



A novel machine learning and deep learning semi-supervised approach for automatic detection of InSAR-based deformation hotspots

Ashutosh Tiwari^{*}, Manoochehr Shirzaei

Department of Geosciences, Virginia Tech, Blacksburg, VA, USA
Virginia Tech National Security Institute, Blacksburg, VA, USA

ARTICLE INFO

Keywords:

InSAR
Time series displacement
Spatio-temporal analysis
Semi-supervised clustering

ABSTRACT

Over the past two decades, Interferometric synthetic aperture radar (InSAR) has been invaluable for studying earth surface deformation and related effects. Deformation maps generated through multi-temporal InSAR processing methods are however difficult to interpret accurately by general individual users, decision-makers, and non-domain experts owing to the volume, variety, and velocity they are produced. This paper proposes a semi-supervised machine learning based information mining approach to simplify these deformation maps and detect hotspots by extracting prominent signals from time series deformation. The approach initially combines two machine learning based clustering methods named time series k-means (TSKM) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithms to derive clusters with unique spatiotemporal deformation behavior, using time series deformation output generated from Wavelet-based InSAR (WabInSAR) method. Clustering results generated from this unsupervised machine learning approach are later used as training labels to develop two deep learning models, one using long short term memory (LSTM) networks alone and another using a combination of LSTM and single-layer perceptron for supervised training. The developed LSTM and LSTM + Perceptron models efficiently learn from the cluster labels, reaching an accuracy of 97.3 %. Further, the deep learning models significantly reduce the computational time from orders of days (~5) to hours (~2) while training and from hours to minutes during prediction. We evaluate the developed approach over Los Angeles, a highly challenging area affected by umpteen deformation events that are challenging to categorize. The outcome of the proposed approach produces hotspots of deforming areas in Los Angeles, providing a generalized and more precise picture of events, much appreciable to non-domain experts. The approach can augment any of the multi-temporal InSAR processing chains and is applicable to different deformation prone sites, aiding in derivation of deformation hotspots from time series deformation maps.

1. Introduction

Over the past two decades, the multi-temporal Interferometric synthetic aperture radar (MT-InSAR) technique and its variants have been effectively used for monitoring earth's surface deformation associated with faulting processes (Bürgmann et al., 2000; Shirzaei and Bürgmann, 2013; Shirzaei et al., 2013a; Whipple et al., 2016; Xu et al., 2021), landslides (Nikolaeva et al., 2014; Rosi et al., 2018; Sun et al., 2015) and land subsidence (Miller et al., 2017; Ojha et al., 2020; Osmanoglu et al., 2011), volcanic activity (Hooper et al., 2004; Pritchard and Simons, 2004; Shirzaei et al., 2013b) and infrastructure resilience and stability (Eppler and Rabus, 2012; Gao et al., 2022; Milillo et al., 2016; Wang et al., 2016). Further, we now sit on SAR data mountains thanks to the

unprecedented temporal revisit (6–12 days) and global spatial coverage (~250 km each scene) provided by the Sentinel-1 mission, and also with data from RADARSAT, ALOS PALSAR, TerraSAR-X, and the upcoming NISAR mission, yielding InSAR products with high spatiotemporal resolution greatly benefitting the scientific community. With the plentiful availability of SAR images, the requirements of devising time-efficient methods of processing huge interferometric data stacks and improving data storage and archival algorithms are also being addressed and considered (Ansari et al., 2017, 2018; De Luca et al., 2018; Lee and Shirzaei, 2023; Ma et al., 2022). However, a remaining issue is the understanding, appreciation, and comprehension of the information within InSAR-derived deformation maps by the general (non-domain) users. These users often require a first-order picture of events occurring in an

^{*} Corresponding author at: Department of Geosciences, Virginia Tech, Blacksburg, VA, USA.

E-mail address: ashutoshtiwari796@gmail.com (A. Tiwari).

area of interest to draw useful conclusions, resulting in critical socio-economic decisions and policies. The problem is exacerbated when the InSAR derived deformation pattern for different events appears similar, confusing decision-makers. Furthermore, different TS-InSAR analysis methods may generate dissimilar deformation maps and time series, making it difficult to draw conclusions required for critical decision-making (Chang and Hanssen, 2015). Even to a domain expert, interpreting InSAR deformation maps produced at a staggering rate, volume, and diversity is often challenging. These maps generally contain time series information corresponding to millions of geolocated InSAR pixels, and visual investigation of even a handful of these pixels consumes a significant amount of time and resources, making it infeasible to analyze even the major features rigorously.

Information mining methods can simplify extracting useful contextual information from InSAR-derived deformation maps. The definition of 'useful contextual information' may vary depending on the investigation's purpose. Initial attempts to assess InSAR-derived time series deformation were presented by (Chang and Hanssen, 2015), where a probabilistic hypothesis testing-based approach is proposed to model the InSAR time series. While yielding satisfactory results, such an approach requires considerable domain knowledge and may be computationally expensive for the present-day InSAR data volumes. We see information mining methods used to alleviate time series noise through combined spatiotemporal filtering (Wnuk, 2021) and to detect outlying deformation signals using DBSCAN clustering and principal component analysis (Bakon et al., 2017). Methods such as PCA and independent component analysis (ICA) are also explored to resolve and relate particular geodetic signals, such as settlement due to groundwater extraction (Karimzadeh et al., 2018), slow slip event modeling (Maubant et al., 2020), tunneling induced land settlement (Wnuk, 2021) and coastal land subsidence (Wang et al., 2022b). The statistical mining methods can segregate different deformation signals, given that the analyzed dataset does not contain significantly distinct information. These methods can help generalize the deformation behavior in space and time, making them comprehensible to general users while saving domain experts' time and effort.

Artificial intelligence (AI), machine learning (ML), and deep learning (DL) have recently evolved to solve many unsolvable and time-challenging problems in earth observation studies (Chen et al., 2023; Hasan et al., 2023; Irrgang et al., 2021; Kubo et al., 2020; Mateo-Garcia et al., 2021; Reichstein et al., 2019; Rolf et al., 2021). InSAR has also benefitted from deep learning architectures, and there have been studies attempting detection of volcano-like structures (Anantrasirichai et al., 2019b), urban area deformation (Anantrasirichai et al., 2019a), landslide zones, and hill slope velocities (Devara et al., 2021; Yuan and Chen, 2022), land subsidence (Radman et al., 2021), selection of elite pixels in time series InSAR processing (Lattari et al., 2022; Tiwari et al., 2020; Wang et al., 2022a), phase unwrapping (Sica et al., 2020; Zhou et al., 2022), and deformation prediction (Rouet-Leduc et al., 2021). For analysing InSAR derived deformation, we see the approach proposed by (Festa et al., 2023), combining PCA (for dimensionality reduction) and k-means clustering to identify natural or anthropogenic deformation signals.

Being cognizant of the significance of detecting useful deformation signals and a few salient observations from the work done till date, we observe that there have been studies using machine learning and dimensionality reduction techniques mostly focused on segregating a single or few deformation events at regional scales. Also, we see that deep learning has not been investigated for clustering InSAR time series displacement. With the rise in the scale of InSAR data both spatially and temporally, it is worth investigating deep learning methods that automate feature engineering and can mine information from gigantic InSAR data volumes with better computational efficiency (Chen and Lin, 2014; Dargan et al., 2020). In this study, for the first time, we investigate a combination of traditional machine learning and deep learning models to address the complexity and tediousness associated with interpreting

spatiotemporal deformation patterns derived from InSAR. We demonstrate the approach over a large study area in Los Angeles, USA, home to umpteen deformation events (Bawden et al., 2001; Brooks et al., 2007; Riel et al., 2018), making it challenging to understand and simplify. The major objectives of this study are (i) to develop an automated semi-supervised ML-DL approach to explore InSAR deformation time series and (ii) to detect hotspot signals characterizing the deformation behavior of the study site.

Section 2 describes the dataset generation process for the study, and Section 3 describes the methodology. Section 4 presents the results and analysis, followed by the conclusion.

2. Study area, datasets and exploratory analysis

For this study, we use a publicly available line-of-sight (LOS) deformation dataset from Los Angeles, CA, published by (Lee and Shirzaei, 2023) using the Wavelet-based InSAR algorithm (Lee and Shirzaei, 2023; Shirzaei, 2012; Shirzaei and Bürgmann, 2012; Shirzaei et al., 2019) (see Data Availability). Several deformation hotspots characterize the InSAR velocity map. In the following, a few major ones are briefly described. A subsidence rate of -1.5 cm/yr affects the Santa Ana area (Longitude: ~ -117.84 , Latitude: ~ 33.73) and an uplift rate of ~ 0.5 cm/yr adjacent to it due to extraction of groundwater and aquifer recharge operations, respectively. The areas near Seal Beach (Longitude: ~ -118.14 , Latitude: ~ 33.76) and Long Beach (Longitude: ~ -118.21 , Latitude: ~ 33.77) are characterized by several subsidence and uplift hotspots with absolute rates up to 1 cm/yr, associated with activities on Newport-Inglewood Fault and oil and gas production from the Wilmington oil field. The Los Angeles central area (Longitude: ~ -118.21 , Latitude: ~ 33.95) subsides at a rate of -0.4 cm/yr. Subsidence and uplift hotspots are also located near the Hollywood Fault (Longitude: ~ -118.39 , Latitude: ~ 34.08) and are associated with the Los Angeles D-line extension transit project. The slow deformation of Santa Ana Mountain (Longitude: ~ -117.60 , Latitude: ~ 33.76) is due to slow landslides in the form of creep affecting the hillslopes.

The LOS deformation time series and velocity of 3,039,151 measurement pixels at ~ 25 m \times ~ 25 m spatial resolution are obtained from interferometric processing of 247 SAR images acquired in descending orbit of Sentinel-1A/B satellites during 2016–2022. Fig. 1 a-d shows each pixel's LOS velocity, mean interferometric coherence, velocity standard deviation, and local incidence angle. Also, Fig. 1e shows the outline of deformation hotspots (marked in red) retrieved from a discussion with authors (Lee and Shirzaei, 2023). Prior information about the deformation events in a study site (as shown in Fig. 1e) is not required to implement the method used herein and the sole purpose is to validate the performance of the proposed method.

