Unsupervised Automatic Detection Of Transient Phenomena In InSAR Time-Series using Machine Learning

ANZA SHAKEEL

Unsupervised Automatic Detection Of Transient Phenomena In InSAR Time-Series using Machine Learning

Anza Shakeel

A thesis presented for the degree of

Doctor of Philosophy

Department of Earth Sciences

Durham University

United Kingdom

1^{st} September 2022

Unsupervised Automatic Detection Of Transient Phenomena In InSAR Time-Series using Machine Learning

Anza Shakeel

Abstract

The detection and measurement of transient episodes of crustal deformation from global InSAR datasets are crucial for a wide range of solid earth and natural hazard applications. But the large volumes of unlabelled data captured by satellites preclude manual systematic analysis, and the small signal-to-noise ratio makes the task difficult. In this thesis, I present a state-of-the-art, unsupervised and event-agnostic deep-learning based approach for the automatic identification of transient deformation events in noisy time-series of unwrapped InSAR images. I adopt an anomaly detection framework that learns the 'normal' spatio-temporal pattern of noise in the data, and which therefore identifies any transient deformation phenomena that deviate from this pattern as 'anomalies'. The deep-learning model is built around a bespoke autoencoder that includes convolutional and LSTM layers, as well as a neural network which acts as a bridge between the encoder and decoder. I train our model on real InSAR data from northern Turkey, and find it has an overall accuracy and true positive rate of around 85% when trying to detect synthetic deformation signals of length-scale $>$ 350 m and magnitude $>$ 4 cm. Furthermore, I also show the method can detect (1) a real M_w 5.7 earthquake in InSAR data from an entirely different region- SW Turkey, (2) a volcanic deformation in Domuyo, Argentina, (3) a synthetic slow-slip event and (4) an interseismic deformation around NAF in a descending frame in northern Turkey. Overall I show that my method is suitable for automated analysis of large, global InSAR datasets, and for robust detection and separation of deformation signals from nuisance signals in InSAR data.

Supervisors: Richard Walters, Noura Al Moubayed, Susanna Ebmeier, Mark Allen and Toby Breckon

Acknowledgements

All Praise to Allah, the Lord Of The Creation, and countless blessings and peace upon our Master Mohammed (PBUH), the leader of the Prophets.

I cannot forget the precious struggling moments of my life when I used to seek opportunities after the hustles of a tiring day. These moments aspire me daily to bring out a better version of myself and most importantly to not lose hope. Firstly, I thank Allah for guiding me on this path and my lovely parents who trusted me (their daughter) to pursue her dreams abroad. Each day of this course has taught me something new, which I will cherish forever. I thank and acknowledge my supervisors Rich, Noura and Susi for always providing me with the best support, and constructive feedback and for motivating me to achieve more. I would like to thank Andrew and Milan for helping me with the InSAR data that is used in this thesis. Last but not the least Mark, who always came to my rescue.

My PhD was funded by the Global Challenges Research Fund (GCRF) at Durham University. I am extremely grateful for this opportunity and would like to thank Douglas, Abir, all my friends from the GCRF cohort (Ayu, Naeema, Raquel, Laksmi, Atreyee, Sufia, Majo, Divya) and my dearest housemate Lucinda. As an international student, I was very lucky to be welcomed by them. Douglas and Abir provided us with all the support an international student can imagine.

Lastly, I cannot thank England enough for finding me the best husband and family-in-law who have been my rock during this write-up. Also, to my beloved niece whom I welcomed during this course.

I hope my efforts serve as a beacon of light for those who have already embarked upon their research journey.

Contents

Declaration

The publications that are pertinent to this thesis are given below. Some of the work described in this thesis has been published in journals. I was in charge of developing the approach, analysing and curating the data, and writing the manuscript in each case. There have been some edits made to both papers' texts to better align them with the thesis.

Publications

Some of the work presented in this thesis has been published in journals - the relevant publications are listed below. In each case, I was responsible for data processing and curation, development of the methodology and preparation of the manuscript. The text of both papers has been edited in some places to fit better within the thesis.

Chapter 3 has been published in the IEEE Transactions of Geo-science and Remote-sensing. A. Shakeel, R. J. Walters, S. K. Ebmeier, and N. Al Moubayed. Aladdin: Autoencoder-lstm-based anomaly detector of deformation in insar. IEEE Transactions on Geoscience and Remote Sensing, 60:1–12, 2022.

Chapter 4 and part of Chapter 5 is currently in review in the Journal of Geophysical Research of the American Geophysical Union.

List of Figures

List of Tables

Nomenclature

EO Earth Observation

GPS Global Positioning System

SAR Synthetic Aperture Radar

InSAR Interferometric SAR

LiCS Looking Inside the Continents from Space

CNN Convolutional Neural Network

FCN Fully Convolutional Network

LSTM Long Short Term Memory

FCN Fully Convolutional Network

TPR True Positive Rate

FPR False Positive Rate

OA Overall Accuracy

ALADDIn Autoencoder-LSTM based Anomaly Detector of Deformation in InSAR

Chapter 1

Introduction

Geohazards, including earthquakes, volcanic eruptions and landslides have major impacts on human lives and livelihoods. An important aspect of understanding and forecasting these phenomena are measurements of how they cause the ground to move. These measurements are critical for both research into geohazards and in monitoring their impact.

While earthquakes occur very rapidly, their aftermaths are devastating and long-lived. The Haiti Earthquake (largest ever recorded in Haiti) in 2010, affected one-third of the overall population (Margesson and Taft-Morales (2010)). In 2021 Haiti was hit by another earthquake of magnitude 7.2, the deadliest earthquake of the year that resulted in 2248 fatalities and an economic loss of USD 1.6 Billion (Aon, 2021 [†]). The risk of social and economic losses remains high for both low and high-hazard areas. Low-hazard areas are those that are often hit by infrequent severe earthquakes and population and assets are vulnerable in this case due to lack of preparation and delayed rescue response (Wei et al. (2015)). Figure 1.1 shows the distribution of around nineteen thousand earthquakes that occurred in the last 10 years. Whereas the Figure 1.2 shows the number of deaths recorded for each year, 222517 deaths out of the 226050 in 2010 are reported from the Haiti earthquake (Gupta and Sadiq (2010)).

[†]https://www.aon.com/empowerresults/helping-countries-recover-from-devastating-earthquakes/

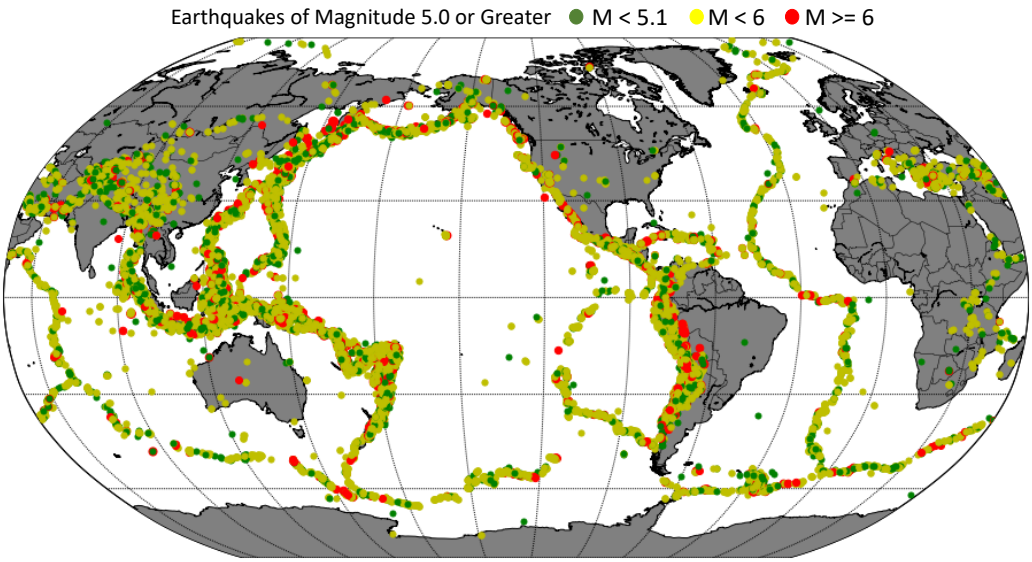

Figure 1.1: Global distribution of 19806 earthquakes between 2010-2020. Circles are coloured by earthquake magnitude. Data from http://earthquake.usgs.gov/earthquakes/

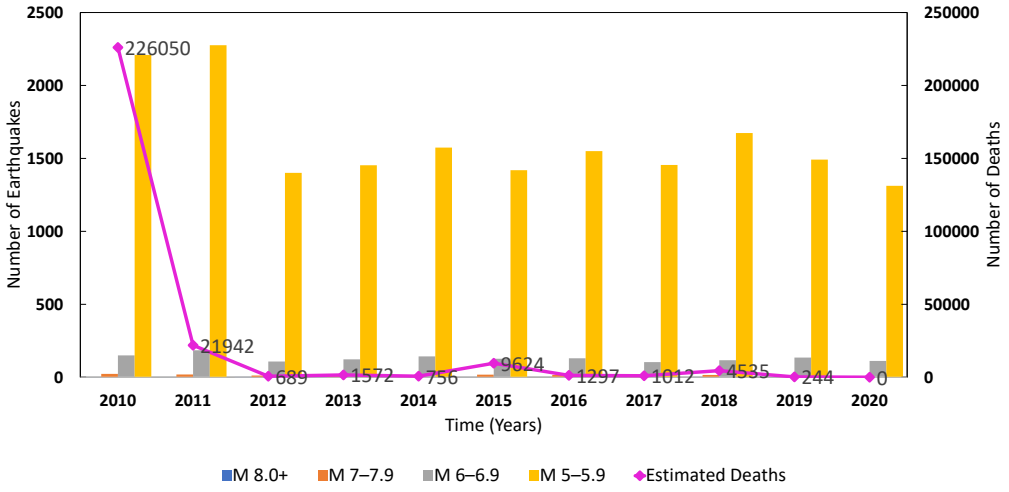

Figure 1.2: Estimated number of deaths for each year between 2010-2020 (right side of the vertical axis). Magnitude-wise number of earthquakes for each year for the past 10 years (left side of the vertical axis). For the year 2020, no data for the deaths is available. The plot is reproduced from the statistics taken from https://www.usgs.gov/programs/earthquake-hazards/lists-maps-and-statistics

To mitigate the risks involved or to speed up the response team, it is important to understand the dynamics of an earthquake and also its triggered aftershocks. The same is true for other geohazards like tsunamis, landslides, volcanoes, etc. Stud-

ies show that slow earthquakes and huge mega-thrust earthquakes have common characteristics and occur in neighbouring regions of the seismogenic zone (Obara and Kato (2016); Cruz-Atienza et al. (2021); Segall et al. (2006)). Monitoring and analysing such slow/silent seismic activities might lead the way to understand and classifying the underlying pattern to assess the probability of larger earthquakes.

When a fault ruptures at a high velocity (several km/s), an earthquake occurs and seismic waves are generated. Faults typically experience repeated semi-periodic failure in earthquakes as part of the earthquake cycle. The earthquake cycle consists of three phases, the inter-seismic, co-seismic, and post-seismic phases. During the inter-seismic phase, elastic strain accumulates in the shallow crust as the fault is 'locked' by friction. This accumulated strain is then released in the co-seismic phase when the fault ruptures seismically. Finally, stress concentrations caused by the earthquake are alleviated by slow, transient slip on the fault and by other aseismic viscous processes during the post-seismic phase. However, in some instances, certain faults may also release strain slowly, by rupturing over hours-weeks rather than seconds during a normal earthquake, this type of event is called a silent or slow earthquake and the energy released during these events is hard to detect by a seismometer. Satellite radar (InSAR) observations can be used instead to detect earthquakes, slow slip events, creep, post-seismic and interseismic deformation by instead measuring the small-magnitude ground deformation they cause (Nissen et al. (2011); Nippress et al. (2017)). Measurements of ground deformation serve as a critical indicator of what is occurring beneath a volcano. The ground usually swells when magma builds up in an underground reservoir before an eruption (called inflation). The earth above the reservoir also sinks as magma leaves the reservoir, possibly erupting (called deflation). Stresses that can cause movement on faults during earthquakes can cause volcanoes to deform (Babb et al. (2011)). Similarly, analyzing ground deformation is another method used to evaluate landslide activity. (Burrows et al. (2019)).

Earth Observation (EO) systems have revolutionised the understanding of the earth

and its dynamics, boosting the opportunity to monitor solid earth hazards (Elliott (2020)). EO satellites measure the ground deformation that is useful for localising or identifying the geographic location of a transient event, which is crucial for flagging regions that might be affected by aftershock events (Calais et al. (2010)). Also, it can be used to understand the temporal and spatial patterns of such activities. These deformation measurements carry useful information not only about the earthquake cycle but for volcanic processes as well. So, these datasets are equally valuable for the monitoring of volcanoes and related topographic changes (Ebmeier et al. (2018)). Pixel-by-pixel, event-based analysis of large, high-resolution EO data is strenuous and time-consuming. On top of that, the low signal-to-noise ratio and data gaps (increased number of missed days to months of image acquisition) makes the job even harder. In this thesis, I have successfully tried to mitigate manual labour exertion and developed a machine learning based event agnostic detection model that learns the spatial-temporal attributes of ground deformation in an unsupervised manner.

1.1 Ground deformation analysis - previous methods

Seismic waves have been used to study and formulate the theory behind earthquakes. Seismology has been used to monitor fractures, study earthquakes or eruptions that occurred in response to stress, and model systems that generate warnings before the occurrence of any such events (Allen and Kanamori (2003)). The installation of GPS stations across the globe opened gates to study ground deformations more deeply and so the motion vectors generated by the GPS combined with seismic waves recorded after an earthquake helped scientists to identify the slip, fault type, source, stress levels and intensity of the hit earthquake more accurately and precisely as compared to using these sources alone (Li et al. (2013)).

