communications
earth & environment

ARTICLE

https://doi.org/10.1038/s43247-023-01107-7 OPEN

Multi-station deep learning on geodetic time series detects slow slip events in Cascadia

Giuseppe Costantino[1✉], Sophie Giffard-Roisin[1], Mathilde Radiguet[1], Mauro Dalla Mura[2,3], David Marsan[1] & Anne Socquet[1]

Slow slip events (SSEs) originate from a slow slippage on faults that lasts from a few days to years. A systematic and complete mapping of SSEs is key to characterizing the slip spectrum and understanding its link with coeval seismological signals. Yet, SSE catalogues are sparse and usually remain limited to the largest events, because the deformation transients are often concealed in the noise of the geodetic data. Here we present a multi-station deep learning SSE detector applied blindly to multiple raw (non-post-processed) geodetic time series. Its power lies in an ultra-realistic synthetic training set, and in the combination of convolutional and attention-based neural networks. Applied to real data in Cascadia over the period 2007–2022, it detects 78 SSEs, that compare well to existing independent benchmarks: 87.5% of previously catalogued SSEs are retrieved, each detection falling within a peak of tremor activity. Our method also provides useful proxies on the SSE duration and may help illuminate relationships between tremor chatter and the nucleation of the slow rupture. We find an average day-long time lag between the slow deformation and the tremor chatter both at a global- and local-temporal scale, suggesting that slow slip may drive the rupture of nearby small asperities.

[1] ISTerre, Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, IRD, Univ. Gustave Eiffel, 38000 Grenoble, France. [2] GIPSA-lab, Univ. Grenoble Alpes, CNRS, Grenoble INP, 38000 Grenoble, France. [3] Institut Universitaire de France (IUF), Paris, France. ✉email: giuseppe.costantino@univ-grenoble-alpes.fr

Slow slip events (SSEs) generate episodic deformation that lasts from a few days to years[1]. Like earthquakes, they originate from slip-on faults but, unlike them, do not radiate energetic seismic waves. In the mid-1990s, Global Navigation Satellite System (GNSS) networks started to continuously monitor the ground displacement, providing evidence that SSEs are a major mechanism responsible for the release of stress in plate boundaries, as a complement to seismic rupture[2–6]. This constituted a change of paradigm for the understanding of the earthquake cycle and of the mechanics of the fault interface. Twenty years later, the characterization of the full slip spectrum and the understanding of the link between slow slip and the associated seismological signals are hindered by our capacity to detect slow slip events in a systematic manner, more particularly those of low magnitude (typically lower than M_w 6), even though a systematic and complete mapping of SSEs on faults is key for understanding the complex physical interactions between slow aseismic slip and earthquakes. Indeed, the small deformation transients associated with an SSE are often concealed in the noise[7,8], making it difficult to precisely characterize the slip spectrum and provide fruitful insights into the fault mechanics[5,9,10]. Studies dealing with the detection and analysis of SSEs often rely on dedicated signal analysis, involving visual inspection of the data, data selection, denoising, filtering, geodetic expertize, dedicated modeling methods with a fine-tuning of the parameters, and also often complementary data such as catalogs of tremor or low-frequency earthquakes (LFEs)[8,11–14]. Tremors and LFEs are weakly emergent micro-seismicity, or micro-seismicity with low-frequency content often accompanying SSEs in certain subduction zones[15,16].

The development of in-situ geophysical monitoring generates nowadays huge data sets, and machine learning techniques have been largely assimilated and used by the seismological community to improve earthquake detection and characterization[6,17–19], generating catalogs with unprecedentedly high quality[20,21] and knowledge shifts[22,23]. However, up to now, such techniques could not be successfully applied to the analysis of geodetic data and slow slip event detection because of two main reasons: (1) too few true labels exist to train machine learning-based methods, which we tackle by generating a realistic synthetic training data set, (2) the signal-to-noise ratio is extremely low in geodetic data[24,25], meaning that we are at the limit of detection capacity. One possibility is to first pre-process the data to remove undesired signals such as seasonal variations, common modes, and post-seismic relaxation signals (via denoising, filtering, trajectory modeling, or independent-component-analysis-based inversion), but this is at the cost of possibly corrupting the data. Trajectory models are often used to subtract the contribution due to seasonal variations in the GNSS time series. Yet, this can lead, in some cases, to poor noise characterization since seasonal variations can have different amplitudes in the whole time series and a signal shape that is not necessarily fully reproduced by a simple sum of sine and cosine functions. This approach can thus introduce some spurious transient signals which could be erroneously modeled as slow slip events. Another possibility would be to use independent-component-analysis-based methods to extract the seasonal variations. However, the extracted components might contain some useful signals as well as some noise characteristics that could not be removed (e.g., specific harmonics or spatio-temporal patterns). Hence, the main motivation of this work is that we do not want to model any GNSS-constitutive signals, but use the non-post-processed GNSS time series as they are, by relying on a deep-learning model which should be able to learn the noise signature, and therefore separate the noise from the relevant information (here, slow slip events). In order to develop an end-to-end model (which does not require any manual intermediate step) capable of dealing with non-post-processed geodetic measurements, it is necessary, on one hand, to set up advanced methods to generate realistic noise, taking into account the spatial correlation between stations as well as the large number of data gaps present in the GNSS time series. On the other hand, it involves developing a specific deep-learning model able to treat multiple stations simultaneously, using a relevant spatial stacking of the signals (driven by our physics-based knowledge of the slow slip events) in addition to a temporal analysis. We address these two major drawbacks in our new approach and present SSEgenerator and SSEdetector, an end-to-end deep-learning-based detector, combining the spatiotemporal generation of synthetic daily GNSS position time series containing modeled slow deformation (SSEgenerator), and a Convolutional Neural Network (CNN) and a Transformer neural network with an attention mechanism (SSEdetector), that proves effective in detecting slow slip events in raw GNSS position time series from a large geodetic network containing more than 100 stations, both on synthetic and on real data.

Overall strategy

In this study, we build two tools: (1) SSEgenerator, which generates synthetic GNSS time series incorporating both realistic noise and the deformation signal due to synthetic slow slip events along the Cascadia subduction zone, and (2) SSEdetector, which detects slow slip events in 135 GNSS time series by means of deep-learning techniques. The overall strategy is to train the SSEdetector to reveal the presence of a slow slip event in a fixed-size temporal window, here 60 days, and to apply the detection procedure on real GNSS data with a window sliding over time. We first create 60,000 (60-day) samples of synthetic GNSS position time series using an SSEgenerator, which are then fed to the SSEdetector for the training process. Half of the training samples contain only noise, with the remaining half containing an SSE of various sizes and depths along the subduction interface. Thanks to this procedure, the SSEdetector can evaluate the probability of whether or not an SSE is occurring in the window, yet it does not allow it to determine its position on the fault. It is important to note that the detection is also linked to the GNSS network in Cascadia. On real data, the detection is applied to each time step and provides the probability of occurrence of a slow slip event over time. We first apply this strategy to synthetic data to evaluate the detection power of the method. Then, we apply SSEdetector on real GNSS time series and compare our results with the SSE catalog from Michel et al.[12], who used a different (independent-component-analysis-based) signal processing technique, and with the tremor rate.

Results

SSEgenerator: construction of the synthetic data set. We choose the Cascadia subduction zone as the target region because (1) a link between slow deformation and tremor activity has been assessed[16] and a high-quality tremor catalog is available[26]; (2) a preliminary catalog of SSEs has recently been proposed during the period 2007–2017 by Michel et al.[12] based on time-series decomposition via independent-component analysis. This proposed catalog will be used for comparison and baseline for our results, which are expected to provide a more comprehensive catalog that will better show the link between slow deformation and tremors.

To overcome the scarcity of cataloged SSEs, we train SSEdetector on synthetic data, consisting of simulated sets of GNSS time series for the full station network. Each group of signals (60 days and 135 stations) is considered a single input unit. In order to be able to detect SSEs in real raw time series, several characteristics need to be present in these synthetics. First, they must contain a wide range of realistic background signals at the level of the GNSS network, i.e., spatially and temporally correlated realistic noise time series. On the other hand, while a subset of the samples (*negative* samples) will only consist of background noise, another subset must also include an SSE signal. Here, we use a training database containing half positive and half negative samples. For this, we model SSE signals that are realistic enough compared to real transients of aseismic deformation. The modeled signals do not propagate over space. Finally, the synthetics should also carry realistic missing data recordings, as many GNSS stations have data gaps in practice.

First, we thus generate realistic synthetic time series, that reproduce the spatial and temporal correlated noise of the data acquired by the GNSS network, based on the method developed by Costantino et al.[25]. This database of 60,000 synthetic time series is derived from real GNSS time series (details in section SSEgenerator: Generation of realistic noise time series). We select data in the periods 2007–2014 and 2018–2022 as sources for the noise generation, while we keep data in the period 2014–2017 as an independent test data set (details in section SSEgenerator: data selection).

In order to create the *positive* samples (time series containing an SSE), we model 30,000 dislocations (approximated as a point source) distributed along the Cascadia subduction interface (see Fig. 1b) following the slab2 geometry[27] (detailed procedure in section SSEgenerator: Modeling of synthetic slow slip events). The focal mechanism of the synthetic ruptures approximates a thrust, with rake angle following a uniform distribution (from 75 to 100°) and strike and dip defined by the geometry of the slab. The magnitude of the synthetic SSEs is drawn from a uniform probability distribution (from M_w 6 to 7). Their depths follow the slab geometry and are taken from 0 to 60 km, with further variability of ± 10 km. Such a variability in the depth allows us to better generalize over different slab models and, most importantly, on their fitting to real GNSS data. We further assign each event a stress drop modeled from published scaling laws[28]. We use the Okada dislocation model[29] to compute static displacements at each real GNSS station. We model synthetic SSE signals as sigmoidal-shaped transients, with duration following a uniform distribution (from 10 to 30 days). Eventually, we compute a database of 30,000 synthetic SSE transients, where the SSE signal is added to the positive samples (placed in the middle of the 60-day window).