We used temporal-spatial clustering to mine information from the dataset shown above. For temporal clustering, we used the LOS deformation time series derived from the WabInSAR algorithm. This is treated as a sequential predictor variable, incorporating temporal nature. In addition, we also used static predictor variables (i.e., no temporal variation) for training the models. The static predictors include the LOS velocities, their standard deviation, local incidence angles, and mean coherence. Since we deal with an unsupervised problem, the target variable is a cluster label representing statistical spatiotemporal similarities within a cluster and dissimilarities among different cluster labels. LOS velocity information helps understand which areas have higher, lower, or negligible surface movement given the associated standard deviations. Local incidence angles distinguish different surface topologies and account for the topographic changes. Mean coherence characterizes pixels affected by noise within the time series. One can introduce other features into the learning approach, such as elevation data and related parameters (e.g., slope, aspect). Note that there is a trade-off between the dimension of the input parameter space used to optimize learning and the overall complexity of the approach, yielding acceptable or similar performance (Belkin et al., 2019; Briscoe and

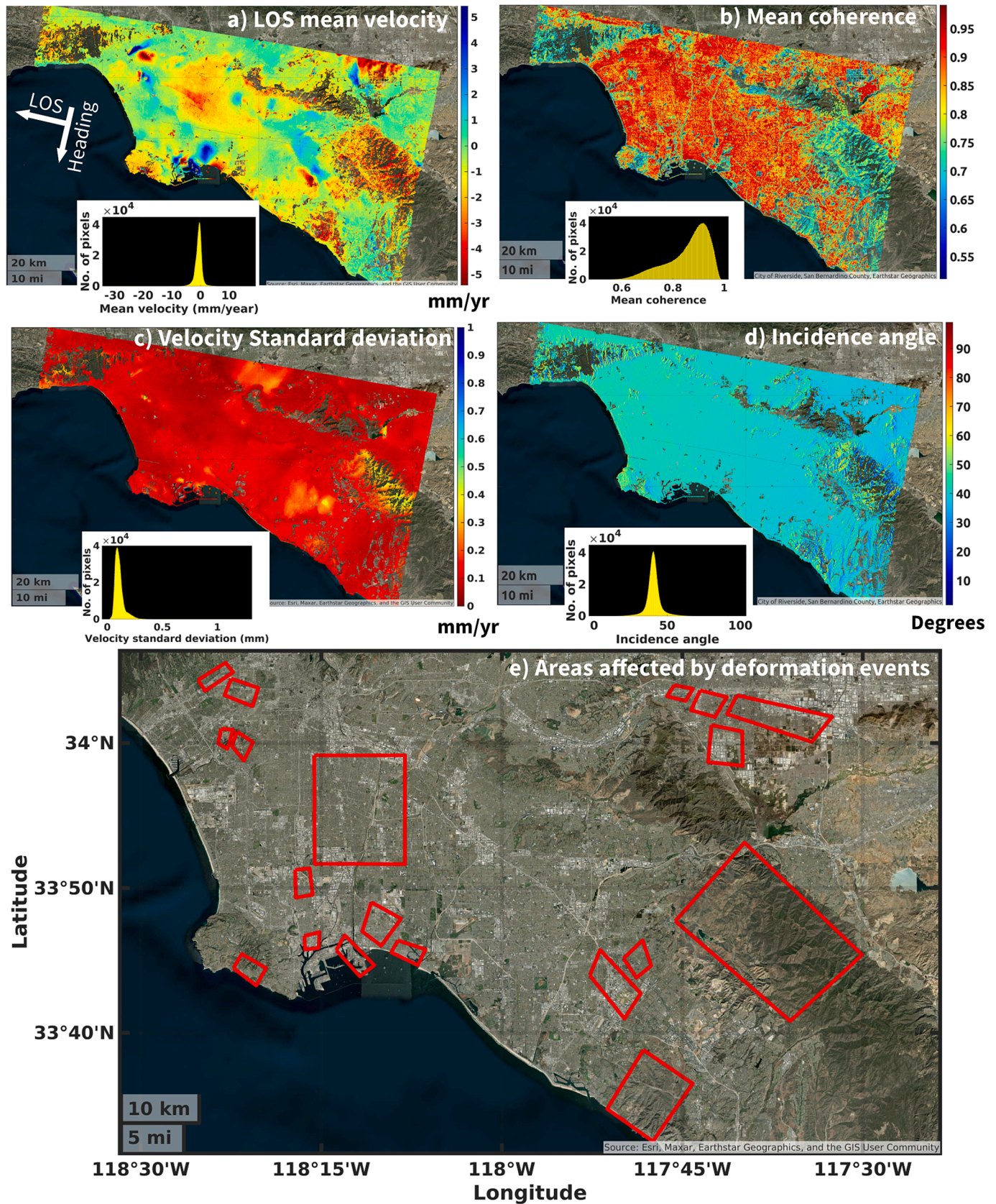


Fig. 1. Study area and dataset used for the proposed approach. a) LOS velocity map of Los Angeles generated using Sentinel-1 SAR images from WabInSAR method, b) Mean coherence for the interferograms, c) Standard deviation corresponding to mean velocities, and d) incidence angle for each selected measurement point, and e) Study area with red polygons denoting areas affected by deformation events. Inset plots in panels a) to d) show histograms of each static variable. Units on horizontal axes are the same as those for the big plots and vertical axes denote the frequency (no. of measurement pixels). Latitude and longitude values are in Decimal degrees. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Feldman, 2011; Geman et al., 1992). Since the problem dealt with is unsupervised, the target variable is a cluster label representing statistical spatio-temporal similarities within a cluster and dissimilarities among different cluster labels.

3. Methodology

The proposed semi-supervised learning approach can be divided into three primary steps. Step 1 involves temporal-spatial unsupervised learning for generating accurate cluster labels. Step 2 uses the labels generated in Step 1 as training inputs for supervised deep learning. Step 3 uses the clusters predicted from Step-2 and a stochastic approach to finally derive hotspots of deformation. The approach begins from unsupervised learning using a combination of temporal and spatial machine learning based clustering methods, and later switches to the development of automated deep learning based time series clustering models. Finally, statistical post-analysis of the predicted clusters yield the planned outcome. The overall approach is illustrated in Fig. 2. InSAR displacement time series obtained from the WabInSAR algorithm enters time-series k-means clustering. We initially label InSAR time series displacements of similar temporal behavior to identify areas undergoing

similar or different deformation phenomena. Subsequently, we test whether the clusters follow spatial separability, and to perform spatial clustering, the individual temporal clusters undergo spatial clustering using the DBSCAN algorithm. The resulting spatiotemporal clusters are used as training labels to develop two deep-learning models for supervised learning. The clusters generated as output from supervised learning are analyzed stochastically to identify the insignificant clusters, and the important clusters are retained.

3.1. Machine learning based spatiotemporal clustering

This method uses statistical criteria to group the time series deformation into different clusters (labels). The static k-means algorithm initializes m clusters at random locations, with m chosen based on the Elbow method or Silhouette score, two widely used metrics for evaluating the number of statistically independent clusters in the data (Pedregosa et al., 2011; Rousseeuw, 1987). For each new data point, the method computes the distance between the point and every cluster. The cluster to which the data point has the minimum distance is assigned to the point. This operation is repeated for all the points, and after each data point is clustered into one of the clusters, the cluster centers move

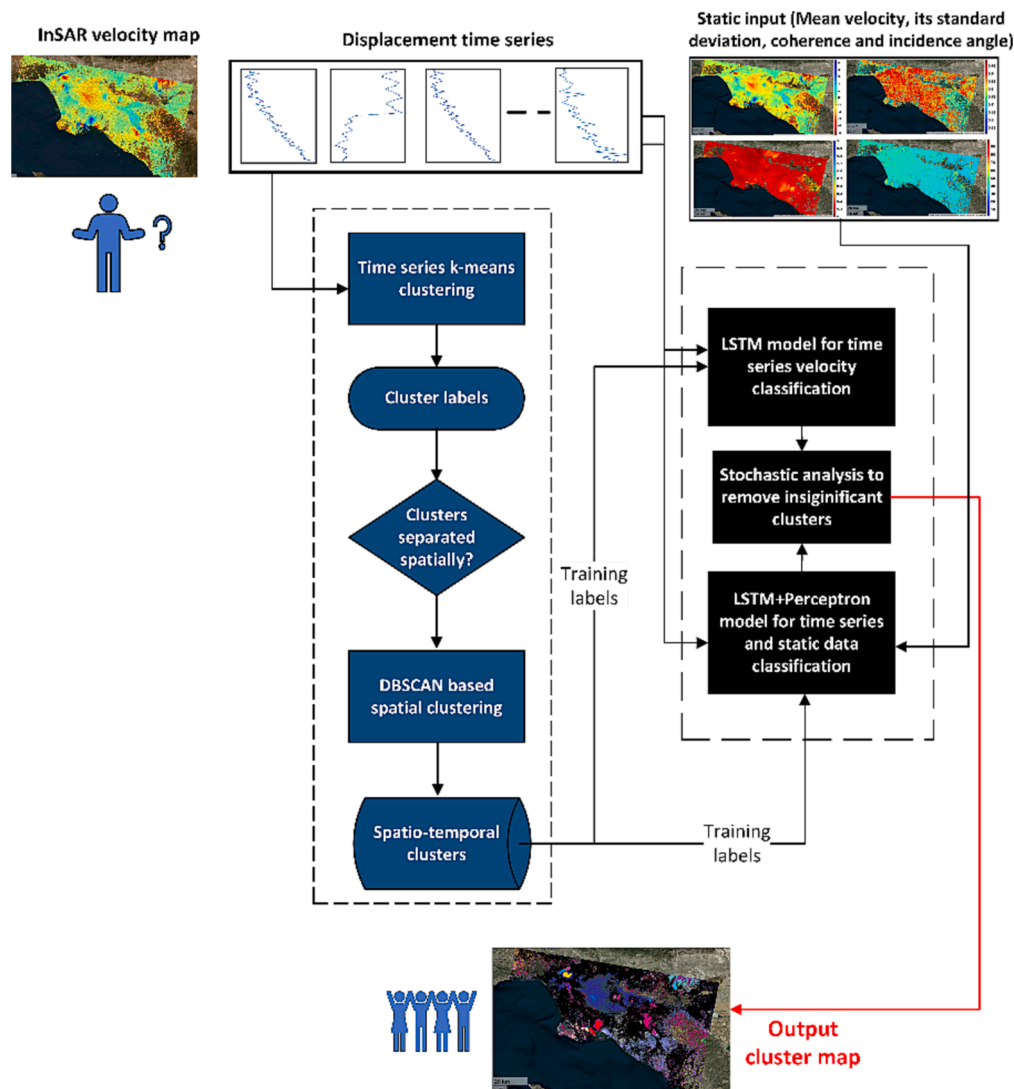


Fig. 2. Workflow for the proposed approach. Input includes displacement time series obtained from WabInSAR algorithm, passed through time series k-means for temporal and DBSCAN algorithm for spatial clustering (unsupervised learning). Output of DBSCAN is passed to deep learning models for supervised learning. Blue shapes correspond to unsupervised learning methods, while black ones denote deep learning methods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and the process ends when it reaches a convergence threshold on the amount of cluster center shifts. The major limitation of using k-means is that it does not account for the temporal nature of the datasets and treats the input time series as a group of t features, where t is the number of time steps in the data. We, therefore, deploy the time series k-means (TSKM) algorithm using dynamic time warping (DTW) following earlier works (Berndt and Clifford, 1994; Huang et al., 2016; Niennattrakul and Ratanamahatana, 2007) as the distance metric for computing time series distances between the InSAR pixels. A unique benefit of using DTW is that it is invariant to signal shifts, scales, or Doppler effects in the time domain (Cai et al., 2021). It measures the similarity between two temporal sequences which do not align precisely in time. The DTW distance computation is shown in Equation (1), where x and y denote two sample time series, d is the Euclidean distance, and $\pi = [\pi_0, \pi_1, \dots, \pi_k]$ denote index pairs.

