate the model is. From the Figure 7, we can see both LR model curve and NN model curve are above the diagonal line, which means these two models are better than random guessing. Furthermore, ROC of NN is always above that of the LR and closer to left top corner. AUC is an intuitive measure in ROC plot. LR model has the AUC of 0.68 but NN has the AUC of 0.79. Therefore, all these prove that NN model indeed have a better prediction ability than LR model.

In the last part, we visualized the predictions by two models on Oklahoma map to make a more intuitive comparison. The mapping results of predictions and mapping are shown in Figure 8. Both LR and NN model predicted the spatial distribution of major earthquake regional clusters mentioned

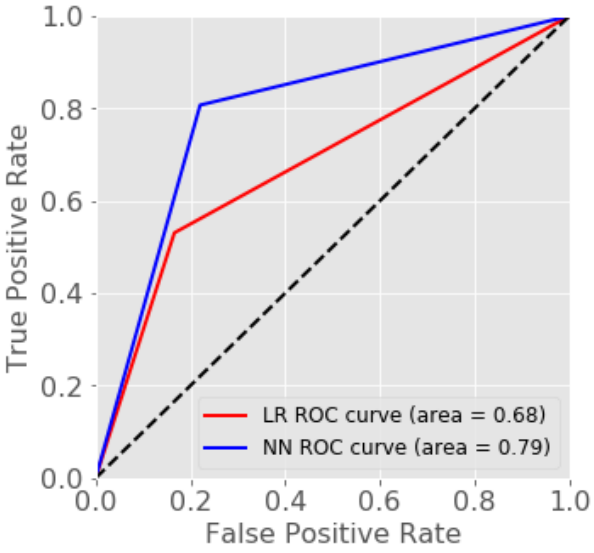


Figure 7: ROC curves by LR model and NN model (Black diagonal line represents random guessing in binary classification problem).