The synthetic data set is thus made of 60,000 samples and labels, equally split into pure noise (labeled as 0) and signal (labeled as 1) with different nuances of signal-to-noise ratio, resulting both from different station noise levels and differences in magnitude and location so that the deep-learning method effectively learns to detect a variety of slow deformation transients from the background noise. The data set is further split into three independent training (60%), validation (20%), and test (20%) sets, with the latter being used after the training phase only.

SSEdetector: high-level architecture. SSEdetector is a deep neural network made of a CNN[30] and a Transformer network[31] that are sequentially connected (detailed structure in section SSEdetector: Detailed architecture). We constructed the CNN to be a deep spatial-temporal encoder, that behaves as a feature extractor. The structure of the encoder is a deep cascade of 1-dimensional temporal convolutional block sequences and

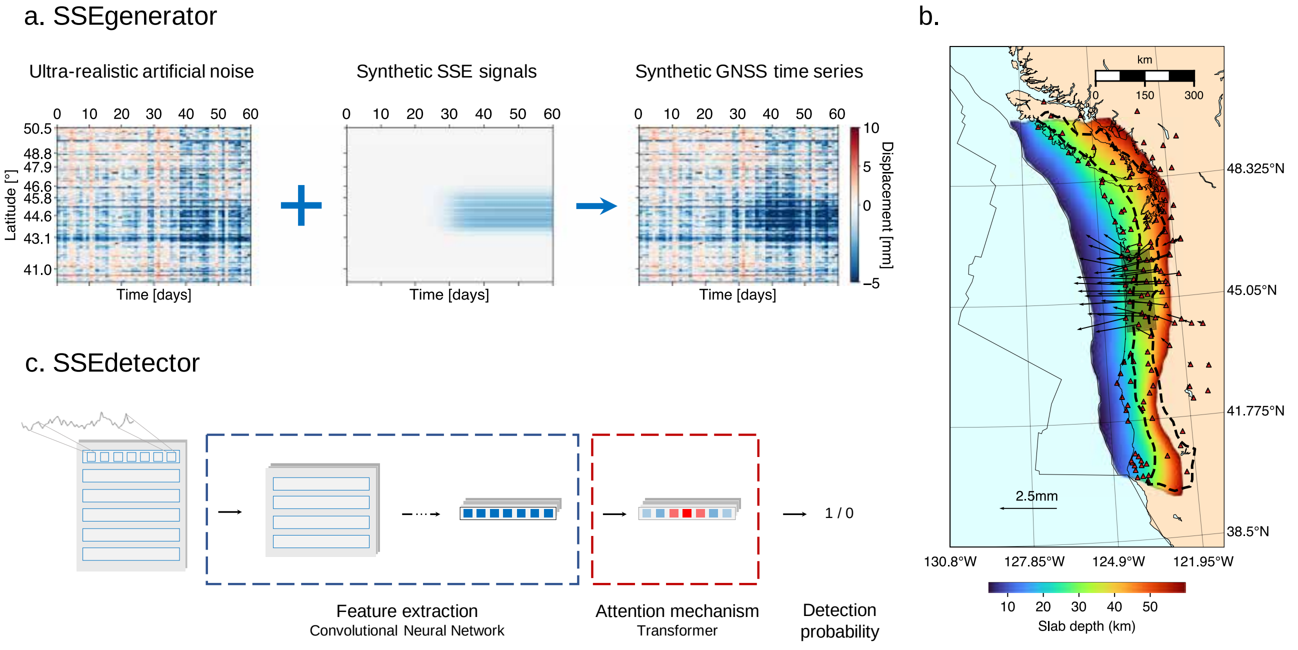

Fig. 1 Schematic architecture of SSEgenerator and SSEdetector. a Overview of the synthetic data generation (SSEgenerator). In the matrix, each row represents the detrended GNSS position time series for a given station, color-coded by the value of the position. The 135 GNSS stations considered in this study are here shown sorted by latitude. The synthetic static displacement model (cf. **b** panel), due to a M_w6.5 event, at each station is convolved to a sigmoid to model the SSE transient, and is added to the ultra-realistic artificial noise to build synthetic GNSS time series. **b** Location of the GNSS stations of MAGNET network used in this study (red triangles). An example of synthetic dislocation is represented by the black rectangle, with arrows showing the modeled static displacement field. The heatmap indicates the locations of the synthetic ruptures considered in this study, color-coded by the slab depth. The dashed black contour represents the tremor locations from the PNSN catalog. **c** High-level representation of the architecture of SSEdetector. The input GNSS time series are first convolved in the time domain. Then, the Transformer computes similarities between samples of each station, learning *self-attention* weights to discriminate between the relevant parts of the signals (here, slow slip transients) and the rest (e.g., background noise), and a probability value is provided depending on whether slow deformation has been found in the data.

spatial pooling layers. The depth of the feature extractor guarantees: (1) a high expressive power, i.e., detailed low-level spatiotemporal features, (2) robustness to data gaps, since their propagation is kept limited to the first layers thanks to a cascade of pooling operators, and (3) limited overfit of the model on the station patterns, thanks to the spatial pooling operation. The decisive component of our architecture is the Transformer network, placed right after the deep CNN encoder. The role of the Transformer is to apply a temporal self-attention mechanism to the features computed by the CNN. As humans, we instinctively focus just on particular fragments of data when looking for any specific patterns. We wanted to replicate such a behavior in our methodology, leading to a network able to enhance crucial portions of the data and neglect the irrelevant ones. This is done by assigning a weight to the data, with those weights being learned by finding significant connections between data samples and by relying on a priori knowledge of the labels, i.e., whether there is an SSE or not. As a result, our Transformer has learned (1) to precisely identify the timing of the aseismic deformation transients in the GNSS time series and (2) to focus on it by assigning a weight close to zero to the rest of the time window. We further guide the process of finding slow deformation transients through a specific supervised learning classification process. First, the disclosed outputs of the Transformer are averaged and passed through a sigmoid activation function. The output values are a detection probability lying in the (0,1) range and can be further interpreted as a confidence measure of the method. Second, we train the SSEdetector by minimizing the binary cross-entropy loss between the target and the predicted labels (details in section Training details). The combination of the two strategies allows SSEdetector to be successfully applied in a real context because (1) we can run our detector on 1-day-shift windows of real data and collect an output value for each day used to build a temporal probability curve, (2) thanks to the Transformer neural network, such a curve will be smooth and the value of probability will gradually increase in time as SSEdetector identifies slow deformation in the GNSS time series.

Application to the synthetic test set: detection threshold. We test the SSEdetector against unseen synthetic samples and we analyze the results quantitatively. We build a synthetic test data set from GNSS data in the period 2018–2022 to limit the influence of data gaps (details in the section “SSEgenerator: data selection”). We generate events down to magnitude 5.5 to better constrain the detection limit at low magnitudes. First, the probability p output by SSEdetector for each test sample must be classified (i.e., either “noise” or “SSE”). We compute the ROC (receiving operating characteristic) curve as a function of the magnitude, with magnitudes higher than a given threshold, as shown in Fig. 2a. The curves represent the false positive rate (FPR, the probability of an actual negative being incorrectly classified as positive) as a function of the true positive rate (TPR, the probability that an actual positive will test positive). They are obtained by selecting the (positive) events whose magnitude is higher than a set of “limit” magnitudes with increments of 0.2. The negative sample set is the same for all the curves so that each curve describes the increment of discriminative SSE power with respect to the noise Each point of the curve is computed with a different detection threshold p_θ, such that a sample is classified as SSE when $p > p_\theta$. Three threshold choices (0.4, 0.5, 0.6) are marked in Fig. 2a. As shown in the figure, the higher the magnitude, the higher the number of true positives. When the threshold is high (e.g., 0.6) the model is more conservative: the false positive rate decreases, yet as well as the number of true positives. On the contrary, when the threshold is lower (e.g., 0.4), more events can be correctly detected, at the cost of introducing more false positives. In this study, we choose $p_\theta = 0.5$ to have a trade-off between the ability to reveal potentially small SSEs (less false negatives, thus more true positives) and the introduction of false positives. This threshold will be used both for further analyses of synthetic data and for the detection of slow slip events in real data.

We obtain a further measure of the sensitivity of our model by computing the true positive rate as a function of the magnitude, as shown in Fig. 2b. On a global scale, the sensitivity increases with the signal-to-noise ratio (SNR), which also shows that there exists an SNR threshold limit for any SSE detection. This threshold is mainly linked to the magnitude of the event, rather than the moment rate. Thus, the ability of SSE detection is mostly influenced by the signal-to-noise ratio rather than the event duration (cf. Supplementary Fig. 1). We compute the sensitivity as a function of the spatial coordinates of the SSE, by deriving a synthetic proxy as the magnitude threshold under which the TPR is smaller than 0.7 on a spatial neighborhood of approximately 50 km. We can see from Fig. 2c that the detection power is related to the density of stations in the GNSS network, as well as to the distance between the rupture and the nearest station, and the rupture depth. When the density of GNSS stations is not high enough, our resolution power decreases as well as the reliability of the prediction. In those cases, we can only detect high-magnitude SSEs. This is also the case on the eastern side of the targeted region where the SSE sources are deeper because of the slab geometry (Fig. 1b), even in locations where the density of stations is higher. In this case, the magnitude threshold increases because these events are more difficult to detect.