The capabilities of satellites have been improved over time and so the interference radar data of successive passes is a useful source to detect earthquakes. (Masson-

net et al. (1993)) detected the Landers earthquake in California-1992 using the interference radar data. Since then over 200 studies on 130 individual earthquakes are published, that have incorporated InSAR data in different models to study the earthquake story above and below the earth's surface (Funning and Garcia (2018)). The radar images captured by the consistent satellites providing global coverage have bypassed the expensive infrastructures that need to be installed and monitored for long times to record the movements, with an approximately equal level of location accuracy. These images have shown the way to the missed ruptures in past events like (Talebian et al. (2004)). The digital elevation model DEM made from stereo satellite images maps the earthquake ruptures and using them and monitoring the fault along past earthquakes around the same location can guide about the pattern of the historic and expected earthquake (Zhou et al. (2015)).

1.2 InSAR dataset

With the back-to-back launch of Sentinel 1A and Sentinel 1B , a minimum of a 6-day interval between SAR images has been established that is well-suited to capturing low-rate deformation with increased numbers of observations (Elliott et al. (2016)). In this thesis, I have focused on differential InSAR data that is interpreted in terms of the displacement of the earth's crust. I elaborate on the mathematics involved in the computation of this data and how it is useful to capture transient, deformation with automatic methods like artificial intelligence and machine learning.

1.2.1 Structure of InSAR data

A SAR image is a 2-dimensional array containing a complex number that has amplitude and phase ϕ information about the microwave radiations that are deflected by the earth's topology (Hanssen (2001)). The 'synthetic aperture' in the SAR allows high spatial resolution by synthetically increasing the aperture window size of the satellite without carrying large antennae (Chen et al. (2000)).

The change in phase $\Delta\phi$ is measured by comparing two SAR images of the same location but captured at a different time and angle tells us about the movement of the earth's surface. The complex representation of two SAR images A and B are given by eq 1.1 and eq 1.2:

$$A = |A|e^{j\phi_A}, \tag{1.1}$$

$$B = |B|e^{j\phi_B}, \tag{1.2}$$

where $j = \sqrt{-1}$, $|A|$ and $|B|$ is the absolute amplitude (Hanssen (2001)). So, an interferogram C can be found by the complex multiplication of A and B, given by eq 1.3:

$$C = |A||B|e^{j(\phi_A - \phi_B)}, \tag{1.3}$$

$\Delta\phi$ can be extracted from eq 1.3 $\Delta\phi = \phi_A - \phi_B$. The phase in the above equations varies from $-\infty$ to $+\infty$, but InSAR wraps these measurements in the range $(-\pi, \pi)$. The wrapped ϕ_w and the original measurements of phase ϕ are related by the given eq 1.4 (Hanssen (2001)):

$$\phi = \phi_w + 2K\pi, \tag{1.4}$$

where K is an integer and ϕ_w is in range $[-\pi, \pi)$. The interferogram generated by InSAR contains the coloured contours of phase change that depict the interference patterns, these wrapped phase ϕ_w contours are called fringes (Hanssen (2001)). To compute the surface deformation, these wrapped phase values ϕ_w need to be reverted to the original phase ϕ. This challenging process is called 'phase unwrapping', which is also the most difficult step in radar interferometry (Ghiglia and Pritt (1998)). To solve for K, it is assumed that the phase gradient between neighbouring pixels is in the range $[-\pi, \pi)$, this is assumption is based on a priori understanding of the topography. To equate phase gradients of ϕ_w (wrapped) to that of ϕ (true), the measured phase gradient is less $-\pi$ than a cycle is added and deducted if it is larger π (Hanssen (2001)). Then the phase gradients are added

using a flood-fill algorithm. The assumptions made to solve this are valid only if the phase gradients of ϕ are not affected by noise. In the case of faulty phase gradients, the error propagates through all pixels making the interferogram faulty (Hanssen (2001)). These errors can pose a problem for processing InSAR images and can hinder the detection of anomalies in it, this is further explained in Chapter 3 where the proposed method detects them as a false positive. In this thesis, I have used unwrapped interferograms to train, evaluate and test my methodology (in Chapters 3, 4 and 5). I have not performed 'phase unwrapping instead, used the processed and unwrapped interferograms from the LiCS Sentinel-1 InSAR portal (Lazecký et al. (2020)).

The total change in phase $\Delta\phi$ is affected by other factors given in eq 1.5 (Walters (2012)):

$$\Delta\theta_{Total} = \Delta\theta_{Deformation} + \Delta\theta_{Orbit} + \Delta\theta_{Topography} + \Delta\theta_{Atmospheric} + \Delta\theta_{Pix}, \quad (1.5)$$

where $\Delta\theta_{Deformation}$ is the phase delay caused by the earth's deformation, while noise contributes to the rest including orbital noise, topographic and atmospheric effects and changes due to the scattering properties of ground at each pixel. Thermal noise also contributes but is insignificant compared to the other four components (Bürgmann et al. (2000)). The change in position of the satellite at time T_1 as compared to time T_0, this known separation and change in the line of sight of the satellite adds a gradient-like effect to the images which is subtracted from them. $\Delta\theta_{Topography}$ is the irregular changes in elevation of the terrain on earth that affect the radar path length and vary with different satellite positions. DEM (Digital Elevation Model) is used to correct these effects. $\Delta\theta_{Atmospheric}$ are the variations in temperature and pressures that add to the ground deformation (Walters (2012); Lazecký et al. (2020)).

The signal of interest is $\Delta\theta_{Deformation}$, which is used to map ground surface movements in the LOS direction (Hanssen (2001)), the rest is considered as noise that is filtered and corrected to the most extent. This technique is called InSAR, it

measures 1D ground motion, however, the ground surface motion is in three spatial dimensions E (east), N (north) and U (up) (Figure 1.3, (Fuhrmann and Garthwaite (2019))). It is important not to neglect any certain dimension of motion or to make assumptions for a specific one, although measurements from polar-orbiting s satellites are necessarily less sensitive to North-South displacements. The deformation in 3D is given by eq 1.6:

$$\Delta\theta_{Def} = \frac{-4\pi\Delta r}{\lambda} = (\frac{4\pi}{\lambda})u \cdot \hat{p}, \tag{1.6}$$

where Δr is the range change when ground motion is away or towards the satellite between two SAR acquisitions and λ is the microwave wavelength (Chen et al. (2000)), $\hat{p}$ is the unit vector pointing from the target to the satellite and u is the 3D vector of motion (Walters (2012)). In this study I use unwrapped Sentinel-1 interferograms obtained from the global LiCSAR processing system developed by the UK's Centre for the Observation and Modelling of Earthquakes, Volcanoes and Tectonics (COMET) (Lazecký et al. (2020); Li et al. (2016))

1.2.2 Acquisition Geometry

Radar acquisitions are made twice, daytime and night-time. The satellite track that is 'descending' or is roughly travelling from N to S captures the daytime ground information. Whereas, when the track is 'ascending', that is travelling from S to N, it captures the night-time data (Figure 1.3). Figure 1.4 shows an example InSAR image that captures a ground deformation caused by an earthquake in China, the Figure shows both 'ascending' 1.4(a) and 'descending' 1.4(b) tracks of the event. As they acquire data from two different directions, they are used together with GPS measurements to mitigate the limitation of recovering 3D displacements from InSAR (Wright et al. (2004); Hu et al. (2014)).

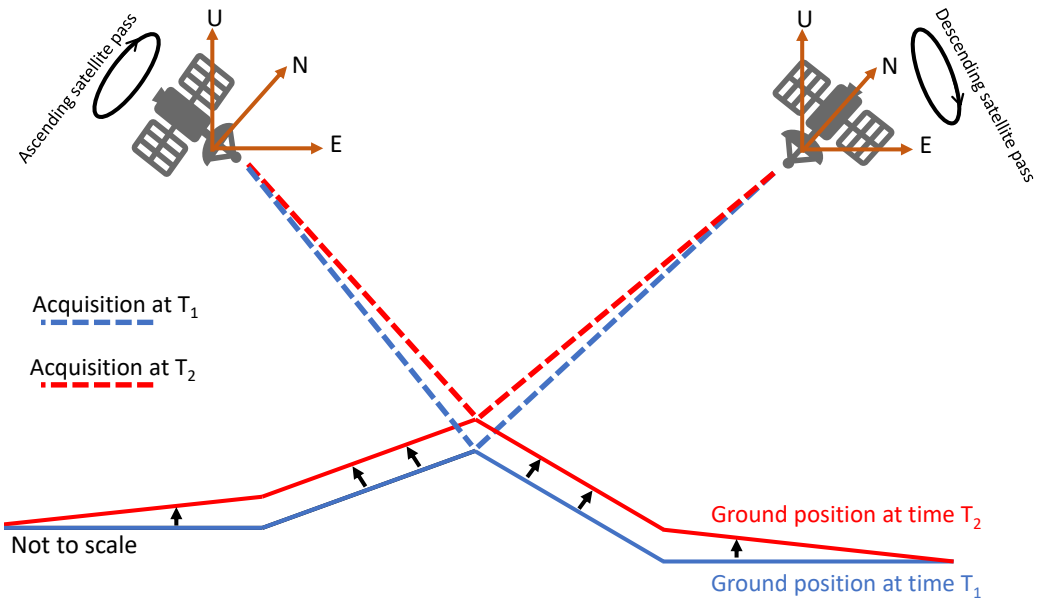

Figure 1.3: Cartoon explaining the measurement procedure of InSAR. Two satellites, (right) is the one following the descending track and (left) is the one following ascending track. The phase difference between acquisitions at time T_1 (blue dotted line) and T_2 (red dotted line) are used to compute the ground movement, with T_1 being pre-event and T_2 being the post-event of any seismic or atmospheric activity that caused the movement.

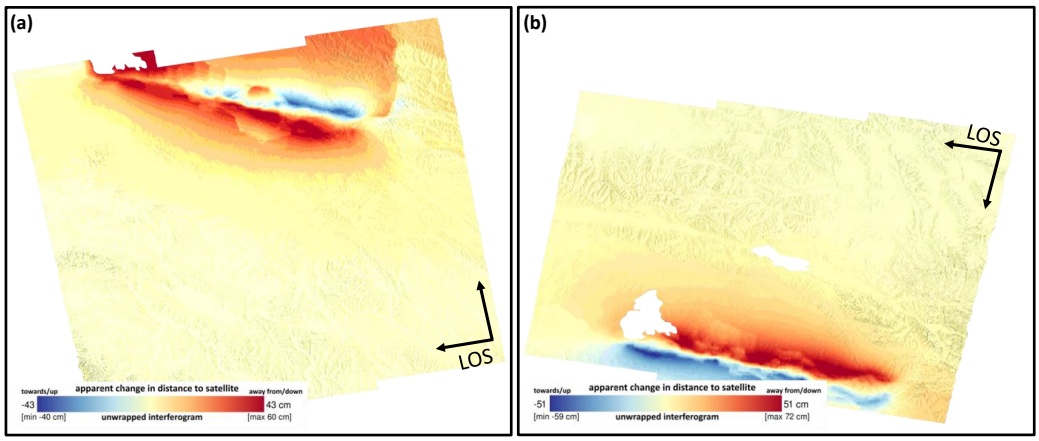

Figure 1.4: Examples of data. InSAR unwrapped images that have captured an earthquake of magnitude 7.4 in Southern Qinghai, China. (a) Image from the ascending track (that is from S to N) and (b) is the image from descending track (from N to S) of the Sentinel-1 satellite. The images are taken from https://comet.nerc.ac.uk/earthquakes/us7000e54r.html

1.3 LiCSAR processing system - Turkey data

Looking Inside the Continents from Space (LiCS) project by the Centre for the Observation and Modelling of Earthquakes, Volcanoes and Tectonics (COMET) is an automated InSAR processing chain for Sentinel-1 data (Lazecký et al. (2020)). The interferometric products (interferograms both wrapped and unwrapped) produced by LicSAR are freely available and easily accessible on the online portal. The aim of this is to map the global tectonic strain and to understand the spatio-temporal patterns when deformation takes place relative to earthquakes and magmatic activity (Lazecký et al. (2020)). Figure 1.5 shows the snapshot of the online portal, displaying the global coverage of the processing system. Turkey is currently the best-covered part of the world, with good interferogram coverage. The interferometric time-series data of Turkey is used to study the deformation field of the North Anatolian Fault (NAF) and East Anatolian Fault (EAF) (Walters (2012); Lazecký et al. (2020)). According to historical seismic records, the NAF has shown generally cyclical seismic behaviour since the seventeenth century. Over 100 years, recurring series of earthquakes rupture progressively from the east to the west (Sengör et al. (2005); Walters (2012)). According to GPS readings, the fault is thought to be shifting at a rate of 241 mm per year (McKlusky et al. (2000)). The NAF's total displacement is generally agreed to be between 80 and 85 kilometres, and it is believed that the fault has been active for 5 Ma (Barka et al. (2000); Bozkurt (2001)). The high rate of active fault and seismic activity makes Turkey a valuable source to study the underlying fault behaviour of the earthquake cycle and learn the spatial and temporal patterns relative to it. In this thesis, I have used interferograms from the area of Turkey to develop the methodology. The idea is to train the model on a generally active region so the learning of the model diversifies and so it can be applied elsewhere. I have provided the exact frame details and network figures of datasets used in Chapters 3, 4 and 5 respectively.