$$DTW(x, y) = \min_{\pi} \sqrt{\sum_{i,j \in \pi} d(x_i, y_j)^2} \quad 1$$

Again, having DTW as a minimization function for computing the distance has the limitation of not being differentiable everywhere. Another variant of the DTW metric is the *softdtw* metric proposed in (Cuturi and Blondel, 2017), which replaces the min function in Equation (1) with the *softmin* function given in Equation (2).

$$\begin{aligned} \text{softdtw}(x, y) &= \underbrace{\text{softmin}}_{\pi} \sqrt{\sum_{i,j \in \pi} d(x_i, y_j)^2}, \pi = [\pi_0, \pi_1, \dots, \pi_n] \\ \text{softmin}(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n) &= -\gamma \log \sum_i e^{-\alpha_i/\gamma} \end{aligned} \quad 2$$

The *softdtw* function uses a log sum exponential formulation to guarantee differentiability everywhere (Cuturi and Blondel, 2017). In Equation (2), γ is a smoothing hyperparameter, e is the exponential operator, and $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$ denote computed distances. More details on DTW can be found in (Portilla et al., 2019) and on *softdtw* in (Cuturi and Blondel, 2017; Petitjean et al., 2011; Sakoe and Chiba, 1978; Zhang et al., 2017). Here, we used the *softdtw* method to do the time-series clustering.

Surface displacement detected by TS-InSAR is spatially and temporally variable, and above we attempt to cluster the measurement pixels primarily using displacement time series. We further note that a similar underlying physical process (such as groundwater pumping) may drive a surface deformation of similar temporal behavior at different locations. However, clusters generated from the TSKM algorithm consider only the temporal nature of the measurement pixels. Thus, two pixels with similar temporal behavior at a significant distance and associated with two different driving mechanisms may still end up in the same cluster. This pertains to the characteristic of the clustering algorithm that is inherently translation invariant. However, to achieve the desired clustering outcomes, we need to split a cluster into multiple smaller clusters if it includes pixels separated at far spatial distances. To estimate the spatial correlation length of the displacement among the measurement points, we generate a co-variogram for the displacement time series. The co-variogram is computed using Equation (3) proposed in (Chiles and Delfiner, 2012), where h_c is the distance bin, d is the vector of displacement time series and N is the number of data point pairs. The terms r_i and s_i denote randomly generated measurement pixel location pairs such that $0 < r < n_p$, $0 < s < n_p$, where n_p is the number of InSAR measurement pixels used in the analysis. We use displacements at r and s locations for computing displacement co-variogram, where $(.)$ represents element-wise multiplication.

$$C(h_c) = \frac{1}{2N} \sum_{i=1}^N d(r_i) \cdot d(s_i) \quad 3$$

We pass the output of the TSKM algorithm to the Density-based spatial

clustering of applications with noise (DBSCAN) algorithm to overturn the clustering done by TSKM in cases of ignoring the spatial separation. DBSCAN algorithm works on clustering points assuming that densely located points in space belong to the same cluster. The algorithm is subjective and includes two input parameters: distance threshold ϵ and a minimum number of points \min (generally $> D + 1$, where D is data dimensionality). We use ϵ as a distance threshold to split clusters obtained by the TSKM algorithm into smaller ones with the same label. To arrive at suitable values of ϵ for each cluster, we use information retrieved from the co-variogram and visual analysis of the individual clusters.

The output from the machine learning approach above is suitable for general users. However, computing the DTW distance requires $O(n^2)$ computational time, where O stands for the big-O notation, showing the worst case complexity of the algorithm run and n stands for the number of data points. The computation is cumbersome for millions of points in the case of SAR interferograms. Further, the prediction time using TSKM is significantly large (order of days). However, one definite advantage here is that we obtain reliable cluster labels which can be used to formulate and perform the DL based supervised training explained in section 3.2, finally yielding mined information from InSAR spatio-temporal displacement estimates.

3.2. Supervised learning with deep learning models

With a rationale to reduce the computation time, we propose two clustering methods that use deep learning models that are quick at training and prediction and suited for analyzing time series input. The first model is based on LSTM, taking the InSAR time series displacement as input and the cluster map generated above through unsupervised learning as training labels. Fig. 3 shows the general structure of an LSTM cell. Equations 4.1–4.4 present the operation of the LSTM cell. Here, x_t , y_t, x_{t-1} , and y_{t-1} denote inputs and outputs for time steps t and $t-1$. The terms w and b represent weights and biases (randomly initialized) tweaked during the model learning. Both the forget and the input gates use the input at the present step (x_t) and the output of the previous time step (y_{t-1}) in a sigmoid activation function σ_{sm} (Equations (4.1) and (4.2)). The forget gate has the job of discarding non-useful signals in the time series, while the input gate retains useful information from the time series. The outputs of the forget and input gates are used to compute the cell state (Equation (4.3)). The cell state is updated by using the predicted cell state \tilde{S}_t , again using x_t and y_{t-1} in a hyperbolic tangent activation, and the cell state at time step $t-1$ (equation (4.2)). The output gate also uses x_t and y_{t-1} in a sigmoid activation function and is used to compute the output of the present time step y_t (Equation (4.4)). This output and cell state go to the next LSTM cell, and in this way, the LSTM cell keeps the information about the time series data and learns from it based on the output label. The symbols '+' and '×' denote pointwise vector addition and multiplication, respectively.

$$f_{g_t} = \sigma_{sm}(w_{fg}[y_{t-1}, x_t] + b_{fg}) \quad \text{Forget gate} \quad 4.1$$

$$\begin{aligned} i_{n_t} &= \sigma_{sm}(w_{in}[y_{t-1}, x_t] + b_{in}) \\ \tilde{S}_t &= \tanh(w_s[y_{t-1}, x_t] + b_s) \end{aligned} \quad \text{Input gate} \quad 4.2$$

$$S_t = f_{g_t} \times S_{t-1} + i_{n_t} \times \tilde{S}_t \quad \text{Cell state} \quad 4.3$$

$$\begin{aligned} out_t &= \sigma_{sm}(w_{out}[y_{t-1}, x_t] + b_{out}) \\ y_t &= out_t \times \tanh(S_t) \end{aligned} \quad \text{Output gate} \quad 4.4$$

Fig. 4 shows the architecture of the proposed LSTM model. The model has an input layer, two hidden LSTM layers, and a fully connected layer as the output layer. This model uses the MT-InSAR displacement as input for training and prediction. The input layer dimension equals the number of pixels times the number of time steps times the number of

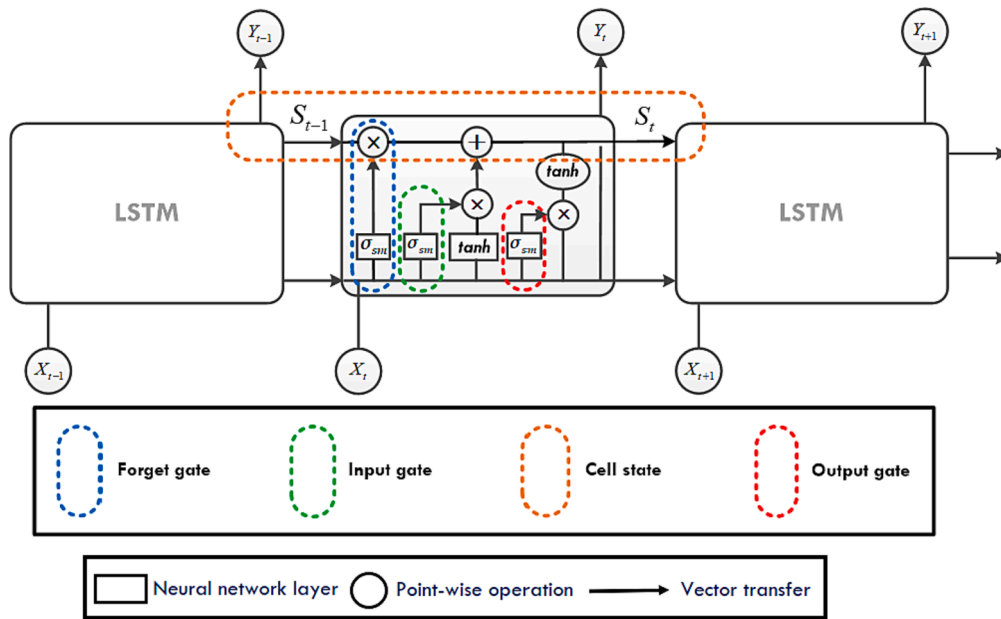


Fig. 3. Structure of an LSTM cell adapted from (Olah, 2015). Operations of input, forget, output gates, and cell state are shown in blue, green, orange and red color respectively. Cells to the left and right also contain the elements shown in the central LSTM cell. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