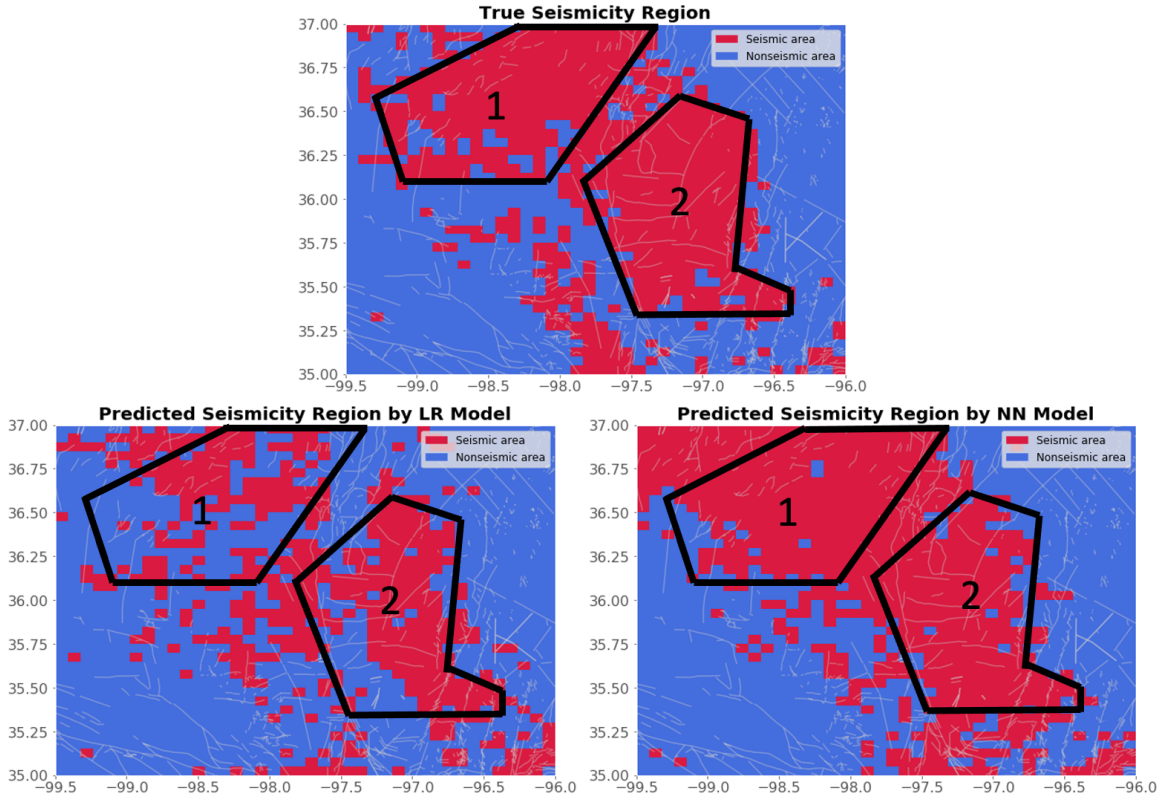


Figure 8: Seismicity prediction by logistic regression model and neural network model (The black line areas represent the interest areas where the performance compared between LR and NN model in Section 5.3).

in Section 5.1, even though the precision of regional outline is poor. However, in terms of TPR, these two models performed greatly differently by predicted maps comparison with true map. We want TPR (true positive rate) to be closer to 1 (equivalently FN closer to 0), which means we want to predict out more seismic regions among true seismic regions. It is because we tend to regard an region at the risk of seismicity as seismic rather than nonseismic, in order to prevent the property damage and casualties. According to Figure 8, all gridding regions in the two blocks marked by 1 and 2 are almost seismic (positive) in true seismic map. NN model indeed selected out almost all true seismic regions in these two blocks, but LR model performs badly and wrongly predicts many seismic regions to be nonseismic.

5.4 Feature Importance Analysis

In this part, we will analyze the influence of each feature on earthquake prediction. According to feature analysis by LR model and NN model in Figure 9, we found the agreed main feature in them is *the depth to basement*. It shows that earthquake occurrence is negatively correlated with its depth to basement in that region. Furthermore, *well_depth_avg* (the average depth of active wells) and *hf_number* (the number of hydraulic fracturing activities) are also important features, which are positively correlated with seismicity in LR model but negatively correlated with seismicity in NN model. This might be due to incomplete of dataset, or because LR model is so poorly fitted that the coefficient sign is wrong. Anyway, these two features should be further investigated in future work. In NN model, in addition to three features above, we found that *hf_base_water_volume_sum* is also considered to be important and negatively correlated with earthquake occurrence based on most attribution algorithms.