Continuous SSE detection in Cascadia from raw GNSS time series during 2007–2022

Overall characteristics of the detected events. In order to evaluate how SSEdetector performs on real continuous data, we applied it to non-post-processed daily GNSS position time series in Cascadia for the period 2007–2022. SSEdetector scans the data with a 60-day sliding window (1-day stride), providing a probability of detection for the central day in each window. Figure 3a shows the probability of slow slip event detection (in blue) together with the tremor activity over the period 2007–2022 (in gray). We consider having a reliable detection when the probability value exceeds 0.5. We find 78 slow slip events over the period 2007–2022, with durations ranging from 2 to 79 days. We find 55 slow slip events in the period 2007–2017, to be compared with the 40 detections of the catalog of Michel et al.[12] (Table 1). We detected 35 of the 40 (87.5%) cataloged SSEs. Three of the missed SSEs have a magnitude smaller than 5.5, and one of them has a magnitude of 5.86. The remaining one has a magnitude of 6.03. We show their location in Supplementary Fig. 2, superimposed on the magnitude threshold map derived for SSEdetector (see Fig. 2b). Given their location, the five missed events have magnitudes that are below the magnitude resolution limit (from 6 to 6.7, see Supplementary Fig. 2). The remaining 20 events may be associated with new undetected SSEs. We also found 23 new events in the period 2017–2022, which were not covered by Michel et al.[12]. We fixed the detection threshold to its default value of 0.5, i.e., the model detects an event with a 50% confidence. Yet, this threshold can be modified to meet specific needs: if high-confidence detections are required, the threshold can be raised; conversely, it can be lowered to capture more events with lower confidence. Interestingly, the two SSEs from Michel et al.[12] with magnitude 5.86 and 6.03, which were missed with a 0.5 probability threshold, can be retrieved when lowering it to 0.4, as shown in Supplementary Fig. 18.

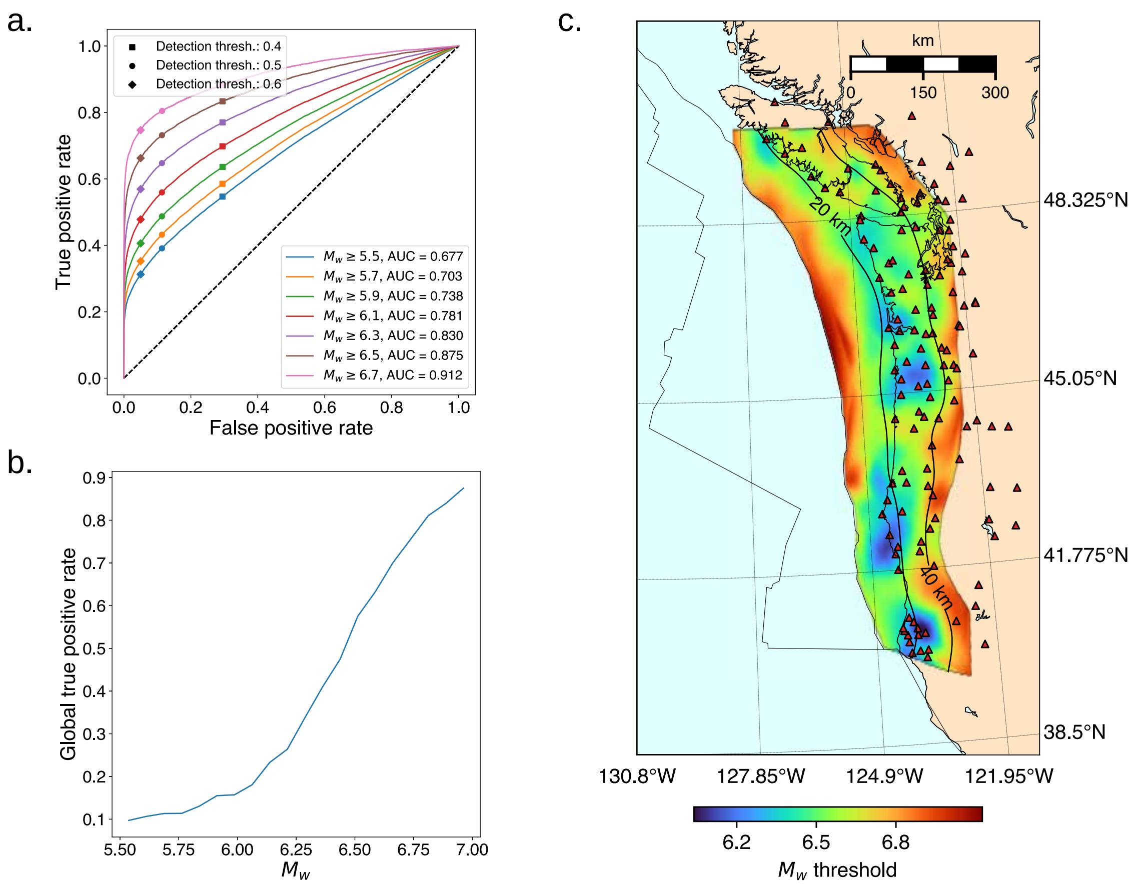

Fig. 2 Performance of SSEdetector on synthetic data. a ROC (receiver operating characteristic) curve as a function of the magnitude greater than an incremental magnitude threshold (see lower legend), shown as different colors. The AUC (area under the curve) is shown in the legend for each curve (an AUC of 1 is associated with the perfect detector). For each curve, a detection threshold equal to 0.4, 0.5, and 0.6 is marked (see upper legend). **b** The blue curve represents the true positive rate (probability that an actual positive will test positive), computed on synthetic data, as a function of the magnitude. **c** Map showing the spatial distribution of the magnitude threshold for reliable detection, computed, for each spatial bin, as the minimum magnitude corresponding to a true positive rate value of 0.7.

We also analyze the shape of the static displacement field in correspondence with the detected SSEs (cf. Supplementary Fig. 3). We compute the static displacement field by taking the median displacement over three days and subtracting the displacement value at each station corresponding to the dates of the SSE. We find a good accordance with independent studies[12,32,33]. Moreover, many of the events found after 2018, as well as the new events detected in the period analyzed by Michel et al.[12], have a displacement field suggesting that they are correct detections.

Analysis of the SSE durations. The shape of the probability curve gives insights into how the SSEdetector reveals slow slip events from raw GNSS time series (see Fig. 3). The probability curve in correspondence with an event has a bell shape: it grows until a maximum value, then it smoothly decreases when the model does not see any displacement associated with slow deformation in the data anymore. We use this property of the probability curve to extract a proxy on the detected SSE duration, based on the time span associated with the probability curve exceeding 0.5. We present the duration distribution in Fig. 4 as well as a summary in Table 1. We detect most of the SSEs found by Michel et al., but we also find 43 additional events, not only in the 2018–2022 period which was not investigated by Michel et al., but also within the 2007–2017 time window that they analyzed, suggesting that our method is more sensitive. We find potential slow slip events at all scales of durations (from 2 to 79 days). Michel et al. hardly detect SSEs that last less than 15 days, probably due to temporal data smoothing[12], while we retrieve shorter events (less than 10 days) since we use raw time series, meaning that our method has a better temporal resolution. However, these short events are associated with higher uncertainties, corresponding to a lower probability value, and the potential implications of this finding should be interpreted with caution. In Fig. 4b, we show a comparison between the SSE durations of Michel et al.'s[12] cataloged events and ours. This plot is made by considering all the combinations between events in our catalog and in the Michel et al. one. Each horizontal alignment represents an event in our catalog that is split into sub-events in the Michel et al. catalog, while vertical alignments show events in the Michel et al. catalog corresponding to sub-events in our catalog. We find that the durations are in good accordance for a large number of events, for which the overlap is often higher than 70%, both for small- and large-magnitude ones. We can also identify, from Fig. 4b, that some events are separated in one method while identified as one single SSE in the other: this is the case for the 55-day-long event from Michel et al.[12], which was paired with 3 SSEdetector sub-events (see Fig. 3d and the rectangle in Fig. 4b). The majority of the points located off the identity line (the diagonal) are thus sub-events for which the grouping differs in the two catalogs. As more points are below the diagonal than above, we can see that the SSEdetector tends to separate the detections more. We interpret this as a possible increase in the detection precision, yet a