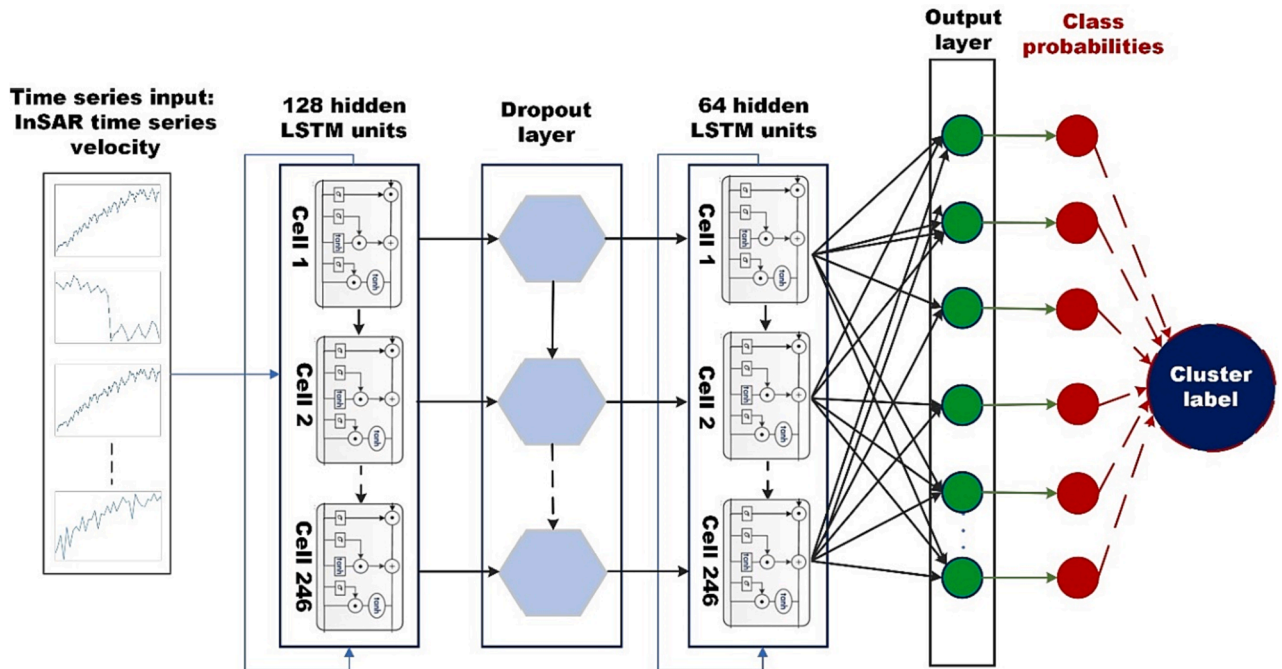


Fig. 4. Architecture of the proposed LSTM model for clustering MT-InSAR time series.

features. We use two LSTM layers which use the temporal input to derive useful information while discarding irrelevant components of the time series. To this end, the first LSTM layer has 128 units and 246 cells. The number of cells corresponds to the number of time steps in the time series data. Each LSTM cell acts as a neural network, having three different gates: forget gate, input gate, and output gate, and a separate cell state. The output of the LSTM layer goes to a dropout layer, where units are switched off randomly to avoid overfitting or chance agreement. This output goes to the next LSTM layer. This LSTM layer has 64 units for learning and is connected to a fully connected layer with the number of units equal to the number of clusters detected from the

DBSCAN output (training labels). The output layer gives the class probabilities based on its learning over multiple epochs, and from these probabilities, the class with the maximum probability and above a threshold (generally 0.5) is chosen as the final cluster label.

LSTMs work well with time series inputs by storing and propagating important contextual information by means of its cell states. However, we also have access to static inputs, which could assist with the clustering. The static inputs include the mean LOS velocity, mean coherence, incidence angle, and velocity standard deviation for each pixel. Further, we initially used a forward LSTM layer for learning, which could learn the forward progression of displacement time series by keeping a record

of important sequence information seen earlier in the series. However, it may be significant to see both forward and reverse flow of the input time series to derive useful contextual information. Therefore, we propose another deep learning model to incorporate static inputs in the learning. We call the new model LSTM + Perceptron, which combines Bidirectional LSTM and perceptron layers for training. The Bidirectional LSTM adds another LSTM layer to the general LSTM layer with reversed input temporal information and combines the general and additional LSTM layers to give a time series output learnt from both forward and backward progressions (Siarni-Namini et al., 2019). The perceptron layer is fully connected, with each layer's neuron connected to every other neuron of the subsequent layer. This layer uses the static inputs (mean LOS velocity, mean coherence, incidence angle, and standard deviation) for learning. The complete model hence performs a combination of temporal and static feature learning. As seen in Fig. 5, the model consists of four LSTM layers and two Perceptron layers. The time series displacement is passed to the LSTM layer as input, while the static input, i.e., mean velocity, the standard deviation of velocity, coherence, and incidence angle, is passed as input to the Fully Connected Perceptron (FCP) layer. The output of two layers of LSTM and one layer of the perceptron is combined at a new layer which is LSTM + FCP. This layer concatenates both the recurrent and static inputs, and outputs go into another combined FCP-connected layer. The output of the two sequential LSTM + FCP layers goes to the output layer, which is again a fully connected layer with the number of units equal to the number of clusters generated by the DBSCAN algorithm. We use the categorical cross entropy as the loss function to evaluate the model error during training for both LSTM and LSTM + Perceptron, as shown in Equation (5), where y_i and \hat{y}_i are the true and predicted cluster, respectively. It is generally used in classifying multi-label (more than two) data. Other training hyperparameters for the model are shown in Table 1. Total number of trainable parameters for the LSTM and LSTM + Perceptron models are 335,265 and 272,673, respectively.

$$CCE = - \sum_{i=1}^N y_i \log(\hat{y}_i) \quad (5)$$

Table 1

Architecture of LSTM and LSTM + Perceptron models for supervised learning. N stands for the number of pixels used for training.

LSTM + Perceptron model		LSTM model	
Layer	Dimension	Layer	Dimension
Input	$n \times 246 \times 1$	Input	$n \times 246 \times 1$
Bidirectional LSTM + Dropout (0.2)	$n \times 246 \times 256$	LSTM + Dropout (0.2)	$n \times 246 \times 128$
Bidirectional LSTM + Dropout (0.2)	$n \times 128$	LSTM + Dropout (0.2)	$n \times 256$
Static Input	$n \times 4$ (features)	Output	$n \times 33$
Perceptron	$n \times 128$		
Concatenated Timeseries and Static (Fully connected)	$n \times 256$		
Fully connected	$n \times 128$		
Output	$n \times 33$		

A confusion (or error) matrix to evaluate the classification quality is not trivial to present for this study due to the possibility of obtaining a large number of clusters. Thus, we try to simplify the visualization of the confusion matrix, highlighting the clusters for which the prediction is incorrect. The confusion matrix is a square matrix of size $m \times m$, with m as the number of clusters. Each row entry represents an actual cluster label, and each column entry represents a predicted cluster label. Diagonal entries (actual label = predicted label) show correct classifications, while off-diagonal entries show incorrect predictions (actual label \neq predicted label). Overall accuracy for clustering is computed based on Equation (6), where TP, TN, FP and FN stand for true positives (label k predicted as label k), true negatives (label not k and also predicted not be k), false positives (label not k but predicted as k) and false negatives (label k but predicted as not k), respectively. More details on multi-class confusion matrix evaluation can be found in (Grandini et al., 2020). To highlight incorrect predictions in a matrix computed for millions of pixels, we normalize the confusion matrix $CM(i,j)$ by converting each entry (i,j) between 0 and 1 via dividing by the total number of pixels in a row. With this arrangement, the sum of each row becomes 1. The terms i

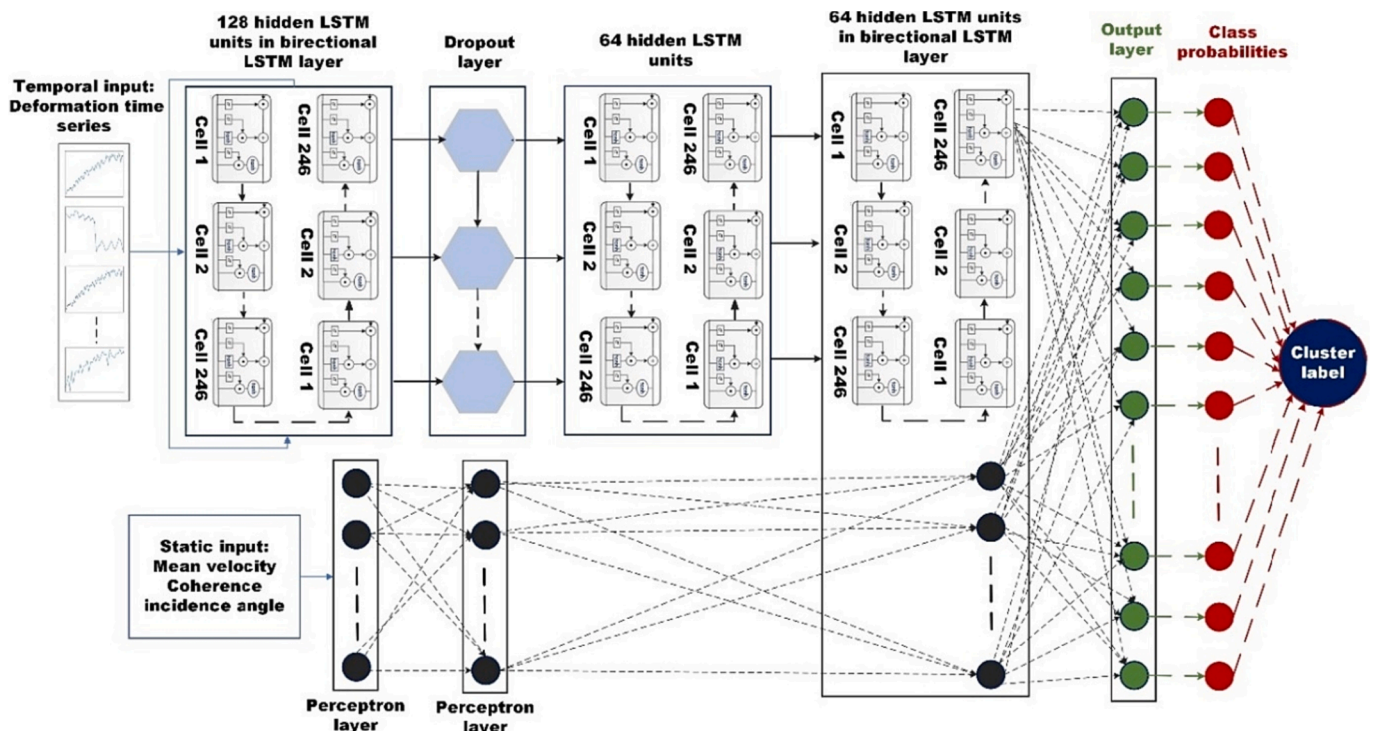


Fig. 5. Architecture of the LSTM + Perceptron model used to cluster InSAR time series.

and j denote the rows and columns. Results and discussions will follow in the next section.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad 6$$

Once we obtain the clustering results from the semi-supervised learning, we try to find important signals and expel the unimportant ones from the generated cluster map. We observe individual cluster behaviors over space by computing standard distance, a measure of spatial dispersion used in Geostatistics. The coordinates of points occurring in a cluster are used to compute the standard distance S_d shown in Equation (7), where (X_i, Y_i) denote point pairs within a cluster and (X_c, Y_c) represent the cluster mean center. From S_{dx} and S_{dy} , the standard distances in X and Y directions, we compute and draw two-dimensional standard deviation ellipses for each cluster, guiding us to examine individual clusters' spatial variability visually. We use the distance threshold of 5 km obtained from the co-variogram on S_d to decide whether a cluster should be retained or removed. The retained clusters will be notified as hotspots of deformation, showing distinct deformation activity in the study area while the removed clusters (high variability) represent unimportant signals treated as noise.

$$S_d = \sqrt{\frac{\sum (X_i - X_c)^2 + \sum (Y_i - Y_c)^2}{n}} \quad X_c = \frac{\sum X_i}{n}, Y_c = \frac{\sum Y_i}{n} \quad 7$$

4. Results and discussion

The time series clustering results using TSKM and subsequent DBSCAN execution are shown in Fig. 6. A total of 7 clusters were formed based on the Silhouette score, which gave the maximum score (0.52) for 7 clusters. As seen in Fig. 6a, the TSKM approach efficiently extracted

the deformation features from the InSAR time series deformation. However, this method cannot separate two similar but spatially apart time series signals from each other. For example, the areas shown with white rectangles in Fig. 6a have the same cluster because of similar time series behavior. However, they are spatially separated and thus are affected by different underlying processes. We subsequently estimated the spatial correlation lengths by randomly generated 10^7 pairs from the dataset for computing the sample co-variogram. The co-variogram is provided in Fig. 7, showing covariance values with increased spatial distance. We find a sharp drop in the covariance value at about 5 km, and it remains low afterward, indicating the spatial correlation between two time series is likely to be insignificant for pixels apart at a distance larger than 5 km.

