Disaster Response Management Using Few-Shot Learning

Isham Sinha
PES University

Siddarth Rao
PES University

Rehan Ahmed
PES University

Sabuj Das
PES University

Dr. Jyothi R
PES University

s_isham@outlook.com supersid43@gmail.com rehanahmed653@gmail.com sabujdasjohn97@gmail.com jyothir@pes.edu

Abstract**—**Disaster management is a critical area that demands efficient rapid response to unforeseen events and crises. Traditional approaches to disaster management often rely on extensive labelled data for training machine learning models, which can be impractical in scenarios where data is scarce or rapidly evolving. In this project, we propose a novel approach utilising few-shot learning techniques to enhance disaster management systems. Few-shot learning enables our system to learn from limited labelled examples, thereby allowing for more adaptable and responsive models. Our presentation will discuss the application of few-shot learning in disaster management, including the methodology, experimental results, potential impact on real-world disaster response scenarios. We aim to demonstrate the efficacy of this approach in addressing the challenges of rapidly disaster situations and improving the overall effectiveness of disaster management systems.

Index Terms**—**Few-shot learning, Disaster response, Contextual understanding, Ethical AI, Adaptability

I. INTRODUCTION

Natural disasters and crisis events generate complex, dynamic scenarios that traditional disaster management systems often struggle to handle effectively. Conventional machine learning (ML) models typically rely on large volumes of historical data to predict and manage responses. However, such models face significant obstacles including the scarcity of labeled data unique to each disaster, and the inability to rapidly adapt to new, unforeseen types of crises. This necessitates an innovative approach to enhance responsiveness and effectiveness in disaster management.

Objective:

In light of these challenges, this paper proposes the adoption of few-shot learning (FSL) strategies to develop a more flexible and adaptable disaster management system. Few-shot learning, a subset of machine learning, is designed to make reliable predictions from a very limited number of training examples. By integrating FSL into disaster management, this research aims to drastically improve the system's ability to function effectively in diverse and unforeseen disaster scenarios.

Few-Shot Learning: A Primer

Few-shot learning involves training a machine to learn from a very limited set of data. Unlike traditional ML approaches that require large datasets to model adequately, FSL techniques utilize prior knowledge and smart data synthesis to achieve remarkable accuracy with far fewer examples. This characteristic is particularly useful in disaster management, where each situation can be unique and historical data may not offer sufficient insights for new crisis types.

II. RELATED WORK

Recent advancements in machine learning, particularly deep learning, have shown great promise in supporting various disaster management tasks like situation awareness, damage assessment, and resource allocation. However, a significant hurdle in utilising these techniques lies in the limited availability of labelled data for specific disaster events. Few-shot learning (FSL) has emerged as a powerful approach to address this data scarcity.

Research in FSL for disaster management applications is multifaceted. Peng et al. [2023] explore FSL for fine-grained disaster tweet classification, while Zhao et al. [2022] demonstrate its effectiveness in a multi-task setting of earthquake detection and phase identification. While not directly employing FSL, Elbes et al. [2023] highlight the trend of deep learning for earthquake prediction, and Murshed et al. [2023] suggest incorporating meta-learning and FSL for real-time seismic intensity prediction with limited data.

FSL's applicability extends beyond earthquakes. Patil and Bhosale [2023] present an FSL approach for tropical cyclone severity prediction using satellite imagery, and Tran-Anh et al. [2023] propose an FSL model for landslide detection that integrates classification and segmentation tasks. Furthermore, a recent study [2023] explores FSL for post-earthquake urban damage detection using remote sensing data, addressing challenges like imbalanced datasets and data pre-processing.

These studies showcase the growing adoption of FSL in disaster management. While effective, further research is needed to ensure generalizability across diverse disaster scenarios and real-world data complexities.

III. DATASET

The Data Collection Process:

