AFRL-RV-PS-
TR-2023-0012

AFRL-RV-PS-
TR-2023-0012

INNOVATIVE DATA SCIENCE APPROACHES TO AUTOMATIC AFTERSHOCK DETECTION

Sheng Zhong and Abdullah Mueen

Department of Computer Science
University of New Mexico
1 University of New Mexico
Albuquerque, NM 87131

28 November 2022

Final Report

AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776

DTIC COPY

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research which is exempt from public affairs security and policy review in accordance with AFI 61-201, paragraph 2.3.5.1. This report is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RV-PS-TR-2023-0012 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNED//	//SIGNED//
1st Lt. Simone M. Smith	Mark E. Roverse, Chief
Program Manager, AFRL/RVB	AFRL Geospace Technologies Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
28-11-2022	Final Report	20 Sep 2021 – 20 Sep 2022

4. TITLE AND SUBTITLE
Innovative Data Science Approaches to Automatic Aftershock Detection

5a. CONTRACT NUMBER
FA9453-21-2-0049

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62212F

6. AUTHOR(S)
Sheng Zhong and Abdullah Mueen

5d. PROJECT NUMBER
2030

5e. TASK NUMBER
EF135630

5f. WORK UNIT NUMBER
V1QG

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Computer Science
University of New Mexico
1 University of New Mexico
Albuquerque, NM 87131

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Space Vehicles Directorate
3550 Aberdeen Avenue SE
Kirtland AFB, NM 87117-5776

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RVBN

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RV-PS-TR-2023-0012

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited (AFRL-2023-0115 dtd 06 Jan 2023).

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The aim of this contract was to explore innovative data mining techniques for automatic aftershock detection in order to intelligently monitor for nuclear explosions. In a nuclear explosion monitoring system, a burst of aftershocks (small earthquakes that occur for days to years following a large earthquake) is uninteresting and could be mislabeled as the target events. Such a burst of uninteresting events overloads the human analysts of the monitoring system. To reduce the load, at the onset of a sequence of events (e.g., aftershocks), a human analyst can label a few of these events and start an online classifier to filter out subsequent uninteresting events.

In this project, we develop an online classification framework for aftershocks. Our specific technique uses a learned model to classify an event as an aftershock and exploits another classifier to decide if the model should be re-trained with the newly detected events. The framework has been tested on two large events: the 7.8Mw earthquake in Gorkha, Nepal on April 2015 and the 8.2Mw earthquake in Chiapas, Mexico on September 2017. We measure that a tolerable false-positive rate of 5% will allow us to automatically remove 83% of the Nepal aftershocks and 90% of the Chiapas aftershocks using the data from only one three-component station. This result shows that the proposed framework can save enough human effort to outweigh the effort needed to carefully judge the false positives. We demonstrate that our method, named FewSig, learns better than existing data science methods from only a few confirmed aftershock instances. We demonstrate that the performance of the framework improves with more confirmed instances and with better-placed stations.

15. SUBJECT TERMS
aftershock sequence, aftershocks, real-time signal classification, few-shot classification

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			1st Lt. Simone M. Smith
Unclassified	Unclassified	Unclassified	Unlimited	50	**19b. TELEPHONE NUMBER** *(include area code),*

This page is intentionally left blank.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

Figure **Page**

LIST OF TABLES

Table **Page**

ACKNOWLEDGMENTS

This material is based on research sponsored by Air Force Research Laboratory under agreement number FA9453-21-2-0049. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

DISCLAIMER

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

This page is intentionally left blank.

Summary: Real-time Signal classification

Seismic monitoring systems sift through a set of seismograms in real time, searching for target events, such as underground explosions. In this monitoring system, a burst of aftershocks (small earthquakes that occur following a large earthquake for days to years) is uninteresting and can be mislabeled as the target events. Such a burst of uninteresting events can overload the human analysts of the monitoring system. To reduce the load, at the onset of a sequence of events (e.g., aftershocks), a human analyst can label a few of these events and start an online classifier to filter out subsequent uninteresting events. We propose an online few-shot classification framework for time series data for the above use case and similar scenarios. Our specific technique uses a Neighborhood Component Analysis (NCA) based model to classify an event as an aftershock and exploits a two-level decision tree to decide if the training set should be extended with the newly detected events. The algorithm demonstrates surprising robustness when tested on seventy univariate datasets from the UEA/UCR archive. Furthermore, we show two case studies where the proposed algorithm can reduces the human effort in monitoring and surveillance applications.

Introduction

In online few-shot classification, an online algorithm learns a classification model from only a small number of positive instances and an arbitrary number of negative instances. The algorithm does not use any unlabeled instances to help with the learning, which is different from PU-learning [24] and other semi-supervised learning approaches [28, 33, 34]. Online few-shot classification is uniquely different from its offline version, which is a popular computer vision method to incorporate a novel class in the model with only a few instances [30]. In the online setting, a human expert labels only the *first few* instances from a stream instead of *a few of the most representative* instances from a large pool of unlabeled instances.

In addition to the small number of training instances, online few-shot learning poses two key challenges. First, the online classification process requires each test instance to be classified before the next instance arrives. This imposes a serious efficiency constraint, challenging computationally expensive algorithms for this task. Second, it must be dtermined whether the newly classified positive instances are classified with sufficiently high confidence that they should be added to the training set before potentially re-training the model.

In order for efficient online processing, we propose a simple two-level framework that identifies *strong* and *weak* positive instances separately and adds only strong ones to the train-ing set. We exploit a pre-computed distance matrix under dynamic time warping (DTW) distance for time series data for efficiency and adopt metric learning under Focal Loss to tackle class imbalance. We develop a decision tree classifier to identify the strong positive instances by bounding the false-positive rate at a maximum, and an NCA-based (Neighborhood Component Analysis) ensemble classifier for the weak instances. We demonstrate that our algorithm significantly outperforms existing semi-supervised algorithms in the online few-shot setting.

Motivation: We consider online few-shot classification for time series data with application to seismic monitoring. Seismic monitoring is an online task essential for national security and public safety. Current seismic monitoring systems are not fully automated and require human analysts to review the information produced by algorithms for safety and security implications. The amount of time an analyst takes to review a block of data (i.e., time series) is largely driven by the number of events in that block and the amount of manual work needed to completely form each event. Large events can take longer to review as they are observed at more stations, and many of these arrivals may not be associated automatically.

For example, prior to the 2011 Tohoku Earthquake and Tsunami event (the strongest earthquake recorded in Japanese history), the Late Event Bulletin (LEB)[1] of the International Data Center (IDC)[1] averaged 120 events per day with approximately 2,000 time-defining associated arrivals recorded on International Monitoring System (IMS) stations. In the immediate aftermath of Tohoku, the LEB contained 830 events per day with approximately 20,000 time-defining associated arrivals. This alone is a 7× to 10× increase in the analyst workload [25]. In addition, the standard STA/LTA [5, 18] detectors become less effective at detecting aftershocks closely spaced in time, requiring the analysts to add more arrivals manually and associate them to these aftershocks.

With an increasing streaming workload, possible mediations are increasing the number of analysts (i.e., resource) and/or delay reporting (i.e., admit vulnerability). A real-time aftershock detector can reduce the workload significantly. However, such a detector poses several computational challenges. First, there is no training data until the main event happens. Note that a historical aftershock sequence helps very little when classifying a new aftershock sequence because their origins are rarely the same as the ongoing shocks. Therefore, we must use only a few training instances of the positive class. Second, the detector must work in real-time to reduce analyst workload as well as to improve itself by learning from recent observations.

The main contributions of this work are summarized below:

- We develop a novel online few-shot time series classifier (FewSig), that can be trained on a few positive signals, in the absence of any unlabeled signals, and adapt to the new positive instances iteratively.
- We evaluate FewSig on 68 datasets and compare them with online versions of semi-supervised time series classification algorithms. The comprehensive evaluation details parameter sensitivity, efficiency, and effectiveness of the proposed framework.
- We demonstrate two case studies where FewSig detects events (aftershocks and abnormal gaits) with a few positive signals in the training data for monitoring and surveillance applications.

[1] www.ctbto.org

Methods, Assumptions, and Procedures

3.1 Related Work

Time Series Classification: Using raw time series data as features for predictive models has some challenges for conventional algorithms, such as taking into account the order of the features and handling instances with different lengths, which leads to a variable number of features for training instances [4]. Thus, embedding the time series into a new space allows using conventional models (e.g., decision trees or support vector machines) for time series data. Such an embedding can be categorized into three approaches [35]: model-based, feature-based, and distance-based.

Model-based algorithms, such as Hidden Markov Models, fit a generative model to each series and then measure similarity between series using similarity between models [8]. However, recent experimental evaluations show that these models are less competitive for time series classification [7]. Feature-based algorithms extract a finite set of features (e.g., statistical properties) from the raw time series to train a predictive model. The state-of-the-art feature-based models such as Hive-COTE-V2[21] show superior performance on time series classification tasks. However, the training time for such a model, as well as the time to calculate the features, makes it impossible to adapt to the online setting. In addition, the performance of these models when trained with only a few positive instances remains unexplored. Finally, distance-based algorithms compare the similarity between unlabeled and labeled series in the classification task using a distance measure such as Euclidean distance or Dynamic Time Warping. Examples of distance-based methods are FDTW [15], DISTF-Tuned [11] and DFDTW_PCA [13].

Our proposed framework is distance-based, balancing between the training efficiency of model-based and classification accuracy of feature-based methods for online few-shot classification.

Semi-Supervised Learning on Time Series: Semi-supervised learning (SSL) meth-ods have been proposed to avoid poorly generalizable models due to the insufficient amount of labeled data to train a predictive model. Wei's Algorithm [33] and DTWD [27] are examples of self-training methods, a well-known semi-supervised learning approach. Souza et al. [29] perform clustering to select the most representative instances from an unlabeled dataset to be labeled by an expert and then performs label propagation to classify the remaining instances.

Nguyen et al. [24] proposed a solution based on clustering and self-training, considering only positive and unlabeled instances. Marussy et al. proposed SUCCESS [19], a model based on constrained hierarchical clustering. Xu et al. proposed [36] based on the graph-theoretic SSL algorithm. SSSL [31] performs self-training with shapelet classification on unlabeled instances. Several state-of-the-art deep-learning models have shown dominance in recent works. For example, Jawed et al. proposed a multi-task learning network (MTL) [14] to jointly train the ConvNet with classification and forecasting by sharing latent representations. SemiTime [10] shows better results than MTL on some datasets, the model learns past-future temporal relations from the unlabeled data, and the backbone feature extractor

is shared with the TSC module that is trained on the labeled dataset. SSTSC [34] increases the richness of the temporal context by splitting a time series into past-anchor-future, making the model learn higher-quality semantic context from the unlabeled dataset.

Even with good results in the semi-supervised scenario, it is worth mentioning that none of these works are adequate for the online few-shot learning setting proposed in this work, in which an unlabeled dataset does not exist to help learn the initial model.

Few-Shot Learning: Due to privacy, safety, ethical issues, high costs for manual analysis, or the nature of the problem (e.g., rare natural events), many applications require training a supervised model from a few labeled instances per class (e.g., five labeled instances). The paradigm that deals with this restriction is few-shot learning [32]. A typical task is object detection in images. In this problem, it is challenging to have a significant number of labeled instances for many classes or even know all classes. In general, existing few-shot learning methods are proposed to tackle image or text data, while time series are still little explored for this learning. For example, [23] considers a meta-learning approach with varying classes using a few labeled instances per class. In contrast, FewSig trains *iteratively* using a few initial positive instances, with no particular assumption about the number of negative instances.

3.2 Background and Notation

In this section, we first formulate the problem and then introduce the technical framework of FewSig. We summarize the notation and abbreviations in Table 1.

We define the online few-shot classification as follows: initially, a labeled set of instances, $L = \{t^1, t^2, ..., t^t\}$, is available with labels $\{y^1, y^2, ..., y^t\}$, which contains only few positive instances. All the instances starting from t^{t+1} belong to O and their true labels are unavailable. New instances are presented in time order, the model needs to classify each instance in O at the time when it appears, and the predicted label $\hat{y}^t$ is immutable. The model is allowed to utilize all the historical instances before t^j to classify t^j.

3.3 FewSig: Online Few-shot Time Series Classification

We propose a general framework for this problem setting, shown in Figure 1. There are three datasets: L, O, T, where $L \subseteq T$ and $T \subseteq L \cup O$. There are two models: a *selective* model to identify the strong positive instances and a *general* classifier to identify the weak positive instances that will not be included in T. Initially, $T = L$ and both models are trained on T, then for each new instance $t^j \in O$, the selective model decides whether to add t^j into T. T will be updated to $T = T \cup \{t^j, \hat{y}^j\}$ if t^j is selected, both models will be retrained on the updated T and finally $\hat{y}^j = c_p$. The general classifier will give the $\hat{y}^j$ if t^j is not selected by the selective model.