The DBSCAN output generated using TSKM generated clusters as input is shown in Fig. 6b. DBSCAN overcomes the issue of spatial separation but creates 33 clusters out of the initial 7 generated by TSKM, some of which carry no useful information or background signals. We observe from the co-variogram that a distance threshold of 5 km holds for spatial correlation in the whole dataset. Further, by visually observing the cluster assignments done by TSKM, we chose ϵ values between a range of 1 to 5 km for individual clusters. We assume that a common deformation activity should not affect areas separated by distance $\geq \epsilon$. The min values were chosen by carefully observing the density of InSAR pixels in individual clusters. Table 2 gives the values of the input parameters chosen for each cluster in the DBSCAN algorithm. Fig. 6c shows mean time series displacements for cluster detected as 1, 2 and 7 by DBSCAN algorithm and cluster 1 (single cluster) by TSKM. We observe that the mean time series curves for these clusters are not identical but labeled as single clusters by TSKM. This probably is due to the displacement variances values falling within the threshold, keeping pixels in the same cluster. Similarly, Fig. 6d shows mean time series

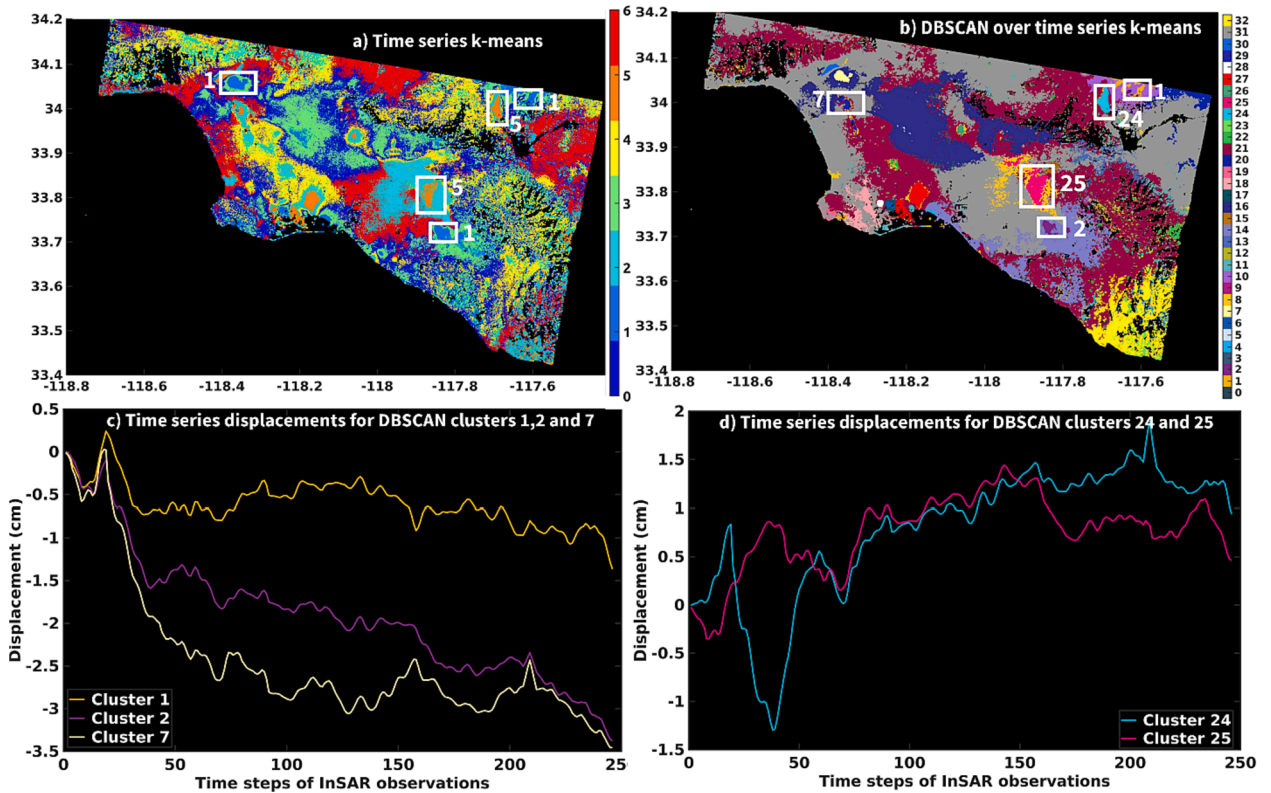


Fig. 6. Unsupervised clustering results from machine learning approach. a) Results obtained from time series k-means (TSKM) algorithm, b) Output of DBSCAN algorithm run over clusters detected with TSKM. White rectangles with numbers marked indicate deformation types classified as single clusters by TSKM and as different clusters by DBSCAN. c) and d) show the clusters identified to be same by TSKM and different by DBSCAN algorithm. c) Mean time series displacements for DBSCAN clusters 1, 2 and 7 detected by TSKM as cluster 1. d) Mean time series displacements for DBSCAN clusters 24 and 25 detected by TSKM as cluster 5.

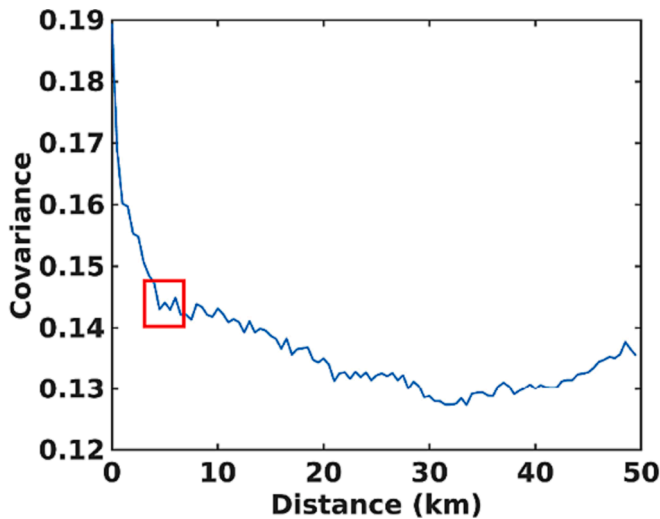


Fig. 7. Co-variogram computed from time series displacement of elite pixels in WabInSAR processing. Red rectangle indicates region where covariance values for corresponding distances show a significant drop and remain low beyond this distance. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Parameters for splitting individual clusters using DBSCAN algorithm.

Cluster No.	Distance threshold (decimal degrees)	Minimum no. of points
1	0.02	500
2	0.01	800
3	0.05	500
4	0.02	1000
5	0.03	300
6	0.03	300
7	0.05	500

displacements for DBSCAN clusters 24 and 25 detected as cluster 5 by TSKM. We again observe significant rise and fall in the two clusters (24 and 25) at time steps near 50 while the rest of the time series is similar. Separating these clusters using TSKM would require smaller thresholds on time series displacements resulting in more no. of clusters and may also result in unnecessary additional clusters. DBSCAN, however, solves the dual purpose of spatial clustering and splitting these cluster assignments. Once these clusters are obtained, we estimate the irrelevant information from the resultant clustered map.

The cluster map obtained above from the spatiotemporal clustering is used as input to LSTM and LSTM + Perceptron networks. Fig. 8 shows the predicted cluster maps obtained using LSTM and LSTM + Perceptron network predictions. Both LSTM and LSTM + Perceptron were trained with the output cluster map generated from the unsupervised learning approach. We divided the complete dataset into training and test datasets with a ratio of 50 % each. The training and test samples were randomly selected in space from the InSAR time series. We obtained 97.3 % and 93.5 % accuracy for the test dataset using the LSTM and LSTM + Perceptron networks, respectively. Fig. 8a and 8b show that the results are very close to those obtained from unsupervised clustering. White rectangles marked in Fig. 8a and 8b show areas with a mismatch between the detected clusters. Fig. 8c and 8d show close up view of the area found to be mismatching.

We visualize the confusion matrix for the predicted clusters generated from LSTM + Perceptron and LSTM models in Fig. 9, highlighting the clusters for which the prediction is incorrect. We observe larger off-diagonal entries in the map corresponding with the LSTM + Perceptron model. We observe that the LSTM network attained better accuracy than the LSTM + Perceptron network. This goes against the general ideology

that deeper architecture with additional features is supposed to learn better. However, relating this to bias variance trade off (Briscoe and Feldman, 2011) and through recent studies (Uzair and Jamil, 2020), we also observe that adding more features can introduce overfitting, and adding more hidden layers to a model can also make the learning curve more complex, resulting in overfitting and vanishing gradients, making the model performance worse. In this study, where we deal with surface deformation, the model performed better with only time series displacement as input, compared to a deeper model with more input features. We observe that making a model deeper does not help improve the model performance for this study. It is also worth mentioning that the latter used both static and temporal data for learning, while LSTM generated clusters using the deformation time series only.

To eventually derive prominent deformation events occurring in LA, we tried to find and expel most of the unimportant signals from the cluster map, giving a much clearer picture of which areas in LA are affected by significant deformation and their differences in spatiotemporal behavior. Standard distances computed for the clusters predicted by LSTM perceptron and LSTM methods are shown in Fig. 10a and 10b respectively. The standard deviation ellipses for the predicted clusters derived from the corresponding S_{dX} and S_{dY} are shown in Fig. 10c and 10d for the LSTM + Perceptron and LSTM methods, respectively, guiding us to examine individual clusters' spatial variability visually. We use the distance threshold of 5 km obtained from the co-variogram on S_d to decide whether a cluster should be retained or removed. We also confirm this from the ellipses shown in Fig. 10c and 10d, where we see a large spread for clusters crossing the 5 km threshold values.