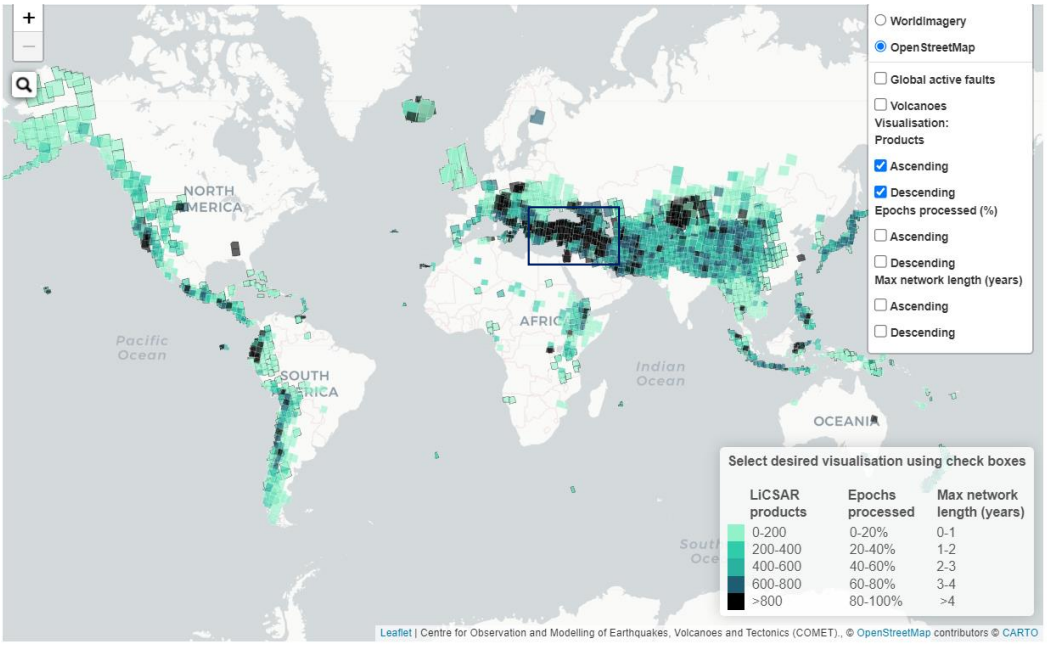

Figure 1.5: A snapshot of the LiCSAR online portal. It shows the global coverage of data processed and available at the portal. A total of 880, 250 including ascending and descending frames are mapped here. The blue box encapsulates the region of Turkey, which is the most captured location, with over 800 products starting from 2014 till 2021. This image is taken from (https://comet.nerc.ac.uk/comet-lics-portal/)

1.4 The aim of this thesis

It is important to fully understand, localize and detect the temporal stamp of any transient event causing major crustal/ground deformation, for example, an earthquake or a volcano, etc. InSAR with its regular, high precision and global coverage, carries useful spatial-temporal patterns that justify the nature of these events. But the large volumes of unlabelled data captured by current and next-generation satellites preclude manual systematic analysis, and the small magnitude of many signals of interest relative to background noise makes the task difficult. Machine learning and deep learning on the other hand the state-of-the-art technique to cater to and learn from such large datasets and these models have showcased high accuracy and efficiency for learning distinguishable patterns. Machine learning in the field of tectonics is still in its infancy and it is either applied to synthetic

data or event-specific labelled InSAR data. Similarly, rigid off-the-shelf models are applied to labelled data. Since the start of my research, I felt the need to develop a machine learning pipeline that is not only unsupervised but also event-agnostic. Because InSAR data captures every kind of deformation, with varying spatio-temporal patterns it should be possible to use this dataset to develop a single, but multi-purpose, diverse model.

I aim to develop, design and test, unsupervised deep learning based event-agnostic models to detect when in time and where in terms of location, transient deformation events occur. The main motive here is to evade manual analysis and present an accurate method for the automatic analysis of InSAR data.

1.4.1 Thesis structure

In this thesis, I will provide an application of machine learning, viable to use with/for InSAR data to broadly detect transient phenomena. The thesis is organised as follows:

- Chapter 2 discusses the machine learning methodology that can be applied to the InSAR data for its automatic analysis. It also presents an in-depth study of traditional and cutting-edge methodologies to detect earthquakes.
- Chapter 3 follows and describes the proposed deep learning based solution for an unsupervised, event agnostic analysis of InSAR data, called ALADDIn, that is adapted from Shakeel et al. (2022).
- Chapter 4 describes an improved methodology and training technique for the detection of transient phenomena. It also provides a comparison based on quantitative and qualitative analysis of both models. To prove the capabilities of my method, it is tested on two real case studies (an earthquake and a volcanic eruption) and a 2D Gaussian-based synthetic test dataset.

- Chapter 5 presents and analyses output when the models are trained on synthetic data. It also explains the process of creating this synthetic data and then evaluates all four models. The deduced best model from the evaluation is applied to a synthetic slow slip event and also to a descending frame capturing deformation around the NAF, North Turkey.
- Chapter 6 summarises the findings based on previous chapters and results. This chapter also discusses future work and research that can be held based on the findings of my thesis.

Chapter 2

Methods - Literature Review

While the satellites are capturing and storing useful information with high frequency and greater precision, it is also becoming impractical to manually analyse and interpret it for diverse attributes. In this chapter, I provide a brief overview of machine learning and deep learning, the mathematical background of these algorithms and operations used in them, general protocols and architectures for training on a variety of tasks, and finally their ability to learn from the complexities of InSAR data. Also, this chapter elaborates on the family of techniques, that are used to design and test deep learning models defined in Chapters 3 and 4. I conclude this chapter, by providing a literature review of existing techniques to process InSAR through deep learning.

2.1 Introduction to machine learning

Over the past decade, machine learning has vastly been developed and has proven itself to be an efficient and accurate technique to answer questions in a variety of fields including remote sensing (Maxwell et al. (2018);Lary et al. (2016)), health care (Qayyum et al. (2020);Ahmad et al. (2018)), finance (Emerson et al. (2019)), etc. Human competitive results for the task of classification in images revolutionized new ways of automation and the need for analysis of data. To bring all experimental data from the past and present to life, algorithms are designed that

are fit for automatic improvement over time through a learning experience or a learning algorithm (Jordan and Mitchell (2015)). A learning experience is defined as: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." (Mitchell et al. (1997)). As per this definition, the process of learning should not be confused with the task T, as learning is the ability to perform a task T (Goodfellow et al. (2016)). A task T can be described as how a learning algorithm processes a 'sample', for example, classification, detection, regression, translation, etc. A 'sample' can be represented by a vector $x \in \mathbb{R}^n$, where each entry x_i is a feature quantified from an event or an object we want to process. For example, the feature of an image, are the values of pixels that construct an image (Goodfellow et al. (2016)). The machine algorithms consist of five main parts: (I) dataset (collection of 'samples' or experimental data), (II) its representation (generated using a feature extractor), (III) a loss function (performance measure P), (IV) an optimizer and (V) a model (Goodfellow et al. (2016)). The metric P is critically selected depending on the task T performed and the capability of an algorithm is judged based on its output on unseen samples, often referred to as test data.

The learning experiences can be learned through three main classes: supervised learning, unsupervised learning, and reinforcement learning (Figure 2.1). Reinforcement learning improves its decision-making metric while interacting with a complex environment, for example, self-driven cars and gaming (Littman and Szepesvári (1996)). In this thesis, the questions I attempt to answer do not require such a form of learning, so, I have developed algorithms that are based on unsupervised and supervised learning and so do the available literature. Hence the topic of reinforcement learning is not discussed further.

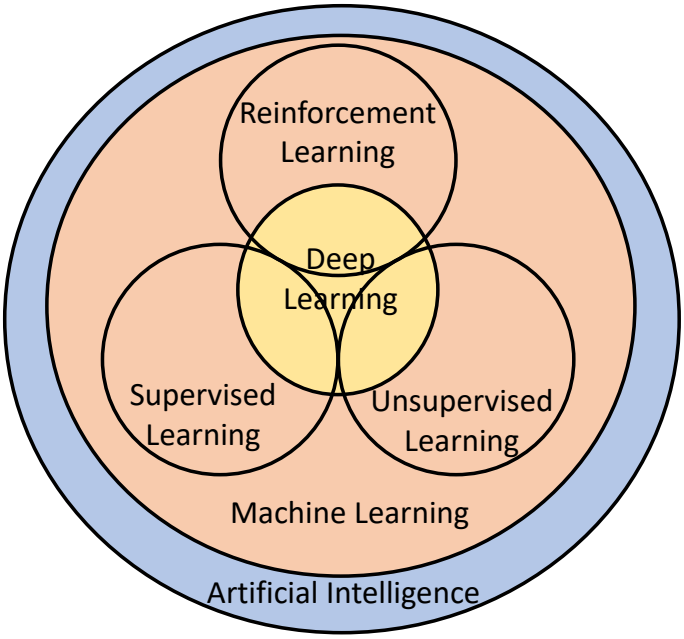

Figure 2.1: A Venn-diagram of artificial intelligence. Showing the relationship between all its branching families including machine learning, deep learning, reinforcement learning, supervised learning, and unsupervised learning. The Venn diagram is adapted from the Figure illustrated in (Bengio et al. (2017)).

2.1.1 Supervised Vs unsupervised learning

Machine learning algorithms E as defined previously are broadly categorized as supervised or unsupervised learning algorithms. The core distinction between these learning experiences lies in the type of dataset. The dataset can be marked or unmarked, marking or tagging a dataset is associating a 'label' with every sample in the data, that defines its attributes of it. For example, the ImageNet dataset (Deng et al. (2009)) contains 3.2 million real-world images, that are annotated and labels are present with each image that represent a specific category.

Generally, when annotations or labels of a dataset are available, then it would not be wrong to say that a 'guide' is present to instruct the algorithm to predict $'y'$ from $'x'$. Where $'y'$ is the associated label of the input $'x'$, this is called supervised learning. The performance metric P evaluates the learning based on the comparison of these predicted labels and the original annotated ones. Whereas, in the case of unsupervised learning, no such 'guide' is available. So, the algorithm attempts

to implicitly or explicitly learn the probability distribution $p(x)$ of the unlabelled data (Goodfellow et al. (2016)). Such algorithms learn the underlying properties and patterns of the data and group the data based on their distinctive patterns, for example, clustering. Structured output problems like classification, segmentation, regression, etc., lie under supervised learning, while density estimation is often categorized as unsupervised learning (Goodfellow et al. (2016)). Another variant, that lie within the boundaries of these learning algorithms is semi-supervised learning. Here the data is partially annotated or while learning for a task T, some useful information is fed with the input x that aids the learning experience.

2.2 Deep learning - a subset of machine learning

Deep learning as compared to many traditional machine learning algorithms, consists of a number of 'hidden' layers between an input and output layer. Conceptually, machine learning algorithms entirely depend on an external feature extractor (for example correlation, PCA, etc, Khalid et al. (2014)), that transforms the input and generates suitable representations from the raw data, for the algorithm to learn from (LeCun et al. (2015)). Whereas, in deep learning, the learning algorithms rely entirely on the number of "hidden" layers to develop features. During the process of learning, these layers do not expose the target output, hence they are called the hidden layers. The number of hidden layers used generally corresponds to the depth of the algorithm and so these learning algorithms are called deep learning models. This form of learning relies on artificial neural networks, that try to replicate the activities of human brain cells called neurons. The hidden layer consists of these neurons and the number of neurons in each hidden layer defines the depth of the model.

2.2.1 Theory

The flow of information from the input layer, passing these hidden layers and resorting to the output layer, can be represented by a function f^*, which comprises a group of functions. Because of this property of composing different functions together, they are called "networks". For example, if we have a network that contains three hidden layers, then the final output would be described by equation 2.1 and equation 2.2 (Goodfellow et al. (2016)).

$$\hat{y} = f^*(x), \tag{2.1}$$

$$f^* = f^3(f^2(f^1(x))), \tag{2.2}$$

where, x is input, f^1 is the first hidden layer, and f^2 and f^3 are the second and third hidden layers respectively. Each function maps its input to the input of the following layer (Figure 2.2). These types of models are called feedforward models because the information is flowing in one direction only and they are referred to as a "network" because of the chain of functions connected with each other (Goodfellow et al. (2016)). The function f^* can be defined as mapping its input to output with two basic parameters, weight w and bias b. So, following the same example of a neural network with three hidden layers (Figure 2.2) elaborated earlier. The output of first hidden layer $h_1 = g(f^1(x; w_1, b_1))$, second hidden layer $h_2 = g(f^2(h_1; w_2, b_2))$ and $h_3 = g(f^3(h_2; w_3, b_3))$. Where g is the activation function on top of each layer to introduce non-linearity in the system of equations. The functions f^1, f^2, f^3 are defined by equation 2.3,2.4,2.5,2.6:

$$h_1 = g(f^1(x; w_1, b_1)) = g(xw_1^\top + b_1), \tag{2.3}$$

$$h_2 = g(f^2(h_1; w_2, b_2)) = g(h_1 w_2^\top + b_2), \tag{2.4}$$

$$h_3 = g(f^3(h_2; w_3, b_3)) = g(h_2 w_3^\top + b_3), \tag{2.5}$$

$$\hat{y} = h_3 = g(f^3(g(f^2(g(f^1(x; w_1, b_1)); w_2, b_2)); w_3, b_3)), \tag{2.6}$$

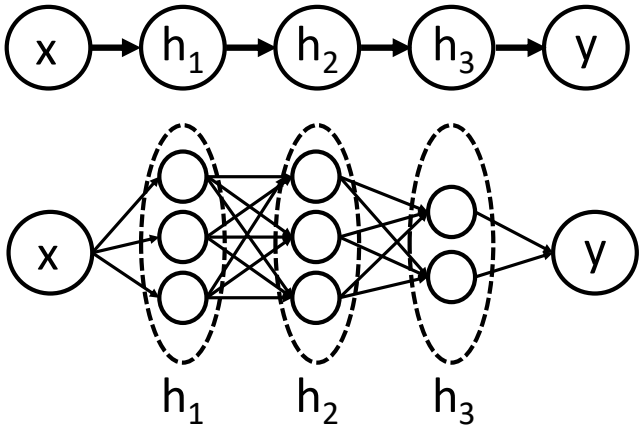

Figure 2.2: An example of a neural network is illustrated in two styles. (top) Shows the input x going forward to hidden layers 1,2 and 3. (bottom) Shows the same neural network, expressing the number of neurons in each hidden layer. The depth and width of this example model can be seen by the number of hidden layers and the number of neurons in each layer.