Table 1: Symbols and notation

Symbol	Definition
t^i	ith time series/instance
t^i_j	jth observation value from t^i
$t^i_{s,m}$	A sub-sequence of length m from t^i that contains $[t^i_s, t^i_{s+1}, \ldots, t^i_{s+m-1}]$
y^i	True label for t^i
$\hat{y}^i$	Predicted label for t^i
L	Labeled set that contains $\{t^1, t^2, \ldots, t^l\}$
O	Online testing set that contains $\{t^{t+1}, t^{t+2}, \ldots, t^\infty\}$
T	Training sets with size N for training the model
D_T	Dissimilarity matrix for each instance in T
c_p	Positive class
c_n	Negative class
C^p	All instances that belong to class c_p, $\{t^i \mid t^i \in T \& y^i = c_p\}$
$d_i^{k,p}$	Average distance between t^i and k nearest neighbor from class c_p
ATDT	Auto-Tuning Decision Tree
NCFAE	Neighborhood Components Focal Analysis Ensemble
tFPR	Target FPR for ATDT

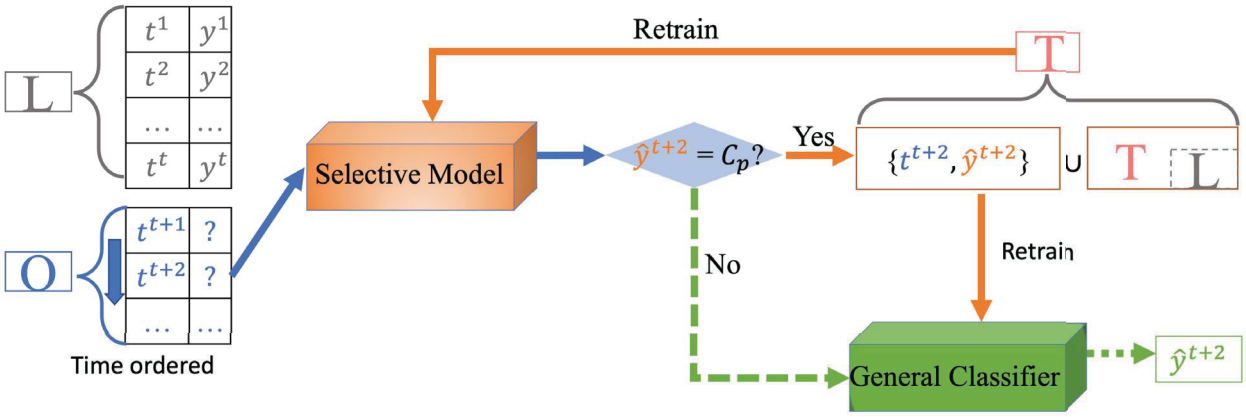

Figure 1: Online classification with self-training

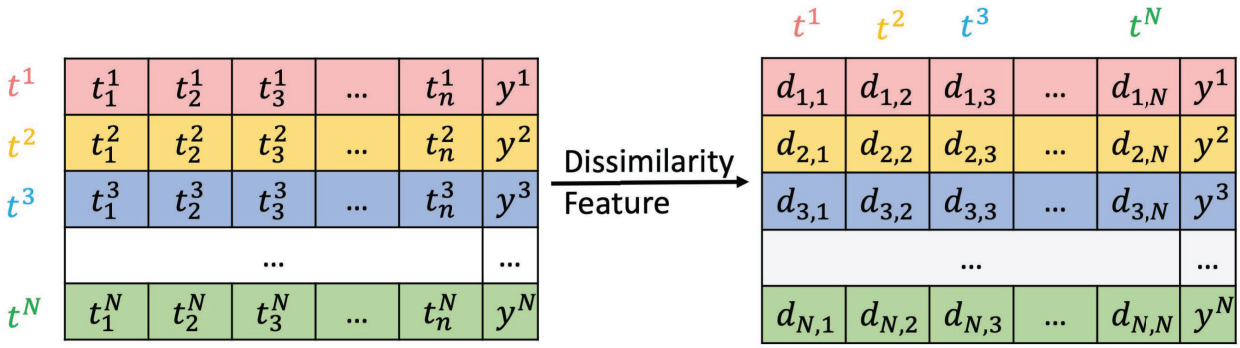

Figure 2: Conversion process of N time series with n observations into a dissimilarity space of $N \times N$, N is the size of the training set T, $N = \mathrm{IR}^T$

Dissimilarity as Features

We represent each time series t^i as a vector of dissimilarity to other time series t^j. This process is shown in Figure 2, and we use D_T to represent the dissimilarity matrix for instances in T. The order of elements in the new feature vector is flexible as long as it is consistent across all D_T.

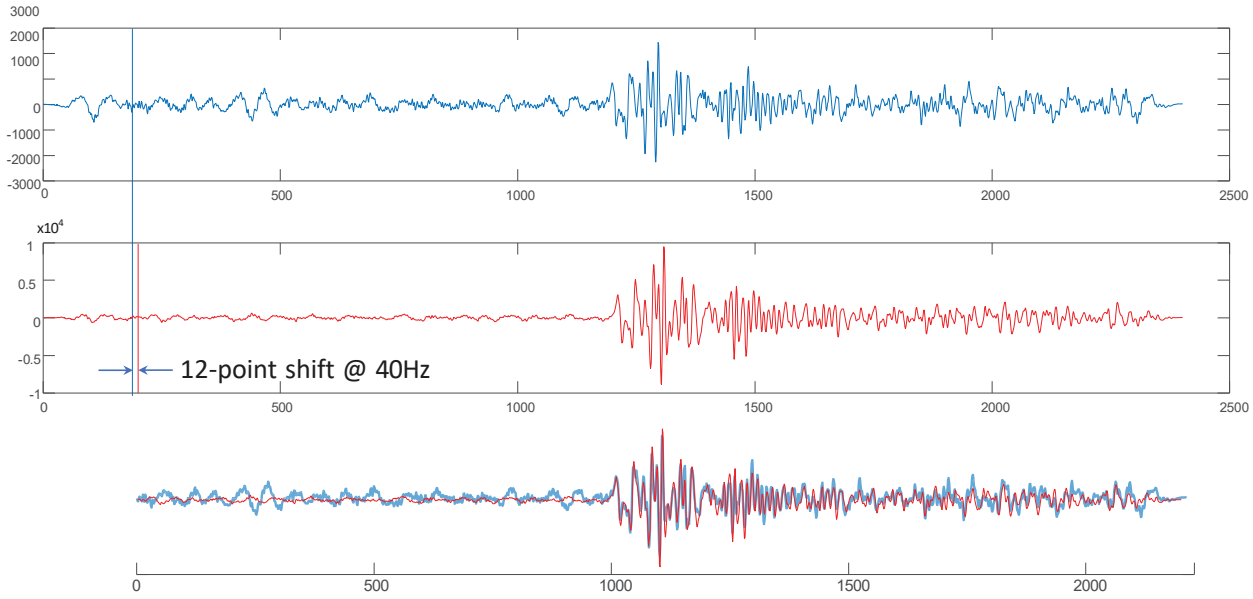

Figure 3: Two real aftershock signals show the highest correlation of 89% when one is shifted in time relative to the other to correct for human error in picking, and 8% correlation if not shifted

We applied the time series sub-sequence similarity search under DTW distance to calculate $d_{i,j}$ [26]. Dissimilarity matrices under sliding DTW have rarely been utilized in the literature, even though the Euclidean distance matrix has been used as a feature [17][4] and sliding Euclidean distances are shown to be effective to correct for errors in alignment [22].

We use sliding DTW distances to address two sources of errors in the time series. The first is due to the error in event alignment. For example, human analysts often pick the onset

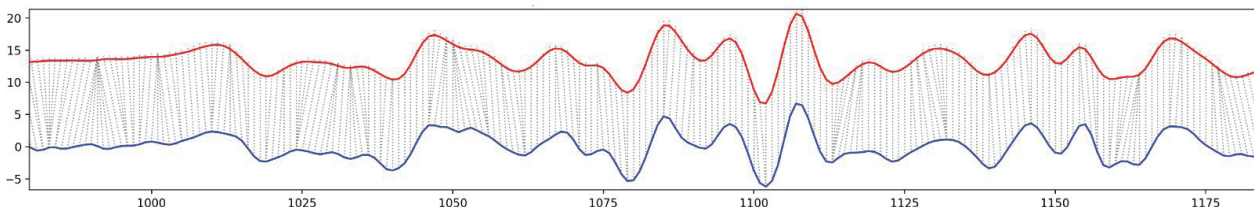

Figure 4: The DTW alignment between two aftershock waveforms (same as in Figure 3). *The blue signal exhibits warping due to the variation in wave propagation from the event origin to the seismometer*

time of an arrival in seismograms, and any error in picking can alter results dramatically, as shown in Figure 3. The second error is due to the inherent complexity of the system being monitored. For example, the non-uniformity in the earth changes the propagation of the waves from neighboring events. Figure 4 shows how DTW deals with warping by allowing one to many alignments to correct tiny variations in the time series.

Finally, we formulate the distance $d_{i,j}$ in Equation 1 and 2, in which a sub-sequence from one time series is taken as the query to search in another time series and vice visa, and the $d_{i,j}$ will be the minimum distance. r is Sakoe-Chiba Band, q, m defines the query sub-sequence, and s_1, s_2 defines the searching range.

$$d_{i,j} = min\{SS(t^i, t^j_{q,m}), SS(t^j, t^i_{q,m})\} \tag{1}$$

$$SS = min\{DTW(t^i_{k,m}, t^j_{q,m}, r), s_1 \le k \le s_2\} \tag{2}$$

To find the best parameter settings, we perform a grid search with some candidate values and calculate the loss defined in Equation 4 on the L to select an optimal parameter configuration. The average distance between t^i and k nearest neighbors from class c_p is notated as $d_i^{k,p}$ defined in Equation 3. To handle the bias of different query lengths m, each distance is normalized by m when computing the loss. The optimal parameter configuration $\{q^*, m^*, s_1^*, s_2^*, r^*\}$ remains constant during the online evaluation.

$$d_i^{k,p} = \frac{\sum_{j=1}^{k} d_{i,j}}{k}, j = i \;\&\; d_{i,j} \le d_{i,j+1} \;\&\; t^j \in C^p \tag{3}$$

$$\text{DistLoss} = \sum_{t_i \in L} \begin{cases} min(0, d_i^{1,c^p} - d_i^{1,c^n}) & \text{if } t_i \in C^p \\ min(0, d_i^{1,c^n} - d_i^{1,c^p}) & \text{if } t_i \in C^n \end{cases} \tag{4}$$

When T is updated by adding instances from O, the D_t also needs to be updated, this process is effortless since the distance $d_{i,j}$, $t^i \in O$, $t^j \in T$ is already computed at the inference phase and the dissimilarity matrix is diagonally symmetric, $d_{i,j} = d_{j,i}$.

ATDT as selective model

The selective model must have a very low FPR since any false-positive instance will affect the performance of the entire framework in the long term. Therefore, we propose a two-level

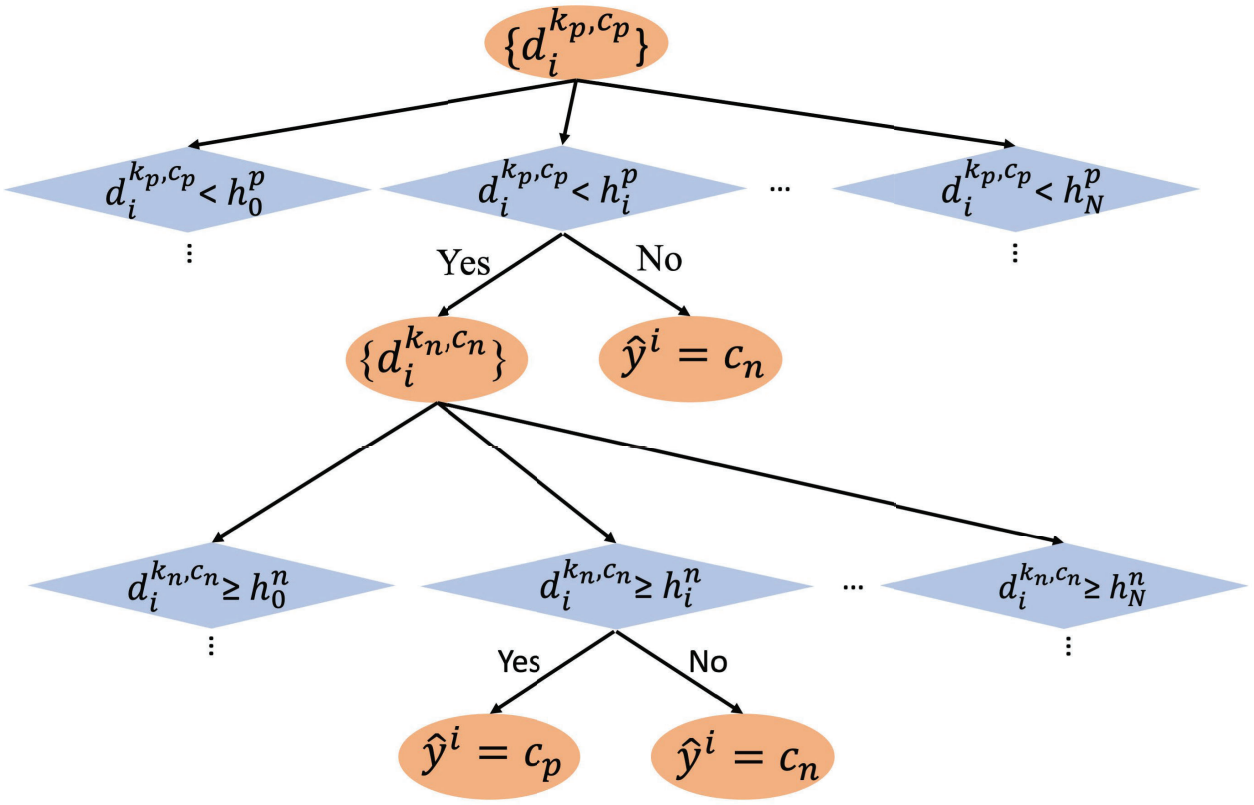

Figure 5: Two-level structure for our proposed auto-tuning decision tree model

auto-tuning decision tree model as shown in Figure 5 as the selective model. This model is small to be interpretable and has a tunable parameter for the target false-positive rate.

The fundamental principle of this model is to select the positive instance that is similar to the positive class, C^p, meanwhile dissimilar to the negative class, C^n. However, such a boundary is hard to judge due to intra-distance and inter-distance overlap in most cases. We introduce a parameter target-FPR (tFPR) to tune the boundary to the desired amount.

The training process can be summarized as finding two thresholds h_p^*, h_n^* such that the final FPR is close to *tFPR* while TPR is highest when classifying the entire training set T. An instance t^i will be classified as positive only when $d_i^{k_p,c_p} < h_p^*$ **and** $d_i^{k_n,c_n} \geq h_n^*$, otherwise the instance yields a negative label. We build the tree with all possible combinations of h_p, h_n. The values for split nodes on the first level are $H_p = \{h_i^p, 0 \leq i \leq \text{IR}^T\}$ which is calculated based on Equation 5, where $ds_i^{k_p,c_p}$ is the ith smallest value in the sorted $d_i^{k_p,c_p}$ array, δ is a very small number, and $N = \text{IR}^T$.

$$h_i^p = \begin{cases} ds_0^{k_p,c_p} - \delta & i == 0 \\ (ds_{i+1}^{k_p,c_p} + ds_i^{k_p,c_p})/2 & 1 \leq i < N \\ ds_{N-1}^{k_p,c_p} + \delta & i == N \end{cases} \quad (5)$$

$H_n = \{h_i^n, 0 \leq i \leq N\}$ can use the same equation by replacing c_p with c_n. k_p^* and k_n^* are decided by performing a *Leave-One-Out* evaluation on the training set. Equation 6

summarizes the whole training process.

$$k_p^*, k_n^*, h_p^*, h_n^* = \underset{k_p,k_n,H_p,H_n}{\arg\max} \; TPR(1 - |FPR - tFPR|) \tag{6}$$

NCFAE as A General Classifier

Considering the challenges in the online classification tasks, we propose an NCFAE model as the general classifier which is based on a K-Nearest Neighbor (K-NN) classifier with the neighborhood component analysis (NCA) technique. The reason for selecting the K-NN as the base model instead of a deep-learned method is the lack of training data. The existing approach of utilizing the K-NN for time series classification is simply to fit the test instance to the model, and the label will be assigned based on the majority class within a certain radius [33]. Other approaches [15] use the full distance matrix to fit the model. However, Jain et al. [13] demonstrated that removing less significant and repeated time series can improve the computational efficiency and the classification performance. We consider removing less significant dimensions and boosting significant dimensions by performing an alternative to the Principal Component Analysis (PCA) applied in [13].