The cluster maps after the removal of unwanted clusters are shown in Fig. 11, where non-domain experts readily interpret the map and hot-spot areas to stand out. We observe that LSTM retains more clusters than the LSTM + Perceptron, indicating that the clusters generated using the latter had higher spatial dispersion. With the confusion matrix showing higher prediction accuracy and the spatial dispersion plots showing lower variability for clusters detected using LSTM compared to LSTM + Perceptron, we find that the LSTM model performed better. The method can generate cluster maps from InSAR time series displacement obtained by any processing software, provided the user has the time series displacements, mean LOS velocity, incidence angle, coherence, and pixel locations. If only the time series are available, the approach can still work with a combination of TSKM + DBSCAN followed by the LSTM model. We intend to release the programs and models related to this work, which can be used to perform unsupervised and subsequent supervised learning over different sites across the globe. The implementation mainly uses Python, with a few postprocessing scripts written in Matlab. We used the skit-learn library for machine learning, while Keras and Tensorflow libraries are used for developing deep learning models (Abadi et al., 2016; Gulli and Pal, 2017; Kramer and Kramer, 2016). A detailed description of the proposed methodology follows next.

The efficiency of unsupervised learning can also be improved by working with the statistical analysis of clusters obtained from TSKM + DBSCAN outputs, as implemented for the deep learning models. Based on spatial correlation lengths and standard distances, the clusters representing distinct statistical behavior for InSAR-derived deformation can be retained while others can be removed. However, TSKM is computationally intensive, and we just use it to generate the initial cluster labels for supervised learning with the LSTM and LSTM + Perceptron models, which deal with the order of data we process with InSAR in significantly less computational time. Another limitation of the machine learning-based unsupervised learning proposed here is the subjectivity involved with the DBSCAN algorithm. The parametric algorithm still requires the user to decide on the distance threshold to separate clusters separated significantly far in space. However, this algorithm may not be required for studies requiring the detection of particular deformation events also occurring at distant locations. For example, if the areas to be studied are affected by only a few types of deformation, such as landslides, land subsidence, volcanic activity, and glacial uplift, it will be useful to

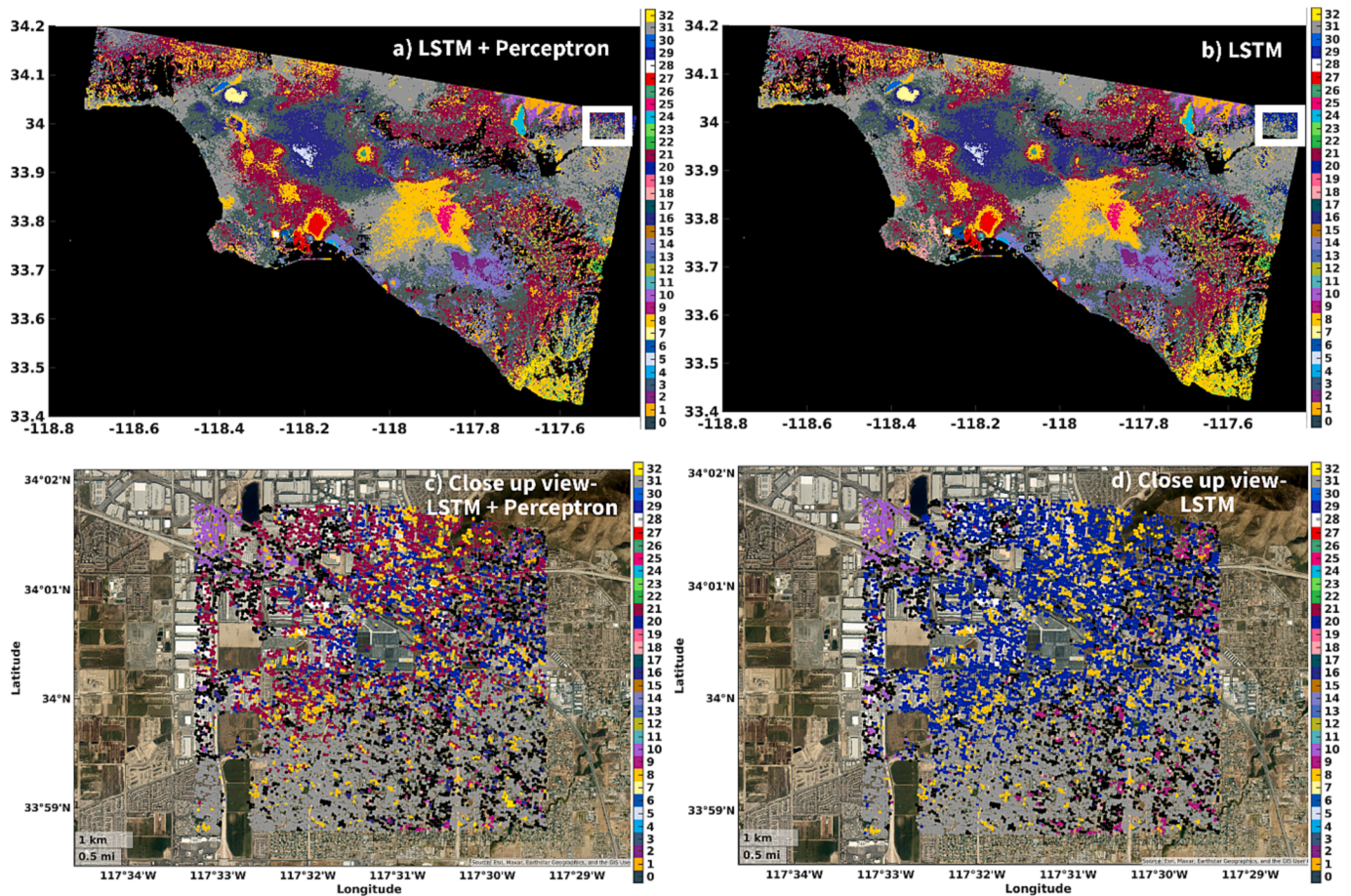


Fig. 8. Results from LSTM + Perceptron (a) and LSTM (b) predictions for the test points. White rectangles in (a) and (b) mark the areas where the predictions differ significantly. (c) and (d) show a close-up view of the area with different predictions made by LSTM + Perceptron and LSTM models, respectively. Units for X and Y axes are decimal degrees for each panel.

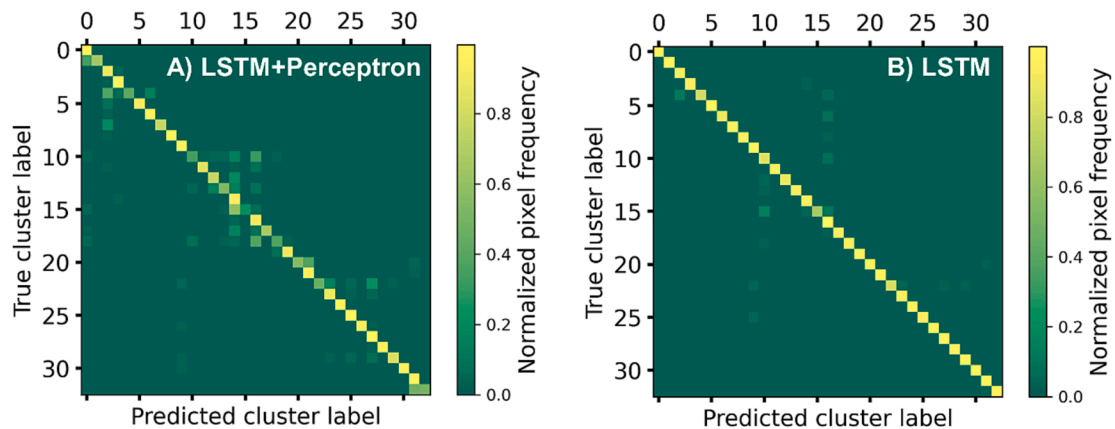


Fig. 9. Confusion matrices generated for the LSTM + Perceptron (a) and LSTM (b) model predictions. The frequency of pixels has been normalized for an easier understanding of differences in prediction. X and Y axes refer to predicted label and true label respectively.

classify landslides as one cluster irrespective of their location in the study area. This would simplify the algorithm, with only TSKM for clustering and LSTM and LSTM + SLP for supervised learning.

As evident in Fig. 11b, the LSTM method successfully identifies and isolates deformation hotspots in the Los Angeles dataset. Concerning the underlying process, clusters 0 and 7 pertain to groundwater pumping and aquifer recharge operations at the Santa Ana system (Riel et al., 2018). Clusters 2, 6, 11, 12, 15, and 16 show deformation associated

with activities on the Newport-Inglewood Fault and oil and gas production from the Wilmington oil field near Seal Beach and Long Beach (Brooks et al., 2007). Clusters 3 and 14 indicate deformation hotspots at the city center, whose cause is not known. Clusters 3 and 10 likely show the post-seismic deformation associated with the 2014 La Habra Earthquake (Donnellan et al., 2015). Clusters 5 and 17 represent deformation caused by the Los Angeles D-line extension transit project. Cluster 8 highlights the deformation due to landslides at Santa Ana