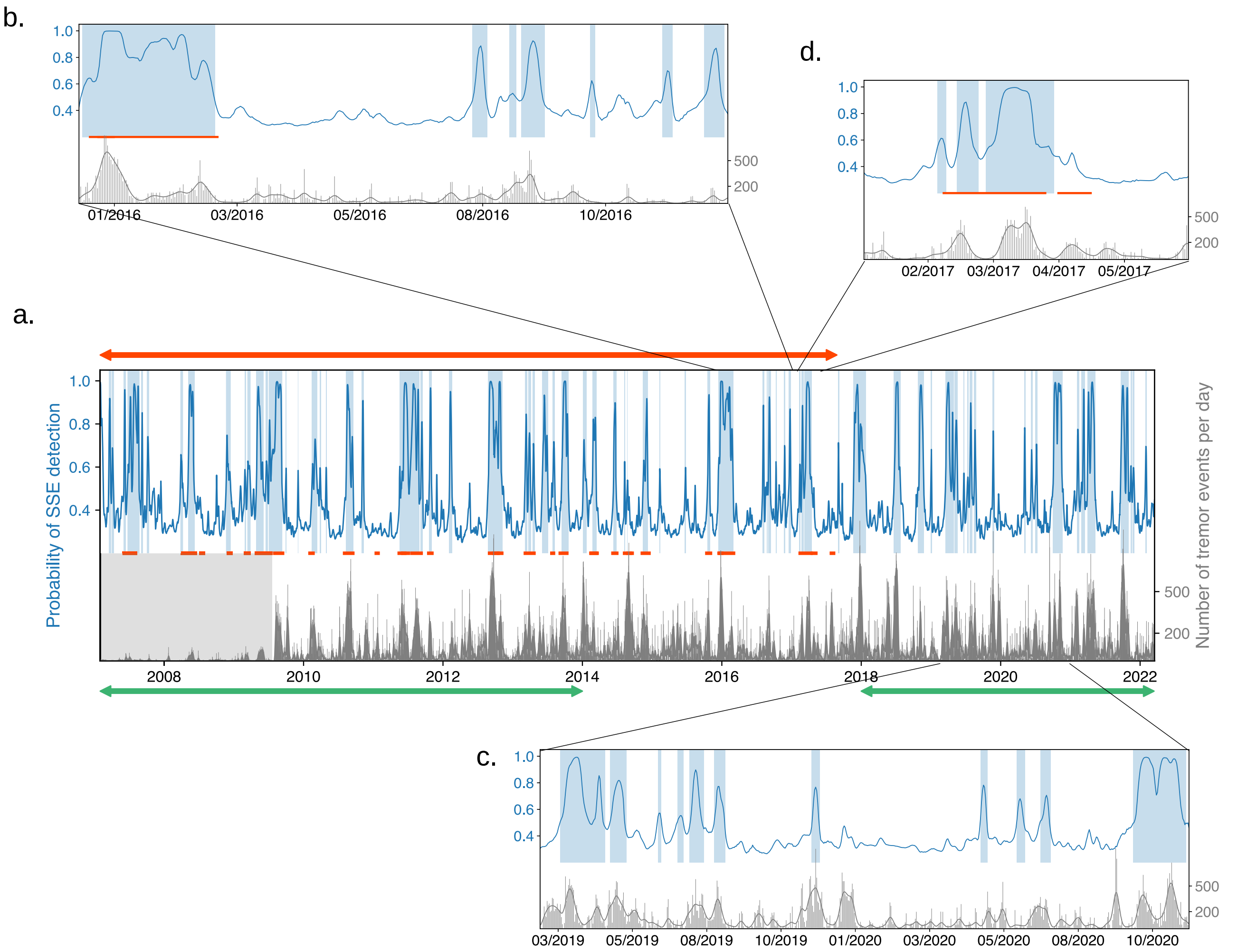

Fig. 3 Overview of the performance of SSEdetector on real raw GNSS time series. The blue curves show the probability of detecting a slow slip event (output by SSEdetector) in 60-day sliding windows centered on a given date. Gray bars represent the number of tremors per day, smoothed (gaussian smoothing, $\sigma = 2$ days) in the gray curve. Red horizontal segments represent the known events cataloged by ref. [12]. The **a** panel shows the global performance of the SSEdetector over 2007–2022. The red arrow indicates the time window analyzed by Michel et al., while the green arrows describe the two periods from which the synthetic training samples have been derived. The gray rectangle indicates the period which was not covered by the PNSN catalog. In this period, data from Ide, 2012[34] has been used. The **b**, **c**, and **d** panels show zooms in 2016–2017, 2019–2021, and 2017 (January to July), respectively.

Table 1 Comparison of the number of detections from SSEdetector with respect to the catalog from Michel et al.[12].

Period		Method	
		Michel et al.	SSEdetector
2007–2017	Common with Michel et al.	40	35
	Not detected by Michel et al.	0	20
2017–2022		0	23

We distinguish detections in 2007–2017 (the same period analyzed by Michel et al.), and in 2017–2022. We further discriminate, in 2007–2017, between events in common with Michel et al., and new events in the same period.

validation with an independent acquisition data set is needed since the separation into sub-events strongly depends on the threshold applied to the detection probability to define a slow slip event (0.5 in this study). Also, a few points above the diagonal represent events that were split into sub-events by Michel et al., which our model tends to detect as a single event. It should be noted that the event duration is related to the choice of the threshold and thus influenced by a priori choices, making it hard to compare different catalogs. Also, the comparison of events in the same catalog might be biased because the threshold may affect the events differently depending on differences in the signal-to-noise ratio.

Validation against tremors. In order to have an independent validation, we compare our results with tremor activity from the Pacific Northwest Seismic Network (PNSN) catalog[26] between 2009-2022 and Ide's catalog[34] catalog between 2006-2009, shown in gray in Fig. 3. We show the location of the tremors in our catalogs with the dashed black contour in Fig. 1b. From a qualitative point of view, we can see that the detection probability curve seems to align well with the number of tremors per day, throughout the whole period. This is also true for the 20 possible newly detected events that were not present in previous catalogs, for example during the period after 2017 (see Fig. 3c), but also in 2016–2017, where we detected 7 possible events that were not previously cataloged (see Fig. 3b). The excellent similarity

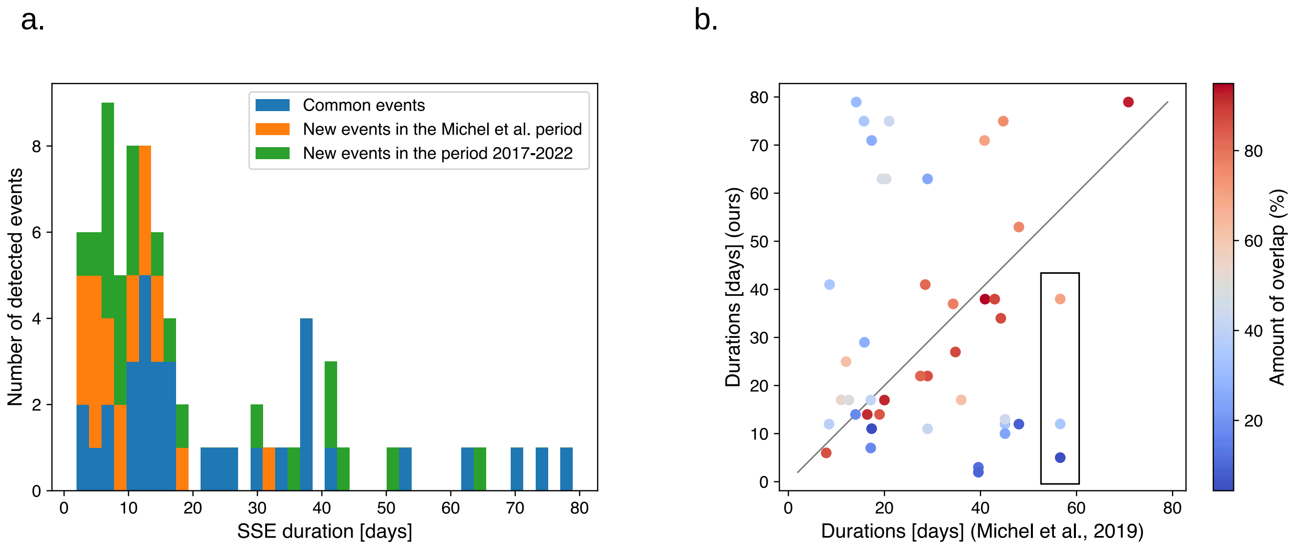

Fig. 4 Distribution of the detected SSEs and comparison with the independent catalog from Michel et al.[12]. a Cumulative histogram of SSEdetector inferred durations. Blue bars represent the 35 cataloged events by Michel et al.[12] that have been successfully retrieved. Orange bars show the 20 additional events that have been discovered within the time window analyzed by Michel et al., while green ones represent the 23 events found in the time period 2017–2022, not covered by the catalog of Michel et al. **b** Event durations from Michel et al.'s catalog with respect to the durations obtained by SSEdetector. Events are color-coded by the overlap percentage (details in section Overlap percentage calculation). The black rectangle represents an example of a large event found by Michel et al.[12], that is split into three sub-events by SSEdetector.

between tremors and our detections is quantitatively assessed by computing the cross-correlation between the probability curve and the number of tremors per day, the latter smoothed with a Gaussian filter ($\sigma = 1.5$ days), as a function of the time shift between the two curves (Fig. 5a). The interval 2007–2010 has been excluded from Fig. 5a in order to consider the period covered by the PNSN catalog only. The maximum correlation value is around 0.58 and is obtained for a time shift between 1 and 2 days. This shows that, at a global scale, the probability peaks are coeval with the peaks of tremor activity.

We also make a further comparison at the local scale for each individual detected SSE. In Fig. 5b we observe that most of the individual detected SSEs show a correlation larger than 0.4 with the coeval peak of tremor. SSE and tremor signals are offset by about 2 days on average (see Fig. 5c). This result, obtained on windows of month-long scale, seems consistent with the decade-long correlation shown in Fig. 5a, suggesting that the found large-scale trend is also true at a smaller scale. This may suggest that the slow deformation, for which the detection probability is a proxy, precedes the tremor chatter by a few days, with potential implications on the nucleation of the slow rupture.

We compare the tremor peak duration (see details in section "Calculation of tremor durations") to the SSE duration in Fig. 5d for all the events that have been also considered in Fig. 5b. The figure shows a correspondence between slow slip duration and coeval tremor activity duration: most of the events are associated with a peak of tremor activity of close duration. This is true also for large events, up to 80 days. This finding gives an insight that our deep-learning-based method, blindly applied to raw GNSS time series, achieves reliable results. Yet, this result should be taken with caution, since it is strongly dependent on the choice of the window of observation (see sections "Calculation of tremor durations" and "Computation of local- and global-scale correlations" for further details).

Validity of the duration proxy against temporal smoothing: the March 2017 slow slip event. We test the first-order effectiveness of the duration proxy computation to assess any potential temporal smoothing effects. We focus on the M_w 6.7 slow slip event that occurred in March 2017. We rely both on the measured time series and on the kinematic model of the slip evolution by Itoh et al.[32]. Figure 6a shows the displacement field output by Itoh et al.'s model and its temporal evolution is shown in Fig. 6b. We use SSEgenerator to build a random noise time series, to which we add the modeled displacement time series to build a synthetic time series reproducing the March 2017 SSE, as shown in Fig. 6c. Figure 6e shows the probability curve output by SSEdetector on the synthetic time series (in blue), to be also compared to the prediction associated with the same event in real GNSS data (in red, see also Fig. 3d). By the analysis of the probability curve, we estimate a duration of 28 days, comparable with the duration inferred by Itoh et al.[32] (30 days). A 3-day time shift is found between the probability curve and the slip evolution, represented by the moment rate function in Fig. 6d, which can be imputed to the specific realization of synthetic noise. When comparing the probability curve associated with the same event as computed from real data (red curve in 6e), we find a better resemblance, consistent also with the catalog from Michel et al. (see Fig. 3d), which demonstrates that SSEdetector does not suffer from any first-order temporal smoothing issues.