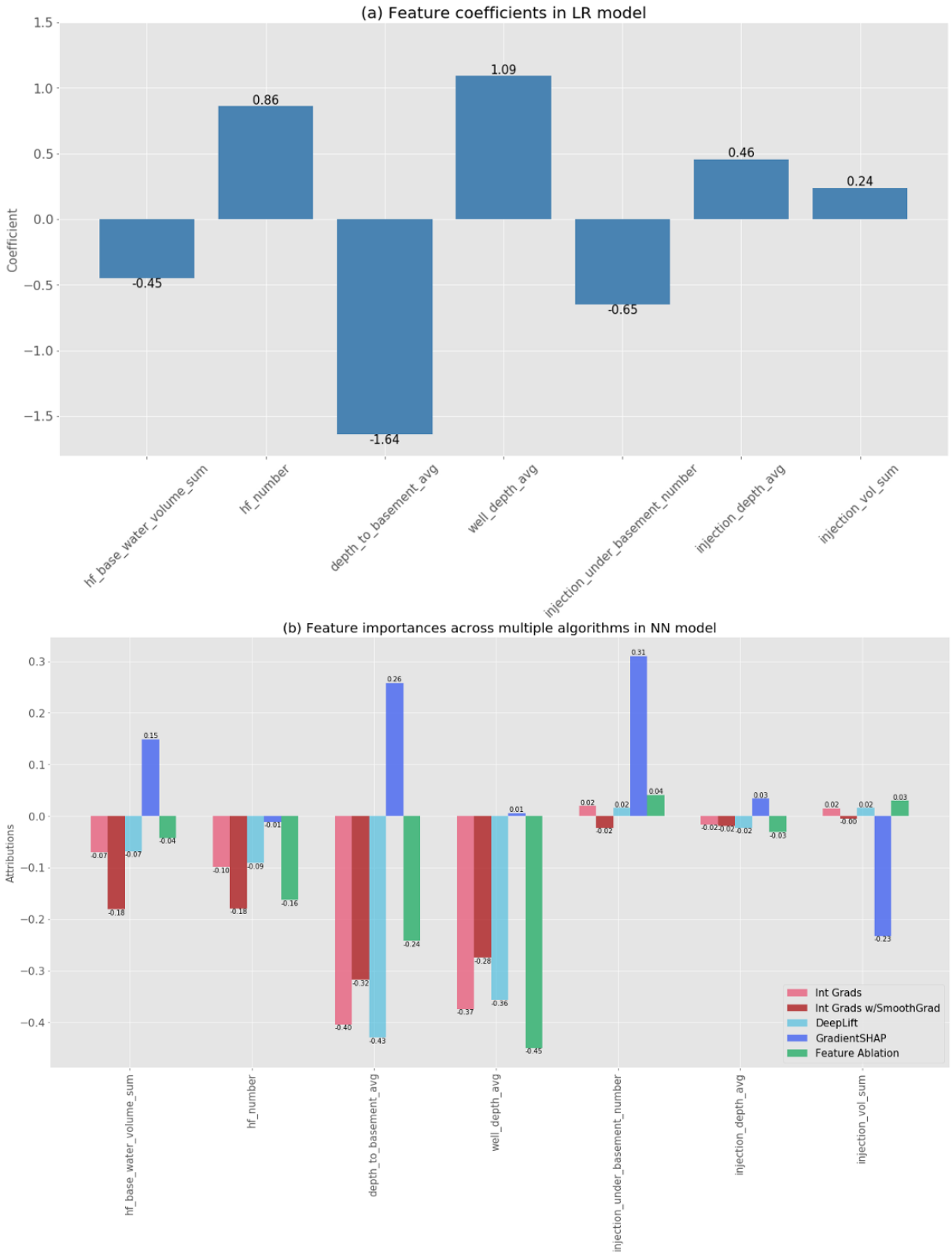


Figure 9: Feature attribution result by LR model and NN model.

6 Conclusion and Future Work

This project visualized various geological and industrial factors in Oklahoma and found some of them have rough correlations with seismicity. We selected out statistically significant features and eliminated multicollinearity by forward stepwise regression approach with VIF checking. These features contain number of hydraulic fracturing activities, volume of water in hydraulic fracturing, depth to basement, depth of wells, volume of injections and number of injections deeper than basement.

Compared with physical model based on fluid flow and seismic physics by (Norbeck & Rubinstein 2018), this project constructed two models using machine learning methods. The logistic model works not badly in predicting non-seismic regions but have deficiencies in predicting seismic regions. The neural network model is good at predicting actual seismic areas in which we concentrate on. According to the result map of interest area we selected, we concluded that neural network model trained can accurately predict out the most actual regions of the earthquake occurrence. This is what logistic regression model cannot do.

Through the feature importance analysis in LR model and NN model, we found the depth to basement is the key predictor of the spatial distribution of seismicity. It is consistent with both models that seismicity is largely promoted with the reduce in depth to basement. The amount of hydraulic fracturing and the depth of wells are also found important but they have opposite correlation with seismicity among these two models, which require to be further investigated. Furthermore, water volume in base of HF is additionally considered to be important and negatively correlated with earthquake occurrence through NN model.

We have implemented and tried to collect various data in a circular region with a radius of 10 km from the center in each rectangular gird point region. For distance calculation in radius mode, we have tried two methods: One was using GeoPy module directly and the other was computing through distance formulas after cartesian coordinate mapping. However, they were time consuming (at least a few days cost) and not suitable for debugging my models. Due to the time limitation of this project, we only implemented the radius mode but actually used rectangular mode for training data generation. Therefore, we can use more reasonable radius mode for collecting features in future study, which may obtain ideal predictive performance and feature importance analysis. It might solve the partial inconsistencies of signs in the feature attribution section.

References

Benesty, J., Chen, J., Huang, Y. & Cohen, I. (2009), Pearson correlation coefficient, *in* 'Noise reduction in speech processing', Springer, pp. 1–4.

Chhikara, R. (1988), *The Inverse Gaussian Distribution: Theory: Methodology, and Applications*, Vol. 95, CRC Press.

De Boer, P.-T., Kroese, D. P., Mannor, S. & Rubinstein, R. Y. (2005), 'A tutorial on the cross-entropy method', *Annals of operations research* **134**(1), 19–67.

Dunne, R. A. & Campbell, N. A. (1997), On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function, *in* 'Proc. 8th Aust. Conf. on the Neural Networks, Melbourne', Vol. 181, Citeseer, p. 185.