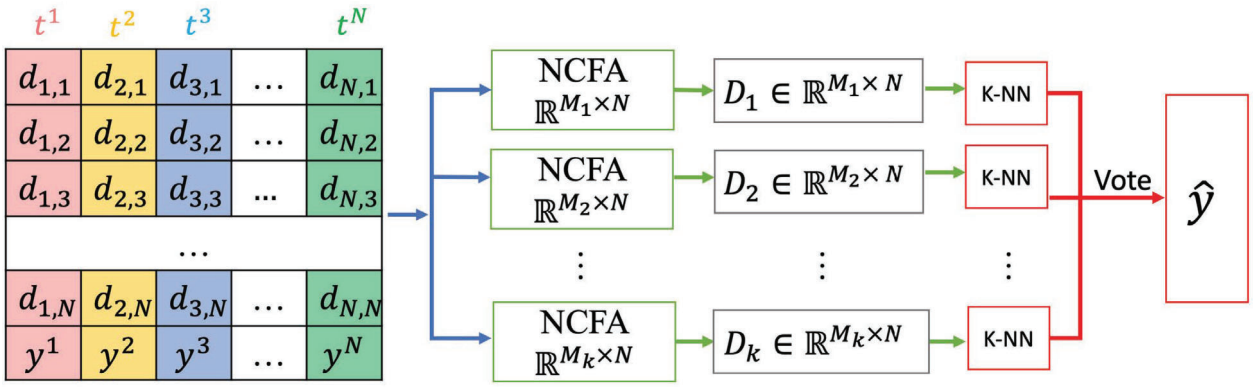

Figure 6: Structure of NCFAE classifier, $\boldsymbol{N = \mathbb{R}^T}$

Neighborhood Components Analysis (NCA) [12] natively supports dimension reduction and can improve classification performance. As shown in the original work, NCA presents better low-dimension visualization than PCA and improved classification results when using K-NN. NCA learns a linear transformation A of the input such that *K-NN* performs well under this transformed space. In our case, the NCA is applied on the D_T before fitting to the K-NN. This process is demonstrated in Figure 6.

$$FL(p_t) = -a(1 - p_t)^{\gamma} \log p_t \tag{7}$$

We replace the Cross-Entropy loss used in the original NCA with the Focal-Loss defined in Equation 7 to address the class imbalance in few-shot classification problems. The Focal-loss was first introduced in [16] for dense object detection where an extreme foreground-background class is an issue. In Equation 7, p_t is the probability that an instance is correctly classified. $a \in [0, 1]$ is a balancing factor for addressing the class imbalance. γ is the focusing

parameter that can make learning more focused on the hard misclassified positive instances rather than numerous simple negative instances.

Instead of using only one matrix $A \in \mathrm{IR}^{M \times N}$, $N = \mathrm{IR}^{T}$, and M is the reduced feature dimension, we learn multiple A with various dimensions $\overline{M} = \{M_1, M_2, \ldots, M_k\}$. $\overline{M}$ can be estimated based on the best *Cross Validation* performance on the training set or based on a certain level of randomness. For this work, we applied the same approach described in [13] that can be expressed in Equation 8.

$$\overline{M} = \{a \mid \frac{N}{20}\rfloor, a \in \{1, 2, 3, \ldots, 19\}\} \tag{8}$$

Each matrix A is learned separately, each transformed space will have an independent K-NN classifier, and the final predicted label $y^{\wedge i}$ is voted for among 19 K-NN classifiers. $vote = v$ means the $y^{\wedge i} = c_p$ as long as there are v K-NN classifiers classifying t^i as c_p.

Results and Discussion

4.1 Experimental Evaluation

All our experiments are reproducible, and the source code, data, and additional results are available on our supporting website [3]. First, We compare FewSig with four semi-supervised models adapted for the online few-shot classification setting on 68 datasets from the UEA/ UCR Time Series Classification Repository covering various domains. Then we show the performance of FewSig for aftershock classification. We performed all the experi-ments on an AMD EPYC 7402 server (24 cores) with a 4xRTX3090 GPU and 128GB RAM.

Models for Comparison

We select three traditional semi-supervised time series classification models: Wei's model [33], DTWD [9], SUCCESS [19], and one state-of-the-art deep learning model SSTSC [34]. Next, we briefly describe the original algorithms and how we modified them to work in the online setting.

Wei's model uses a one-nearest-neighbor (1-NN) classifier as a base model. Initially, L contains only a few labeled positive instances and $T = L$. To expand T, the algorithm iteratively selects the most confident instances from the unlabeled set accompanied by a pseudo-label c_p. The confidence is measured by the Euclidean distance of the selected instance to all the instances in T. When iteration stops, all instances in T are considered as c_p, and the remaining instances in the unlabeled set are c_n. We take the following procedures to adapt the model for the online setting: 1) The model starts with an L that has both c_p and c_n. 2) Initially $T = L$, the 1-NN model will be retrained every time T updates. 3) For each instance $t^i \in O$, the 1-NN will first classify with label $\hat{y}^j$, if $\hat{y}^j = c_p$ then t^i will be added to T, then all the previous $t^i \in O$ that are not in T will be reconsidered to determine whether to put them in T. The selection condition is simply whether the nearest neighbor from T is c_p. As in the offline setting, at most one will be added to T during each iteration. The iteration will stop when all the rest of the instances in O have NN belonging to c_n.

DTWD considers a similar approach for augmenting T via self-training on an unlabeled set. However, it employs a new distance measure and a one-class classifier. The distance measure is the ratio of DTW distance to Euclidean distance. The one-class classifier relies on the entire unlabeled set to pick the k value, however, we do not have access to such

information in the online scenario. Thus, we consider Wei's approach with the distance measure employed by DTWD. **SUCCESS** is based on the constrained single-linkage hierarchical agglomerative clustering algorithm with DTW distance. One constraint when linking the instances is instances from L can not be linked. The final label of a cluster is decided by the majority class of instances in L. The unlabeled instances will be labeled by the cluster label. Then final 1-NN classifier will be trained for testing. To work in the online setting, we adapt SUCCESS to classify each $t^j \in O$, then rebuild the cluster with L and the existing O, and finally retrain the 1-NN classifier.

SSTSC learns an encoder that can capture the temporal context based on the unlabeled dataset in a self-supervised manner. This encoder will be used as a backbone for the supervised TSC module that is trained on the labeled dataset L. We consider the same approach employed for SUCCESS to adapt SSTSC for the online setting.

Online Experimental Protocols

For each dataset, we re-assign the labels of the instances to positive and negative labels. We label the instances from the minority class as positive. For balanced datasets, we randomly chose a class as c_p. In both cases, the instances of the remaining classes are considered as c_n. We randomly split each dataset into L used for the initial training and online testing set O. Both L and O have the same number of negative instances, but L contains 5 positive instances, while the remaining positive instances are in O. The instances in O are ordered and fed to the model sequentially to simulate real-world online scenarios. At time t, the model needs to classify the current instance t^t from O. The true label y^t is hidden from the model.

Experimental Results

We select 68 out of 128 univariate time series datasets from the UEA/UCR repository [6]. The selection is based on the size of a dataset, and the threshold is 800 since we are focusing on a problem with limited data. We perform 30 trials with random shuffling of each dataset and verify that none of the trials share more than two positive instances in L. The L, O are consistent when evaluating different models.

There is no parameter configuration for Wei's model, DTWD, and SUCCESS. For SSTSC, we use the same configuration described in the original paper. We use the following settings for FewSig across all datasets: tFPR $= 1\%$ for ATDT, $a = 0.5, \gamma = 2$ for Focal loss, and SDG with $lr = 0.02$, $k = 1$ for k-NN in NCFAE, and the final F1 score is averaged for vote=$\{1, 2, 3, 4\}$.

Ranking Comparison

We conduct the Friedman test and Nemenyi post-hoc test on all 68 datasets across five models. The ranking is computed with the mean F1 scores of 30 trials per dataset. We show the critical difference diagram in Figure 7 in which FewSig shows statistical differences in the ranking compared to the rival models. The complete results for each dataset are available on the supporting webpage.

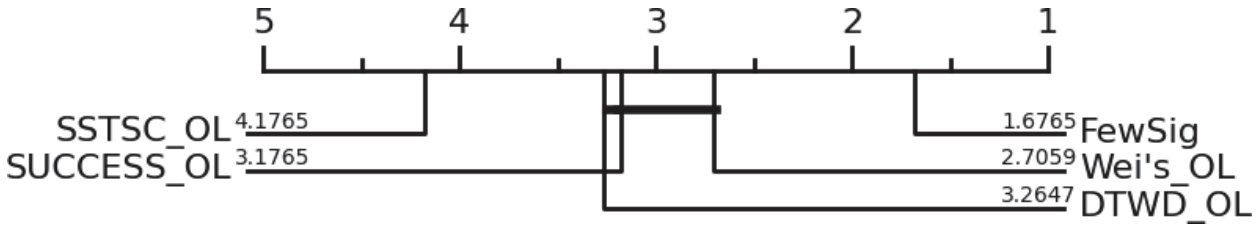

Figure 7: Critical difference diagram of 5 models on the 68 UEA&UCR benchmark datasets

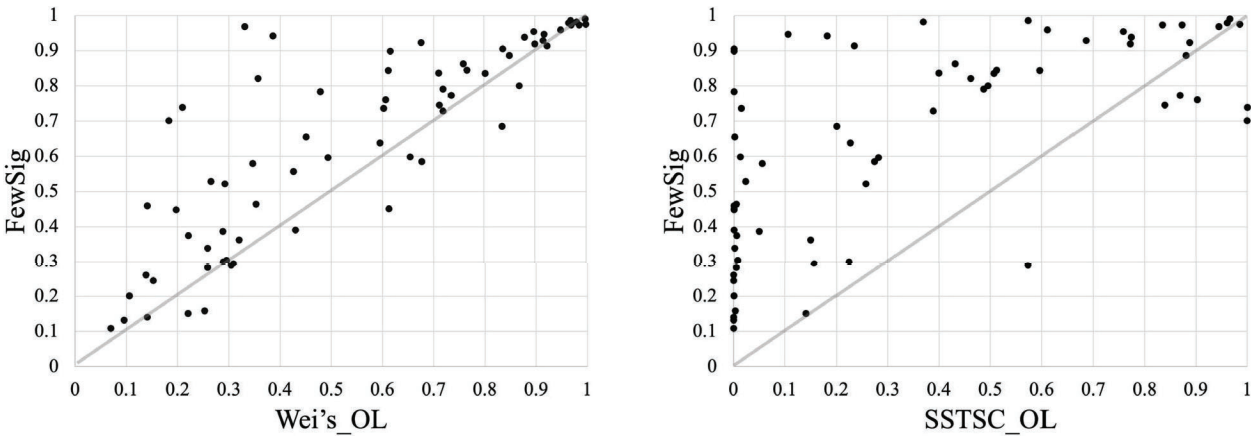

Figure 8: F1 scores of 68 datasets comparison between FewSig and Wei's model (left) and SSTSC (right)

In Figure 8, we compare the FewSig with Wei's model and SSTSC with respect to the F1 of 68 datasets. FewSig has a higher F1 for most of the datasets in both cases.

Table 2: Average running time in seconds. *k* is the grid search round for finding optimal hyperparameters

Model	Initial time		Ave. Time per t_i	
	Distance	Train	Distance	Train+Infer
Wei's	1.49	0.0004	0.019	0.006
DTWD	5.01	0.0005	0.049	0.006
SUCCESS	3.517	0.0003	0.041	0.050
SSTSC	0	26.78	0	36.68
FewSig	8.571*k	14.46	0.057	0.79

We present the average running time of 68 datasets for each model in Table 2. We consider the time of initial training on L, the time for inferring each instance in O, and the time for model retraining. Since the model does not need to be retrained on each instance $t_i \in O$ for Wei's model, DTWD, and FewSig, the time is amortized on the entire testing set. FewSig takes several minutes to initiate and processes roughly one event per second. Most reference methods are faster than FewSig, however, the amortized time taken per event by FewSig is enough for our target domains with a statistically significant gain in accuracy.

Parameter Sensitivity Test

In this section, we discuss the FewSig sensitivity to three design parameters: i) different vote numbers for NCFAE, ii) tFPR for ATDT, iii) the initial positive instances in *L*.

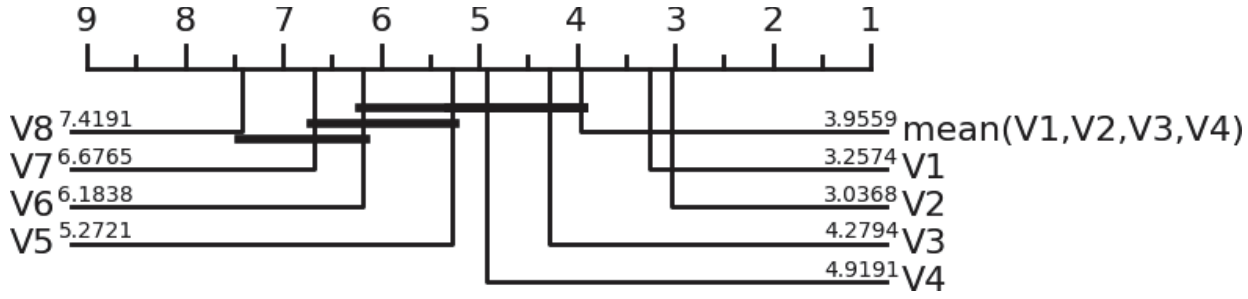

Figure 9: Critical difference diagram of FewSig with the different number of votes on 68 datasets

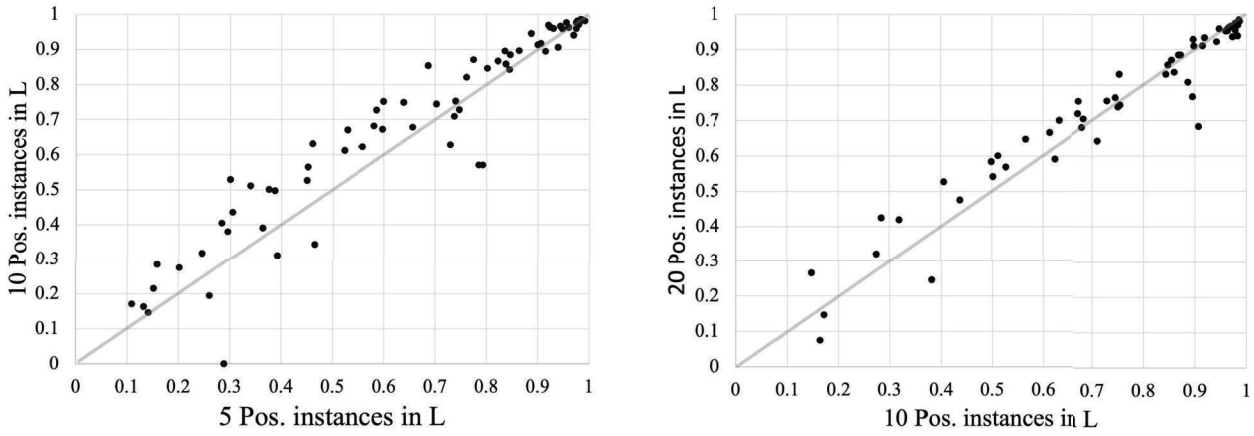

Figure 10: F1 score comparison with the different numbers of positive instances in *L* on all 68 datasets.