Robustness of SSEdetector to variations in the source characteristics. We perform an extensive study to test how the SSEdetector performs on SSEs that exhibit characteristics in the source parameters that differ from the synthetic data set used for the training. We train the SSEdetector on synthetic dislocations with fixed aspect ratios and shapes. However, SSEs in Cascadia generally have aspect ratios between 3 and 13 and exhibit heterogeneous slip along strike with pulse-like propagation[12,35]. We create a synthetic test sample containing three slow slip events that propagate in space and time, to simulate a real large SSE that lasts 60 days and propagates southwards, as shown in Supplementary Fig. 19. The detection probability and the associated duration proxy suggest that the probability curve can still be used to retrieve SSEs that last longer than 30 days. Thanks to its multi-station approach, our method detects slices of slow slip events

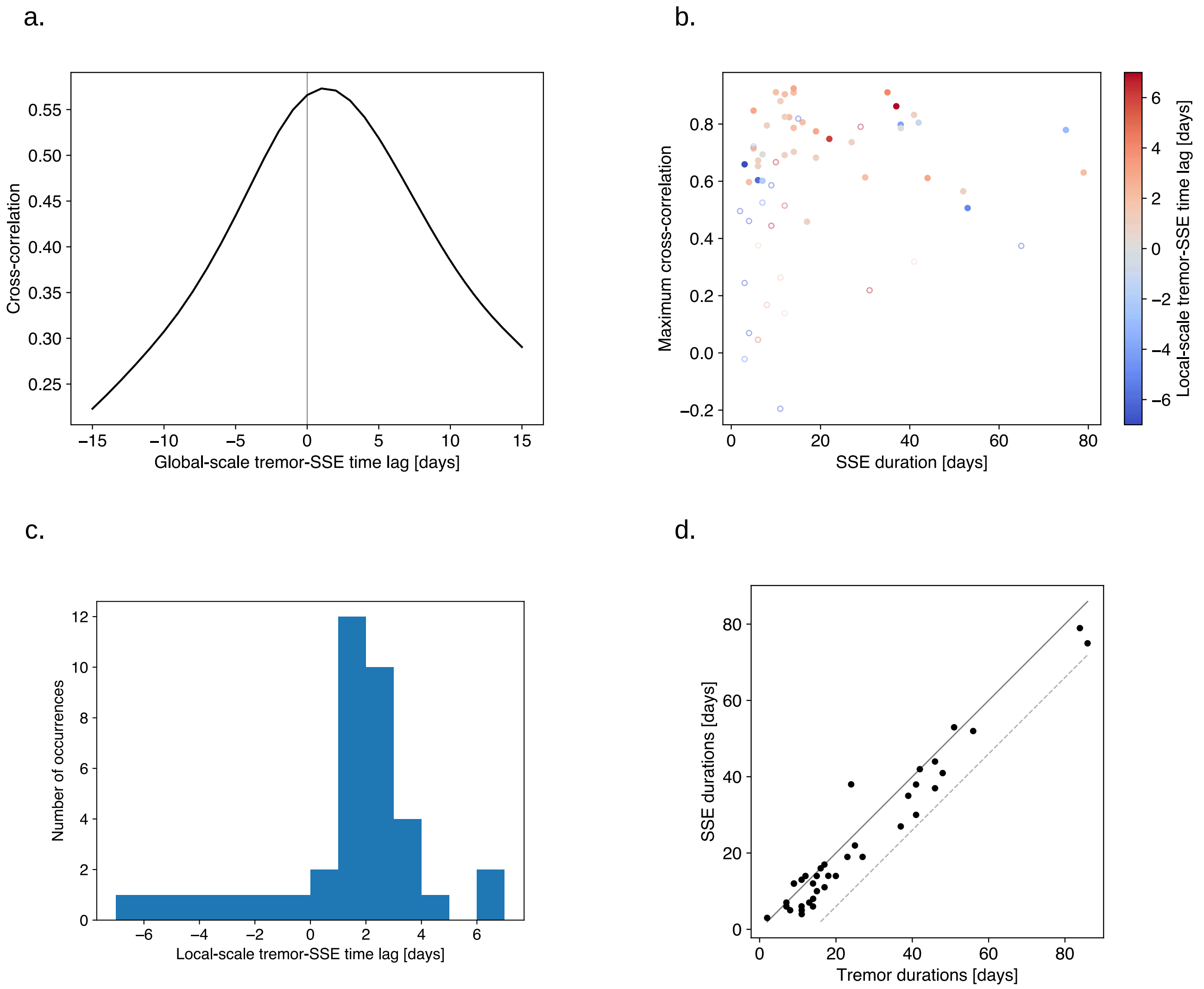

Fig. 5 Validation of SSEdetector performance against tremor activity in 2010–2022. a Global-scale cross-correlation between the full-length SSEdetector output probability and the number of tremors per day, as a function of the time shift between the two curves. **b** Local-scale maximum value of cross-correlation for each SSE and tremor window, centered on the SSE duration, as a function of the SSE duration, color-coded by the associated time lag between tremor and SSE (positive lag means deformation precedes tremor). Events having a zero-lag cross-correlation (correlation coefficient) lower than 0.4 are marked with an empty point. **c** Histogram of the time lags computed in the **b** panel. **d** SSE durations as a function of the tremor durations for the events in the **b** panel which have a correlation greater than 0.4. The solid black line represents the identity line, while the dashed gray line is the maximum tremor duration that can be attained for a given SSE duration, that is SSE duration + 14 days (see section "Computation of local- and global-scale correlations").

through time and space (station dimension), regardless of the training configuration. This can be seen as a stage in which some building blocks are provided to the model, that combines them during the testing phase to detect events having different shapes and sizes. For this reason, SSEdetector can be used to generalize, at first order, over more complex events in real data without the need to build a complex train data set, which should also account for realistic event propagation mechanisms, yet difficult to model.

We also test the generalization ability of SSEdetector over shorter (less than 10 days) events. We build a synthetic sample of a 3-day slow slip event. As shown in Supplementary Fig. 20, the SSEdetector is able to correctly detect it, with an inferred duration of 4 days, suggesting that it has learned what the first-order temporal signature of the signals of interest, being able to detect them also at shorter scales.

SSEdetector is applied to real data by following a sliding-window strategy. In real data, two events could be found in the same window, most likely one in the northern and the other in the southern part. For this, we build two synthetic tests, the first test with two 20-day-long events that do not occur at the same time, and the second one with two 20-day-long synchronous events. Supplementary Figs. 20 and 21 show the results for both cases, which suggest that SSEdetector can generalize well when there are two events in the same window, also corroborating the results found on real data.

Sensitivity study. We analyze the sensitivity of SSEdetector with respect to the number of stations. We construct an alternative test selecting 352 GNSS stations (see Supplementary Fig. 5), which is the number of stations used by Michel et al.[12]. The 217 extra stations have larger percentages of missing data compared to the initial 135 stations (cf. Supplementary Fig. 4). We train and test SSEdetector with 352 time series and we report the results in Supplementary Figs. 6–7. We observe that the results are similar, with an excellent alignment with tremors and similar correlation and lag values, although with this setting the detection power

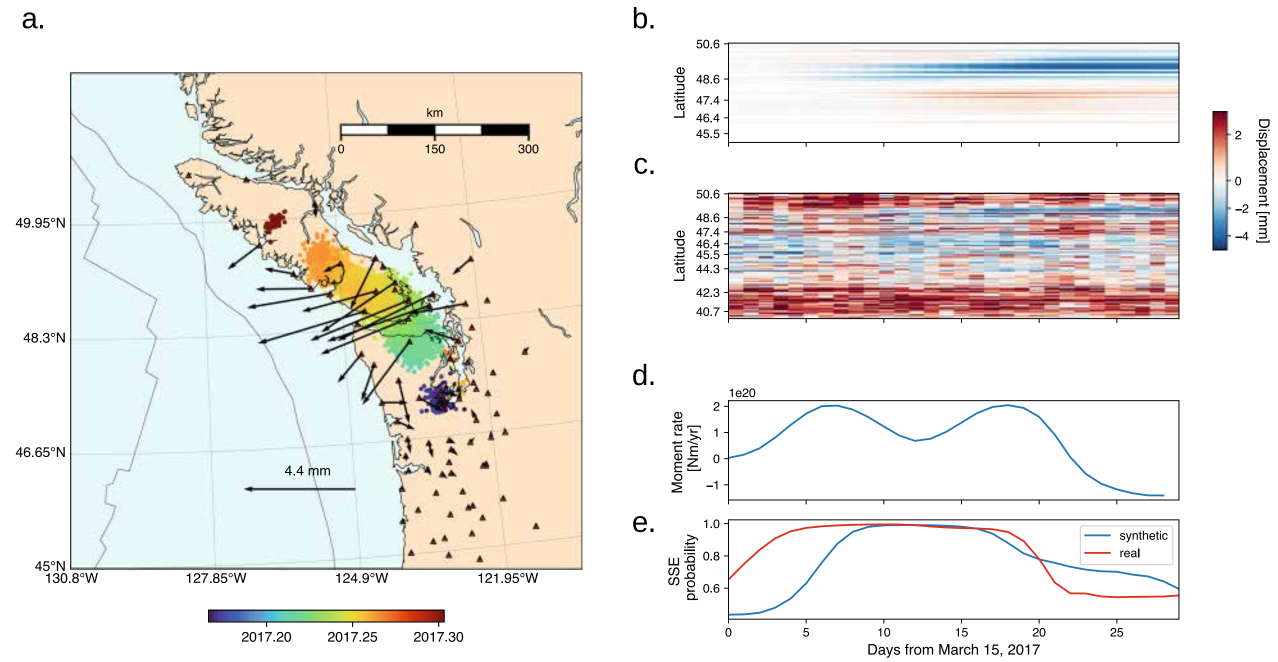

Fig. 6 Temporal smoothing analysis on a synthetic model of the March 2017 slow slip event. **a** Model of the static displacement associated with the March 2017 slow slip event by Itoh et al.[32] (black arrows). The red triangles represent the 135 stations used in this study and the points show the tremor location from March 1 to April 30, 2017, color-coded by date of occurrence. **b** Matrix showing the temporal evolution of Itoh et al.'s model. Each row represents the detrended E-W GNSS position time series for a given station, color-coded by the position value. **c** Detrended E-W synthetic time series (sum of the Itoh et al.'s model and a realization of artificial noise output by SSEgenerator, color-coded by position value. **d** Moment rate function associated with the slip evolution of the model by Itoh et al. **e** The blue curve represents the daily probability output by the SSEdetector from the synthetic time series shown in the **c** panel. The red curve shows the probability curve associated with the prediction of SSEdetector on the March 2017 slow slip event on real GNSS data (see also Fig. 3d).

slightly decreases, probably due to a larger number of missing data, suggesting that there is a trade-off between the number of GNSS stations and the number of missing data.