Ellsworth, W. L. (2013), 'Injection-induced earthquakes', *Science* **341**(6142).

Hincks, T., Aspinall, W., Cooke, R. & Gernon, T. (2018), 'Oklahoma's induced seismicity strongly linked to wastewater injection depth', *Science* **359**(6381), 1251–1255.

Hough, S., Tsai, V., Walker, R. & Aminzadeh, F. (2017), 'Was the mw 7.5 1952 kern county, california, earthquake induced (or triggered)?', *Journal of Seismology* **21**, 1613 – 1621.

Howard, B., Christopher, S., Dane, S. & Josh, L. (2007), 'epsg projection 4326 - wgs 84'.
URL: *https://spatialreference.org/ref/epsg/wgs-84/*

Ioannidis, J. P. (2018), 'The proposal to lower p value thresholds to. 005', *Jama* **319**(14), 1429–1430.

Lundberg, S. M. & Lee, S.-I. (2017), A unified approach to interpreting model predictions, *in* 'Proceedings of the 31st international conference on neural information processing systems', pp. 4768–4777.

Norbeck, J. & Rubinstein, J. L. (2018), 'Hydromechanical earthquake nucleation model forecasts onset, peak, and falling rates of induced seismicity in oklahoma and kansas', *Geophysical Research Letters* **45**(7), 2963–2975.

OklahomaCorporationCommission (2021), 'Oil and gas data files'.
URL: *https://spatialreference.org/ref/epsg/wgs-84/*

Patro, S. & Sahu, K. K. (2015), 'Normalization: A preprocessing stage', *arXiv preprint arXiv:1503.06462* .

PyTorch (2021), 'Stepwise regression tutorial in python'.
URL: *https://pytorch.org/docs/stable/index.html*

Registry, F. F. C. D. (2021), 'Data download'.
URL: *https://fracfocus.org/data-download*

Shrikumar, A., Greenside, P. & Kundaje, A. (2017), Learning important features through propagating activation differences, *in* 'International Conference on Machine Learning', PMLR, pp. 3145–3153.

Sundararajan, M., Taly, A. & Yan, Q. (2017), Axiomatic attribution for deep networks, *in* 'International Conference on Machine Learning', PMLR, pp. 3319–3328.

Walter, J. I., Ogwari, P., Thiel, A., Ferrer, F., Woelfel, I., Chang, J. C., Darold, A. P. & Holland, A. A. (2020), 'The oklahoma geological survey statewide seismic network', *Seismological Research Letters* **91**(2A), 611–621.