Figure 9 shows the F1-score ranking of FewSig with the different number of votes for the ensemble. If two classifiers agree on a positive instance, it gives the best overall results across all 68 datasets. In Figure 10 we show the F1 comparison with the different number of positive instances in *L*. The figure shows that doubling the positive instances from 5 to 10 has more impact than doubling from 10 to 20. The complete results for all 68 datasets and ranking comparison with other models are available on our supporting website[3].

4.2 2015 Mw 7.8 Nepal (Gorkha) Earthquake Aftershock Sequence

Data Preparation

In this section, we utilize FewSig to perform online classification on the aftershock sequence of the 2015 M_w 7.8 Nepal (Gorkha) earthquake. We use a highly calibrated aftershock

sequence catalog [20] as the ground truth. This catalog uses multiple-event hypocenter relocation analysis by local seismic stations and a geodetic rupture model based on InSAR and GPS data. All the ground truth aftershock events have magnitudes greater than M2.0. We select the non-aftershock events from distinct geographical regions and we only select those that share similar origin-to-station distances as the aftershock events. Figure 13 shows the selected non-aftershock origins when we experiment on the MKAR station. Information about the non-aftershock events is collected from the Late Event Bulletin (LEB) [1]. Note that our database contains records from 2009 to 2018, hence, the non-aftershock events happened in this period of time. The arrivals are selected at a specific station, hence, the experimental results are also specific to a station. In the following sections, we first discuss the general arrival selection process, and we demonstrate the results subsequently.

Algorithm 1: Ground truth aftershock and LEB events association

```
   Function Association(d, δ_t)
        //d is user defined distance threshold.
        //δ_t is user-defined time difference threshold.
1       GT_ori ← all the events from ground truth list
2       LEB_ori ← all the events from the LEB database
3       results ← {}
        for each e ∈ GT_ori do
4           tmp ← {}
5           for each e' ∈ LEB_ori do
                if great-circle-distance(e, e') < d then
6                   tmp ← tmp ∪ e'
                end
            end
            //Select a temporally closest one from tmp as the association.
7           bsf_t ← δ_t
8           assoc_e ← null
            for each e' ∈ tmp do
9               dt ← abs(time(e) − time(e'))
                if dt ≤ bsf_t then
10                  assoc_e ← e'
11                  bsf_t ← dt
                end
            end
12          results ← results ∪ assoc_e
        end
13      return results
    end
```

To retrieve the waveform of an event at a station, we need the arrival time. The arrival times for the ground truth aftershock events at a station are identified by associating the location of the ground truth events with the origins in the LEB database. Two events can be considered identical when they are both temporally and spatially close to each other.

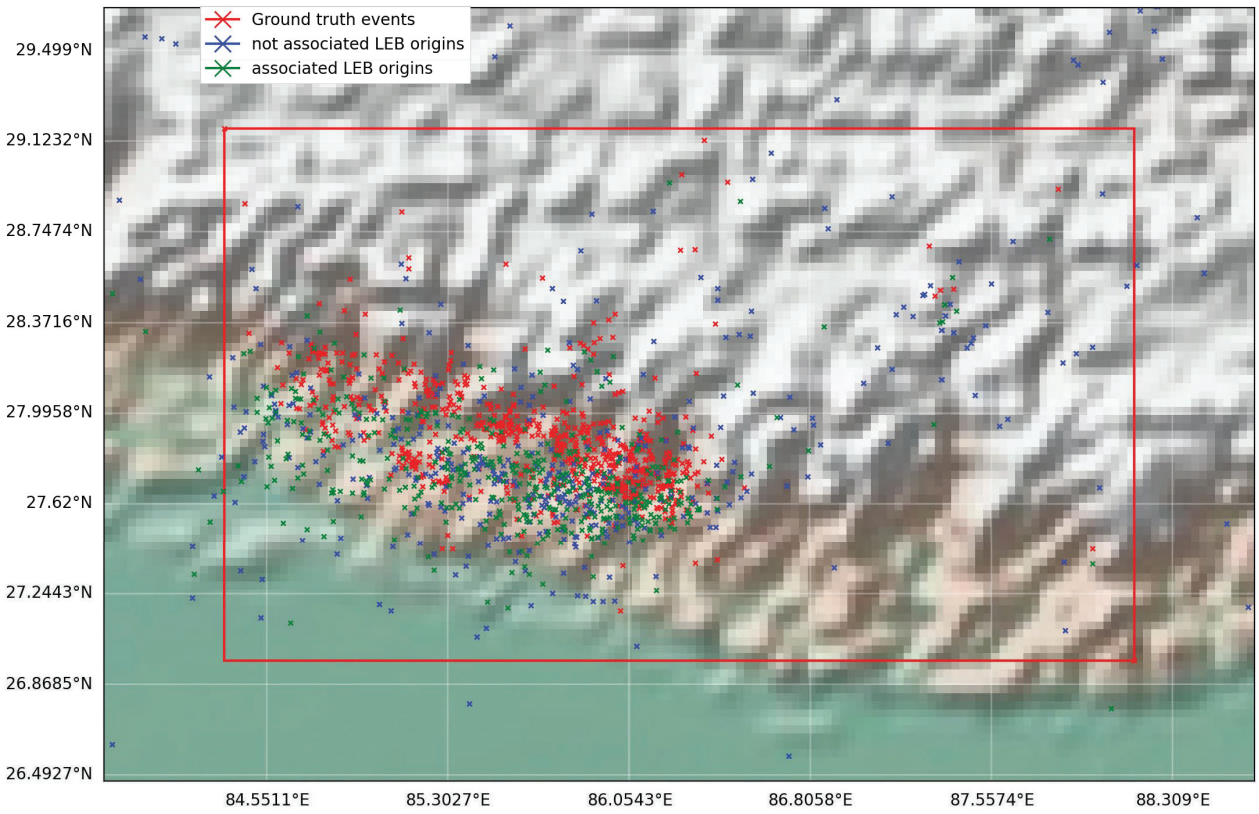

Figure 11: The origin distribution of events from ground truth and the LEB bulletin. ***The events from the LEB bulletin are selected by limiting the origins to the red rectangular region and limiting the origin time between 2015-04-25T06:11:24.290000Z and 2016-05- 14T22:45:53.330000Z according to the ground truth. We applied d = 200KM and δ_t = 5Sec. as the association threshold***

Table 3: Top 5 IMS stations with the most valid aftershock arrivals

Station	number of valid arrivals
MKAR	217
KURK	95
ZALV	72
NRIK	69
AAK	57

Algorithm 1 describes this association process. The distribution of ground truth events and the LEB events are shown in Figure 11. The associated arrivals in the LEB database are demonstrated in Figure 12. Note that the actual number of arrivals used for the experiments is less than the number of green dots shown in this map due to the lack of availability of waveform data and further selection criteria - 1. Only P phase arrival. 2. Waveform Signal to Noise Ratio (SNR) no less than two. The number of valid aftershock arrivals are listed in Table 3.

The waveform data are retrieved from the IMS database. For both aftershocks and non-

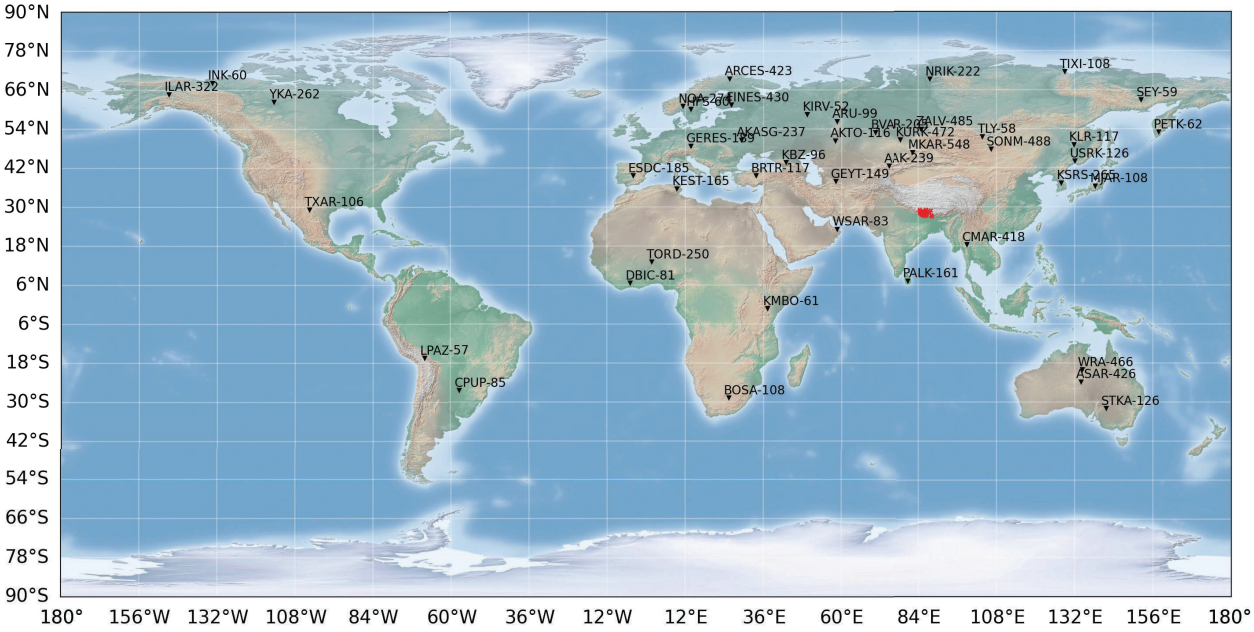

Figure 12: Stations and the corresponding number of arrivals in the LEB for the associated events. *Only the stations with more than 50 arrivals are displayed*

aftershocks, we select only P phase arrivals and extract three 60-second windows from the continuous waveforms of three broadband channels BHZ (vertical), BHN (north-south), BHE (east-west). Thus, the time series contains 30 seconds of pre-arrival and 30 seconds of the post-arrival signal. This time window is large enough to capture the initial compressional seismic waves generated by any regional earthquake. If the waveforms are sampled at 40Hz, the length of each waveform is 2,400 real numbers.

Following conventional seismic signal preprocessing techniques, we remove the trend of each signal, remove the mean and taper the waveform before filtering. To filter out the noise, we applied a 0.4Hz to 10Hz second-order Butterworth bandpass filter in both directions to cancel the phase shift. Next, we compute the SNR on the filtered waveforms. SNR is the ratio of the standard deviation of the signal part (post-30 seconds) over the noise part (pre-30 seconds). For the following experiments, we use waveforms that have SNR ≥ 2.

To simulate an online aftershock detection system, all the instances in both L and O are time ordered based on the event time. The training set L contains the first 5 aftershock arrivals and all the historical non-aftershock arrivals with labels available; the remaining instances are in the online testing set O without any knowledge of their labels.

The dissimilarity matrix described in section 1.3.3 is defined for signals from one channel, however, we are using waveforms from 3 different channels of a station. To generate a dissimilarity matrix for multiple channels, we change Equation 2 to Equation 9, i_z, i_h, i_e represent waveforms from Z, N, and E channels for arrival i. a_z, a_n, a_e are the peak values of the corresponding waveforms as described in Equation 10. Equation 1 remains the same.

$$SS = min\{a_z DTW(t^z_{k,m}, t^{j_z}_{q,m}, r) + a_n DTW(t^n_{k,m}, t^{j_n}_{q,m}, r) + a_e DTW(t^e_{k,m}, t^{j_e}_{q,m}, r), s_1 \leq k \leq s_2\} \quad (9)$$

$$\alpha_h^{ii} = \frac{mmmmmm\blacksquare|tt^{iih}|\blacksquare}{mmmmmm(|tt^{iizz}|) + mmmmmm(|tt^{iinn}|) + mmmmmm(|tt^{iiee}|)}, h \in \{zz, nn, ee\} \quad (10)$$

Table 4: Optimal parameters selected by the loss function defined in Equation 4 for computing the DTW distances on each station, Q25 indicates the query signal starts at arrival time minus 25 seconds and ends at arrival time plus 25 seconds

Earthquake, Station	parameters for equation 9
Nepal, MKAR	Q25-S26-W1
Nepal, KURK	Q25-S26-W1
Nepal, ZALV	Q25-S26-W8
Chiapas, TXAR	Q25-S26-W1
Chiapas, ROSC	Q25-S26-W1
Chiapas, CMIG	Q25-S26-W5

S26 indicates the search range starts at arrival time minus 26 seconds and ends at arrival time plus 26 seconds. W1 is the warping band of one timestamp. For this experiment, we fix the query time series as Q25 and vary the search range of three different values S25, S26, then we compute the loss of Equation 4 for each search range with ten different warping bands from W1 to W10. Finally, we apply the loss function to select the combination that gives the minimum loss.

Parameters for FewSig are constant for all the experiments: tFPR $= 0.5\%$, Vote $= 2, k = 1$, learning rate(lr) $= 0.02$, epoch $= 250, \gamma = 2, a = 0.5$. The warping banding for computing the DTW varies on each station, we report the exact number in Table 4.

In the following section, we examine the FewSig on three stations with the most number of valid aftershock arrivals based on Table 3.

4.2.1 Experimental Results on MKAR

We extracted 217 aftershocks and 1182 non-aftershocks at the MK31 station based on the conditions described in the data preparation section. Figure 13 shows the origins of the selected arrivals.

The performance of FewSig and reference models described in section 1.4.1 are demonstrated in Figure 14. We can conclude that FewSig consistently leads the other models by at least 0.15 on F1, 20% on TPR, and essentially maintains the lowest FPR. The F1-score of FewSig rapidly increases when more aftershocks are used to train models. It reaches the peak point 157.2 hours after the main shock for an F1-score of 89%. F1-score gradually decreases when there are more non-aftershock events. Finally, F1-score reaches 85%, TPR equals 82.55% and FPR equals 5.4% for the FewSig.