To obtain a foundational dataset for our disaster response management project, we conducted web scraping from the website [usgs.gov] (https://www.usgs.gov/programs/earthquake-hazards) collecting earthquake data for earthquakes in the time frame of 1 year (2013-14).


```
     type                                         properties  \
0  Feature  {'mag': 1.07, 'place': '6km NW of The Geysers,...
1  Feature  {'mag': 1.4, 'place': '54 km WNW of Ninilchik,...
2  Feature  {'mag': 0.94, 'place': '9km N of Morgan Hill, ...
3  Feature  {'mag': 0.8, 'place': '23 km ENE of Minto, Ala...
4  Feature  {'mag': 0.59, 'place': '13 km N of Naches, Was...
5  Feature  {'mag': 1.5, 'place': '57 km SSW of Nanwalek, ...
6  Feature  {'mag': 1.21, 'place': '20km ESE of Julian, CA...
7  Feature  {'mag': 0.68, 'place': '13km N of Borrego Spri...
8  Feature  {'mag': 1.70000005, 'place': '13 km ESE of P?h...
9  Feature  {'mag': 4.1, 'place': '92 km W of San Antonio ...

                                            geometry           id
0  {'type': 'Point', 'coordinates': [-122.8040009...    nc73840701
1  {'type': 'Point', 'coordinates': [-152.6119, 6...  ak0231hb492i
2  {'type': 'Point', 'coordinates': [-121.6403333...    nc73840691
3  {'type': 'Point', 'coordinates': [-148.9112, 6...  ak0231hb399b
4  {'type': 'Point', 'coordinates': [-120.7065, 4...    uw61904971
5  {'type': 'Point', 'coordinates': [-152.1984, 5...  ak0231hb18r9
6  {'type': 'Point', 'coordinates': [-116.4208333...    ci40165191
7  {'type': 'Point', 'coordinates': [-116.3901667...    ci40165183
8  {'type': 'Point', 'coordinates': [-155.3621673...    hv73317307
9  {'type': 'Point', 'coordinates': [-67.227, -24...    us6000jkv8
```


JSON data extracted through web-scraping

This involved extracting a comprehensive range of natural disaster data, with a specific focus on earthquake events due to their significant impact on disaster management.

	ID	Magnitude	Location	Event_occured	is_Tsunami	Type	logitude	Latitude	Depth
0	nc73840701	1.07	6km NW of The Geysers, CA	1675305555804	0	earthquake	-122.804001	38.817501	1.50
1	ak0231hb492i	1.40	54 km WNW of Ninilchik, Alaska	1676511181847	0	earthquake	-152.611900	60.187600	91.70
2	nc73840691	0.94	9km N of Morgan Hill, CA	1675461731417	0	earthquake	-121.640333	37.213500	5.60
3	ak0231hb399b	0.80	23 km ENE of Minto, Alaska	1676084327768	0	earthquake	-148.911200	65.260300	5.20
4	uw61904971	0.59	13 km N of Naches, Washington	1675299658230	0	earthquake	-120.706500	46.852833	19.05
...	...	...	...	...	...	...	...	...	...
12006	ci40152455	0.84	14km N of Warner Springs, CA	1672782126029	0	earthquake	-116.616833	33.408000	7.52
12007	ak0231nosyg	1.70	31 km NE of Paxson, Alaska	1674061617007	0	earthquake	-145.178900	63.297000	2.80
12008	ak0231nm7k5	1.20	7 km NNW of Meadow Lakes, Alaska	1674061620788	0	earthquake	-149.669200	61.685400	39.90
12009	nc73827346	0.69	1km NNE of The Geysers, CA	1673374637080	0	earthquake	-122.754500	38.783500	1.69
12010	nc73827341	0.09	1km NNE of The Geysers, CA	1673334674067	0	earthquake	-122.753667	38.783667	1.62

Earthquake Classification Based on Magnitude:

Using the scraped earthquake data, we implemented a classification process to categorise the earthquakes according to their magnitude. This step is important in establishing distinct classes of earthquakes based on varying magnitudes, enabling a more granular analysis of earthquake data.

Creation of Earthquake Classes:

Following the classification based on magnitude, we further segmented the earthquake data using additional relevant factors to create specific classes of earthquakes. These factors could include geographical location, depth, and any other pertinent parameters to ensure a comprehensive and nuanced categorization of earthquake events.

By meticulously curating and categorising the earthquake data, we established a robust foundation for training and validating our few-shot learning models for disaster response management.

IV. METHODOLOGY

A. Data Collection and Preprocessing

The seismic dataset utilised in this study was sourced from the United States Geological Survey (USGS) website (https://earthquake.usgs.gov) via web-scraping techniques. The data was originally available in JSON format on the USGS platform.

Upon retrieval, the seismic data was converted into a structured tabular format using the `pandas` library in Python. The preprocessing steps involved were:

- **Filtering Earthquake Events**: All seismic events classified as earthquake-related were retained, while non-earthquake events were filtered out from the dataset.
- **Normalisation of Tsunami Information**: Boolean values indicating the occurrence of an accompanying tsunami were converted to categorical variables, with "True" denoting the

presence of a tsunami and "False" representing its absence.