The parameters weight w and bias b are initialized randomly and are of different shapes depending on the number of neurons in each layer.

The activation function that provides non-linearity to the network is essential because it allows learning the difference concerning the loss. The most commonly used activation functions (Figure 2.3) are a $sigmoid(z) = \frac{1}{1+e^{-z}}$ (Cybenko (1989)), $Tanh(z) = \frac{1-e^{-2x}}{1+e^{-2x}}$ (Malfliet and Hereman (1996)), rectified linear unit (RELU) $Relu(z) = max(0, z)$ (Glorot et al. (2011)) and $softmax(z) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K$ (Bridle (1989)). Over the years these functions have been modified and adapted as per network and task requirements. Softmax and sigmoid are generally used with the output layer, to transform activations from neural layers to probabilities or class scores for classification purposes.

After developing the network, the training protocols need to be set up. The foremost protocol is to learn while minimizing the loss $P(\theta)$. The loss or cost function is a comparison between the predicted output $\hat{y}$ and the original label y (for the case of supervised learning). $P(\theta)$ is also one of the sources of performance measures. This means of iterative training is called gradient-based optimized learning (Goodfellow et al. (2016)). To apply this form of learning, it is important to selectively

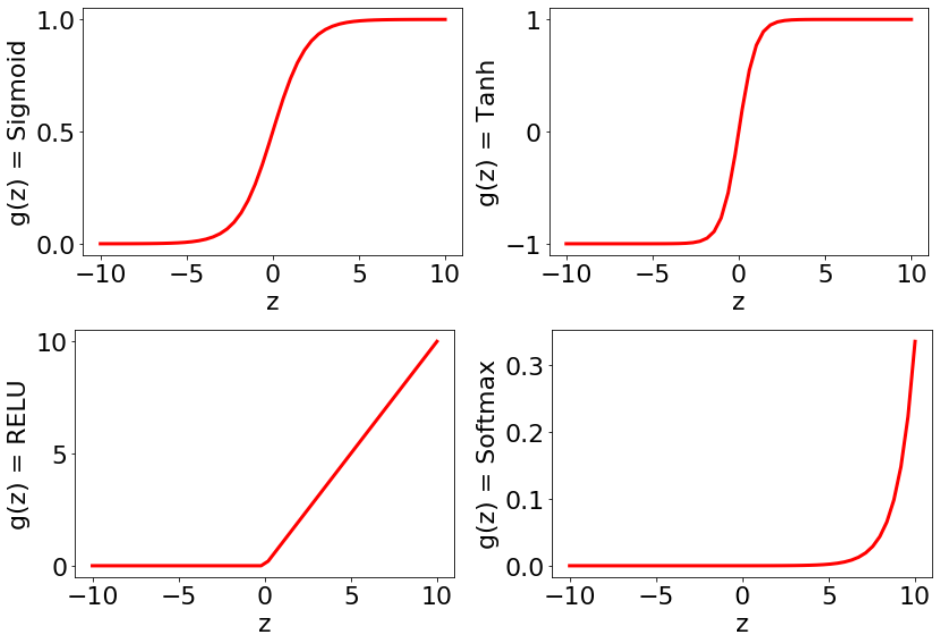

Figure 2.3: Some of the commonly used activation functions. (Top Left) Sigmoid function $g(z) = \frac{1}{1+e^{-z}}$ plotted on range of values z from (-10 to 10). (Top Right) Tanh function $g(z) = \frac{1-e^{-2x}}{1+e^{-2x}}$ plotted on range of values z. (Bottom Left) RELU function $g(z) = max(0, z)$ plotted on a range of values z. (Bottom Right) softmax function $g(z) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K$ plotted on range of values z.

choose or compose from scratch a cost function, that best suits the model and task's requirements. The most popular is the binary cross-entropy (BCE) (equation 2.7), this cost function is used for binary classification, depending on probabilities ($\hat{y}$ needs to be transformed to probabilities for example, by using the softmax activation function). For regression, mean absolute error (MAE) and mean squared error (MSE) is used. They are defined in the equation 2.8 and 2.9 respectively. In this thesis, I have used MSE and MAE as loss functions to design my models for the purpose of unsupervised anomaly detection, concisely explained in the Chapters 3 and 4.

$$BCE = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})), \tag{2.7}$$

$$MAE = \sum_{i=1}^{D} |y_i - \hat{y}_i|, \tag{2.8}$$

$$MSE = \sum_{i=1}^{D} (y_i - \hat{y}_i)^2, \tag{2.9}$$

Where y_i is the ground-truth (input), $\hat{y}_i$ is the output and i is iterated over total number of samples D.

2.2.1.1 Back-Propagation

While the information is propagated forward from the input to hidden layers and finally to predict output and compute cost $P(\theta)$. The flow of information until here is one-sided, back-propagation (Hecht-Nielsen (1992); Rumelhart et al. (1995)) allows this flow backward, from a cost function to each hidden layer in the reverse direction. The algorithm back-propagation is a way to compute gradients that assists iterative training for gradient-based optimization (Goodfellow et al. (2016)). It uses chain-rule calculus to compute the derivatives, in a specific order of propagation.

To define the algorithm, it is important to identify the dimensions of the parameters, that is the weight w and bias b for each layer. The array consisting of weights should be of size $(n \times j)$, where n is the number of neurons in the hidden layer and j are the input units. Following the example, the neural network shown above (Figure 2.2), for the first hidden layer h_1, the weights w_1 are of size (3×1). Respectively, for h_2 and h_3, the size of w_2 and w_3 is, (3×3) and (2×3). And for the output layer, the size of weights would be (1×2). Weights are assigned for each connection (represented by arrows in Figure 2.2 (bottom)), but while computations, they are converted to an array format. Each row of array w_n^l, corresponds to the weights of n^{th} neuron of the l^{th} layer. Whereas, bias b^l is a constant value added for each layer l.

Backpropagation works by updating every weight parameter concerning the cost function. Let's say the cost function or the total loss $P(\theta)$ is computed using MAE given in the equation 2.8. Then, using the chain rule and partial derivatives, the gradient for each weight, concerning the cost $P(\theta)$ can be calculated. For example, the gradient for w_1^2 (1^{st} neuron in 2^{nd} layer) is given by 2.10:

$$\frac{\partial P}{\partial w_1^2} = \frac{\partial P}{\partial h_3} \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial h_2}{\partial w_1^2}, \tag{2.10}$$

where h_3 and h_2 are layer outputs computed using equation 2.4 and 2.5. The weights are then subtracted by their respective gradients by multiplying with a value α, called the learning rate. This is shown in the equation 2.11:

$$\hat{w_1^2} = w_1^2 - \alpha \frac{\partial P}{\partial w_1^2}, \tag{2.11}$$

The learning rate is an optimizing parameter, which is the step size with which the weights are updated at each iteration (Goodfellow et al. (2016)). This hyperparameter controls the extent of the cost function that needs to be sent back to update the weights. The value can be selected from a range of 0.0 to 1.0 It is correlated with the number of training epochs. A training epoch is when the network is iterated over all data. So, a low learning rate means, a greater number of epochs as the weights are updated slowly, this ensures that no local-minima is missed while the model slowly converges towards optimal accuracy. Whereas, a high learning rate means rapid changes achieved within less number of epochs (Bengio (2012)), these drastic changes might cause a divergent behavior and instead of converging to local-minima, the model weights might diverge far from it. While designing the models elaborated in Chapters 3 and 4, I started with training a batch of data with a relatively high learning rate and then decreased it until the model achieved optimal accuracy. This can be engineered by monitoring the training as well as validation cost and accuracy for each epoch. Deep learning platforms like TensorFlow (Abadi et al. (2015)) provide algorithms to automatically schedule the learning rate while training. It can also be tuned cyclically by providing a range of learning rates and the algorithm finds the best value by monitoring the convergence and divergence of cost function (Smith (2017)).

2.2.2 Convolutional Neural Network (CNN)

Convolutional neural networks are a type of neural network that can process spatial, multi-dimensional data, for example, images which is a 2D or 3D array of pixels and videos, etc (LeCun et al. (1995)). The difference between a neural and a convolutional neural network is that of matrix multiplication, the latter employs the mathematical operation of convolution instead of general matrix multiplication. The convolution operation is denoted by an $'*'$, it is applied between the input x and a kernel K and the output is often called a feature map. Equation 2.12 defines this operation for 2D input x (Goodfellow et al. (2016)):

$$z(i,j) = (x * K)(i,j) = \sum_m \sum_n x(m,n)K(i-m, j-n), \tag{2.12}$$

The kernel K can be of size much smaller than the input x as its purpose is t sort each element of x separately. In the equation, 2.12, m and n are the pixel location of x, and i and j are the ones for the output feature map z. For a 3D x, a 3D K computes feature maps in each dimension. When implementing the convolution, it is performed as a dot product, as shown in Figure 2.4. To ensure each element in a multi-dimensional input x is processed, other parameters like stride S, and padding P are also considered while applying this operation. Stride is the gap with which K is applied to x. So, the size of the feature map can be deduced by the equation 2.13,

$$n_{out} = \frac{n_{in} + 2P - k}{S} + 1, \tag{2.13}$$

where n_{out} is the output size of the feature map, n_{in} is the size of input x, P represents padding that is the number of rows and columns of zeros appended at the corners of x to maintain the feature size and to ensure kernel K is fully applied x, k is the size of filter applied (Figure 2.4). For example, if x is of size (3×4) and size of K is (2×2), and convolution is applied with $S = 1$ and $P = 0$ then

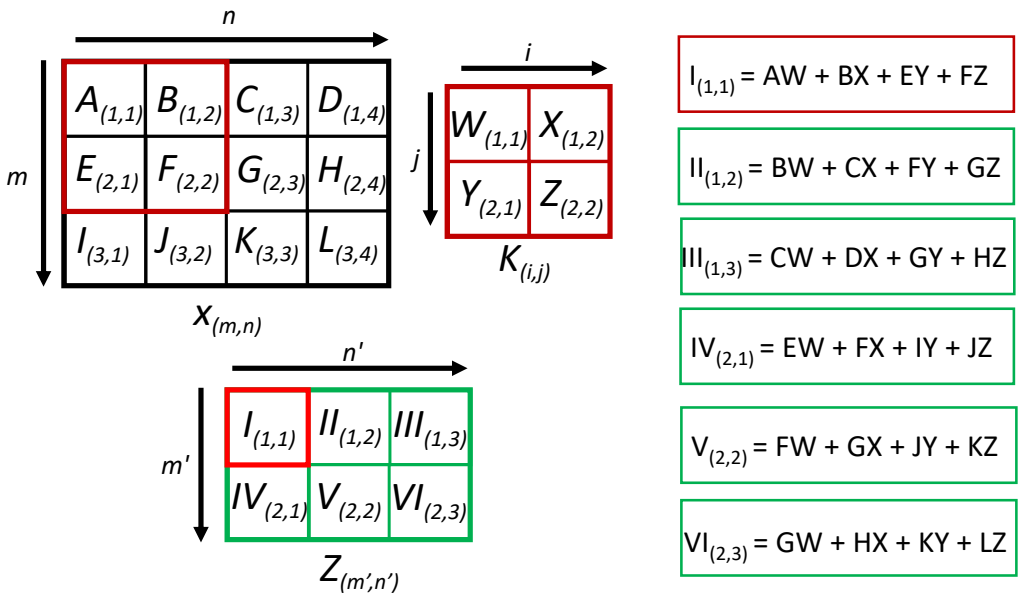

Figure 2.4: The mathematical operation of convolution. It is performed between input $X_{(m,n)}$ and kernel $K_{(i,j)}$ to produce feature map $Z_{(m',n')}$ using equation 2.12. Here the operation is performed with stride $S = 1$ and padding $P = 0$. The output size of the feature map Z can be deduced by the equation 2.13. The red boundary shows K and its application on X. The illustration is adapted from the Figure illustrated in (Goodfellow et al. (2016)).

as per equation 2.13 (applied on each dimension), the size of output feature map would be (2×3). The kernel K can be initialized with randomly generated weights w or designed as per needs. The most popular and effective technique in terms of accuracy is to generate them randomly (Saxe et al. (2011); Cao et al. (2018)).

As seen in the previous section, every neuron in the neural network is connected to every other unit (see Figure 2.2 (bottom)). Whereas convolutional neural network has 'sparse connectivity, this is achieved by making the kernel K smaller than input x (Goodfellow et al. (2016)). This representation is computationally inexpensive, as it requires fewer operations and parameters as compared to the neural network. Another advantage of convolution is that it is equivariance to translation. If the output of the function changes in the same way as the input, then that function is called equivariant ($f(v(x) = v(f(x))$). Let's say, $v(x)$ is a function that shifts input image x to the right with one unit, then $x'(i, j) = v(x) = x(i - 1, j)$. The output would be the same when convolution is applied to x before transformation

or when it is applied to x' (Goodfellow et al. (2016)).

2.2.3 Recurrent neural networks (RNN)

This form of the neural network is capable to process sequential data, for example, 1D temporal sequence, sentences of a particular language, audio data, etc, it is often called sequential learning. Recurrent neural networks (RNNs) can learn from and process a sequence of values ($x^1 ... x^T$) of varying lengths (Rumelhart et al. (1986)). A recurrent system can be defined by the equation 2.14 and s^t is the state of the system and t is the time-step that may or may not be infinite (Goodfellow et al. (2016)):

$$s^t = f(s^{t-1}; \theta), \tag{2.14}$$

The equation can be rewritten with a neural network perspective by replacing the state s^t with a hidden state h^t and input x, equation 2.15:

$$h^t = f(h^{t-1}, x^t; \theta), \tag{2.15}$$

For a feed-forward recurrent neural network, the system of equations can be developed by introducing the weights $w^{'}, w^{''}, w^{'''}$ and biases $b^{'}, b^{''}$ with $g(h; x)$ being the activation function, for example, a softmax can be applied on output and sigmoid or tanh can be applied elsewhere. This system is defined in the equation 2.16 and 2.17:

$$h^t = g(h^{t-1} w^{'} + b^{'} + x^t w^{''}), \tag{2.16}$$

$$\hat{y} = g(h^t w^{'''} + b^{''}), \tag{2.17}$$

There are different ways to group recurrent networks and there are different types of such networks. In this thesis, the nature of the input InSAR data set is sequential and contains spatio-temporal patterns in a multi-dimensional setting. So, I have worked on the Long Short-term memory (LSTM) that is called gated RNNs and is capable of retaining features in memory.