During the online evaluation process, the selective model ATDT picked 127 arrivals as aftershocks, 11 of them are actually non-aftershocks which are shown in Figure 15, this yields 54.72% recall and 2.3% FPR for the ATDT.

Figure 16 shows the origins of the false positives and false negatives obtained by FewSig. The detailed scores for FewSig are listed in Table 5.

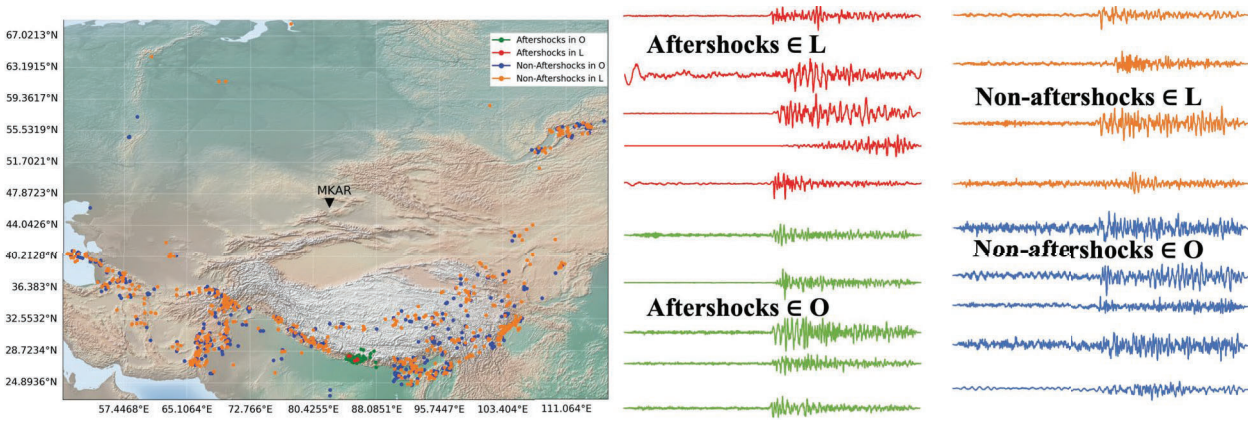

Figure 13: Left figure shows the geographical distribution of origins for valid arrivals at MKAR. *The right figure shows some example waveforms for aftershocks and non-aftershocks from L and O. The first 5 aftershocks are in red. The BHZ waveforms recorded at MKAR are shown. They were filtered with a 0.4Hz to 10Hz, two pass, Butterworth bandpass filter*

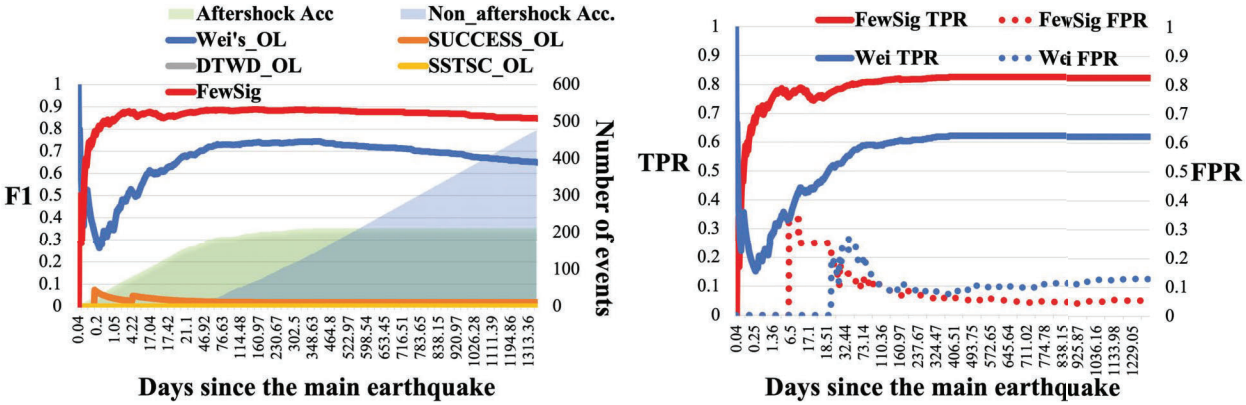

Figure 14: Online performance for classifying Nepal aftershock sequence at MKAR. *TPR, FPR, and F1 scores of different models are shown on the solid or dotted curves. A point on a curve shows the score when testing the events at and before the time on the x-axis. The accumulated number of testing aftershocks and non-aftershocks are represented by the light green and blue shaded areas respectively. Note that the F1 curve for DTWD_OL is overlain by SSTSC_OL since their F1 scores are zero throughout*

Table 5: Nepal, MKAR: Detailed scores of the overall performance for FewSig, each row represents the vote count in the NCFAE module

num_Vote	TP	TN	FP	FN	Recall	Precision	FPR	F1
1	176	444	33	36	0.8302	0.8421	0.0692	0.8361
2	175	451	26	37	0.8255	0.8706	0.0545	0.8475
3	169	453	24	43	0.7972	0.8756	0.0503	0.8346
4	166	456	21	46	0.7830	0.8877	0.0440	0.8321
5	166	458	19	46	0.7830	0.8973	0.0398	0.8363
6	162	459	18	50	0.7642	0.9000	0.0377	0.8265
7	159	460	17	53	0.7500	0.9034	0.0356	0.8196
8	157	460	17	55	0.7406	0.9023	0.0356	0.8135
9	153	460	17	59	0.7217	0.9000	0.0356	0.8010
10	153	461	16	59	0.7217	0.9053	0.0335	0.8031

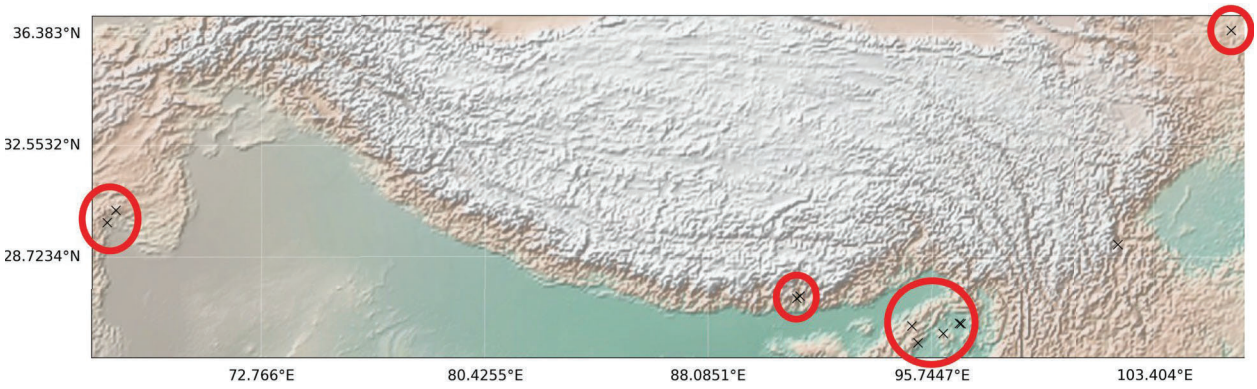

Figure 15: Nepal, MKAR: 11 false positive non-aftershock arrivals selected by the ATDT

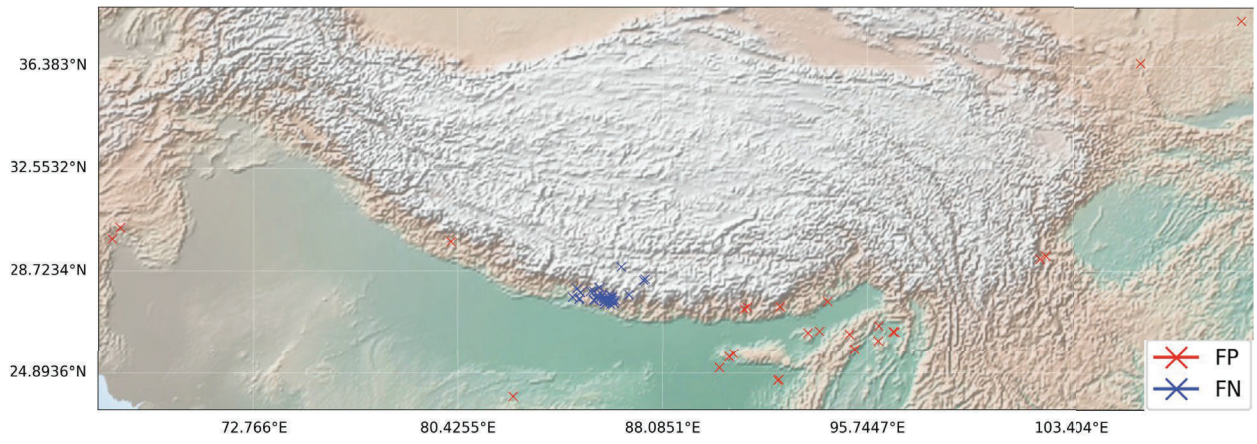

Figure 16: Nepal, MKAR: False positives and false negatives obtained by FewSig

4.2.2 Experimental Results on KURK

The station with the second most number of valid arrivals is KURK, we show the same set of experimental results in Figure 17, 18, 19, 20 and Table 6, 7. The overall performance is around 5% worse than that on the MKAR data with respect to the F1 score.

During the online testing process, the ATDT selective model picked 50 arrivals as aftershocks, 7 of them are actually non-aftershocks which are shown in Figure 19. This yields 47.78% recall and 2.3% FPR for the ATDT.

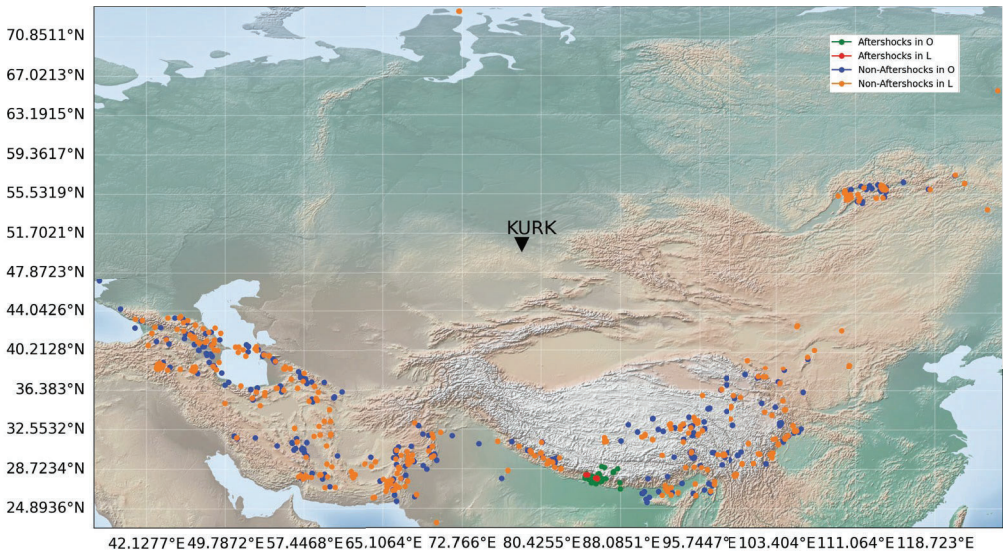

Figure 17: Geographical distribution of origins for valid arrivals at KURK station

Table 6: Number of valid arrivals on KURBB for training and testing

Number of arrivals	Training set	Online Testing set
Aftershock	5	90
Non-aftershock	445	303

Table 7: Nepal, KURK: Detailed scores of the overall performance for FewSig, each row represents the vote count in the NCFAE module

num_Vote	TP	TN	FP	FN	Recall	Precision	FPR	F1
1	75	263	40	15	0.8333	0.6522	0.1320	0.7317
2	74	281	22	16	0.8222	0.7708	0.0726	0.7957
3	72	283	20	18	0.8000	0.7826	0.0660	0.7912
4	69	285	18	21	0.7667	0.7931	0.0594	0.7797
5	68	285	18	22	0.7556	0.7907	0.0594	0.7727
6	68	285	18	22	0.7556	0.7907	0.0594	0.7727
7	67	286	17	23	0.7444	0.7976	0.0561	0.7701
8	67	286	17	23	0.7444	0.7976	0.0561	0.7701
9	66	287	16	24	0.7333	0.8049	0.0528	0.7674
10	64	287	16	26	0.7111	0.8000	0.0528	0.7529

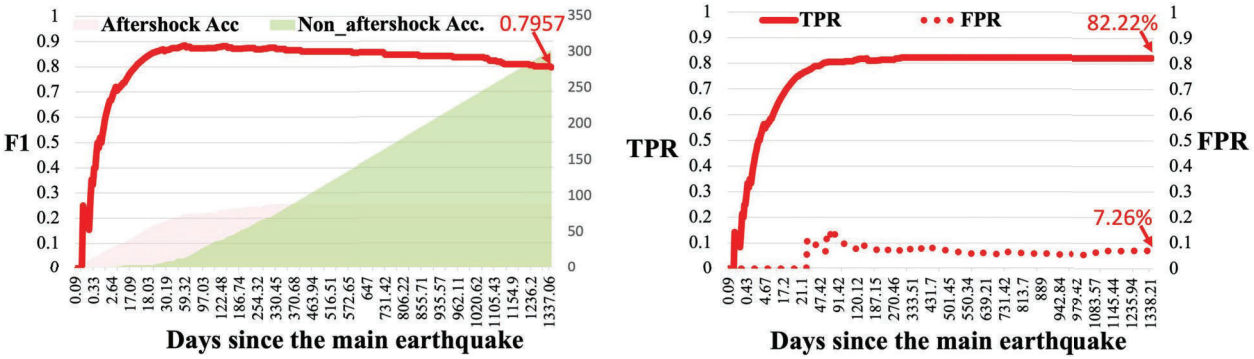

Figure 18: Nepal, KURK: Online performance while classifying Nepal aftershock sequence at KURK. *TPR, FPR, and F1-score of FewSig are shown in solid or dotted curves. A point on a curve shows the score when testing the events at and before the time on the x-axis. The accumulated number of testing aftershocks and non-aftershocks are represented by the light red and green shaded areas respectively*