We also test the ability of SSEdetector to identify SSEs in a sub-region only (even if it is trained with a large-scale network. For that, we test SSEdetector (trained on 135 stations), without re-training, on a subset of the GNSS network, situated in the northern part of Cascadia. To this end, we replace with zeros all the data associated with stations located at latitudes lower than 47 degrees (see Supplementary Fig. 8). Similarly, we find that SSEdetector retrieves all the events that were found by Michel et al.[12] and the correlation with tremors that occur in this sub-region is still high, with a global-scale cross-correlation of 0.5 (cf. Supplementary Figs. 9–10). This means that the model is robust against long periods of missing data and, thanks to the spatial pooling strategy, can generalize over different settings of stations and obtain some information on the localization. We found similar results for SSEs in the southern part, with a lower correlation with tremors, probably because of the higher noise level in south Cascadia, as shown in Supplementary Figs. 16–17.

Finally, we test the SSEdetector against other possible deep-learning models that could be used for detection. We report in Supplementary Figs. 11–12, the results were obtained by replacing the one-dimensional convolutional layers with two-dimensional convolutions on time series sorted by latitude (as shown in Fig. 1a). This type of architecture was used in studies having similar multi-station time-series data[36]. We observe that the results on real data are not satisfactory because of too high a rate of false detections and a lower temporal resolution than SSEdetector (in other words, short SSEs are not retrieved). This suggests that our specific model architecture, handling in different ways the time dimension and the station dimension, might be more suited to multi-station time-series data sets.

Discussion

In this study, we use a multi-station approach that proves efficient in detecting slow slip events in raw GNSS time series even in the presence of SSE migrations[12,32,33]. Thanks to SSEdetector, we are able to detect 87 slow slip events with durations from 2 to 79 days, with an average limit magnitude of about 6.4 in north Cascadia and 6.2 in south Cascadia computed on the synthetic test set (see Fig. 2b). The magnitude of the smallest detected SSE in common with Michel et al. is 5.42, with a corresponding duration of 8.5 days. One current limitation of this approach is that the location information is not directly inferred. In this direction, some efforts should be made to develop a method for characterizing slow slip events after the detection in order to have information on the location, but also on the magnitude, of the slow rupture.

We apply our methodology to the Cascadia subduction zone because it is the area where independent benchmarks exist and it is thus possible to validate a new method. However, the applicability of SSEgenerator and SSEdetector to other subduction zones is possible. The current approach is, however, region-specific. In fact, the characteristics of the targeted zone affect the structure of the synthetic data, thus a method trained on a specific region could have poor performance if tested on another one without re-training. This problem can be addressed by generating multiple data sets associated with different regions and combining them for the training. Also, we focus on the Cascadia subduction zone, where not much regular seismicity occurs, making it a prototypical test zone when looking for slow earthquakes.

When addressing other regions, such as Japan, for example, the influence of earthquakes or post-seismic relaxation signals could make the problem more complex. This extension goes beyond the scope of this study, yet we think that it will be essential to tackle this issue in order to use deep-learning approaches for the detection of SSEs in any region.

Conclusions

We developed a powerful pipeline, consisting of a realistic synthetic GNSS time-series generation, SSEgenerator, and a deep-learning classification model, SSEdetector, aimed to detect slow slip events from a series of raw GNSS time series measured by a station network. We built a new catalog of slow slip events in the Cascadia subduction zone by means of SSEdetector. We found 78 slow slip events from 2007 to 2022, 35 of which are in good accordance with the existing catalog[12]. The detected SSEs have durations that range between a few days to a few months. The detection probability curve correlates well with the occurrence of tremor episodes, even in time periods where we found new events. The duration of our SSEs, for the 35 known events, as well as for the 43 new detections, are found to be similar to the coeval tremor duration. The comparison between tremors and SSEs also shows that, both at a local and a global temporal scale, the slow deformation may precede the tremor chatter by a few days, with potential implications on the link between a slow slip that could drive the rupture of nearby small seismic asperities. This is the first successful attempt to detect SSEs from raw GNSS time series, and we hope that this preliminary study will lead to the detection of SSEs in other active regions of the world.

Methods

SSEgenerator: data selection. We consider the 550 stations in the Cascadia subduction zone, belonging to the MAGNET GNSS network, and we select data from 2007 to 2022. We train the SSEdetector with synthetic data whose source was affected by different noise and data-gap patterns. We divide the data into three periods: 2007–2014, 2014–2018, and 2018–2022. In order to create a more diverse training set, data in the period 2007–2014 and 2018–2022 has been chosen as a source for synthetic data generation. The period 2014–2018 was left aside and used as an independent validation set for performance assessment on real data. Nonetheless, since synthetic data is performed by applying a methodology based on randomization (SSEgenerator), a test on the whole sequence 2007–2022 is possible without overfitting.

For the two periods 2007–2014 and 2018–2022, we sorted the GNSS stations by the total number of missing data points and we chose the 135 stations affected by fewer data gaps as the final subset for our study. We make sure that stations having too high a noise do not appear in this subset. We selected 135 stations since it represents a good compromise between the presence of data and the longest data-gap sequence in a 60-day window. However, we also train and test SSEdetector on 352 stations (the same number used in the study by Michel et al.[12]) to compare with a setting equivalent to the one from Michel et al.[12]. We briefly discuss the results in the section “Sensitivity study”.

SSEgenerator: generation of realistic noise time series. Raw GNSS data is first detrended at each of the 135 stations, i.e., the linear trend is removed, where the slope and the intercept are computed, for each station, without taking into account the data gaps, i.e., for each station the mean over time is calculated without considering the missing data points, and is removed from the series. Only the linear trend is removed from the data, without accounting for seasonal variations, common modes, or co- and post-seismic relaxation signals, which we want to keep to have a richer noise representation. We remove the linear trend for each station to avoid biases in our noise generation strategy since the variability due to the linear trend would be captured as a principal component thus introducing spurious piece-wise linear signals in the realistic noise. We build a matrix containing all station time series $\mathbf{X} \in \mathbb{R}^{N_t \times N_s}$, where N_t is the temporal length of the input time series and N_s is the number of stations. In this study, we use 2 components (N-S and E-W) and we apply the following procedure for each component independently. Each column of $\mathbf{X}$ contains a detrended time series. We proceed as follows. The $\mathbf{X}$ matrix is then re-projected in another vector space through a Principal Component Analysis (PCA), as follows. First, the data is centered. The mean vector is computed $\boldsymbol{\mu} \in \mathbb{R}^{N_s}$, such that μ_i is the mean of the i-th time series. The centered matrix is considered $\tilde{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}$, and is decomposed through Singular Value Decomposition (SVD) to obtain the matrix of right singular vectors $\mathbf{V}$, which is the rotation matrix containing the spatial variability of the original vector space. We further rotate the data by means of this spatial matrix to obtain spatially uncorrelated time series $\hat{\mathbf{X}} = \tilde{\mathbf{X}}\mathbf{V}$. Then, we produce $\hat{\mathbf{X}}_R$, a randomized version of $\hat{\mathbf{X}}$, by applying the iteratively refined amplitude-adjusted Fourier transform (AAFT) method[37], having globally the same power spectrum and amplitude distribution of the input data. The number of AAFT iterations has been set to 5 based on preliminary tests. The surrogate time series are then back-projected in the original vector space to obtain $\mathbf{X}_R = \hat{\mathbf{X}}_R\mathbf{V}^T + \boldsymbol{\mu}$. We further enrich the randomized time series $\mathbf{X}_R$ by imprinting the real pattern of missing data for 70% of the synthetic data. We shuffle the data gaps before imprinting them to the data, assigning a station a data-gap sequence belonging to another station, such that SSEdetector can better generalize over unseen test data for the same station, which necessarily would have a different pattern of data gaps. We leave the remaining 30% of the data as it is. We prefer not to use any interpolation method to avoid introducing new values in the data. Thus, we set all the missing data points to zero, which is a neutral value with respect to the trend of the data and the convolution operations performed by SSEdetector.

After this process, we generate sub-windows of noise time series as follows. Given the window length W_L, a uniformly distributed random variable is generated $s \sim \mathcal{U}(-\frac{W_L}{2}, \frac{W_L}{2})$ and the data is circularly shifted, in time, by the amount s. Then, $\lfloor \frac{N_t}{W_L} \rfloor$ contiguous (non-overlapping) windows are obtained. The circular shift is needed in order to prevent SSEdetector from learning a fixed temporal pattern of data gaps. Finally, by knowing the desired number N of noise windows to compute, the surrogate generation $\left\{\mathbf{X}_R\right\}_i$ can be repeated $\lceil \frac{N}{\lfloor N_t/W_L \rfloor} \rceil$ times. In our study, we generate $N = 60,000$ synthetic samples, by calling the surrogate data generation 1,429 times and extracting 42 non-overlapping noise windows from each randomized time series.