2.2.3.1 Long Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a part of gated-RNNs. Each LSTM unit is referred to as a 'cell', these cells are recurrently connected through paths called gates (Goodfellow et al. (2016)). RNNs are generally used for speech recognition, language modelling, translation, and image captioning (Miao et al. (2015); Cho et al. (2014); Mao et al. (2014)) as they are more short-term sequence dependant due to their time-based connections within the network. In contrast, LSTMs are capable of retaining information for longer intervals so they have been successfully used for capturing changes that are prolonged in time, for example in analyzing CCTV videos for surveillance (Shah et al. (2018)). The LSTM cells are connected together with gates through which they share information (as shown in Figure 2.5). This includes the cell state, that can go unchanged or can be updated or deleted according to the input. A sigmoid function σ is used at the output of each gate, predicting how much information should be allowed to pass further. The output of the LSTM is based on equations 2.18, 2.19 and 2.20, where f is the forget gate, i is input gate, o is the output gate, C is the cell state, W and b are the weight and bias matrices of each gate while h^{t-1} is output from previous time frame and x^t is the input at present time frame.

$$f^t = \sigma(W^f.[h^{t-1} + x^t] + b^f), \tag{2.18}$$

$$\begin{aligned} i^t &= \sigma(W^i.[h^{t-1}, x^t] + b^i), \\ C^{t'} &= tanh(W^C.[h^{t-1}, x^t] + b^C), \\ C^t &= f^t \times C^{t-1} + i^t \times C^{t'}, \end{aligned} \tag{2.19}$$

$$\begin{aligned} o^t &= \sigma(W^o.[h^{t-1}, x^t] + b^o), \\ h^t &= o^t \times tanh(c^t) \end{aligned} \tag{2.20}$$

The forget gate shown in the equation 2.18 is a neural network that outputs based on previous response and current input, it controls how much information to retain.

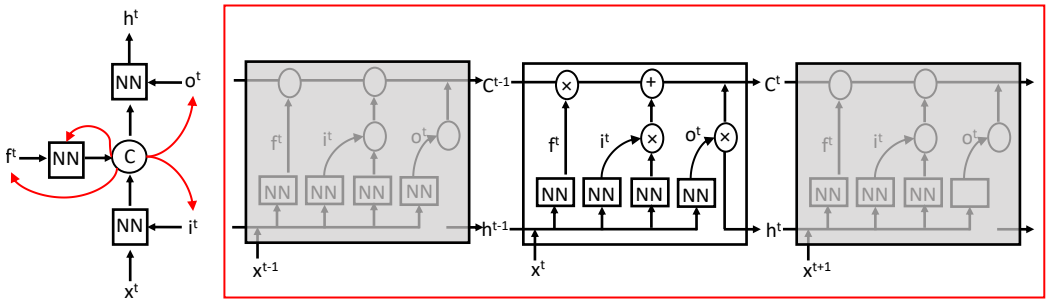

Figure 2.5: (Left) LSTM cell. (Right) Unrolled recurrent network of a LSTM. Where x^t is input at time t, f^t, i^t and o^t are forgotten (2.18), input (2.19) and output gates (2.20). The box called NN is a neural network that is bounded by an activation function, sigmoid σ in case of f^t, i^t, and tanh for o^t. Red lines corresponding in (Left) correspond to the recurrent paths unrolled in (Right). Mathematical operations of matrix multiplication and addition are shown in circles.

The input gate is also a neural network that controls what new information is required to update the cell state C^t as shown in the equation 2.19. The cell state is updated based on the previous cell state and the information passed through the input gate. The output gate that is shown in the equation 2.20 works similarly to equation 2.18. Finally, the output at a particular time frame h^t depends on the cell state and the output gate O^t. As neural networks are the basis of LSTM cells, they often miss the spatial information/neighborhood context of multidimensional data.

In order to learn both colloquial/informal, the structure of the input data in the space as well as in time, these LSTM cells are applied with convolutions instead of matrix multiplications in equations 2.18, 2.19 and 2.20. These are called Convolutional LSTMs and can maintain the dimensions of the input data in the case of an image or a video. Where there is no ground truth available and it is both expensive and time-consuming to mark, label, or caption abundant videos/images data, unsupervised or semi-supervised deep learning techniques involving convolutional LSTMs (Xingjian et al. (2015)) are used to understand changes and track object movements (Ning et al. (2017)) in them. Such techniques are often categorized as anomaly detection, either using CNNs and RNNs separately or using them together in a layer as one operation.

2.2.4 Designing network architectures

Deep learning architectures, refers to the combination of layers that are best suited to solve the problem, considering the available data set. The final architecture that is trained is also referred to as a model. As mentioned in previous sections as well, the depth of a model refers to the number of layers and the width of the model refers to the number of neurons (in the case of a NN), number of filters/kernels (for a CNN) and recurrent units (for a RNN). The goal is to design an architecture that (I) best generalize the problem or the function to be learned and (II) that perfectly fits the training data. Studies show, that a deeper model better generalizes for a variety of tasks (Bengio et al. (2007); Szegedy et al. (2015)). To overcome these challenges, the hyper-parameters are tuned; hyperparameters are parameters explicitly involved in the training process. Variables like learning rate, number of epochs, and architectural design are the hyperparameters for deep learning algorithms.

The two kinds of training scenarios one should be careful about are the under-fitting and over-fitting of the deep learning network. These issues are often identified during the validation process. Validation is the process of evaluating the model after a set of training epochs. The performance metrics P are observed and if the validation loss exceeds after certain epochs and training loss keeps on decreasing is when the model has over-fitted and learned/generalized only on the training data. This occurs due to training a deep architecture on relatively small and simple data. The performance metrics P converges quickly when the complex architecture tries to fit uniform data on a simple task resulting in the model to over-fit. To overcome this, firstly, the architecture is rechecked and re-engineered by removing unnecessary operations, and secondly by lowering the learning rate. So, the optimization of the cost function is slow. Also, "early stopping" can be evoked, an algorithm that monitors validation loss and terminates training just before over-fitting (Goodfellow et al. (2016)). On the other hand, under-fitting is

when the model poorly performs on all portions of data sets (training, validation, and testing), it fails to generalize the data. It is usually improved by increasing model complexity, cleaning noisy data, and increasing the number of data samples, data augmentation techniques can be incorporated that assist in model training (Perez and Wang (2017)).

Different types of architectures are designed and available for varying tasks and datasets. The most common ones apart from the classical deep CNNs (AlexNet (Krizhevsky et al. (2012)), ResNet (He et al. (2016))) are fully convolutional networks (FCN) (Dai et al. (2016)), and the output layer is a convolution layer predicting a 2D map of labels. Such models are mainly used for image generation. Generative adversarial networks (GAN) are also used for image generation or sequence-to-sequence learning (Goodfellow et al. (2014)). They are based on accompanied learning of two sub-models, a generator, and a discriminator. Generator G generates images that are passed on to a discriminator that classifies them for being real or fake. Autoencoders and Variational Autoencoders (VAE) (Doersch (2016)) are groups of CNNs divided into two branches within a model, an encoder that maps or compresses the input data into a latent space (which is learned by training the model) and a decoder that converts data using this learned latent space. Autoencoders are fully discussed in this thesis, as they are the backbone of the methodology developed (explained in Chapters 3 and 4).

2.2.4.1 Task-oriented deep learning

One of the most important aspects to focus on, while designing the architecture of a deep learning model is the task T to learn. Some of the most common tasks are given as follows:

1. Classification: This type of task, involves categorizing labeled data into multiple categories, often called 'multi-task classification'. The other one is where the model is tasked to differentiate between two classes or identify one class

in noisy data, this is called 'binary classification'. Object recognition also lies under this task, where objects are identified in images and then categorized according to assigned labels (Simonyan and Zisserman (2014); Uçar et al. (2017); Sun et al. (2018)). The task of segmentation is also part of this group, where objects are recognized and boundaries of the identified objects are marked, this is done on a pixel-by-pixel basis in the case of videos or image data (Ronneberger et al. (2015); Badrinarayanan et al. (2017)).

2. Regression: This type of task is very similar to the task of classification, instead the predicted output is a numerical value (Goodfellow et al. (2016); Kendall et al. (2017); Shakeel et al. (2019)).

3. Structured output: This family of tasks involves predicting a series or a vector that holds an objective. This type of task includes machine translation and transcription (from one language to another), language modelling, captioning, and annotating data (Goodfellow et al. (2016); Wang et al. (2016); Tenney et al. (2019)).

4. Anomaly detection: This type of task involves parsing through a series of data and flagging unusual or atypical events (Goodfellow et al. (2016)). In this thesis, I have tried to solve the problem of analysing the InSAR data through the technique of anomaly detection. Therefore, it is elaborated in the following sections.

2.2.4.2 Unsupervised anomaly detection

The task of anomaly detection is based on separating outliers from a set of data that is either labeled or assumed to be normal. The first and foremost issue regarding anomaly detection lies in the data. Anomalies are rare or unusual events; they are unknown, only until they occur; they are irregular and demonstrate abnormal behavior (Pang et al. (2021)). Methods that are used to solve anomaly detection

involve, firstly, feature extraction, and then those features are passed on to statistical modelling (Dukkipati et al. (2016); Eskin (2000)) or neighbourhood methods, that rely on the assumption that normal data has fewer neighbours as compared to normal data (Bosman et al. (2017); Liu et al. (2021)). Secondly, learning feature representation of normality, and thirdly, end-to-end anomaly score learning (Pang et al. (2021)).

For feature extraction, principal component analysis (PCA) is the most common extraction technique (Candès et al. (2011)). However, the CNN-based feature extraction method has outperformed such models, CNNs are capable of extracting semantic-rich, discriminative features that help separate anomalies from normal data and reduce false positives (Goodfellow et al. (2016)). Pre-trained, off-the-shelf models like AlexNet (Krizhevsky et al. (2012)), and ResNet (He et al. (2016)) are used to generate low-dimensional features.

The deep learning architecture of the autoencoder is generally used for normality representation learning (Pang et al. (2021)). The latent space learned by the encoder, best represents the normal samples in the data and so, it accurately reconstructs the normal data. This leads to low reconstruction error when an anomalous sample is processed through the system (Ionescu et al. (2019); Theis et al. (2017); Goodfellow et al. (2016)). GANs are also widely used for representation learning. The discriminator loss, which identifies whether the generated output is real or fake is moulded as a reconstruction error and acts differently when an anomaly occurs (Schlegl et al. (2017)). The disadvantage of using GANs is that they are computationally expensive and difficult to converge on complex data, their generated output can be biased and misled anomaly score as the objective behind its generator is image synthesis (Pang et al. (2021)).

I have designed and developed a deep learning network architecture that performs the task of anomaly detection using an autoencoder. An in-depth explanation of this methodology is provided in the Chapters 3 and 4 of this thesis.

2.3 Deep learning for InSAR

Learning patterns of earth deformation from the InSAR data is an unsupervised learning problem. The InSAR images contain the change in phase that is captured by the satellite between two-time acquisitions. In contrast to video data, where an individual image or frame contains information on the position of objects at a single acquisition time, an individual radar interferogram contains information on the difference in position between two acquisition times. If a satellite has N_t time acquisitions then, we can generate a maximum of N_{IFG} interferograms using the equation 2.21:

$$N_{IFG} = \frac{N_t(N_t - 1)}{2}, \tag{2.21}$$

Each interferogram contains the change in phase between two acquisition times, denoted here as 'pre' and 'post'. Of these N_{IFG} interferograms, only N_t are independent. Figure 2.6 shows the difference between a regular time series of video frames and a sequence of changes captured by a series of short time-span of interferograms.