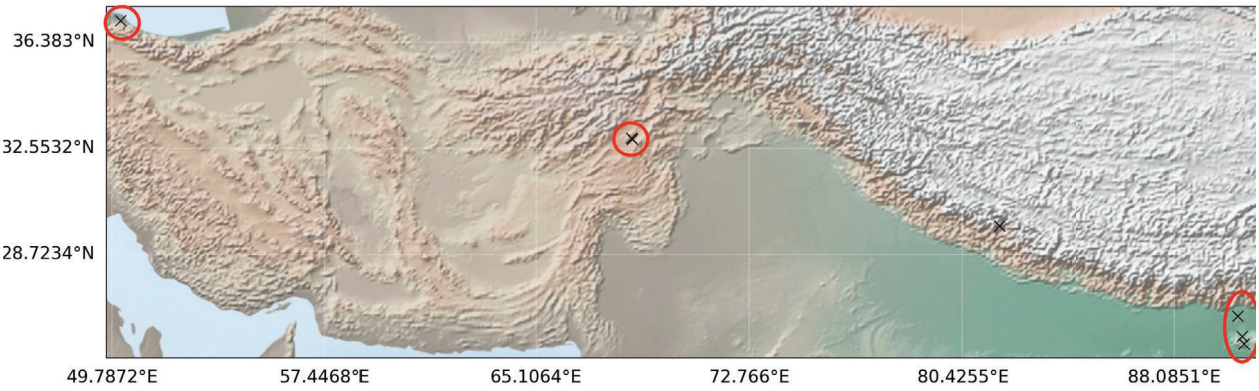

Figure 19: Nepal, KURK: Seven false positive non-aftershock arrivals selected by the ATDT

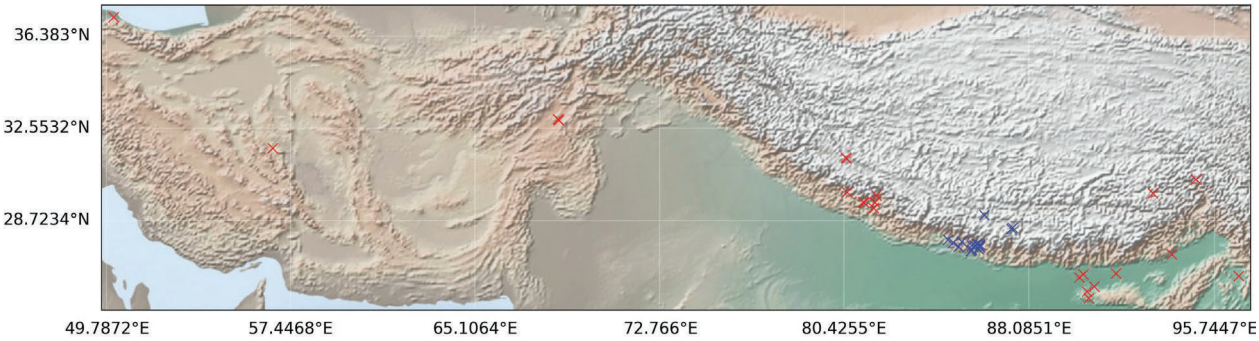

Figure 20: Nepal, KURK: False positives and false negatives obtained by FewSig. *Red dots are false positives and blue dots are false negatives*

4.2.3 Experimental Results on ZALV

Lastly, we evaluate FewSig on the arrivals at the station ZALV. The same set of experimental results are shown in Figure 21, 22, 23, 24 and Table 8, 9. FewSig achieves unexpectedly poor performance, a mere 22% F1-score. Based on the output of ATDT, it is evident that the waveform similarities among aftershocks are very low. The selective model could pick only one aftershock arrival out of 67 in the testing set.

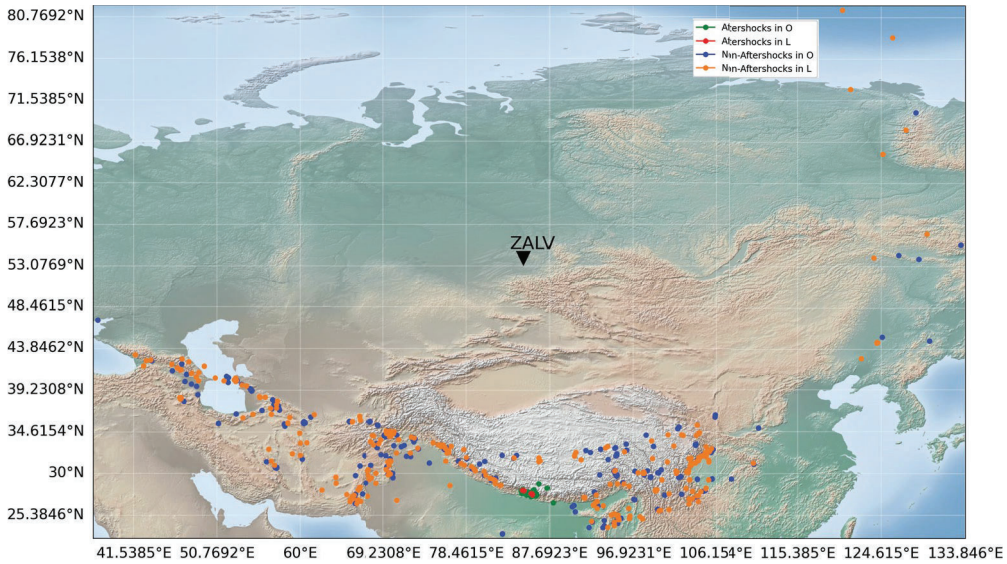

Figure 21: Geographical distribution of origins for valid arrivals at ZALV station

Table 8: Number of valid arrivals at ZAAOB for training and testing

Number of arrivals	Training set	Online Testing set
Aftershock	5	67
Non-aftershock	330	229

Table 9: Nepal, ZALV: Detailed scores of the overall performance for FewSig, each row represents the vote count in the NCFAE module

num_Vote	TP	TN	FP	FN	Recall	Precision	FPR	F1
1	12	222	7	55	0.1791	0.6316	0.0306	0.2791
2	9	224	5	58	0.1343	0.6429	0.0218	0.2222
3	7	224	5	60	0.1045	0.5833	0.0218	0.1772
4	7	225	4	60	0.1045	0.6364	0.0175	0.1795
5	5	225	4	62	0.0746	0.5556	0.0175	0.1316
6	5	225	4	62	0.0746	0.5556	0.0175	0.1316
7	4	225	4	63	0.0597	0.5000	0.0175	0.1067
8	4	225	4	63	0.0597	0.5000	0.0175	0.1067
9	4	225	4	63	0.0597	0.5000	0.0175	0.1067
10	3	225	4	64	0.0448	0.4286	0.0175	0.0811

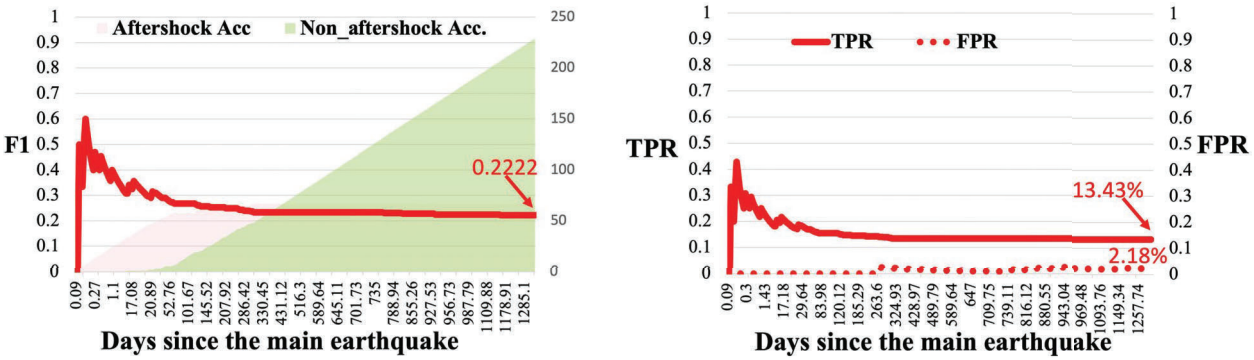

Figure 22: Nepal, ZALV: Online performance while classifying Nepal aftershock sequence at ZALV. *TPR, FPR, and F1-score of FewSig are shown in solid or dotted curves. A point on a curve shows the score when testing on the events at and before the time on the x-axis. The accumulated number of testing aftershocks and non-aftershocks are represented by the light red and green shaded areas respectively*

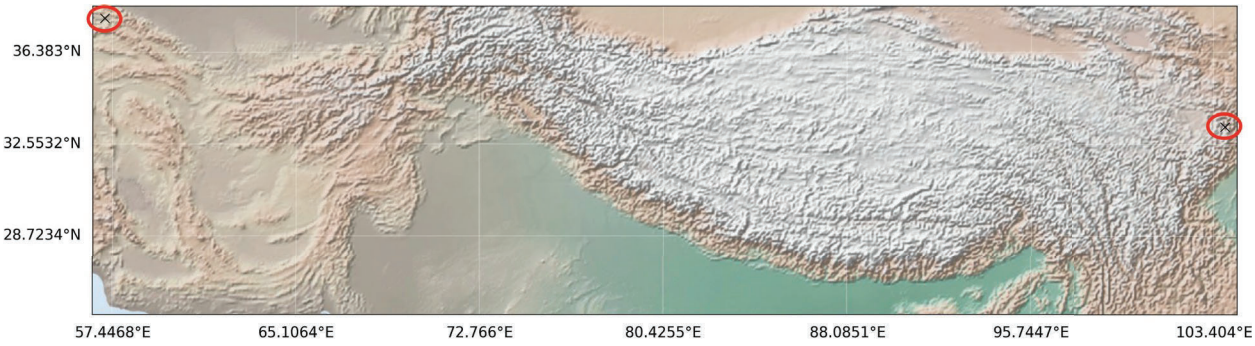

Figure 23: Nepal, ZALV: Two false positive non-aftershock arrivals selected by the ATDT

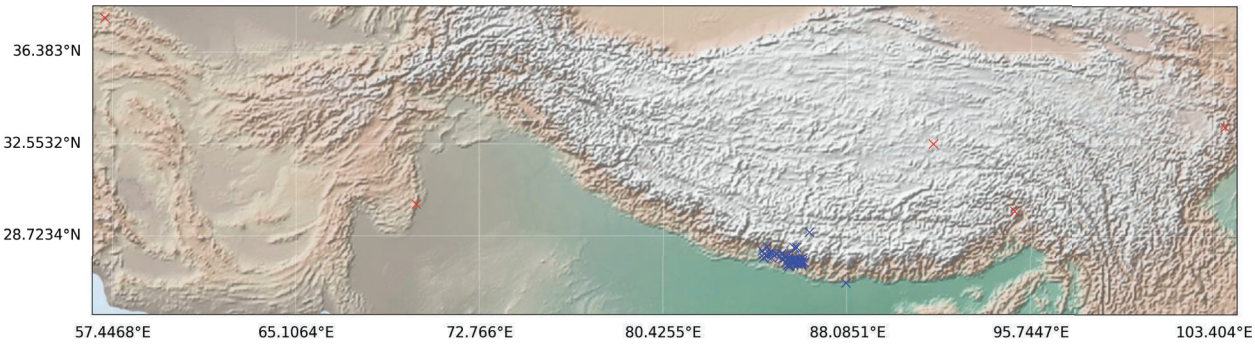

Figure 24: Nepal, ZALV: False positives and false negatives classified by FewSig. *Red dots are false positives and blue dots are false negatives*

4.3 2017 Mw 8.2 Chiapas Earthquake Aftershock Sequence

Data Preparation

To test the universality of FewSig, we further examine the 2017 Chiapas aftershock sequence with FewSig. We use the catalogs from the Mexican Servicio Sismológico Nacional (SSN) [2] as the ground truth, the selection criteria we made are 1. The period from 2017-09-08 to 2018-03-08. 2. The Area is 14 to 17 for latitude and -96 to -93 for longitude. All the ground truth events have a magnitude greater than M2.5. The arrival selection procedure is the same as the Nepal earthquake. The distribution of ground truth events and the LEB events are shown in Figure 25. The associated arrivals at some IMS stations in the LEB database are demonstrated in Figure 26. Table 10 lists the number of valid aftershock arrivals.

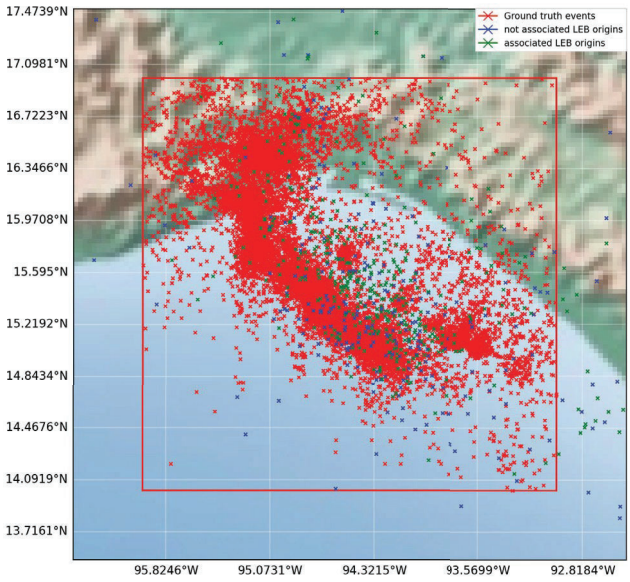

Figure 25: The origin distribution of events from ground truth and the LEB bulletin. ***The events from the LEB bulletin are selected by limiting the origins to the red rectangular region and limiting the origin time between 2017-09-08T04:49:17.000000Z and 2018-03- 08T22:18:24.000000Z according to the ground truth. We applied d = 200KM and δ_t = 5Sec. as the association threshold***

Table 10: Top 5 IMS stations with the most valid aftershock arrivals

Station	number of valid arrivals
TXAR	139
CMIG	109
ROSC	100
ILAR	72
NVAR	64

The parameters used for FewSig remain the same as in section 1.4.2. The training and testing sets are split differently. For the Chiapas aftershock sequence, the number of non-

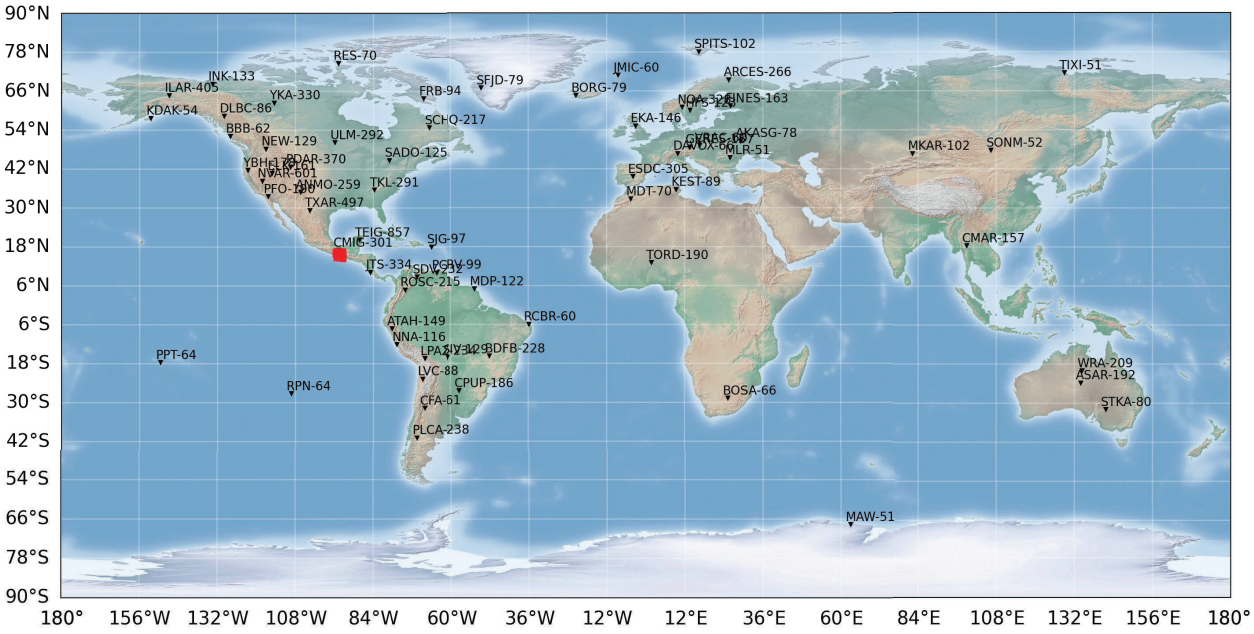

Figure 26: Stations and the corresponding number of arrivals in the LEB for the associated events. *Only the stations with more than 50 arrivals are displayed*

aftershock arrivals in the online testing set is insufficient because our dataset only covers 2009 to 2018. Thus we move some later non-aftershock arrivals from the training set to the tail of the online testing set such that the number of non-aftershock arrivals in both the training set and online testing set is equal. The arrivals in the online testing set are time ordered except for the arrivals transferred from the training set.