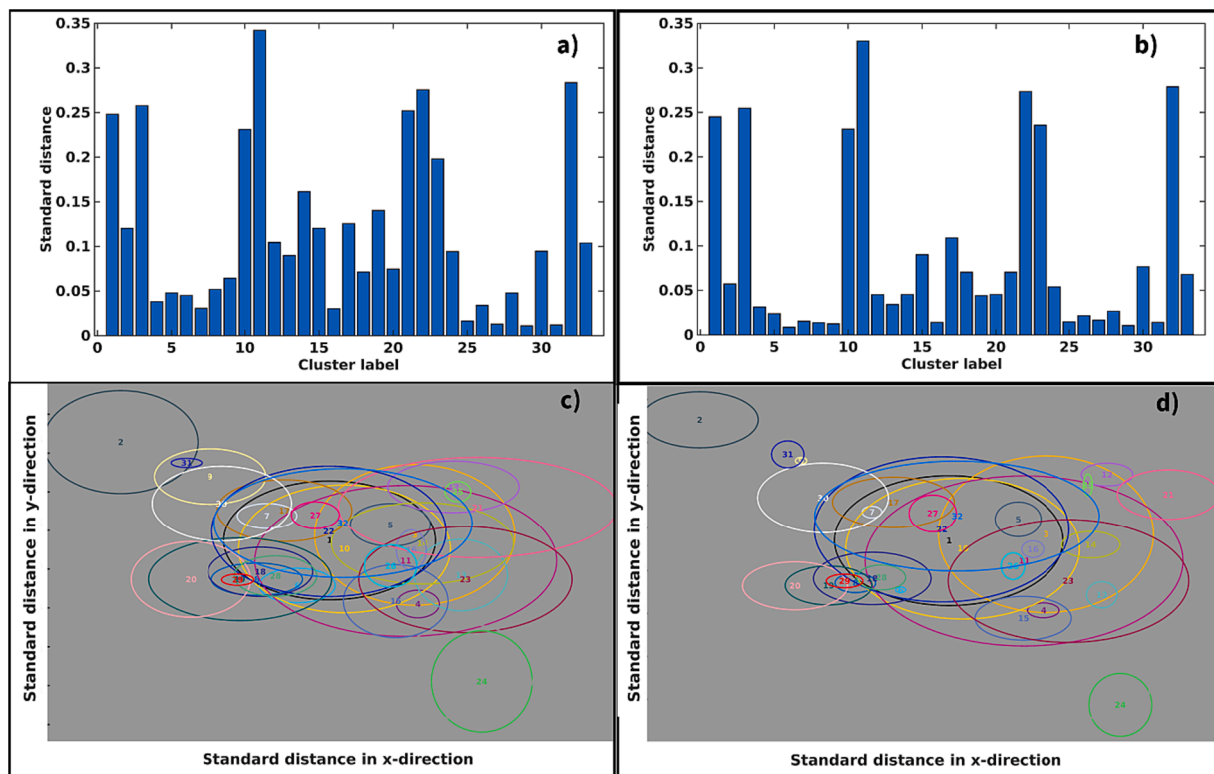


Fig. 10. Spatial statistics computed over individual clusters detected by LSTM + Perceptron and LSTM methods. a) and b) show Bar charts of standard distances for LSTM + Perceptron and LSTM clusters, respectively. c) and d) show ellipses depicting the two-dimensional standard deviation of individual clusters for the LSTM + Perceptron and LSTM methods, respectively.

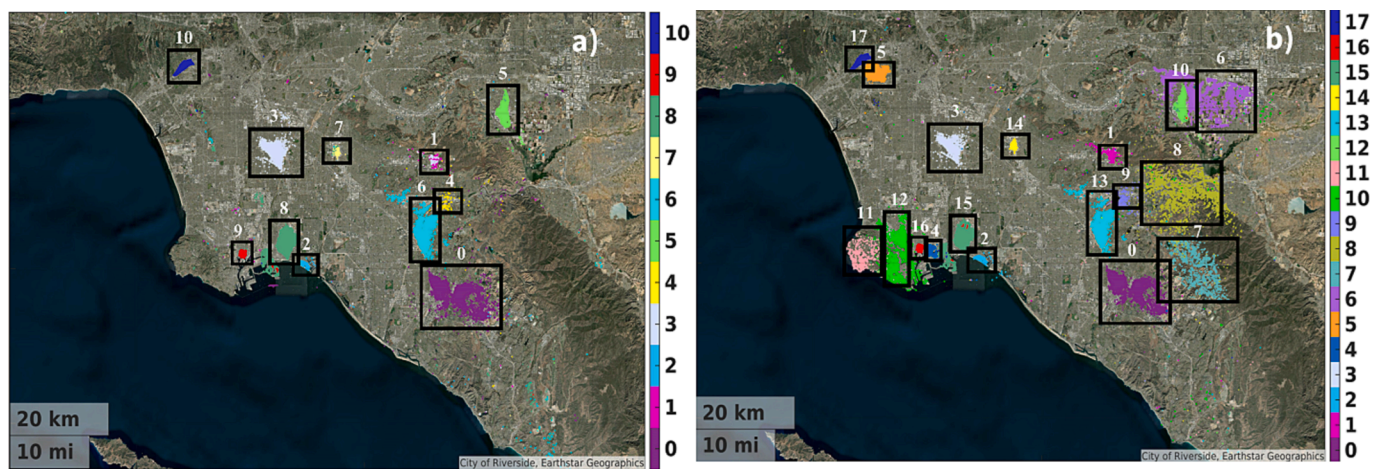


Fig. 11. Final cluster maps from a) LSTM + Perceptron and b) LSTM methods, respectively. These maps are derived after discarding clusters that do not carry useful information based on the threshold values of standard distances and co-variogram analysis. Rectangles with black boundaries and cluster ids on top mark the deformation hotspots detected by the respective methods.

Mountain (Morton et al., 2006). Clusters 1 and 9 seem to show deformation along the Whittier Fault (Donnellan et al., 2015).

5. Conclusion

We present a novel semi-supervised learning approach combining time series and spatial clustering, LSTM, and perceptron-based deep learning to extract useful information from time series deformation derived from InSAR. The unsupervised approach works by initially clustering the elite pixels detected in InSAR processing based on their temporal behavior and subsequently applying spatial clustering to

separate signals left unseparated in space by the time series clustering. The cluster labels generated using the spatiotemporal clustering act as training labels for the LSTM and LSTM + Perceptron based deep learning architectures, which minimize the time requirements for the overall clustering task. These methods are trained to establish a mathematical relation between the deformation time series and the associated spatiotemporal clusters.

The results from spatiotemporal clustering are convincing, efficiently distinguishing the deformation events in LA. Further to these results, the LSTM and LSTM + Perceptron methods also learn well from the training labels generated from spatiotemporal clustering, achieving 96 % and 93

% accuracy, respectively, and significantly reducing the computational time. Observing the results obtained from (i) LSTM using only time series and (ii) LSTM + Perceptron using both temporal and static data, we see slightly better results for LSTM, trained only with time series. This signifies that we can derive significant deformation hotspots using only time series deformation data. The proposed information extraction approach can be integrated with any multi-temporal InSAR processing chains and shows a novel way of presenting InSAR deformation maps, more interpretable to non-domain experts, general users, and decision-makers.

CRedit authorship contribution statement

Ashutosh Tiwari: Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft. **Manoochehr Shirzaei:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Dataset link has been provided in the manuscript.

Acknowledgment

The authors are grateful to the ESA for providing Sentinel-1 IW images. NSF and USGS have provided funding for this work. Authors thank Department of Energy for supporting this study.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., 2016. Tensorflow: a system for large-scale machine learning. *Osdi. Savannah, GA, USA*, pp. 265–283.
- Anantrasirichai, N., Biggs, J., Albino, F., Bull, D., 2019a. The application of convolutional neural networks to detect slow, sustained deformation in InSAR time series. *Geophys. Res. Lett.* 46, 11850–11858.
- Anantrasirichai, N., Biggs, J., Albino, F., Bull, D., 2019b. A deep learning approach to detecting volcano deformation from satellite imagery using synthetic datasets. *Remote Sens. Environ.* 230, 111179.
- Ansari, H., De Zan, F., Bamler, R., 2017. Sequential estimator: Toward efficient InSAR time series analysis. *IEEE Trans. Geosci. Remote Sens.* 55, 5637–5652.
- Ansari, H., De Zan, F., Bamler, R., 2018. Efficient phase estimation for interferogram stacks. *IEEE Trans. Geosci. Remote Sens.* 56, 4109–4125.
- Bakon, M., Oliveira, I., Perissin, D., Sousa, J.J., Papco, J., 2017. A data mining approach for multivariate outlier detection in postprocessing of multitemporal InSAR results. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sensing* 10, 2791–2798.
- Belkin, M., Hsu, D., Ma, S., Mandal, S., 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proc. Natl. Acad. Sci.* 116, 15849–15854.
- Briscoe, E., Feldman, J., 2011. Conceptual complexity and the bias/variance tradeoff. *Cognition* 118, 2–16.
- Brooks, B., Merrifield, M., Foster, J., Werner, C., Gomez, F., Bevis, M., Gill, S., 2007. Space geodetic determination of spatial variability in relative sea level change, Los Angeles basin. *Geophys. Res. Lett.* 34.
- Cai, B., Huang, G., Samadiani, N., Li, G., Chi, C.-H., 2021. Efficient time series clustering by minimizing dynamic time warping utilization. *IEEE Access* 9, 46589–46599.
- Chang, L., Hanssen, R.F., 2015. A probabilistic approach for InSAR time-series postprocessing. *IEEE Trans. Geosci. Remote Sens.* 53, 421–430.
- Chen, X.-W., Lin, X., 2014. Big data deep learning: challenges and perspectives. *IEEE Access* 2, 514–525.
- Chen, M., Qian, Z., Boers, N., Jakeman, A.J., Kettner, A.J., Brandt, M., Kwan, M.-P., Batty, M., Li, W., Zhu, R., 2023. Iterative integration of deep learning in hybrid Earth surface system modelling. *Nat. Rev. Earth Environ.* 4, 568–581.
- Chiles, J.-P., Delfiner, P., 2012. *Geostatistics: modeling spatial uncertainty*. John Wiley & Sons.
- Cuturi, M., Blondel, M., 2017. Soft-dtw: a differentiable loss function for time-series. *Int. Conf. Mach. Learn. PMLR* 894–903.
- Dargan, S., Kumar, M., Ayyagari, M.R., Kumar, G., 2020. A survey of deep learning and its applications: a new paradigm to machine learning. *Arch. Computat. Methods Eng.* 27, 1071–1092.
- De Luca, C., Bonano, M., Casu, F., Manunta, M., Manzo, M., Onorato, G., Zinno, I., Lanari, R., 2018. The parallel SBAS-DInSAR processing chain for the generation of national scale sentinel-1 deformation time-series. *Procedia Comput. Sci.* 138, 326–331.
- Devara, M., Tiwari, A., Dwivedi, R., 2021. Landslide susceptibility mapping using MT-InSAR and AHP enabled GIS-based multi-criteria decision analysis. *Geomat. Nat. Haz. Risk* 12, 675–693.
- Donnellan, A., Grant Ludwig, L., Parker, J.W., Rundle, J.B., Wang, J., Pierce, M., Blewitt, G., Hensley, S., 2015. Potential for a large earthquake near Los Angeles inferred from the 2014 La Habra earthquake. *Earth Space Sci.* 2, 378–385.
- Eppler, J., Rabus, B., 2012. Monitoring urban infrastructure with an adaptive multilooking InSAR technique. *Fringe* 2011 (697), 68.
- Festa, D., Novellino, A., Hussain, E., Bateson, L., Casagli, N., Confuorto, P., Del Soldato, M., Raspini, F., 2023. Unsupervised detection of InSAR time series patterns based on PCA and K-means clustering. *Int. J. Appl. Earth Obs. Geoinf.* 118, 103276.
- Gao, Q., Crosetto, M., Monserrat, O., Palama, R., Barra, A., 2022. Infrastructure monitoring using the interferometric synthetic aperture radar (InSAR) technique. *Int. Arch. Photogram. Remote Sens. Spat. Inform. Sci.* 43, 271–276.
- Geman, S., Bienenstock, E., Doursat, R., 1992. Neural networks and the bias/variance dilemma. *Neural Comput.* 4, 1–58.
- Gulli, A., Pal, S., 2017. *Deep learning with Keras*. Packt Publishing Ltd.
- Hasan, M.F., Smith, R., Vajedian, S., Pommerenke, R., Majumdar, S., 2023. Global land subsidence mapping reveals widespread loss of aquifer storage capacity. *Nat. Commun.* 14, 6180.
- Hooper, A., Zebker, H., Segall, P., Kampes, B., 2004. A new method for measuring deformation on volcanoes and other natural terrains using InSAR persistent scatterers. *Geophysical Research Letters* 31.
- Irrgang, C., Boers, N., Sonnewald, M., Barnes, E.A., Kadow, C., Staneva, J., Saynisch-Wagner, J., 2021. Towards neural Earth system modelling by integrating artificial intelligence in Earth system science. *Nat. Mach. Intell.* 3, 667–674.
- Karimzadeh, S., Matsuoka, M., Ogushi, F., 2018. Spatiotemporal deformation patterns of the Lake Urmia Causeway as characterized by multisensor InSAR analysis. *Sci. Rep.* 8, 5357.
- Kramer, O., Kramer, O., 2016. Scikit-learn. *Mach. Learn. Evolut. Strategies* 45–53.
- Kubo, H., Kunugi, T., Suzuki, W., Suzuki, S., Aoi, S., 2020. Hybrid predictor for ground-motion intensity with machine learning and conventional ground motion prediction equation. *Sci. Rep.* 10, 11871.
- Lattari, F., Rucci, A., Matteucci, M., 2022. A deep learning approach for change points detection in InSAR time series. *IEEE Trans. Geosci. Remote Sens.* 60, 1–16.
- Lee, J.-C., Shirzaei, M., 2023. Novel algorithms for pair and pixel selection and atmospheric error correction in multitemporal InSAR. *Remote Sens. Environ.* 286, 113447.
- Ma, Z., Liu, J., Aoki, Y., Wei, S., Liu, X., Cui, Y., Hu, J., Zhou, C., Qin, S., Huang, T., 2022. Towards big SAR data era: An efficient Sentinel-1 Near-Real-Time InSAR processing workflow with an emphasis on co-registration and phase unwrapping. *ISPRS J. Photogramm. Remote Sens.* 188, 286–300.
- Mateo-Garcia, G., Veitch-Michaelis, J., Smith, L., Oprea, S.V., Schumann, G., Gal, Y., Baydin, A.G., Backes, D., 2021. Towards global flood mapping onboard low cost satellites with machine learning. *Sci. Rep.* 11, 7249.
- Maubant, L., Pathier, E., Daout, S., Radiguet, M., Doin, M.P., Kazachkina, E., Kostoglodov, V., Cotte, N., Walpersdorf, A., 2020. Independent component analysis and parametric approach for source separation in InSAR time series at regional scale: application to the 2017–2018 Slow Slip Event in Guerrero (Mexico). *Journal of Geophysical Research: Solid Earth* 125, e2019JB018187.
- Millillo, P., Perissin, D., Salzer, J.T., Lundgren, P., Lacava, G., Milillo, G., Serio, C., 2016. Monitoring dam structural health from space: Insights from novel InSAR techniques and multi-parametric modeling applied to the Pertusillo dam Basilicata, Italy. *Int. J. Appl. Earth Obs. Geoinf.* 52, 221–229.
- Miller, M.M., Shirzaei, M., Argus, D., 2017. Aquifer mechanical properties and decelerated compaction in Tucson, Arizona. *J. Geophys. Res. Solid Earth* 122, 8402–8416.
- Morton, D.M., Miller, F.K., Cossette, P.M., Bovard, K.R., 2006. *Geologic Map of the San Bernardino and Santa Ana 30' X 60' Quadrangles*. California, Citeseer.
- Nikolaeva, E., Walter, T., Shirzaei, M., Zschau, J., 2014. Landslide observation and volume estimation in central Georgia based on L-band InSAR. *Nat. Hazards Earth Syst. Sci.* 14, 675–688.
- Ojha, C., Werth, S., Shirzaei, M., 2020. Recovery of aquifer-systems in Southwest US following 2012–2015 drought: evidence from InSAR, GRACE and groundwater level data. *J. Hydrol.* 587, 124943.
- Olah, C., 2015. *Understanding lstm networks*.
- Osmanoğlu, B., Dixon, T.H., Wdowinski, S., Cabral-Cano, E., Jiang, Y., 2011. Mexico City subsidence observed with persistent scatterer InSAR. *Int. J. Appl. Earth Obs. Geoinf.* 13, 1–12.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., 2011. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Petitjean, F., Ketterlin, A., Gançarski, P., 2011. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recogn.* 44, 678–693.
- Portilla, R., Heintz, B., Lee, D., 2019. *Understanding Dynamic Time Warping*. Databricks.
- Pritchard, M., Simons, M., 2004. An InSAR-based survey of volcanic deformation in the central Andes. *Geochemistry, Geophysics, Geosystems* 5.