Anomaly detection techniques can be used for these unlabelled combinations of images across time. The patterns in images or videos are learned by these models and are trained for reconstruction. The reconstruction loss is thus an indicator of any anomaly. Weakly labelled regular video frames are used with added features from HOG (Histogram of oriented gradient) and HOF (Histogram of optical flow) to train a fully connected autoencoder and to preserve the spatial information a fully convolutional autoencoder is also trained by (Hasan et al. (2016)). The model then segments the irregular time frames within a video clip based on its regularity score. To learn the spatial changes concerning time frames, convolutional Long Short Term Memory cells (LSTM) are used. LSTM is a type of recurrent neural network that retains information in its memory cell (Xingjian et al. (2015)), the details of

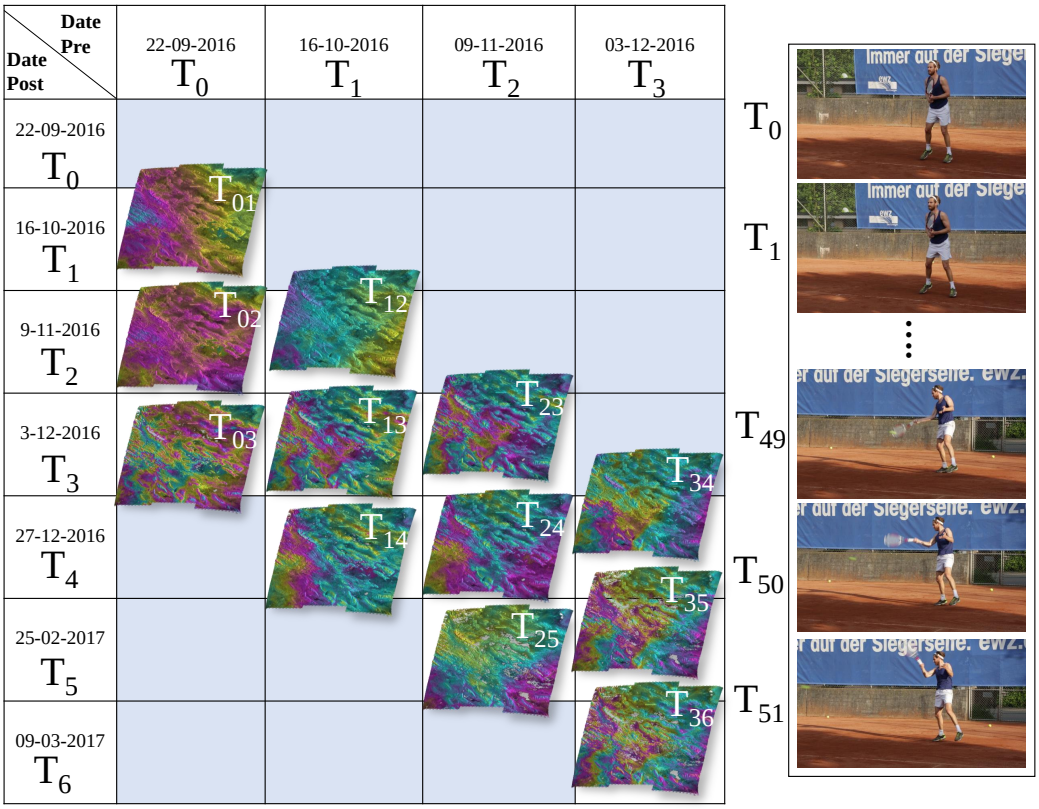

Figure 2.6: Comparison between the sequential nature of InSAR data and regular real-world videos dataset. (Right) Frames are taken from a video data set (Perazzi et al. (2016)). (Left) Combination of interferograms, capturing the phase changes of a location within dates-pre and dates-post.

LSTM are explained in a section 2.2.3.1 of this chapter. (Medel and Savakis (2016)) uses the convolutional LSTMs in an autoencoder that has one encoder trained on a time sequence for example $(T_2 - T_1 - T_0)$ and has two decoders to predict response at present frame $(T_0 - T_1 - T_2)$ and future time frame $(T_1 - T_2 - T_3)$. The adversarial learning in the GANs also helps to classify the normal and abnormal patterns, where these models are trained solely on either normal examples and tested on abnormal/mixed examples or vice versa (Zenati et al. (2018)).

2.3.1 Existing methods of Deep learning using InSAR data

As an emerging field, machine learning and deep learning have been developed over remotely sensed data and are designed, deployed and tested as a successful

approach in some recently published groundbreaking works (Anantrasirichai et al. (2018) and DeVries et al. (2018)). (DeVries et al. (2018)) Trains a neural network on 131,000 mainshock-aftershock pairs and predicts the aftershock location with an accuracy of 0.849 without any prior assumptions about the orientation of the slip. Whereas (Anantrasirichai et al. (2018)) uses computer vision-based techniques like canny edge detection to pre-process the volcanic eruption captured in a wrapped InSAR image. These edge contours in a grey-scaled image are then passed to AlexNet (Krizhevsky et al. (2012)) which classifies volcanic events. These two papers have invoked the conventional convolutional neural networks for event classification (Anantrasirichai et al. (2018)) and neural networks for location prediction (DeVries et al. (2018)), on comparison with traditional approaches the machine learning based technique shows a drastic improvement in accuracy.

The improvements achieved by these techniques have motivated and opened the gates for machine learning in the field of geology. The convolutional neural networks extract and learn context/features from the training set. The context carries important information about the objects and their relationships with each other in an image. The initial layers of a CNN capture low-level features (lines, blobs, edges, gradient orientation etc) whereas the final layers of a deep CNN capture the high-level features which are the position, colour and type of object. (Anantrasirichai et al. (2018)) Uses transfer learning and so applies the edges/contours information learned by AlexNet (Krizhevsky et al. (2012)) pre-trained on the ImageNet dataset (Deng et al. (2009)) to classify the two-dimensional map of edges of the wrapped InSAR images of volcanic eruptions.

Chapter 3

ALADDIn: Autoencoder-LSTM based Anomaly Detector of Deformation in InSAR

In this chapter, I present a novel method to address the challenging problem of automatic detection of transient deformation of the Earth's crust in time-series of differential satellite radar (InSAR) images. The work presented in this chapter was published in IEEE Transactions of Geo-science and Remote-sensing in May 2022 (Shakeel et al. (2022)). The text and figures from this publication are reproduced under a creative commons license and changes have been made accordingly for the work to better fit the thesis.

When I initiated this research, I felt a need to develop an unsupervised and event-agnostic method for the analysis of InSAR. As, all of the literature reviews at that time (Anantrasirichai et al. (2018); DeVries et al. (2018)), focused on targeted supervised tasks that require either a labelled synthetic data set to meet the requirement or marked wrapped interferograms. I, therefore, present an unsupervised and event agnostic method which uses unwrapped interferograms to detect anomalous events.

3.1 Introduction

Two new radar satellites, the Sentinel-1 constellation, are revolutionizing the way deformation of the Earth's surface is measured, generating high-spatial-resolution, near-global imagery of on-shore crustal deformation on a daily-to-weekly basis (Elliott et al. (2016)). This new data set of Sentinel-1 InSAR (Interferometric Synthetic Aperture Radar) images affords a major opportunity to investigate the prevalence of transient deformation phenomena that may have remained undetected in previous data sets (Floyd et al. (2016)). InSAR data sets have a unique 3D structure, where individual images (interferograms) are in fact the difference in phase between two individual radar images taken of the same area but at different times. This 3D interferogram (referred to as IFG) dataset is very different to the real-world video time-series that is a common dataset for deep-learning analysis, including anomaly detection, where an individual image instead captures information at a particular instant in time.

Detecting and measuring transient episodes of crustal deformation is important for a wide range of solid earth and natural hazard applications, e.g. for improving understanding of seismic and volcanological hazards and for monitoring anthropogenic deformation. It is important to characterize when and where such events have occurred in order to illuminate the basic physics of these deformation processes and to accurately estimate the hazards they pose to human populations. However, such a large data set of satellite images (10TB/day, 1000-2000 images/day) (Lazecký et al. (2020)) precludes systematic manual analysis, and the large magnitude of atmospheric and other nuisance signals relative to deformation signals of interest makes this task difficult. Therefore, this important objective requires the development of new automatic-detection tools based on cutting-edge machine-learning methods.

Machine learning has been successfully applied to a wide variety of remotely-sensed satellite data sets for scene classification, object detection and mapping purposes

(Youssef et al. (2020); Ma et al. (2019)). However, the application of machine learning and deep learning algorithms for the analysis of InSAR data is still in its infancy. The majority of the early attempts to apply machine-learning to the detection and extraction of deformation signals in InSAR data sets have involved relatively inflexible, off-the-shelf and supervised solutions, for example, AlexNet (Krizhevsky et al. (2012)) was used for supervised classification of volcanic signals in 2D images (Anantrasirichai et al. (2018)), a VGG (Simonyan and Zisserman (2015)) network was employed to detect volcanic unrest in 1D time-series (Gaddes et al. (2019)), a supervised FCN (Long et al. (2015)) was designed based on UNet (Ronneberger et al. (2015)) to separate volcanic signals from time-consecutive InSAR-derived 2D time-series (Sun et al. (2020)) and a supervised autoencoder (Rouet-Leduc et al. (2021)) was trained to reconstruct accumulated ground deformation.

These existing approaches all have one or more of several major limitations: I) they are limited to analysis in space (2D) or time (1D) only, or else use higher-level InSAR-derived products that involve filtering or modelling constraints that make a priori assumptions about the signal; II) they require resource-intensive pixel-wise labelling on a limited dataset of real-world examples; III) they are restricted to focus on a single type of deformation only (e.g. volcanoes); IV) they preclude the important ability to detect deformation signals with previously unobserved spatial or temporal structure. To overcome all these issues I take full advantage of the unique, differential and multi-linked 3D nature of InSAR datasets. I have developed a new, unsupervised, event-agnostic, and state-of-the-art deep-learning based approach for the automatic detection of transient deformation.

In this novel approach, I adopt an anomaly detection framework, based on convolutional neural networks (CNNs) and neural networks (NNs). Under this framework, anomalies correspond to any transient phenomena that deviate from the 'normal' spatio-temporal patterns in the dataset. Such 'normal' phenomena arise from a combination of atmospheric signals, satellite orbital errors and other unwanted 'nuisance' signals (Simons and Rosen (2007); Emardson et al. (2003)). I exploit

the fact that the unknown 2D fields of nuisance non-deformation signals associated with individual SAR acquisition dates (these are termed 'epoch images' or *EP* here, following the domain nomenclature, and are not to be confused with the typical machine-learning definition of epoch) map into signals in interferograms with a fundamentally different temporal pattern to 'anomalous' deformation signals (Figure 3.1). By training a deep-learning algorithm to map common noise signals in inteferograms into the unknown EP time-series (Figure 3.1 a) I am then able to detect rare deformation events that map into the EP time-series differently (Figure 3.1 b). The approach, therefore, allows not only to estimate a background time-series of the unknown non-deformation signals but also to identify deformation and effectively and accurately separate it from this background. In comparison to the past work, the main contributions of this chapter are:

- Here I have established a novel network architecture using CNNs and NNs that transforms InSAR data into an *EP* image sequence. It models the spatial and temporal patterns and the connection between interferograms and their corresponding *EP* images.
- The model is unsupervised and is event-agnostic anomaly detection, where anomalies correspond to any transient phenomena that deviate from the 'normal' spatio-temporal pattern.
- I have successfully trained the framework on a set of interferogram sequences with multiple outputs. First, is the automatic prediction of *EP* image responses (that are originally unknown). Second, the reconstruction of interferograms using this predicted emph EP responses, and last but not least the detection of anomalies within the sequence.
- I have developed a novel detection-and-extraction approach, that flags anomalies, estimates their spatial structure and separates them from noise.
- Finally, in this chapter I present an accurate analysis of a test set with and without synthetic anomaly with spatial extent and amplitude similar to the

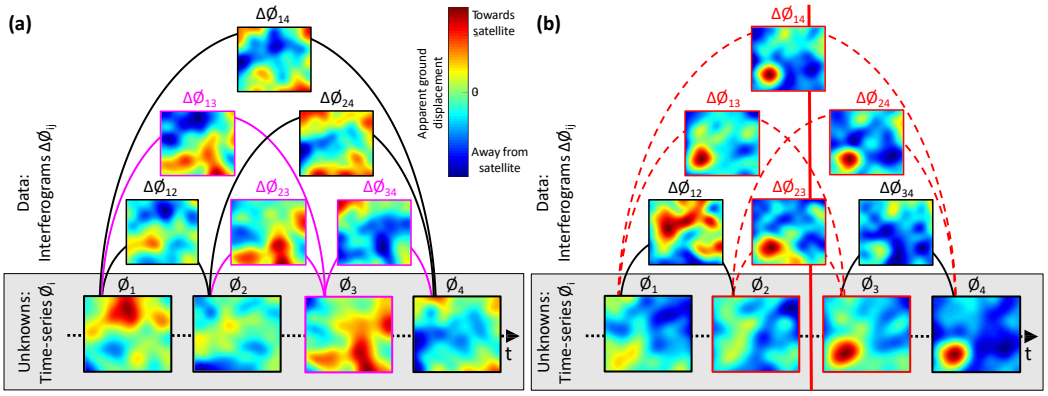

Figure 3.1: Cartoon illustrating the unique structure of the InSAR dataset. In particular the relationship between the measured inteferograms (IFGs, phase-change $\Delta\emptyset_{ij}$ shown in top part) and the unknown epoch images (EP, $\emptyset_i$ shown in grey box in (a) and (b)). Pink outlined images in (a) show how nuisance signals associated with an individual epoch (e.g. the red signal in the bottom-right of the $\emptyset_3$ image) contribute to all linking interferograms, some in a positive sense (e.g. a similar red signal in $\Delta\emptyset_{13}$, $\Delta\emptyset_{23}$) and some in a negative sense (e.g. a blue signal of similar shape in $\Delta\emptyset_{34}$). Vertical red line in (b) represents a transient episode of deformation taking place between $\emptyset_2$ and $\emptyset_3$. Red outlined images in (b) show how this deformation contributes in a positive sense to any interferogram that spans this event (e.g. a similar circular structure that is always red in the bottom left of $\Delta\emptyset_{13}$, $\Delta\emptyset_{23}$, $\Delta\emptyset_{14}$ and $\Delta\emptyset_{24}$).

background noise, achieving a true positive rate of 81.25% and an overall accuracy of 91.25%, and also successfully demonstrate method's ability to detect a real earthquake of Magnitude 5.7 that occurred in south-east Turkey (a region outside the training set).

3.1.1 Chapter overview

In this chapter, I first provide an introduction to InSAR data in Section 3.2 with literature review in the following Section 3.3 and how it can be used with deep learning. Then I present the methodology explaining all three network architectures (Compact, Deep and Bi-Deep) and detecting mechanism in Section 3.4. As ALADDIn is designed to cater for unique InSAR-like data structures, it is erroneous to compare it with existing off-the-shelf deep learning based anomaly detectors that are trained on frame-frame video data. Finally, as proof of efficiency and accuracy,

I put forward a detailed experimental analysis of test results for real, normal and synthetic test cases when passed through all three models (Compact, Deep and Bi-Deep) in Section 3.6. Discussion about how these results can be improved is given in Section 3.7 and how should the output be judged. In conclusion, I discuss the key contributions of this paper in Section 3.8.