SSEgenerator: modeling of synthetic slow slip events. We first generate synthetic displacements at all 135 selected stations using Okada’s dislocation model[29]. We draw random locations (fault centroids), and strike and dip angles using the slab2 model[27] following the subduction geometry within the area of interest (see Fig. 1b). We let the rake angle be a uniform random variable from 75 to 100 degrees, in order to have a variability around 90 degrees (thrust focal mechanism). For each (latitude, longitude) couple, we extract the corresponding depth from the slab and we add further variability, modeled as a uniformly distributed random variable from −10 to 10 km. We allow for this variability if the depth is at least 15 km, to prevent the ruptures from reaching the surface. We associate each rupture with a magnitude M_w, uniformly generated in the range (6, 7), and we compute the equivalent moment as $M_0 = 10^{1.5M_w+9.1}$. As for the SSE geometry, we rely on the circular crack approximation[38] to compute the SSE radius as:

$$R = \left(\frac{7}{16}\frac{M_0}{\Delta\sigma}\right)^{1/3} \tag{1}$$

where $\Delta\sigma$ is the static stress drop. We compute the average slip on the fault as:

$$\bar{u} = \frac{16}{7\pi}\frac{\Delta\sigma}{\mu}R \tag{2}$$

where μ is the shear modulus. We assume $\mu = 30$ GPa. By imposing that the surface of the crack must equal a rectangular dislocation of length L and width W, we obtain $L = \sqrt{2\pi}R$. We assume that $W = L/2$. Finally, we model the stress drop as a lognormally distributed random variable. We assume the average stress drop to be $\overline{\Delta\sigma} = 0.05$ MPa for the Cascadia subduction zone[28]. We also assume that the coefficient of variation c_V, namely the ratio between the standard deviation and the mean, is equal to 10. Hence, we generate the static stress drop as $\Delta\sigma \sim \text{Lognormal}(\mu_N, \sigma_N^2)$, where μ_N and σ_N are the mean and the standard deviation of the underlying normal distribution, respectively, that we derive as:

$$\sigma_N = \sqrt{\ln(c_V^2 + 1)} \tag{3}$$

and

$$\mu_N = \ln(\overline{\Delta\sigma}) - \sigma_N^2/2. \tag{4}$$

We thus obtain the (horizontal) synthetic displacement vector $\mathbf{D}_s = (D_s^{N-S}, D_s^{E-W})$ at each station s. We model the temporal evolution of slow slip events as a logistic function. Let D be the E-W displacement for simplicity. In this case, we model an SSE signal at a station s as:

$$d_s(t) = \frac{D}{1 + e^{-\beta(t-t_0)}} \tag{5}$$

where β is a parameter associated with the growth rate of the curve and t_0 is the time corresponding to the inflection point of the logistic function. We assume $t_0 = 30$ days, so that the signal is centered in the 60-day window. We derive the parameter β as a function of the slow slip event duration T. We can rewrite the duration as $T = t_{\max} - t_{\min}$, where $t_{\max}$ is the time corresponding to the post-SSE value of the signal (i.e., D), while $t_{\min}$ is associated to the pre-SSE displacement (i.e., 0). Since these values are only asymptotically reached, we introduce a threshold γ, such that $t_{\max}$ and $t_{\min}$ are associated with $d_s(D - \gamma D)$ and $d_s(\gamma D)$, respectively. We choose $\gamma = 0.01$. By rewriting the duration as $T = t_{\max} - t_{\min}$ and solving for β, we obtain:

$$\beta = \frac{2}{T}\ln\left(\frac{1}{\gamma} - 1\right). \tag{6}$$

Finally, we generate slow slip events having uniform duration T between 10 and 30 days. We take half of the noise samples (30,000) and we create a positive sample (i.e., time series containing a slow slip event) as $\mathbf{X}_R + \mathbf{d}(t)$, where $\mathbf{d}(t)$ is a matrix containing all the modeled time series $d_s(t)$ for each station. We let $\mathbf{X}_R$ contain missing data. Therefore, we do not add the signal $d_s(t)$ where data should not be present.

SSEdetector: detailed architecture. SSEdetector is a deep neural network obtained by the combination of a convolutional and a Transformer neural network. The full architecture is shown in Supplementary Fig. 13. The model takes input GNSS time series, which can be grouped as a matrix of shape (N_s, N_t, N_c), where N_s, N_t, N_c are the number of stations, window length and number of components, respectively. In this study, $N_s = 135, N_t = 60$ days and $N_c = 2$ (N-S, E-W). The basic unit of this CNN is a Convolutional Block. It is made of a sequence of a one-dimensional convolutional layer in the temporal (N_t) dimension, which computes N_f feature maps by employing a 1×5 kernel, followed by a Batch Normalization[39] and a ReLu activation function[40]. We will refer to this unit as ConvBlock(N_f) for the rest of the paragraph (see Supplementary Fig. 13). We alternate convolutional operations in the temporal dimension with pooling operations in the station dimension (max-pooling with a kernel of 3) and we replicate this structure as long as the spatial (station) dimension is reduced to 1. To this end, we create a sequence of 3 ConvBlock($\cdot$) + max-pooling. As an example, the number of stations after the first pooling layer is reduced from 135 to 45. At each ConvBlock($\cdot$), we multiply by 4 the number of computed feature maps N_f. At the end of the CNN, the computed features have shape (N_t, N_f^{final}), with $N_f^{final} = 256$.

This feature matrix is given as input to a Transformer neural network. We first use a Positional Embedding to encode the temporal sequence. We do not impose any kind of pre-computed embedding, but we use a learnable matrix of shape (N_t, N_f^{final}). The learned embeddings are added to the feature matrix (i.e., the output of the CNN). The embedded inputs are then fed to a Transformer neural network[31], whose architecture is detailed in Supplementary Fig. 14. Here, the global (additive) self-attention of the embedded CNN features is computed as:

$$\eta_{t_1,t_2} = W_a \tanh\left(W_1^T h_{t_1} + W_2^T h_{t_2} + b_h\right) + b_a, \tag{7}$$

$$a_{t_1,t_2} = softmax\left(\eta_{t_1,t_2}\right) = \frac{e^{\eta_{t_1,t_2}}}{\sum_{t_2} e^{\eta_{t_1,t_2}}}, \tag{8}$$

$$c_{t_1} = \sum_{t_2} a_{t_1,t_2} \cdot h_{t_2}, \tag{9}$$

where W represents a learnable weight matrix and b a bias vector. The matrices h_{t_1} and h_{t_2} are the hidden-state representations at time t_1 and t_2, respectively. The matrix a_{t_1,t_2} contains the attention scores for the time steps t_1 and t_2. Here, a context vector is computed as the weighted sum of the hidden-state representations by the attention scores. The context vector contains the importance at a given time step based on all the features in the window. The contextual information is then added to the Transformer inputs. Then, a position-wise Feed-Forward layer (with a dropout rate of 0.1) is employed to add further nonlinearity. After the Transformer network, a Global Average Pooling in the temporal dimension (N_t) is employed to gather the transformed features and to output a vector summarizing the temporal information. A Dropout is then added as a form of regularization to reduce overfitting[41], with dropout rate $\delta = 0.2$. In the end, we use a fully connected layer with one output, with a sigmoid activation function to express the probability of SSE detection.

Training details. We perform a mini-batch training[42] (batch size of 128 samples) by minimizing the binary cross-entropy (BCE) loss between the target labels y and the predictions $\hat{y}$ (a probability estimate):

$$BCE(y, \hat{y}) = -y\ln(\hat{y}) - (1 - y)\ln(1 - \hat{y}). \tag{10}$$

The BCE loss is commonly used for binary classification problems (detection is a binary classification). We use the ADAM method for the optimization[43] with a learning rate $\lambda = 10^{-3}$ which has been experimentally chosen. We schedule the learning rate such that it is reduced during training iterations and we stop the training when the validation loss does not improve for 50 consecutive epochs. We initialized the weights of the SSEdetector with a uniform He initializer[44]. We implemented the code of SSEdetector in Python using the Tensorflow and Keras libraries[45,46]. We run the training on NVIDIA Tesla A100 Graphics Processing Units (GPUs). The training of SSEdetector takes less than 2 h. The inference on the whole 15-year sequence (2007–2022) takes a few seconds.

Calculation of tremor durations. We compute the durations of tremor bursts using the notion of topographic prominence, explained in the following. We rely on the software implementation from the SciPy Python library[47]. Given a peak in the curve, the topographic prominence is informally defined as the minimum elevation that needs to be descended to start reaching a higher peak. The procedure is graphically detailed in Supplementary Fig. 15. We first search for peaks in the number of tremors per day by comparison with neighboring values. In order to avoid too many spurious local maxima, we smooth the number of tremors per day with a Gaussian filter ($\sigma = 1.5$ days). For each detected SSE, we search for peaks of tremors in a window given by the SSE duration ± 3 days. For each peak of tremors that is found, the corresponding width is computed as follows. The topographic prominence is computed by placing a horizontal line at the peak height h (the value of the tremor curve corresponding to the peak). An interval is defined, corresponding to the points where the line crosses either the signal bounds or the signal at the slope towards a higher peak. In this interval, the minimum values of the signal on each side are computed, representing the bases of the peak. The topographic prominence p of the peak is then defined as the height between the peak and its highest base value. Then, the local height of the peak is computed as $h_L = h - \alpha \cdot p$. We set $\alpha = 0.7$ in order to focus on the main tremor pulses, discarding further noise in the curve. From the local height, another horizontal line is considered and the peak width is computed as the intersection point of the line with either a slope, the vertical position of the bases, or the signal bounds, on both sides. Finally, the total width of a tremor pulse in an SSE window is computed by considering the earliest starting point on the left side and the latest ending point on the right side. It must be noticed that the derivation of the tremor duration depends on the window length. In fact, the inferred tremor duration can saturate to a maximum value equal to the length of the window. For this reason, we added in Fig. 5c a dashed line corresponding to the window length (SSE duration + 14 days) (see section "Computation of local- and global-scale correlations").

Computation of local- and global-scale correlations. We compute the time-lagged cross-correlation between the SSE probability and the number of tremors per day (Fig. 5a, b). We smooth the number of tremors per day with a Gaussian filter ($\sigma = 1.5$ days). We consider a lag between –7 and 7 days, with a 1-day stride.

In the case of Fig. 5a, we compute the global correlation coefficient by considering the whole time sequence (2010–2022). As for Fig. 5b, we make a local analysis. For each detected SSE, we first extract SSE and tremor slices from intervals centered on the SSE dates $[t_{SSE}^{start} - \Delta t, t_{SSE}^{end} + \Delta t]$, where $\Delta t = 30$ days. We first compute the cross-correlation between the two curves to filter out detected SSEs whose similarity with tremors is not statistically significant, i.e., if their correlation coefficient is lower than 0.4. We build Fig. 5b after this process.

We compute Fig. 5d by comparing the SSE and tremor durations for all the events that had a cross-correlation higher than 0.4. For those, we infer the tremor duration, using the method explained in the section "Calculation of tremor durations" on the daily tremor rate cut from an interval, with $\Delta t' = 7$ days.