- Radman, A., Akhoondzadeh, M., Hosseini, B., 2021. Integrating InSAR and deep-learning for modeling and predicting subsidence over the adjacent area of Lake Urmia, Iran. *Gisci. Remote Sens.* 58, 1413–1433.
- Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., Prabhat, f., 2019. Deep learning and process understanding for data-driven Earth system science. *Nature* 566, 195–204.
- Riel, B., Simons, M., Ponti, D., Agram, P., Jolivet, R., 2018. Quantifying ground deformation in the Los Angeles and Santa Ana Coastal Basins due to groundwater withdrawal. *Water Resour. Res.* 54, 3557–3582.
- Rolf, E., Proctor, J., Carleton, T., Bolliger, I., Shankar, V., Ishihara, M., Recht, B., Hsiang, S., 2021. A generalizable and accessible approach to machine learning with global satellite imagery. *Nat. Commun.* 12, 4392.
- Rosi, A., Tofani, V., Tanteri, L., Tacconi Stefanelli, C., Agostini, A., Catani, F., Casagli, N., 2018. The new landslide inventory of Tuscany (Italy) updated with PS-InSAR: geomorphological features and landslide distribution. *Landslides* 15, 5–19.
- Rouet-Leduc, B., Jolivet, R., Dalaison, M., Johnson, P.A., Hulbert, C., 2021. Autonomous extraction of millimeter-scale deformation in InSAR time series using deep learning. *Nat. Commun.* 12, 6480.
- Rousseuw, P.J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20, 53–65.
- Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* 26, 43–49.
- Shirzaei, M., 2012. A wavelet-based multitemporal DInSAR algorithm for monitoring ground surface motion. *IEEE Geosci. Remote Sens. Lett.* 10, 456–460.
- Shirzaei, M., Bürgmann, R., 2012. Topography correlated atmospheric delay correction in radar interferometry using wavelet transforms. *Geophys. Res. Lett.* 39.
- Shirzaei, M., Bürgmann, R., 2013. Time-dependent model of creep on the Hayward fault from joint inversion of 18 years of InSAR and surface creep data. *J. Geophys. Res. Solid Earth* 118, 1733–1746.
- Shirzaei, M., Bürgmann, R., Foster, J., Walter, T., Brooks, B., 2013a. Aseismic deformation across the Hilina fault system, Hawaii, revealed by wavelet analysis of InSAR and GPS time series. *Earth Planet. Sci. Lett.* 376, 12–19.
- Shirzaei, M., Walter, T.R., Bürgmann, R., 2013b. Coupling of Hawaiian volcanoes only during overpressure condition. *Geophysical Research Letters* 40, 1994–1999.
- Shirzaei, M., Manga, M., Zhai, G., 2019. Hydraulic properties of injection formations constrained by surface deformation. *Earth Planet. Sci. Lett.* 515, 125–134.
- Siarni-Namini, S., Tavakoli, N., Namin, A.S., 2019. The performance of LSTM and BiLSTM in forecasting time series, 2019 IEEE International Conference on Big Data (Big Data). IEEE, pp. 3285–3292.
- Sica, F., Calvanese, F., Scarpa, G., Rizzoli, P., 2020. A CNN-based coherence-driven approach for InSAR phase unwrapping. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5.
- Sun, Q., Zhang, L., Ding, X., Hu, J., Li, Z., Zhu, J., 2015. Slope deformation prior to Zhouqu, China landslide from InSAR time series analysis. *Remote Sens. Environ.* 156, 45–57.
- Tiwari, A., Narayan, A.B., Dikshit, O., 2020. Deep learning networks for selection of measurement pixels in multi-temporal SAR interferometric processing. *ISPRS J. Photogramm. Remote Sens.* 166, 169–182.
- Uzair, M., Jamil, N., 2020. Effects of hidden layers on the efficiency of neural networks, 2020 IEEE 23rd international multitopic conference (INMIC). IEEE 1–6.
- Wang, D., Even, M., Kutterer, H., 2022a. Deep learning based distributed scatterers acceleration approach: Distributed scatterers prediction Net. *Int. J. Appl. Earth Observ. Geoinform.* 115, 103112.
- Wang, G., Li, P., Li, Z., Liang, C., Wang, H., 2022b. Coastal subsidence detection and characterization caused by brine mining over the Yellow River Delta using time series InSAR and PCA. *Int. J. Appl. Earth Obs. Geoinf.* 114, 103077.
- Wang, Y., Zhu, X.X., Zeisl, B., Pollefeys, M., 2016. Fusing meter-resolution 4-D InSAR point clouds and optical images for semantic urban infrastructure monitoring. *IEEE Trans. Geosci. Remote Sens.* 55, 14–26.
- Whipple, K.X., Shirzaei, M., Hodges, K.V., Ramon Arrowsmith, J., 2016. Active shortening within the Himalayan orogenic wedge implied by the 2015 Gorkha earthquake. *Nat. Geosci.* 9, 711–716.
- Wnuk, K.C., 2021. Post-Processing of InSAR Persistent Scatterer Time-Series to Investigate Deformational Processes Induced by Subsurface Excavation. Colorado School of Mines.
- Xu, X., Sandwell, D.T., Klein, E., Bock, Y., 2021. Integrated Sentinel-1 InSAR and GNSS time-series along the San Andreas Fault system. *Journal of Geophysical Research: Solid Earth* 126, e2021JB022579.
- Yuan, R., Chen, J., 2022. A hybrid deep learning method for landslide susceptibility analysis with the application of InSAR data. *Nat. Hazards* 114, 1393–1426.
- Zhang, Z., Tavenard, R., Bailly, A., Tang, X., Tang, P., Corpetti, T., 2017. Dynamic time warping under limited warping path length. *Inf. Sci.* 393, 91–107.
- Zhou, L., Yu, H., Pascacio, V., Xing, M., 2022. PU-GAN: a one-step 2-D InSAR phase unwrapping based on conditional generative adversarial network. *IEEE Trans. Geosci. Remote Sens.* 60, 1–10.