3.2 InSAR data

A Synthetic Aperture Radar (SAR) satellite image is a 2D array of complex numbers encoding amplitude and phase information of microwave radar waves that are emitted by the satellite, backscattered from the Earth's surface, and recorded again by the satellite's antenna. In order to measure the movement of the Earth's surface, two SAR images of the same location but captured at different times can be used to construct an unwrapped Interferometric SAR (InSAR) image. This image is called an unwrapped interferogram (hereafter referred to simply as an interferogram or InSAR image) and represents a map of how the ground has moved towards or away from the satellite (i.e. a 1D displacement in the satellite's 'line-of-sight') in the time interval between the two SAR measurements. The largest nuisance signals in interferograms arise from uncertainties in satellite orbits and from changes in atmospheric conditions and are commonly considered as noise when trying to measure ground motion. In this study, I use unwrapped Sentinel-1 interferograms obtained from the global LiCSAR processing system developed by the UK's Centre for the Observation and Modelling of Earthquakes, Volcanoes and Tectonics (COMET) (Lazecký et al. (2020)). These images typically cover a region of the Earth's surface $\sim$250 km $\times$ 250 km, with pixels of size 80 $\times$ 80 m. There is a major difference between regular video data that is commonly analysed using deep-learning methods and InSAR data. In video data, an individual image contains information on the position of objects at a single acquisition time, but an individual interferogram instead contains information on the difference in position

between two acquisition times. The unique structure of the dataset is illustrated in Figure 3.1a; in this simple example, six interferograms (curved lines) capture the differences between four epoch images (circles) that are each associated with an individual SAR satellite image. These epoch images are always unknown; due to the way in which unwrapped interferograms are constructed, these 2D fields cannot be directly calculated from the SAR images themselves. Nuisance signals associated with an epoch (e.g. Ep_3 in Figure 3.1a) can be mapped into associated interferograms (pink outlined images and lines) via a simple spatio-temporal relationship. But permanent ground displacement that takes place between two Epochs (e.g. between Ep_2 and Ep_3 in Figure 3.1b) maps into a different set of interferograms (red outlined images and lines) according to a different relationship. In the following section, I describe how it is possible to use CNNs to exploit this key difference and therefore detect deformation.

3.3 Literature review

InSAR data have inherent inter-dependent spatial and temporal patterns associated with background nuisance signals due to the unique data structure (Figure 3.1a). This prominent feature of the data can be learned so that anomalous signals corresponding to deformation (Figure 3.1b) are identified. Long Short Term Memory (LSTM) cells (Gers et al. (1999)) are often used in similar cases to learn from time-dependent data. LSTMs are a type of Recurrent Neural Network (RNN) that directly model the temporal dynamics in the data stream for more accurate prediction. RNNs are commonly used for speech recognition, language modelling, translation and image captioning (Miao et al. (2015); Cho et al. (2014); Mao et al. (2014)), but they suffer from a vanishing-gradients problem which limits how much memory they can hold. Information is propagated through each time-point in a RNN and so gradients are computed for each hidden layer (all across time) using backpropagation starting from the final layer to the initial layer. Depending on

the length of time and the number of layers, the small derivatives when multiplied together (causing a ripple effect) decreases (vanishes) the gradients exponentially (Hochreiter (1998)). In contrast, LSTMs are capable of retaining information for longer intervals so they have been successfully used for capturing changes that are prolonged in time, e.g. CCTV surveillance (Shah et al. (2018)). LSTM models tend to learn from the temporal dynamics of the sequence in 1D, whereas multiple filters in a convolutional layer span and perform convolutions on 2D or 3D data, preserving pixel-based spatial information. For problems like the one tackled in this study, were, to detect the time-stamp and location of anomaly it is important to learn both the spatial structures and temporal patterns of input data, the LSTM cells are applied with convolutions as a mathematical operator. The internal $1D$ matrix multiplications in the LSTM layer are converted in convolution operations. These are termed convolutional LSTMs (Donahue et al. (2015)) and are able to maintain the dimensions of the input data for images or videos. Where there is no ground truth available and it is both expensive and time-consuming to mark, label or caption abundant video or image data, unsupervised or semi-supervised deep learning techniques involving convolutional LSTMs (Xingjian et al. (2015)) are used to understand changes and track object movements (Ning et al. (2017)).

In contrast to the existing approaches to the machine-learning analysis of InSAR data (Anantrasirichai et al. (2018); Sun et al. (2020); Gaddes et al. (2019); Rouet-Leduc et al. (2021)), I develop this method starting with three building blocks: the encoder that models spatio-temporal patterns in the interferogram sequence; the fully connected (FC) layers that transition these encoded features to corresponding epoch responses; and the decoder that then up-samples these epoch responses. A FC layer is a 1D layer containing feed-forward neurons. Each neuron in them is connected to every single feature encoding of preceding and succeeding layers, representing every pixel in time and space. This strictly ensures that the model learns: (I) the spatio-temporal patterns within the interferogram set (while encoding); (II) the relationship within epoch responses and their difference (while transitioning

using the fully connected layers and also constrained by the loss function, that is defined in eq 3.1); and (III) the spatio-temporal patterns within the sequence of epochs (while decoding). In order to encode the distribution of an input sequence, so that the LSTM layers can learn spatio-temporal patterns from it, the input images are fed to convolution blocks each block includes a time-distributed convolution and a layer normalization. The benefit of the time-distributed layer is that it ensures the same convolution is applied to each temporal instance in the input sequence. For example, as I have $X\ IFG$ in the sequence, then in the time-distributed layer there would be X number of convolutional filters applied individually on each interferogram, giving X number of features in time that each contains 2D spatial information. The weights of this layer are distributed among the X filters, this helps to connect the features learned for each temporal instance within an input sequence and makes it computationally manageable for backpropagation.

The approach of (Rouet-Leduc et al. (2021)) is in some respects similar to ours, although the time-series (in this case epochs that are generated by the model) they used for training is computed using SBAS inversion and topography is added as an extra channel mid-way in the model. Although, the same length of epoch time-series is used, (Rouet-Leduc et al. (2021)) select a much lower (48 x 48) spatial resolution. Unlike this approach, the autoencoder is supervised and trained on synthetic data. Although (Rouet-Leduc et al. (2021)) predicts the accumulated ground deformation of an InSAR time-series, the model does not know if that deformation is anomalous or not. In addition, the model prediction in (Rouet-Leduc et al. (2021)) provides only the spatial structure of cumulative deformation, without allowing the exact timing or duration of the event to be retrieved.

3.4 Methodology

The solution to an anomaly detection problem can be developed by first understanding the data (InSAR data in this case) and defining the 'anomalous' class

(crustal deformation) in it. The low absolute numbers of interferograms containing transient deformation, and even lower numbers where deformation has been labelled make this problem better suited to unsupervised learning. The lack of data with real-world anomalies in it and then the lack of labelled anomalies in the available data hinders accurate development of solutions, but this encourages to advance of the field of unsupervised learning (Pang et al. (2020)). I have thus developed a methodology by training and testing different network architectures, the backbone of which is an autoencoder.

The development of backbone of the methodology is a process of implementing and experimenting with multiple deep learning models that are capable to capture the attributes of InSAR and the problem at hand, which is anomaly detection. Some of these architectures include Generative adversarial networks GAN (Goodfellow et al. (2014)) , Autoencoders and Variational Autoencoders (VAE) (Doersch (2016)). Initially I planned to automate the whole process of SAR interferometery explained in Section 1.2.1, by using the 2 SAR images as input and the relevant interferogram as an output. This did not work because it is wrong to expect from an autoencoder to replicate all of the processes involved in generating a filters unwrapped interferogram. Then I navigated to using the readily available unwrapped interferograms from the LiCS Sentinel-1 InSAR portal. I decided to use unwrapped interferograms because these image contain the actual change in phase (deformation) value as compared to the wrapped data. Wrapped data masks and overpowers the actual anomalous deformation, making it harder to detect. Also, the wrapped interferograms are in range of $[-\pi, \pi)$, this restricts the learning space, leaving greater room for prediction error and model over-fitting. So, I organized the time-series of unwrapped interferograms into small data sequences or data batches to process. I started implementing and experimenting with VAEs first, so that the model learns features that are regularized over the encoded latent space. Despite of being advantageous as compared to the autoencoders, VAEs turned out to be difficult to train. This is majorly because of encoded space being different in dimension

to that of decoded space. The input and output of the autoencoder is supposed to be same. In this case, the output (epoch *EP* time-series is indirectly related to the input but with different dimensions and different spatio-temporal patterns. Though, I do convert the *EP* back to interferograms to compute reconstruction loss and to meet the definition of the autoencoder. But still the other cost function is dependant on the latent representation that is being computed entirely from the encoded interferograms. For these reasons the model did not converge and keep on outputting random irrelevant spatial patterns. I then tried GANs, the generator of which is also an autoencoder. It comprises of two components, a generator G model and a discriminator D model, both of these are trained in a simultaneously, G predicts and then D decides how close it is to the input by classifying both the input and the output as real or fake. The cost function of D is back-propagated through G and D. During my experiments with GANs, I observed that the discriminator model D, instead of helping to bring reconstructed data closer to the input, added more noise to them (making them more fake). Again, this is because of the fact the G is predicting *EP* time-series where as the D is focusing on the reconstructed interferograms being real or fake. And because the cost function of D is also back-propagated through G, it hindered its learning. This led me to develop the network architecture that is similar to a generator G and the backbone of which is a autoencoder.

3.4.1 Autoencoders

Autoencoders are a type of artificial neural network that is designed to understand the underlying distribution of a dataset by learning to reconstruct the data from a transformed version of them. The transformation of the input is referred to as an encoding and usually results in a compressed representation of the data. The reconstruction of the data is referred to as decoding and usually involves up-sampling of the encoded data. This has proven to be a very powerful approach with applications in image denoising, segmentation, 2D reconstruction and image

generation purposes (Zhu et al. (2016); Vincent et al. (2010); Sameen and Pradhan (2017)). Autoencoder-based anomaly detection in deep learning often refers to training a model to learn the normality underlying a given labelled dataset. The autoencoder learns to reconstruct the input, which can be an image or a video sequence (Xu et al. (2015)) or multivariate sequence data (Lu et al. (2017) ; Marchi et al. (2015)). So when fed with new input data the anomalies are identified with high error. The autoencoder based anomaly detection can be designed using different combinations of deep learning layers for example convolutional (for spatial data) (Zhang et al. (2019)), LSTM (for temporal data) (Malhotra et al. (2016)) or combined Convolutional-LSTM (for spatio-temporal data) (Luo et al. (2017)).

3.4.2 Network architectures

3.4.2.1 Compact model

We start by training a Compact model (Figure 3.2a), with just four of these blocks, two with time-distributed convolution layers with strides and two with conv-LSTM. For an input of $X\ IFG$ that is made from $Y\ EP$, the encoded features, which are ordered in time and are passed to LSTM blocks that learn and retain the spatio-temporal pattern. To reconstruct the EP responses from these features, they are first transformed to Y representations in temporal order by the FC layers, where Y is the number of EP responses. At this stage the features are of size $X \times 16 \times 16 \times F$ (where F is the number of filters) and the number of neurons in the FC layers are used in accordance with the size of features required after transformation, i.e. $Y \times 16 \times 16 \times F$. Likewise, in the decoder, transpose convolution layers are used to upsample the reshaped NN features. The NN with 2304 neurons are used for the transformation of $X\ IFG$ to $Y\ EP$, where X is 26, Y is 9, so, the number of neurons are $9 \times 16 \times 16 = 2304$.

3.4.2.2 Deep model

I further developed these building blocks and increase the overall depth of the model, creating what is called the Deep model (Figure 3.2b). This is designed with a balance of time-distributed convolutions and the conv-LSTM layers in the encoder as well as the decoder. Each of the convolution blocks in the Deep model includes a max-pooling layer that removes invariances like shift and scale in the feature representation and ensures computationally manageable trainable parameters across the model by down-scaling and extracting the most important features. Apart from depth, other major changes are the addition of upsampling layers instead of the stride, smaller filter size (to precisely capture local features) in the convolutions and *tanh* (Kalman and Kwasny (1992)) as an activation layer after each layer except the neural network. An activation function is applied on the top of layers to introduce non-linearity and the output of *tanh* is zero-centred and ranges from -1 to 1, hence strongly mapping both negative and positive inputs.

3.4.2.3 Bi-Deep model

Finally, for the Bi-Deep model (Figure 3.2c) I added separate skip connections for the encoder and decoder and also included a bidirectional-conv-LSTM layer. These skip connections ensure the forwarding of any residual feature representation in the previous layer and also helps in merging features learned by a time-distributed convolution and a conv-LSTM layer and other combinations like a FC layer and a time-distributed convolution layer etc. Unlike U-Net (Ronneberger et al. (2015)), long skip connections cannot be used because of different feature sizes, as the model reconstructs $Y\ EP$ from the encoded $X\ IFG$. So, to ensure the flow of information between layers of the autoencoder, I perform feature concatenation via short skip connections separately for the encoder and decoder. The bidirectional-conv-LSTM layer retains information by spanning the features propagating both forward and backwards in time, to prevent bias in the predicted *EP* sequence associated with

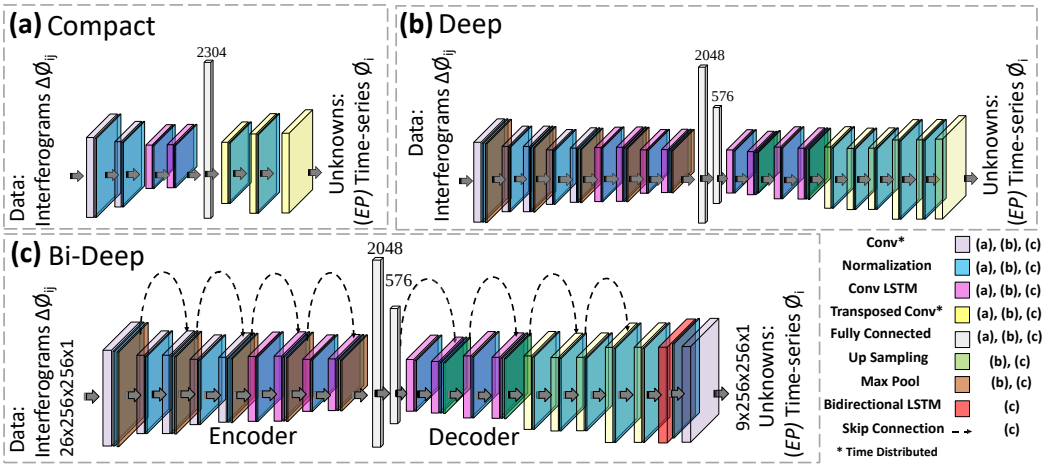

Figure 3.2: The network architecture of all models. (a) Compact, (b), Deep and (c) Bi-Deep models are shown. Compact includes time-distributed 2D convolutional layers with a stride in place of max-pooling, followed by layer normalization, 2D convolutional LSTM, a fully connected layer and transpose convolutions to decode the encoded features. Whereas the Deep includes a greater number of conv-LSTM layers in the encoder as well as the decoder, with an extra fully connected layer, max-pooling layers are replaced by stride in convolutions and upsampling layers instead of transpose convolution. The Bi-Deep architecture is similar to the Deep one but with a major difference in the input of each layer, here skip connections are placed to merger features and a bi-directional LSTM layer is added at the end.

their order in time.