- **Magnitude-Based Impact Classification**: The severity of each earthquake event was categorised based on its magnitude on the Richter scale. Impact levels were defined as follows:
    - **Light**: Magnitude < 4.0
    - **Strong**: 4.0 <= Magnitude <= 6.0
    - **Great Impact**: Magnitude > 6.0

	ID	Magnitude	Location	Event_Occured	Is_Tsunami	Type	Longitude	Latitude	Depth	Earthquake_impact
0	nc73840701	1.07	6km NW of The Geysers, CA	Thursday at 02: 39: 15 (02February2023)	False	earthquake	-122.804001	38.817501	1.50	MINOR
1	ak0231hb492i	1.40	54 km WNW of Ninilchik, Alaska	Thursday at 01: 33: 01 (16February2023)	False	earthquake	-152.611900	60.187600	91.70	MINOR
2	nc73840691	0.94	9km N of Morgan Hill, CA	Friday at 22: 02: 11 (03February2023)	False	earthquake	-121.640333	37.213500	5.60	MINOR
3	ak0231hb399b	0.80	23 km ENE of Minto, Alaska	Saturday at 02: 58: 47 (11February2023)	False	earthquake	-148.911200	65.260300	5.20	MINOR
4	uw61904971	0.59	13 km N of Naches, Washington	Thursday at 01: 00: 58 (02February2023)	False	earthquake	-120.706500	46.852833	19.05	MINOR
...	...	...	...	...	...	...	...	...	...	...
12006	ci40152455	0.84	14km N of Warner Springs, CA	Tuesday at 21: 42: 06 (03January2023)	False	earthquake	-116.616833	33.408000	7.52	MINOR
			31 km NE of Paxson,	Wednesday at 17: 06: 57						

Dataset after preprocessing null values

The features selected for earthquake impact classification included:

- Magnitude of the earthquake
- Depth of the seismic event
- Geographic coordinates (Latitude and Longitude) of the epicentre

These features were chosen due to their known influence on earthquake severity and impact assessment.

The target variable for classification was "Earthquake Impact," representing the severity level of each seismic event based on its magnitude classification.

B. Few-shot model using RandomForestClassifier:

I. Data Selection

The feature and target variables for the seismic dataset were carefully chosen. The selected feature variables included 'Magnitude', 'Depth', 'Latitude', and 'Longitude', while the target variable was identified as "Earthquake_impact".

II. Dataset Splitting

Initially, the dataset was divided into training and testing subsets using a standard train-test split. Subsequently, to facilitate fine-tuning and validation during few-shot learning, the training dataset was further partitioned into training and validation sets.

III. Few-Shot Learning with Random Forest

- **Definition of Few-Shot Size**: The size of the few-shot samples was determined, specifying the number of data points to be included in each few-shot task.
- **Training Random Forest Classifier**: For both the training and validation datasets, random forest classifiers were trained using the few-shot learning approach. This involved training the classifier on a limited amount of labelled data, simulating a scenario where only a small amount of labelled data is available.
- **Validation and Fine-Tuning**: The trained random forest models were validated using the validation dataset. Fine-tuning techniques were applied to optimise the model's performance based on the validation results.

IV. Evaluation

Accuracy Calculation: Following the training and fine-tuning process, the accuracy of the random forest classifiers was calculated. This accuracy metric quantified the models' effectiveness in predicting earthquake impact severity based on the selected features.

C. Few-shot Modelling using StratifiedKFold:

I. Data Selection and Splitting

The process of selecting feature and target variables from the seismic dataset remained consistent with the approach outlined in the previous section.

C. Cross-Validation with Random Forest for Few-Shot Learning

- **Determining Number of Folds**: For robust evaluation, the number of folds for cross-validation was set to 5. This value was chosen for balancing computational efficiency with the need for reliable performance estimation.
- **Initialization of Accuracy Scores**: A list was initialised to store the accuracy scores obtained from each fold during cross-validation.
- **Cross-Validation Procedure**: Cross-validation was performed, involving the partitioning of the dataset into training and validation subsets across multiple folds. Stratified k-fold cross-validation

was employed to maintain class distribution balance across folds.