In the following section, we examine FewSig on three stations with the most aftershock events based on Table 10.

4.3.1 Experimental Results on TXAR

We examine the arrivals at TXAR since it has the most valid arrivals. The results are shown in Figure 27, 28, 29, 30 and Table 11, 12.

During the online testing process, the ATDT selective model picked 86 events as aftershocks; one of them is actually a non-aftershock which is shown in Figure 19. This yields 63.43% recall and 0.36% FPR for the ATDT. FewSig accomplished the online classification task well at TXAR with an overall 91% F1 score, 90.3% TPR, and 3.57% FPR.

Table 11: Number of valid arrivals at TX32 for training and testing

Number of arrivals	Training set	Online Testing set
Aftershock	5	134
Non-aftershock	280	280

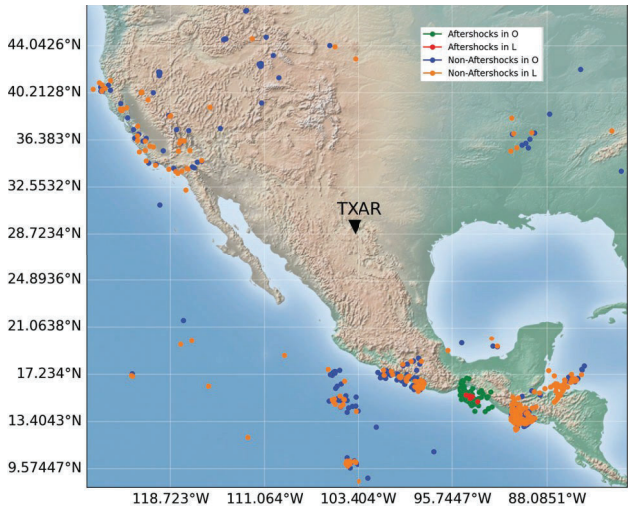

Figure 27: Geographical distribution of origins for valid arrivals at TXAR station

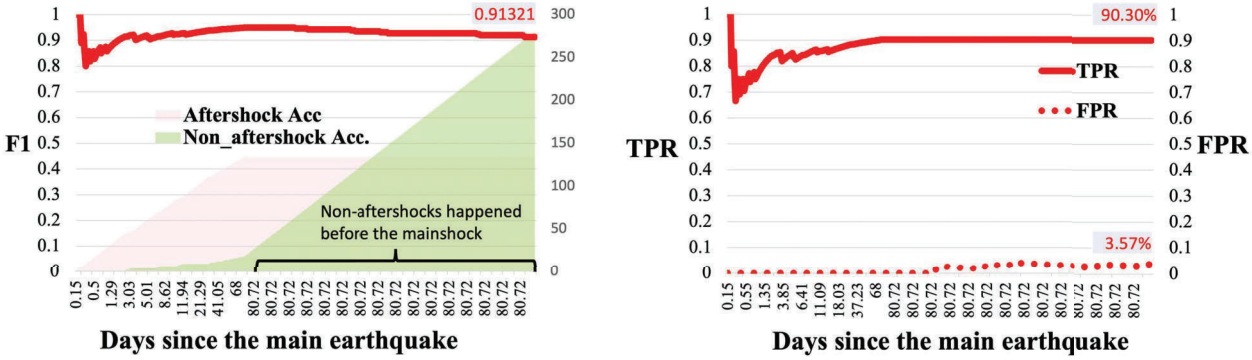

Figure 28: Chiapas, TXAR: Online performance while classifying the Chiapas aftershock sequence at TXAR. *TPR, FPR, and F1-score of FewSig are shown in solid or dotted curves. A point on a curve shows the score when testing the events at and before the time on the x-axis. The accumulated number of testing aftershocks and non-aftershocks are represented by the light red and green shaded areas respectively*

4.3.2 Experimental Results on CMIG

Next, we examine the arrivals at CMIG. The same set of results are demonstrated as for TXAR in Figures 31, 32, 33, and 34 and Tables 13 and 14.

During the online testing process, the ATDT selective model picked 63 events as aftershocks, 2 of them are actually non-aftershocks which are shown in Figure 19. This yields 58.65% recall and a 0.586% FPR for the ATDT.

FewSig achieved sub-optimal performance at this station, as we noticed in Figure 32, the F1 score can reach above 0.8 before evaluating the chunk of non-aftershock arrivals that were transferred from the training set for balancing, However, the F1 decreases to 0.64 when testing that non-aftershocks. This is due to the poor performance of the NCFAE module since the ATDT maintains excellent FPR. Most of the false positives are

Figure 29: Chiapas, TXAR: The one false positive non-aftershock arrival selected by the ATDT

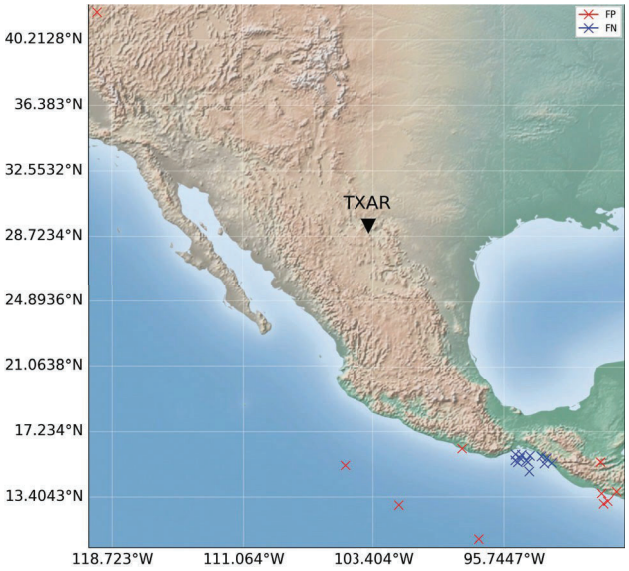

Figure 30: Chiapas, TXAR: False positives (red) and false negatives (blue) classified by FewSig

classified by the NCFAE module. Such poor performance of NCFAE could be caused by noise at CMIG and the non-aftershock origins that are close to aftershocks.

Table 12: Chiapas, TXAR: Detailed scores of the overall performance for FewSig, each row represents the vote count in the NCFAE module

num_Vote	TP	TN	FP	FN	Recall	Precision	FPR	F1
1	122	268	12	12	0.9104	0.9104	0.0429	0.9104
2	121	270	10	13	0.9030	0.9237	0.0357	0.9132
3	120	271	9	14	0.8955	0.9302	0.0321	0.9125
4	120	274	6	14	0.8955	0.9524	0.0214	0.9231
5	118	274	6	16	0.8806	0.9516	0.0214	0.9147
6	116	275	5	18	0.8657	0.9587	0.0179	0.9098
7	115	275	5	19	0.8582	0.9583	0.0179	0.9055
8	114	275	5	20	0.8507	0.9580	0.0179	0.9012
9	113	275	5	21	0.8433	0.9576	0.0179	0.8968
10	111	275	5	23	0.8284	0.9569	0.0179	0.8880

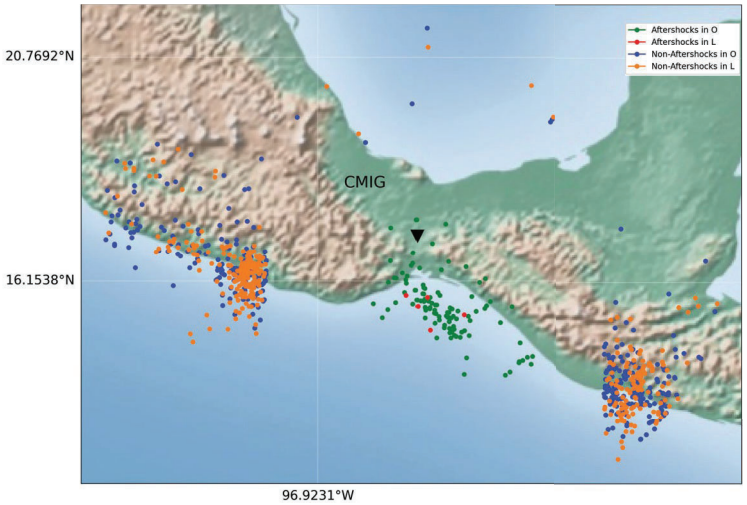

Figure 31: Geographical distribution of origins for selected arrivals at CMIG station

Table 13: Number of valid arrivals at CMIG for training and testing

Number of arrivals	Training set	Online Testing set
Aftershock	5	104
Non-aftershock	340	341

Table 14: Chiapas, CMIG: Detailed scores of the overall performance for FewSig, each row represents the vote count in the NCFAE module

num_Vote	TP	TN	FP	FN	Recall	Precision	FPR	F1
1	85	271	70	19	0.8173	0.5484	0.2053	0.6564
2	77	281	60	27	0.7404	0.5620	0.1760	0.6390
3	75	292	49	29	0.7212	0.6048	0.1437	0.6579
4	73	329	12	31	0.7019	0.8588	0.0352	0.7725
5	72	333	8	32	0.6923	0.9000	0.0235	0.7826
6	72	336	5	32	0.6923	0.9351	0.0147	0.7956
7	69	336	5	35	0.6635	0.9324	0.0147	0.7753
8	68	337	4	36	0.6538	0.9444	0.0117	0.7727
9	68	337	4	36	0.6538	0.9444	0.0117	0.7727
10	67	338	3	37	0.6442	0.9571	0.0088	0.7701

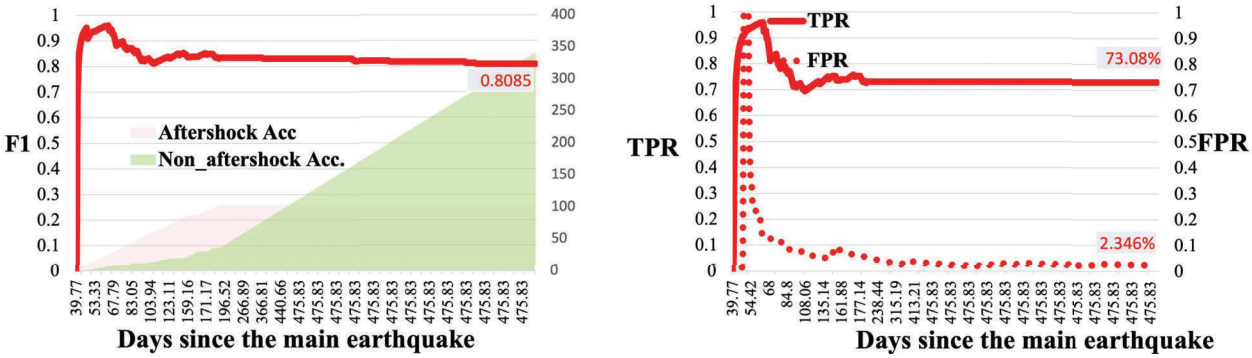

Figure 32: Chiapas, CMIG: Online performance while classifying the Chiapas aftershock sequence at CMIG. *TPR, FPR, and F1-score of FewSig are shown in solid or dotted curves. A point on a curve shows the score when testing the events at and before the time on the x-axis. The accumulated number of testing aftershocks and non-aftershocks are represented by the light red and green shaded areas respectively*

Figure 33: Chiapas, CMIG: Two false positive non-aftershock arrivals selected by the ATDT

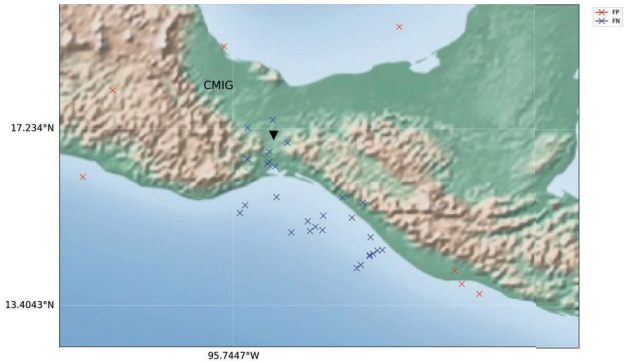

Figure 34: Chiapas, CMIG: False positives (red) and false negatives (blue) classified by FewSig

4.3.3 Experimental Results on ROSC

Finally, we examine the arrivals at ROSC. The same set of results as before are demonstrated in Figures 35, 36, 37, and 38 and Tables 15 and 16.

The ATDT selective model picked 35 events as aftershocks, 5 of them are actually non-aftershocks which shows in Figure 19. This yields 34.48% recall and 2.53% FPR for the ATDT. FewSig has an overall 0.74 F1 score, 74.7% TPR, and 11.62% FPR. The FPR value is slightly higher because there is a group of non-aftershocks that are next to the aftershocks, with similar back-azimuths as the aftershock. These would be expected to have similar propagation effects to the aftershocks This can be further confirmed by the distribution of the false positives shown in Figure 38, where the majority of the false positives share similar back-azimuths to the aftershocks.