Appendix

6.1 Supplementary Figures

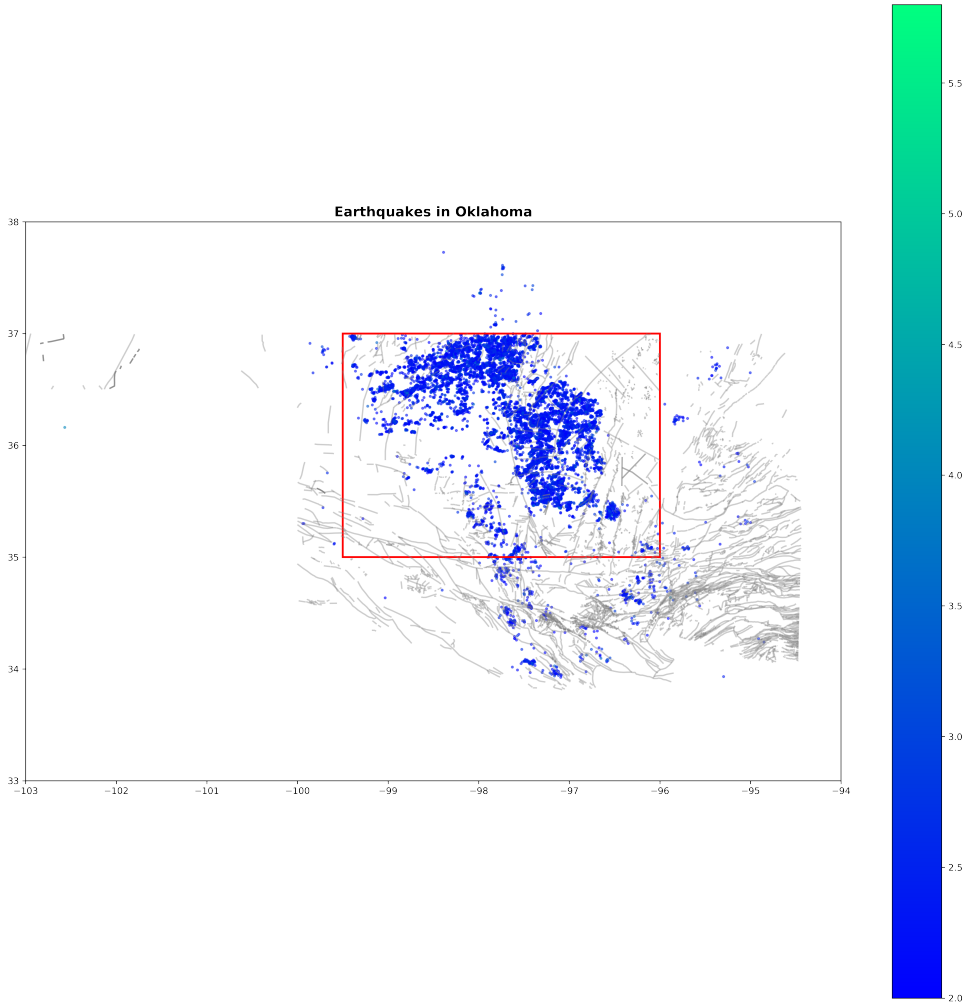


Figure 10: Earthquake occurrences in Oklahoma. The color of marker represents the magnitude of earthquake. The red box represents the interest area we studied in this project. The longitude ranges in (-99.5W, -96W) and latitude ranges in (35.0N, 37.0N)

```
Optimization terminated successfully.
         Current function value: 0.631770
         Iterations 7
                           Logit Regression Results
==============================================================================
Dep. Variable:                  class   No. Observations:                 1600
Model:                          Logit   Df Residuals:                     1593
Method:                           MLE   Df Model:                            6
Date:                Wed, 28 Jul 2021   Pseudo R-squ.:                 0.04317
Time:                        17:38:31   Log-Likelihood:                -1010.8
converged:                       True   LL-Null:                       -1056.4
Covariance Type:            nonrobust   LLR p-value:                 1.699e-17
===========================================================================================
                              coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------
injection_vol               0.1912      0.119      1.609      0.108      -0.042       0.424
injection_psi              -0.6618      0.206     -3.219      0.001      -1.065      -0.259
injection_depth_sum         0.3869      0.130      2.976      0.003       0.132       0.642
depth_to_basement          -0.4143      0.059     -7.030      0.000      -0.530      -0.299
HF_number                   0.7876      0.149      5.299      0.000       0.496       1.079
HF_Base_Water_Volume       -0.0275      0.135     -0.204      0.838      -0.292       0.237
HF_Base_NoWater_Volume     -0.1966      0.119     -1.659      0.097      -0.429       0.036
===========================================================================================
```

Figure 11: Feature selection in stepwise regression with p-value (It is an intermediate step in stepwise logistic regression process. The 'injection_vol', 'HF_Base_Water_Volume', 'HF_Base_NoWater_Volume' both have a p-value greater than threshold 0.05 in logistic regression results, which are not statistically significant and some of them should be eliminated properly in later process.)