- **Evaluation and Accuracy Calculation**: Within each fold, the random forest model was trained on the training subset and evaluated on the validation subset. The accuracy score of the model on the validation subset was computed and appended to the list of accuracy scores.
- **Average Accuracy Calculation**: After completing cross-validation across all folds, the average accuracy across the folds was calculated. This average accuracy score served as the overall performance metric for the random forest model in few-shot learning for disaster management.

D. Few-Shot Learning with Meta-Learning (Model-Agnostic Meta-Learning)

I. **Library Imports**:

- The implementation of Few-Shot Learning using Meta-Learning required the import of relevant libraries:
    - torch: PyTorch library for tensor computation and neural network training.
    - torch.optim.Adam: Optimizer from PyTorch for stochastic optimization, specifically the Adam algorithm.
    - torch.nn: Module from PyTorch for building neural networks.
    - innerloop_ctx from higher: Context manager from the Higher library, facilitating higher-order gradients computation for meta-learning.

II. **Data Preprocessing**:

- Feature Selection: The columns to be used as features and the target variable were selected. The feature columns included 'Magnitude', 'Depth', 'Latitude', and 'longitude', while the target variable was the column named "Earthquake_impact".
- Standardization and Encoding: Preprocessing steps like standardisation of feature columns and encoding of the target column as integers were performed using standard techniques to prepare the data for training.

III. **Neural Network Model Definition**:

- The architecture for a simple neural network called **SimpleNN** was defined using PyTorch. This model is suitable for classification tasks and consists of two fully connected layers with a ReLU activation function between them. It takes input features and produces output logits for a specified number of classes.

IV. **Model Initialization**:

- Input and output dimensions for the model were instantiated based on the data. An optimizer, specifically Adam, with a learning rate of 0.001, was defined. Additionally, the loss function, CrossEntropy, was specified for classification tasks.

V. **Training with MAML**:

A function named **train_maml** was defined for training the model using the Model-Agnostic Meta-Learning (MAML) algorithm. This function takes the model, data, number of iterations, inner learning rate, and outer learning rate as inputs. It implements the MAML algorithm by iteratively adapting the model's weights to different tasks and updating the meta-model's weights based on the performance on query sets.

VI. **Task Creation and Training**:

- The data was split into multiple tasks, each containing a subset of the original data, to facilitate few-shot learning. These tasks were created by dividing the length of the data by a specified task size.
- The **train_maml** function was called with the model and the list of tasks to train the model using the MAML algorithm, resulting in a trained model.

VII. **Evaluation**:

- Assuming **test_data** is available, the input features of the test set were converted to a PyTorch tensor. The accuracy of the trained model was then calculated on the entire dataset after meta-training, providing insights into the model's performance in disaster management scenarios.

E. Implementation of Few-Shot Learning using Meta-Learning (Prototypical Networks)

I. Library Imports and Data Preprocessing

- Similar to the MAML implementation, the relevant libraries such as **torch, torch.optim.Adam, torch.nn,** and **innerloop_ctx** from **higher** are imported.
- The preprocessing steps, including feature selection, standardisation, and encoding of the target variable, are carried out as in the MAML implementation.

II. PrototypeNetwork Class Definition

- A class named **PrototypeNetwork** is defined to implement prototype-based classification.
- During the **fit** method, prototypes (mean vectors) for each class are calculated using the training data.
- During prediction, the distances between test instances and the prototypes are computed, and the class label of the nearest prototype is assigned to each test instance.

III. Training and Evaluation

- The PrototypeNetwork class is instantiated, and the model is trained on the training data using the **fit** method.
- After training, predictions are made on the test data using the **predict** method
- The accuracy of the predictions is evaluated using the accuracy score from scikit-learn library, providing insights into the effectiveness of the Prototypical Networks for disaster management applications.

V. RESULTS AND DISCUSSION

Results and Discussions:

In this section, we present and discuss the results obtained from employing various few-shot learning methods, including Random Forest, Model-Agnostic Meta-Learning (MAML), and Prototypical Networks, for designing a disaster management system.

Few-shot Learning with Random Forest:

- **Test Accuracy:** When employing the Random Forest algorithm for few-shot learning, we achieve a test accuracy of 97%. This indicates that Random Forest is capable of effectively learning from a small amount of labelled data and generalising well to unseen instances.
- **Validation Accuracy:** Upon introducing a validation set, the model's performance improved significantly, with both validation and test accuracies reaching 99%. This suggests that incorporating a validation set for hyperparameter tuning and model selection can lead to better generalisation performance.
- **Cross-Validation Accuracy:** However, when employing cross-validation using StratifiedKFold, the average accuracy dropped to 90%. This could be attributed to the inherent variability in the data and the limited amount of labelled data available for training the model in each fold.