Overlap percentage calculation. In Fig. 4b we color-code the SSE durations by the overlap percentage between a pair of events, which we compute as the difference between the earliest end and the latest start, divided by the sum of the event lengths. Let E_1 and E_2 be two events with start and end dates given by (t_1^{start}, t_1^{end}) and (t_2^{start}, t_2^{end}) and with durations given by $D_1 = t_1^{end} - t_1^{start}$ and $D_2 = t_2^{end} - t_2^{start}$, respectively. We compute their overlap π as:

$$\pi = \frac{\max\left(0, \min\left(t_1^{\text{end}}, t_2^{\text{end}}\right) - \max\left(t_1^{\text{start}}, t_2^{\text{start}}\right)\right)}{D_1 + D_2} \tag{11}$$

Data availability

We downloaded the daily GNSS time series from the Nevada Geodetic Laboratory solution (http://geodesy.unr.edu). The script to download the data as well as the final slow slip event catalog are available at https://gricad-gitlab.univ-grenoble-alpes.fr/costangi/sse-detection.

Code availability

The source code of the SSEgenerator and SSEdetector along with the pre-trained model of the SSEdetector are available at https://gricad-gitlab.univ-grenoble-alpes.fr/costangi/sse-detection.

Received: 14 March 2023; Accepted: 13 November 2023;
Published online: 24 November 2023

References

1. Behr, W. M. & Bürgmann, R. What's down there? the structures, materials and environment of deep-seated slow slip and tremor. *Philos. Trans. R. Soc. A* **379**, 20200218 (2021).
2. Dragert, H., Wang, K. & James, T. S. A silent slip event on the deeper cascadia subduction interface. *Science* **292**, 1525–1528 (2001).
3. Lowry, A. R., Larson, K. M., Kostoglodov, V. & Bilham, R. Transient fault slip in guerrero, southern mexico. *Geophys. Res. Lett.* **28**, 3753–3756 (2001).
4. Schwartz, S. Y. & Rokosky, J. M. Slow slip events and seismic tremor at circum-pacific subduction zones. *Rev. Geophys.* **45** https://onlinelibrary.wiley.com/doi/full/10.1029/2006RG000208, https://onlinelibrary.wiley.com/doi/abs/10.1029/2006RG000208, https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2006RG000208 (2007).
5. Ide, S., Beroza, G. C., Shelly, D. R. & Uchide, T. A scaling law for slow earthquakes. *Nature* **447**, 76–79 (2007).
6. Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y. & Beroza, G. C. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nat. Commun.* **11**, 3952 (2020).
7. Rousset, B. et al. A geodetic matched filter search for slow slip with application to the mexico subduction zone. *J. Geophys. Res. Solid Earth* **122**, 10,498–10,514 (2017).
8. Frank, W. B. et al. Uncovering the geodetic signature of silent slip through repeating earthquakes. *Geophys. Res. Lett.* **42**, 2774–2779 (2015).
9. Gomberg, J., Wech, A., Creager, K., Obara, K. & Agnew, D. Reconsidering earthquake scaling. *Geophys. Res. Lett.* **43**, 6243–6251 (2016).
10. Hawthorne, J. C. & Bartlow, N. M. Observing and modeling the spectrum of a slow slip event. *J. Geophys. Res. Solid Earth* **123**, 4243–4265 (2018).
11. Frank, W. B. & Brodsky, E. E. Daily measurement of slow slip from low-frequency earthquakes is consistent with ordinary earthquake scaling. *Sci. Adv.* **5** https://www-science-org.insu.bib.cnrs.fr/doi/10.1126/sciadv.aaw9386 (2019).
12. Michel, S., Gualandi, A. & Avouac, J. P. Similar scaling laws for earthquakes and cascadia slow-slip events. *Nature* **574**, 522–526 (2019).
13. Bartlow, N. M., Miyazaki, S., Bradley, A. M. & Segall, P. Space-time correlation of slip and tremor during the 2009 cascadia slow slip event. *Geophys. Res. Lett.* **38** https://onlinelibrary.wiley.com/doi/full/10.1029/2011GL048714, https://onlinelibrary.wiley.com/doi/abs/10.1029/2011GL048714, https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2011GL048714 (2011).
14. Radiguet, M. et al. Slow slip events and strain accumulation in the guerrero gap, mexico. *J. Geophys. Res. Solid Earth* **117**, 4305 (2012).

15. Obara, K. Nonvolcanic deep tremor associated with subduction in southwest Japan. *Science* **296**, 1679–1681 (2002).
16. Rogers, G. & Dragert, H. Episodic tremor and slip on the cascadia subduction zone: The chatter of silent slip. *Science* **300**, 1942–1943 (2003).
17. Kong, Q. et al. Machine learning in seismology: turning data into insights. *Seismol. Res. Lett.* **90**, 3–14 (2019).
18. Zhu, W. & Beroza, G. C. Phasenet: a deep-neural-network-based seismic arrival-time picking method. *Geophys. J. Int.***216**, 261–273 (2019).
19. Woollam, J. et al. Seisbench-a toolbox for machine learning in seismology. *Seismol. Res. Lett.* **93**, 1695–1709 (2022).
20. Ross, Z. E., Trugman, D. T., Hauksson, E. & Shearer, P. M. Searching for hidden earthquakes in southern California. *Science* https://www-science-org.insu.bib.cnrs.fr/doi/10.1126/science.aaw6888 (2019).
21. Tan, Y. J. et al. Machine-learning-based high-resolution earthquake catalog reveals how complex fault structures were activated during the 2016-2017 central italy sequence. *Seismic Record* **1**, 11–19 (2021).
22. Ross, Z. E., Cochran, E. S., Trugman, D. T. & Smith, J. D. 3d fault architecture controls the dynamism of earthquake swarms. *Science* **368**, 1357–1361 (2020).
23. Tan, Y. J. & Marsan, D. Connecting a broad spectrum of transient slip on the san andreas fault. *Sci. Adv.* **6**, 2489–2503 (2020).
24. Rouet-Leduc, B., Jolivet, R., Dalaison, M., Johnson, P. A. & Hulbert, C. Autonomous extraction of millimeter-scale deformation in insar time series using deep learning. *Nat. Commun.* **12**, 1–11 (2021).
25. Costantino, G. et al. Seismic source characterization from gnss data using deep learning. *J. Geophys. Res. Solid Earth* **128**, e2022JB024930 (2023).
26. Wech, A. G. Interactive tremor monitoring. *Seismol. Res. Lett.* **81**, 664–669 (2010).
27. Hayes, G. P. et al. Slab2, a comprehensive subduction zone geometry model. *Science* **362**, 58–61 (2018).
28. Gao, H., Schmidt, D. A. & Weldon, R. J. Scaling relationships of source parameters for slow slip events. *Bull. Seismol. Soc. Am.* **102**, 352–360 (2012).
29. Okada, Y. Surface deformation due to shear and tensile faults in a half-space. *Bull. Seismol. Soc. Am.* **75**, 1135–1154 (1985).
30. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
31. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 03762 (2017).
32. Itoh, Y., Aoki, Y. & Fukuda, J. Imaging evolution of cascadia slow-slip event using high-rate gps. *Sci. Rep.* **12**, 1–12 (2022).
33. Bletery, Q. & Nocquet, J.-M. Slip bursts during coalescence of slow slip events in cascadia. *Nat. Commun.* **11**, 2159 (2020).
34. Ide, S. Variety and spatial heterogeneity of tectonic tremor worldwide. *J. Geophys. Res. Solid Earth* **117**, JB008840 (2012).
35. Schmidt, D. & Gao, H. Source parameters and time-dependent slip distributions of slow slip events on the cascadia subduction zone from 1998 to 2008. *J. Geophys. Res. Solid Earth* **115**, JB006045 (2010).
36. Licciardi, A., Bletery, Q., Rouet-Leduc, B., Ampuero, J.-P. & Juhel, K. Instantaneous tracking of earthquake growth with elastogravity signals. *Nature* **606**, 319–324 (2022).
37. Schreiber, T. & Schmitz, A. Surrogate time series. *Phys. D Nonlinear Phenom.* **142**, 346–382 (2000).
38. Lay, T. & Wallace, T. C. *Modern Global Seismology* (Elsevier, 1995).
39. Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning*. 448–456 (PMLR, 2015).
40. Agarap, A. F. Deep learning using rectified linear units (relu). https://arxiv.org/abs/1803.08375 (2018).
41. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
42. Bottou, L., Curtis, F. E. & Nocedal, J. Optimization methods for large-scale machine learning. *Siam Rev.* **60**, 223–311 (2018).
43. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. https://arxiv.org/abs/1412.6980 (2014).
44. He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE International Conference on Computer Vision*, 1026–1034 (IEEE, 2015).
45. Chollet, F. et al. Keras https://github.com/fchollet/keras (2015).
46. Abadi, M. et al. Tensorflow: large-scale machine learning on heterogeneous distributed systems. https://arxiv.org/abs/1603.04467 (2016).
47. Virtanen, P. et al. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nat. Method.* **17**, 261–272 (2020).

Acknowledgements

This work has been funded by ERC CoG 865963 DEEP-trigger. Most of the computations presented in this paper were performed using the GRICAD infrastructure (https://gricad.univ-grenoble-alpes.fr), which is supported by Grenoble research communities.

Author contributions

G.C. developed the SSEgenerator and SSEdetector and produced all the results and figures presented here. A.S. designed the study and provided expertise for the geodetic data analysis. S.G.-R. provided expertise for the deep-learning aspects. G.C. wrote the first draft of the paper. G.C., S.G.-R., M.R., M.D.M., D.M., and A.S. contributed to reviewing the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43247-023-01107-7.

Correspondence and requests for materials should be addressed to Giuseppe Costantino.

Peer review information *Communications Earth & Environment* thanks the anonymous reviewers for their contribution to the peer review of this work. Primary handling editors: Teng Wang, Joe Aslin, and Clare Davis. A peer review file is available.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.