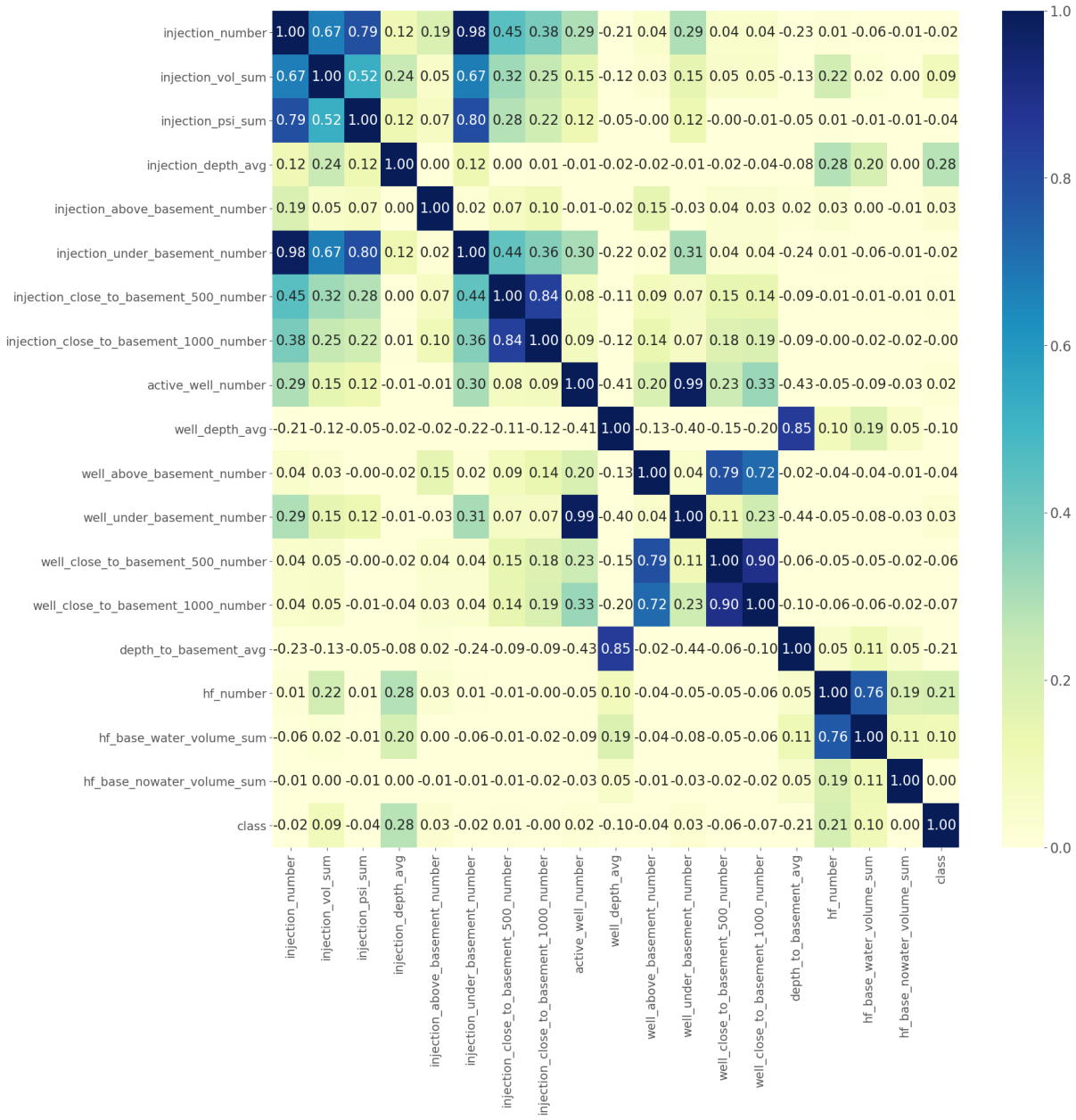


Figure 12: Correlation heatmap generated in Section 4.3. We noted that there exists multicollinearity (dark blocks in figure) among these candidate features.