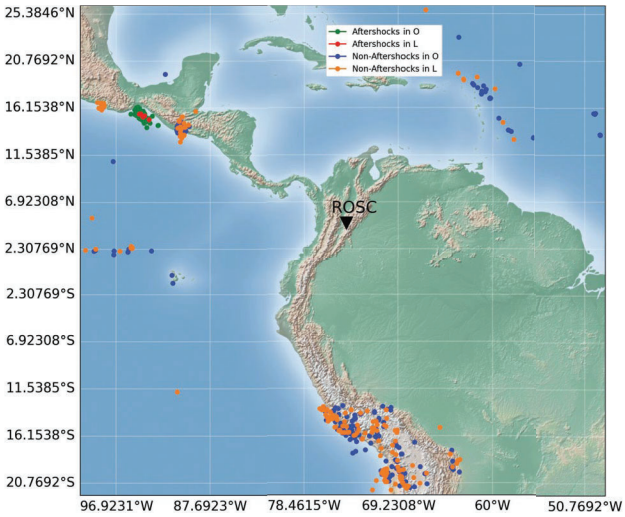

Figure 35: Geographical distribution of origins for valid arrivals at ROSC station

Table 15: Number of valid arrivals at ROSC for training and testing

Number of arrivals	Training set	Online Testing set
Aftershock	5	87
NNon-aftershock	197	198

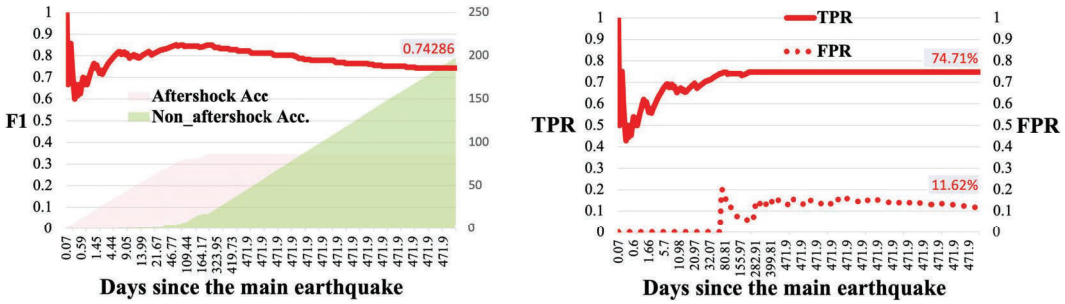

Figure 36: Chiapas, ROSC: Online performance for classifying Chiapas aftershock sequence at ROSC

Figure 37: Chiapas, ROSC: Five false positive non-aftershock arrivals selected by the ATDT

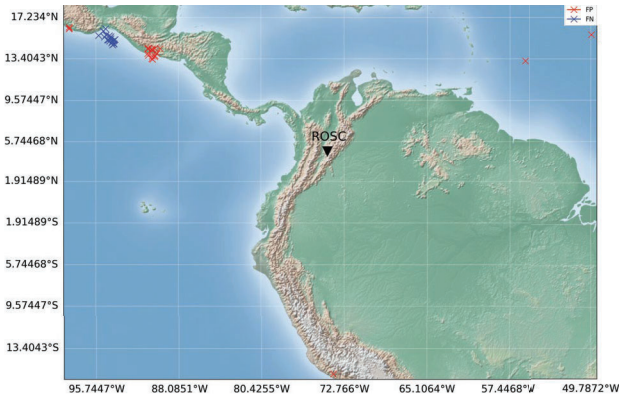

Figure 38: Chiapas, ROSC: False positives (red) and false negatives (blue) by FewSig

Table 16: Chiapas, ROSC: Detailed scores of the overall performance for FewSig, each row represents the vote count in the NCFAE module

num_Vote	TP	TN	FP	FN	Recall	Precision	FPR	F1
1	69	165	33	18	0.7931	0.6765	0.1667	0.7302
2	65	175	23	22	0.7471	0.7386	0.1162	0.7429
3	65	178	20	22	0.7471	0.7647	0.1010	0.7558
4	63	181	17	24	0.7241	0.7875	0.0859	0.7545
5	63	181	17	24	0.7241	0.7875	0.0859	0.7545
6	62	182	16	25	0.7126	0.7949	0.0808	0.7515
7	58	182	16	29	0.6667	0.7838	0.0808	0.7205
8	57	182	16	30	0.6552	0.7808	0.0808	0.7125
9	56	182	16	31	0.6437	0.7778	0.0808	0.7044
10	55	183	15	32	0.6322	0.7857	0.0758	0.7006

4.4 Utilizing Back Azimuth as an Additional Feature

Figure 13 shows the origin locations of both aftershock and non-aftershock arrivals. Some false positives from Figure 16 could be easily corrected by checking their back azimuth values. This feature is easy to acquire when detecting a new arrival. We insert the following decision process to the framework in Figure 1 - arrival i will be directly classified as a non-aftershock as the final label if the inequality 11 holds, otherwise, it follows the same decision route as shown in Figure 1. μ is the mean of the back azimuth value of the corresponding origins of aftershock arrivals in the training set, and δ is the user-defined threshold. The selection of δ depends on many factors such as the location of a station and the aftershock fault, and also relies on domain knowledge. We set $\delta = 21$ just for this proof of concept exercise.

$$|\text{BackAzimuth}(i) - \mu| > \delta \tag{11}$$

We compare results with and without back azimuth features in Table 17 for Nepal aftershocks on MKAR and Table 18 for Chiapas aftershocks on TXAR.

Table 17: Nepal, MKAR: Effect of back azimuth on overall performance

Nepal MKAR	TPR	FPR	F1	ATDT selected arrivals	#FP by ATDT
With back-azimuth	82.075%	1.26%	0.8877	121	1
Without back-azimuth	83.49%	5.45%	0.8475	127	11

Table 18: Chiapas, TXAR: Effect of back azimuth on overall performance

Chiapas TXAR	TPR	FPR	F1	ATDT selected arrivals	#FP by ATDT
With back-azimuth	89.55%	2.5%	0.92	85	1
Without back-azimuth	90.30%	3.57%	0.9132	86	1

We conclude from Tables 17 and 18 that using back azimuth can improve the overall performance with respect to the F1 score and can help decrease the false positive rate for both the selective model and general model. However, such a feature has a fundamental limitation when the back azimuth of aftershocks and non-aftershocks are similar, for example, the origins shown in Figure 35. In addition, we also observe that non-array stations generally estimate back azimuth poorly, reducing the gain from this feature in our model.

Conclusion

In this project, we develop an online few-shot classification framework for seismic signals. Our model can be trained on a few positive signals and adapt to the new positive instances iteratively. This setting is essential for online nuclear explosion monitoring systems, as evaluated in this work. In addition, we focus on classifying one new repeated class. However, a different emerging class can also happen in this process, and we leave this as future work.

References

[1] IDC Documentation, Processing of Seismic, Hydroacoustic, and Infrasonic Data (1999), (IDC/OPS/MA/001/Rev.1).

[2] Mexico earthquake catalog, http://www2.ssn.unam.mx:8080/catalogo/.

[3] Support web site, https://sites.google.com/view/onlinefewshot/home.

[4] Abanda, A., Mori, U., and Lozano, J. A., A review on distance-based time series classification, *Data Mining and Knowledge Discovery 33* (3 2019), pp. 378–412.

[5] Allen, R. V., Automatic earthquake recognition and timing from single traces, *Bulletin of the Seismological Society of America 68*, 5 (1978), pp. 1521–1532.

[6] Bagnall, Anthony, Lines, Jason, Vickers, W., and Keogh, E., The UEA & UCR time series classification repository, www.timeseriesclassification.com.

[7] Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E., The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances, *Data Mining and Knowledge Discovery 31* (5 2017), pp. 606–660.

[8] Chen, H., Tang, F., Tino, P., and Yao, X., Model-based kernel for efficient time series analysis, In *Proceedings of the KDD 2013* (2013), pp. 392–400.

[9] Chen, Y., Hu, B., Keogh, E., and Batista, G. E., Dtw-d: Time series semi-supervised learning from a single example, In *Proceedings of the KDD 2013* (New York, NY, USA, 2013), KDD '13, Association for Computing Machinery, pp. 383–391.

[10] Fan, H., Zhang, F., Wang, R., Huang, X., and Li, Z., Semi-supervised time series classification by temporal relation prediction, IEEE, pp. 3545–3549.

[11] Giusti, R., Silva, D. F., and Batista, G. E. A. P. A., Improved time series classification with representation diversity and svm, In *Proceedings of the ICMLA 2016* (2016), pp. 1–6.

[12] Goldberger, J., Roweis, S., Hinton, G., and Salakhutdinov, R., Neighbourhood components analysis, In *International Conference on Neural Information Processing Systems* (Cambridge, MA, USA, 2004), NIPS'04, MIT Press, pp. 513–520.

[13] Jain, B. and Spiegel, S., Dimension reduction in dissimilarity spaces for time series classification, In *Advanced Analysis and Learning on Temporal Data* (Cham, 2016), Springer, pp. 31–46.

[14] Jawed, S., Grabocka, J., and Schmidt-Thieme, L., Self-supervised learning for semi-supervised time series classification, In *Pacific-Asia Conference on KDD* 2020), Springer, pp. 499–511.

[15] Kate, R. J., Using dynamic time warping distances as features for improved time series classification, *Data Mining and Knowledge Discovery 30* (3 2016), pp. 283–312.

[16] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollar, P., Focal loss for dense object detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence 42* (2 2020), pp. 318–327.

[17] Lines, J., Davis, L., Hills, J., and Bagnall, A., A shapelet transform for time series classification, In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (2012), KDD, pp. 289–297.

[18] Lomax, A., Satriano, C., and Vassallo, M., Automatic picker developments and optimization, *Seismological Research Letters 83*, 3 (2012), pp. 531–540.

[19] Marussy, K. and Buza, K., Success: A new approach for semi-supervised classification of time-series, In *Artificial Intelligence and Soft Computing* (Berlin, Heidelberg, 2013), Springer Berlin Heidelberg, pp. 437–447.

[20] McNamara, D., Yeck, W., Barnhart, W., Schulte-Pelkum, V., Bergman, E., Adhikari, L., Dixit, A., Hough, S., Benz, H., and Earle, P., Source modeling of the 2015 mw 7.8 nepal (gorkha) earthquake sequence, *Tectonophysics 714-715* (2017), 21–30, Special Issue on the 25 April 2015 Mw 7.8 Gorkha (Nepal) Earthquake.

[21] Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., and Bagnall, A., Hive-cote 2.0: a new meta ensemble for time series classification, *Machine Learning 110* (12 2021), pp. 3211–3243.

[22] Mueen, A., Keogh, E., and Young, N., Logical-shapelets: an expressive primitive for time series classification, *the 17th ACM SIGKDD international conference* (2011), pp. 1154–1162.

[23] Narwariya, J., Malhotra, P., Vig, L., Shroff, G., and Vishnu, T., Meta-learning for few-shot time series classification, In *ACM IKDD CoDS*. 2020, pp. 28–36.

[24] Nguyen, M. N., Li, X.-L., and Ng, S.-K., Positive unlabeled learning for time series classification, In *International Joint Conference on Artificial Intelligence* (2011), IJCAI'11, AAAI Press, pp. 1421–1426.

[25] Pope, B., Air force technological advancement center. Personal Communication, March 9, 2022, at 12:33 PM.

[26] Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and Keogh, E., Searching and mining trillions of time series subsequences under dynamic time warping, In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2012), pp. 262–270.

[27] Shokoohi-Yekta, M., Wang, J., and Keogh, E., *On the Non-Trivial Generalization of Dynamic Time Warping to the Multi-Dimensional Case,* ch. 33, pp. 289–297.

[28] Sousa, C. A. R., Souza, V. M. A., and Batista, G., Time series transductive classification on imbalanced data sets: an experimental study, In *International Conference on Pattern Recognition* (2014), IEEE, pp. 3780–3785.

[29] Souza, V. M. A., Rossi, R. G., Batista, G., and Rezende, S. O., Unsupervised active learning techniques for labeling training sets: an experimental evaluation on sequential data, *Intelligent Data Analysis 21*, 5 (2017), pp. 1061–1095.

[30] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M., Learning to compare: Relation network for few-shot learning, In *IEEE conference on computer vision and pattern recognition* (2018), pp. 1199–1208.

[31] Wang, H., Zhang, Q., Wu, J., Pan, S., and Chen, Y., Time series feature learning with labeled and unlabeled data, *Pattern Recognition 89* (2019), pp. 55–66.

[32] Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M., Generalizing from a few examples: A survey on few-shot learning, *ACM computing surveys 53*, 3 (2020), pp. 1–34.

[33] Wei, L. and Keogh, E., Semi-supervised time series classification, In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06* (New York, New York, USA, 2006), ACM Press, p. 748.

[34] Xi, L., Yun, Z., Liu, H., Wang, R., Huang, X., and Fan, H., Semi-supervised time series classification model with self-supervised learning, *Engineering Applications of Artificial Intelligence 116* (2022), p. 105331.

[35] Xing, Z., Pei, J., and Keogh, E., A brief survey on sequence classification, *ACM SIGKDD Explorations Newsletter 12* (2010).

[36] Xu, Z. and Funaya, K., Time series analysis with graph-based semi-supervised learning, In *IEEE international conference on data science and advanced analytics* (2015), IEEE, pp. 1–6.

List of Abbreviations

ATDT	Auto-Tuning Decision Tree
DTW	Dynamic Time Warping
FPR	False Positive Rate
GPS	Global Positioning System
IDC	International Data Center
IMS	International Monitoring System
InSAR	Interferometric Synthetic Aperture Radar
LEB	Late Event Bulletin
MTL	Multi-Task Learning
NCA	Neighborhood Component Analysis
NCFAE	Neighborhood Components Focal Analysis Ensemble
NN	Nearest Neighbor
NNC	Neighborhood Component Analysis
PCA	Principal Component Analysis
PU Learning	Positive Unlabeled Learning
SSL	Semi-Supervised Learning
SSTSC	Semi-Supervised Time Series Classification
STA/LTA	Short Term Average / Long Term Average
TPR	True Positive Rate
TSC	Time Series Classification

DISTRIBUTION LIST

DTIC/OCP
8725 John J. Kingman Rd, Suite 0944
Ft Belvoir, VA 22060-6218 1 cy

AFRL/RVIL
Kirtland AFB, NM 87117-5776 1 cy

Official Record Copy
AFRL/RVB/1st Lt. Simone M. Smith 1 cy

This page is intentionally left blank.