Few-shot Learning with MAML:

- **Accuracy after MAML:** The Model-Agnostic Meta-Learning (MAML) algorithm yielded an accuracy of approximately 93.52%. MAML's ability to learn task-agnostic initializations enables fast adaptation to new tasks, making it suitable for few-shot learning scenarios.

Few-shot Learning with Prototypical Networks:

- **Accuracy:** The Prototypical Networks achieved an accuracy of around 71.98%. While Prototypical Networks offer a simple yet effective approach for few-shot learning by learning class prototypes, their performance may be limited by the complexity and variability of the underlying data.

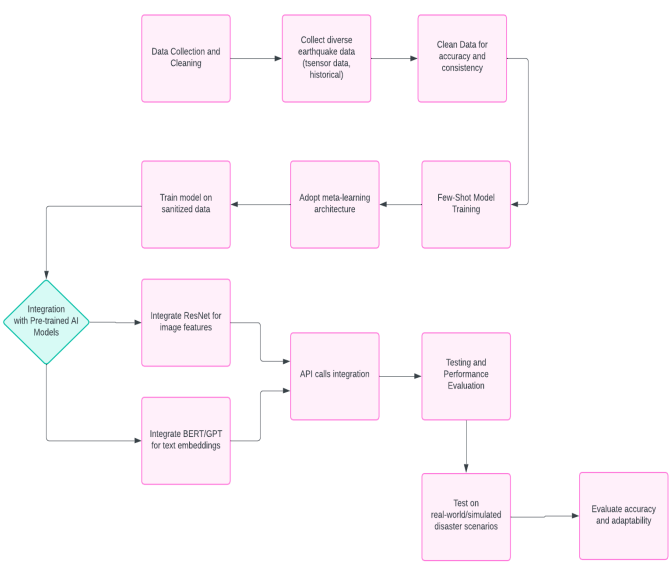
Discussion:

- **Advantages of Random Forest:** Random Forest demonstrates robust performance in few-shot learning tasks, particularly when provided with sufficient labelled data. It offers high accuracy and is relatively insensitive to overfitting, making it suitable for real-world disaster management applications.
- **Advantages of MAML:** MAML excels in scenarios where adapting quickly to new tasks with limited labelled data is crucial. By learning task-agnostic initializations, MAML provides a

flexible framework for few-shot learning across diverse disaster management scenarios.

- **Advantages of Prototypical Networks:** Prototypical Networks offer a simple yet effective approach for few-shot learning by learning class prototypes. While they may not achieve the highest accuracy compared to other methods, their simplicity and interpretability make them valuable in certain contexts.
- **Drawbacks:** Each method has its drawbacks. Random Forest may struggle with capturing complex patterns in the data. The high accuracy on the scores may indicate overfitting of the data, or can indicate skewness of the dataset. MAML's performance heavily relies on the choice of hyperparameters and the availability of sufficient task-specific data, and Prototypical Networks may be limited by the assumptions of prototype-based classification.

Architecture:



VI. CONCLUSION AND FUTURE WORK

Conclusion:

In this study, we explored the application of few-shot learning techniques for designing a disaster management system. Through experimentation with various methods, including Random Forest, Model-Agnostic Meta-Learning (MAML), and Prototypical Networks, we gained insights into their effectiveness in learning from limited labelled data and adapting to new disaster scenarios.

Our results demonstrate that Random Forest exhibits robust performance in few-shot learning tasks, achieving high accuracy when provided with sufficient labelled data. However, the introduction of a validation set for hyperparameter tuning further improved model performance.

Furthermore, Model-Agnostic Meta-Learning (MAML) showed promise in rapidly adapting to new disaster scenarios with limited labelled data, achieving competitive accuracy rates. Prototypical Networks, while simpler in approach, offered a viable alternative for few-shot learning by learning class prototypes.