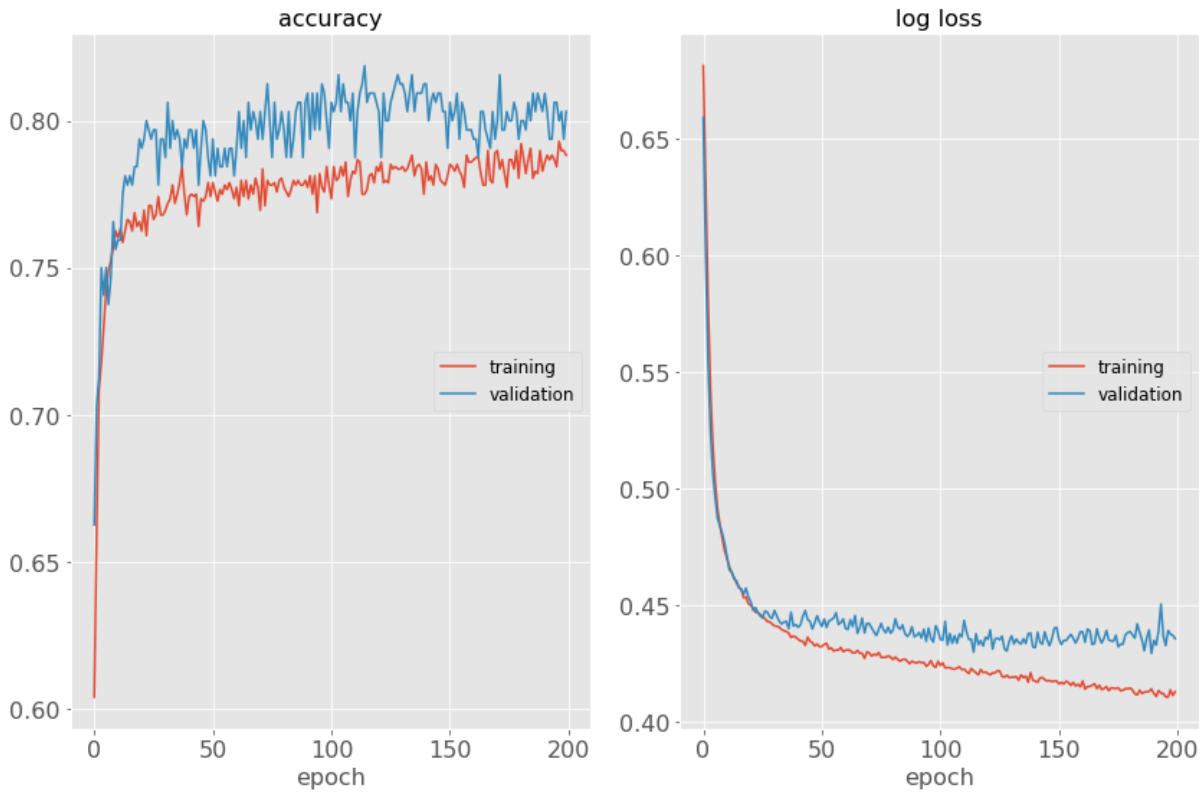


Figure 13: Neural network model training process in Section 4.5.3. We stopped training when the log loss of the validation is no longer reduced. It means NN model is currently in an optimal trained state.

6.2 Supplementary Tables

Table 2: The dependencies of this project

Module	Version	Purpose
os	3.7.11	Search the data files (csv or xlsx) in the directory
numpy	1.18.1	Dataset process
pandas	1.0.1	
Shapely	1.7.1	Map the activities on map in spatial
geopandas	0.9.0	
matplotlib	3.1.3	Plot the figure
seaborn	0.10.0	
scipy	1.6.2	Load and process the geological formation data and plot the figure
xarray	0.18.2	
netCDF4	1.5.7	
statsmodels	0.12.2	Used in stepwise regression
torch	1.9.0	Neural network model implementation and training, and visualization of intermediate information
scikit-learn	0.22.1	
livelossplot	0.5.4	
captum	0.4.0	Perform the feature attribution

Table 3: Features generated by spatial gridding and data collecting for regions

Feature	Interpretation
injection_number [entry]	the number of injections
injection_vol_sum [BPD]	the sum of injection volume
injection_psi_sum [PSI]	the sum of injection pressure
injection_depth_avg [m]	the mean of injection depth
injection_above_basement_number [entry]	the number of injections whose depth are shallow than the basement
injection_under_basement_number [entry]	the number of injections whose depth are deeper than the basement
injection_close_to_basement_500_number [entry]	the number of injections whose depth is within 500 metres of basement
injection_close_to_basement_1000_number [entry]	the number of injections whose depth is within 1000 metres of basement
active_well_number [entry]	the number of working wells
well_depth_avg [m]	the mean of well depth
well_above_basement_number [entry]	the number of working wells whose depth are shallow than the basement
well_under_basement_number [entry]	the number of working wells whose depth are deeper than the basement
well_close_to_basement_500_number [entry]	the number of wells whose depth is within 500 metres of basement
well_close_to_basement_1000_number [entry]	the number of wells whose depth is within 1000 metres of basement
depth_to_basement_avg [m]	the mean of the depth to the basement
hf_number [entry]	the number of hydraulic fracturing activities
hf_base_water_volume_sum [BPD]	the sum of volume by hydraulic fracturing with water
hf_base_nowater_volume_sum [BPD]	the sum of volume by hydraulic fracturing with nowater
earthquake_occurrence [0 or 1]	0: not occurred; 1: occurred.