Future Work:

Moving forward, several avenues for future research and development emerge from this study:

- **Exploration of Hybrid Approaches:** Investigating hybrid approaches that combine the strengths of different few-shot learning methods, such as integrating MAML with Prototypical Networks or leveraging ensemble methods, could lead to improved performance and robustness.
- **Data Augmentation Techniques:** Exploring advanced data augmentation techniques tailored to disaster management scenarios could help enhance the generalisation and robustness of few-shot learning models, particularly in handling diverse and complex disaster-related data.
- **Integration of Domain Knowledge:** Incorporating domain-specific knowledge and expert insights into the few-shot learning process could enhance model interpretability and adaptability to real-world disaster management scenarios.
- **Scalability and Deployment:** Addressing scalability challenges and optimising few-shot learning models for deployment in resource-constrained environments, such as disaster-affected areas with limited infrastructure, remains a critical area for future work.
- **Evaluation on Real-world Datasets:** Conducting extensive evaluations on large-scale, real-world disaster datasets can provide valuable insights into the practical utility and effectiveness of few-shot learning techniques in actual disaster management scenarios.

In summary, the exploration and advancement of few-shot learning techniques hold great potential for revolutionising disaster management practices, enabling more effective and adaptive response strategies in the face of evolving and complex disaster scenarios. Continued research and innovation in this field are essential for realising these transformative benefits and enhancing disaster resilience worldwide.

VII. REFERENCES

[1] Peng et al. Crisis Match Semi-Supervised Few-Shot Learning for Fine-Grained Disaster Tweet Classification: Henry Peng Zhou, Cornelia and Donea Caragea, Yue Zhou - 2023

[2] CCFE: A Few-Shot Learning Model for Earthquake Detection and Phase Identification: Peng Zhao, Shao Yongqian and Xia Shian - 2022

[3] Deep Learning-Based Earthquake Prediction Technique Using Seismic Data: M. Elbes, S. AlZu'bi and T. Kanan - 2023

[4] Real-time Seismic Intensity Prediction using Self-supervised Contrastive GNN for Earthquake Early Warning: R. U. Murshed, K. Noshin, M. A. Zakaria, M. F. Uddin, A. F. M. S. Amin and M. E. Ali - 2024

[5] Enhancing Few-Shot Learning for Tropical Cyclone Severity Prediction: A Deep Learning Approach: Harshal Patil, Snehal Bhosale

[6] Integrative Few-Shot Classification and Segmentation for Landslide Detection Dat Tran-Anh, BAO BUI-QUOC, ANH VU-DUC, Trung-Anh Do,
Hung Nguyen Viet, Hoai-Nam Vu, AND Cong Tran

[7] Koukouraki, Eftychia, Leonardo Vanneschi, and Marco Painho. "Few-shot learning for post-earthquake urban damage detection." Remote Sensing 14, no. 1 (2021): 40.

[8] Hou, James, and Susu Xu. "Near-Real-Time Seismic Human Fatality Information Retrieval from Social Media with Few-Shot Large-Language Models." In Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems, pp. 1141-1147. 2022.

[9] L. Fang et al., “Aftershock Observation and Analysis of the 2013Ms 7.0 Lushan Earthquake,” Seismological Research Letters, vol. 86, no. 4, pp. 1135–1142, Jul. 2015, doi: https://doi.org/10.1785/0220140186.

[10] C. E. Yoon, O. O’Reilly, K. J. Bergen, and G. C. Beroza, “Earthquake detection through computationally efficient similarity search,” Science Advances, vol. 1, no. 11, p. e1501057, Dec. 2015, doi: https://doi.org/10.1126/sciadv.1501057

[11] S. Yuan, J. Liu, S. Wang, T. Wang and P. Shi, "Seismic Waveform Classification and First-Break Picking Using Convolution Neural Networks," in IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 2, pp. 272-276, Feb. 2018, doi: 10.1109/LGRS.2017.2785834.

[12] N. Iqbal, Azzedine Zerguine, SanLinn Kaka, and Abdullatif Al-Shuhail, “Automated SVD filtering of time-frequency distribution for enhancing the SNR of microseismic/microquake events,” Journal of geophysics and engineering, vol. 13, no. 6, pp. 964–973, Nov. 2016, doi: https://doi.org/10.1088/1742-2132/13/6/964.

[13] H. Soto and B. Schurr, "DeepPhasePick: a method for detecting and picking seismic phases from local earthquakes based on highly optimized convolutional and recurrent deep neural networks," in Geophysical Journal International, vol. 227, no. 2, pp. 1268-1294, Aug. 2021, doi: 10.1093/gji/ggab266.

[14] 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, Aug. 2020, doi: https://doi.org/10.1038/s41467-020-17591-w.

[15] W. Fang, L. Fu and H. Li, "Unsupervised CNN Based on Self-Similarity for Seismic Data Denoising," in IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1-5, 2022, Art no. 8022205, doi: 10.1109/LGRS.2021.3131046.