6.3 Supplementary Codes and Results

6.3.1 The Code of Defining NN Model

```
    class Net(nn.Module):
    def __init__(self,n_input, n_output):
        super(Net,self).__init__()
        self.linear1 = nn.Linear(n_input, 8)
        self.relu1 = nn.ReLU()
        self.linear2 = nn.Linear(8, 8)
        self.relu2 = nn.ReLU()
        self.linear3 = nn.Linear(8, n_output)

    def forward(self, x):
        lin1_out = self.linear1(x)
        relu_out1 = self.relu1(lin1_out)
        relu_out2 = self.relu2(self.linear2(relu_out1))
        return self.linear3(relu_out2)
```

6.3.2 The Process in Stepwise Regression with VIF Checking

- *Iteration 1*
  Feature selected: hf_base_nowater_volume_sum
  P-value >0.05 check: hf_base_nowater_volume_sum
  Feature removed: hf_base_nowater_volume_sum
  VIF>5 check: None
  Feature removed: None

- *Iteration 2*
  Feature selected: hf_base_water_volume_sum
  P-value >0.05 check: None
  Feature removed: None
  VIF>5 check: None
  Feature removed: None

- *Iteration 3*
  Feature selected: hf_number
  P-value >0.05 check: None
  Feature removed: None
  VIF>5 check: None
  Feature removed: None

- *Iteration 4*
  Feature selected: depth_to_basement_avg
  P-value >0.05 check: None
  Feature removed: None
  VIF>5 check: None
  Feature removed: None

- *Iteration 5*
  Feature selected: well_close_to_basement_1000_number
  P-value >0.05 check: None
  Feature removed: None
  VIF>5 check: None
  Feature removed: None

- *Iteration 6*
  Feature selected: well_close_to_basement_500_number
  P-value >0.05 check: well_close_to_basement_500_number(0.647), well_close_to_basement_1000_number(0.065)
  Feature removed: well_close_to_basement_500_number
  VIF>5 check: None
  Feature removed: None

- *Iteration 7*
  Feature selected: well_under_basement_number
  P-value >0.05 check: None
  Feature removed: None
  VIF>5 check: None
  Feature removed: None

- *Iteration 8*
  Feature selected: well_above_basement_number
  P-value >0.05 check: well_above_basement_number(0.46)
  Feature removed: well_above_basement_number
  VIF>5 check: None
  Feature removed: None

- *Iteration 9*
  Feature selected: well_depth_avg
  P-value >0.05 check: well_close_to_basement_1000_number(0.352)
  Feature removed: well_close_to_basement_1000_number
  VIF>5 check: None
  Feature removed: None

- *Iteration 10*
  Feature selected: active_well_number
  P-value >0.05 check: well_under_basement_number(0.477), active_well_number(0.772)
  Feature removed: active_well_number
  VIF>5 check: None
  Feature removed: None

- *Iteration 11*
  Feature selected: injection_close_to_basement_1000_number
  P-value >0.05 check: injection_close_to_basement_1000_number(0.876)
  Feature removed: injection_close_to_basement_1000_number
  VIF>5 check: None
  Feature removed: None

- *Iteration 12*
  Feature selected: injection_close_to_basement_500_number
  P-value >0.05 check: injection_close_to_basement_500_number(0.876)
  Feature removed: injection_close_to_basement_500_number
  VIF>5 check: None
  Feature removed: None

- *Iteration 13*
  Feature selected: injection_under_basement_number
  P-value >0.05 check: well_under_basement_number(0.102)
  Feature removed: well_under_basement_number
  VIF>5 check: None
  Feature removed: None

- *Iteration 14*
  Feature selected: injection_above_basement_number
  P-value >0.05 check: None
  Feature removed: None
  VIF>5 check: None
  Feature removed: None

- *Iteration 15*
  Feature selected: injection_depth_avg
  P-value >0.05 check: injection_above_basement_number
  Feature removed: injection_above_basement_number
  VIF>5 check: None
  Feature removed: None

- *Iteration 16*
  Feature selected: injection_psi_sum
  P-value >0.05 check: injection_psi_sum(0.969)
  Feature removed: injection_psi_sum
  VIF>5 check: None

Feature removed: None

- *Iteration 17*
  Feature selected: injection_vol_sum
  P-value >0.05 check: None
  Feature removed: None
  VIF>5 check: None
  Feature removed: None

- *Iteration 18*
  Feature selected: injection_number
  P-value >0.05 check: injection_under_basement_number(0.065), injection_number(0.194)
  Feature removed: injection_number
  VIF>5 check: None
  Feature removed: None