3.4.3 Experimental details

In general all these models (Compact, Deep and Bi-Deep) attempt to learn the relationship between normal spatio-temporal patterns of background noise in a set of related interferograms and the unknown 2D fields of that same noise in their constituent epochs (e.g. Figure 3.1a). The model transforms $X\ IFG$ in the encoding feature space to $Y\ EP$ in the decoding feature space. The transformation is constrained through the loss function

$$Loss = MSE(IFG2TS(Output), Input) \tag{3.1}$$

, where $IFG2TS$ is a custom layer that converts the sequence of Y estimated epochs into X interferograms by simple subtraction in each case of the 1st constituent

epoch from the 2nd, as per Figure 3.1a. Therefore, the mean squared error (MSE) is computed between the input set of interferograms and the reconstructed ones.

The data values in the interferograms are of varying ranges and positive and negative values hold equal importance, so activation layers like $ReLU$ (ranges from 0 to ∞) and $Sigmoid$ (ranges from 0 to 1) cannot be used. Therefore a $tanh$ activation function is applied for all convolutions except the output time distributed convolution layer so that the prediction of EP values are not bound by any range.

I train and test the method using Sentinel-1 InSAR data from Turkey (Figure 3.3), obtained from the COMET LiCSAR processing system (Lazecký et al. (2020)). Turkey has been a focus area for the initial development of this processing system, so has the largest dataset available for training and testing the method (Li et al. (2016)).

The data frame on the northern coast of Turkey (LiCSAR Frame name: 014A_04939_131313, spatial extent $\sim$ 250 km $\times$ 250 km) is used for training. In order to manage the complexity of the model and the memory required to train a large number of parameters, the frame is divided into cubes of size $256 \times 256 \times 26$ pixels (a spatial extent of approximately 20.5 km $\times$ 20.5 km) with a fifty percent spatial overlap (in both E-W and N-S directions) and instead of passing the whole time-series in every training iteration, a set of 26 interferograms that cover 9 EP is passed. The temporal sliding window is 9 EP in length and moves with a stride of 4 ensuring a temporal overlap of $> 50\%$ between successive input sequences. The 26 interferograms link each EP with all successive and preceding EP within the sequence, up to a maximum distance of 4 forwards and backwards in time. For example, the central EP is linked by 8 interferograms to all other EP, but all other EP in the sequence are linked with less than 8, to a minimum of 4 interferograms for the EP at the start of the sequence and the EP at the end of the sequence. The order with which the 26 IFG are passed to the model is sequential, for example, the first sequence passed is in order:

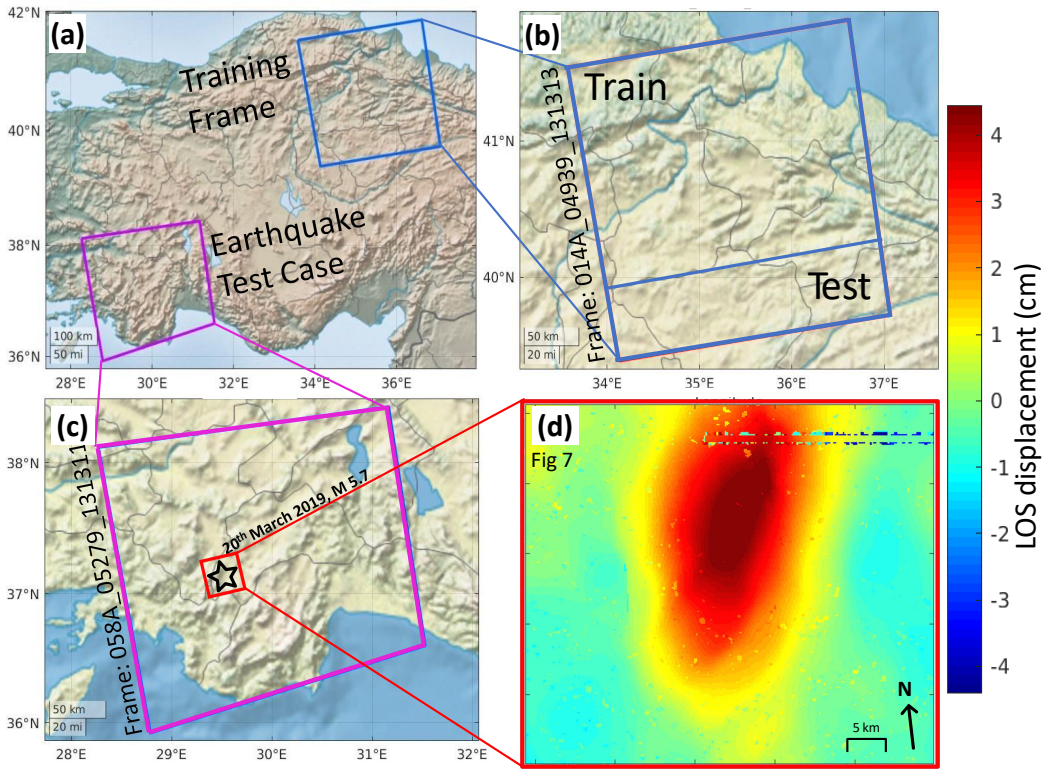

Figure 3.3: Data set details. (a) The map shows the geolocation of the train and test data. (b) The training frame is located in the north-east of Turkey, whereas the real earthquake test case, shown in (c) and (d) is located in the south-western part of Turkey. (c) Shows the zoomed-in image of the region where the earthquake occurred on the 20th of March 2019. The spatial structure of the earthquake which is estimated by the model is shown here in (d) (shown in 3.7).

- 26 Interferograms: IFG_{12}, IFG_{13}, IFG_{14},IFG_{15}, IFG_{23}, IFG_{24}, IFG_{25}, IFG_{26}, IFG_{34}, IFG_{35}, IFG_{36}, IFG_{37}, IFG_{45}, IFG_{46}, IFG_{47}, IFG_{48}, IFG_{56}, IFG_{57}, IFG_{58}, IFG_{59}, IFG_{67}, IFG_{68}, IFG_{69}, IFG_{78}, IFG_{79}, IFG_{89}.

- Covering initial 9 Epochs: EP_1, EP_2, EP_3, EP_4, EP_5, EP_6, EP_7, EP_8, EP_9.

The dataset details are given in Table 3.1. The model is trained using Keras (Chollet et al. (2015)) with the TensorFlow (*Abadiet al.* (2015)) backend. Due to the large size of the images in memory the batch size was set to 1 and because of computational constraints no experiments are done with batch size greater than 1. Adam optimizer (Kingma and Ba (2014)) was used with a learning rate of

0.00001. A lower learning rate gives the model a chance to learn features through steady changes in a loss instead of rapid fluctuations. During my experiments, in order to find the optimal value for this hyper-parameter, I initially start with a comparatively higher learning rate and gradually decrease it by monitoring the training and validation loss. This is discussed in more detail in section 2.2.1.1.

Table 3.1: Dataset Details

Data	**No. of patch locations**	**No. of sequences for each patch location**	**Total sequences**
Train	365	25	9125
Validation	62	25	1550
Synth Test	1	10	10
Real EQ	1	7	7

3.5 ALADDIn: Autoencoder-LSTM based Anomaly Detector of Deformation in InSAR

The deep learning models are trained on background atmospheric noise so that in case of an anomaly there must be a misfit. The training data has been manually reviewed for any anomalous events, thus confirming that the training patch sequences contain only the 'normal' background atmospheric noise. Once the test data are passed through the models, the residuals are computed between the reconstructed interferograms and the original data. Because any anomaly will appear in multiple interferograms (and therefore also multiple residuals), I first reduce the residual dataset down to a mutually exclusive set of N_{EI} (Figure 3.4) "epoch intervals". An epoch interval is a different image that spans two successive epochs, so therefore N_{EI} is equal to one less than the number of epochs. In order to estimate this set of N_{EI} residual epoch intervals, I perform a linear least squares inversion on

a pixel by pixel basis of N_{IFG} residuals as follows (based on the SBAS approach from (Berardino et al. (2002))):

$$d_{IFG} = G.m_{EI}, \tag{3.2}$$

where d_{IFG} is a $N_{IFG} \times 1$ array containing pixel values of all N_{IFG} residual interferograms, m_{EI} is the $N_{EI} \times 1$ vector of epoch intervals that I wish to solve for, and G is the $N_{IFG} \times N_{EI}$ sized design matrix for this system of equations, containing 1s and 0s only. Eq 3.3 shows an example of matrix G for a set of six residual interferograms (IFG_{12}, IFG_{13}, IFG_{14}, IFG_{23}, IFG_{24}, IFG_{34}) corresponding to the simplified cartoon structure shown in Figure 3.1, constructed from four epochs (EP_1, EP_2, EP_3, EP_4), which will output three epoch intervals (EI_{12}, EI_{23}, EI_{34}) based on the residuals. These intervals are essentially equivalent to the shortest spanning set of residual interferograms (e.g. EI_{12} is equivalent to $IFG_{12} = EP_2$ - EP_1) but are instead estimated from the full set of residual interferograms so are more robust to noise in any one residual image.

$$\begin{pmatrix} IFG_{12} \\ IFG_{13} \\ IFG_{14} \\ IFG_{23} \\ IFG_{24} \\ IFG_{34} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} EI_{12} \\ EI_{23} \\ EI_{34} \end{pmatrix} \tag{3.3}$$

Due to the overlap between successive sequences, most epoch intervals occur twice in the time series and some appear three times. Epoch intervals for every sequence are computed by solving the equation 3.2 for m_{ET} and then they are concatenated together to make one overlapping time-series of residual epoch intervals. These intervals are then automatically analysed for the presence of spatial anomalies. This is achieved by two complementary analysis methods: semivariogram analysis (Wackernagel (2013)) and density-based clustering (Kriegel et al. (2011)).

3.5.1 Detection framework

3.5.1.1 Semivariogram

An empirical semivariogram is an estimate of how pairs of samples within a dataset differ as a function of distance. The semivariance $\gamma(h)$ for distance h is:

$$\gamma(h) = \frac{\sum_{N(h)} [Z_i - Z_{i+h}]^2}{2(|N(h)|)}, \tag{3.4}$$

where Z_i is the value at pixel location i and $N(h)$ is the total number of pairs that lie at distance h. The spatial variability measured by the semivariogram can account for deformation that affects only certain spatial frequencies, e.g. capturing deformation that is spread over small regions, and separating such anomalies from larger areas that are 'normal'. These small but significant changes are less likely to be detected by simply computing bulk differences between actual and reconstructed images (e.g. by a Mean Squared Error).

The epoch intervals containing no anomaly are all expected to have similar spatial structures and therefore will also have similar empirical semivariograms, whilst epoch intervals containing anomalies will have semivariograms that differ substantially from this normal structure. This can be seen in Figure 3.5 b, where the semivariograms for residual EI that contain synthetic anomalies (red lines) are significantly different from semivariograms corresponding to 'normal' epoch intervals (blue lines). A semivariogram is calculated for each residual epoch interval, and the root-mean-squared-error is computed between each semivariogram and all others in the entire set of residual epoch intervals across all sequences. The threshold used to separate the anomalous values varies per study and is not fixed a-priori, as the spatial structure of the background noise will vary depending on the dataset, resulting in varying semivariance values. But in each case, the key assumption is that deformation events are rare. The threshold is computed using the inter-quantile range of the average error values.

3.5.1.2 Density-Based Spatial Clustering (DBSCAN)

The second detection operation I perform is density-based clustering (DBSCAN) (Kriegel et al. (2011)) of the residual epoch intervals. Under normal circumstances, the residual epoch intervals are expected to be similar with values near to zero (as they are accurately reconstructed by the model, e.g. see Figure 3.4 bottom row), but in case of an anomaly or multiple anomalies within a sequence, there must be an interval containing the spatial structure of that anomaly. So, the goal is to separate all normal intervals in one cluster and anomalies in other clusters without any prior knowledge of the instances of anomalies in a sequence. This algorithm performs clustering based on the density of data points and has the advantage for this unsupervised problem of not requiring a-priori specification of the number of clusters. As I have an overlapping sequence of residual epoch intervals, so each time interval occurs at least twice. I use the prior knowledge of this overlap in epoch intervals to set the minimum points in a cluster to be two. Due to the varying nature of data points, I compute the search radius (epsilon) for the algorithm separately for each sequence, by sorting and plotting the distance to the nearest n points for each point. Epoch intervals are classed as anomalous if they do not fall within the predominant cluster (Figure 3.5 c).

Finally, the classified anomalies from the semivariogram and clustering analysis of epoch interval time series are combined using an AND operation in order to reduce the number of false positives. DBSCAN clustering is prone to false positives due to its sensitivity to the distance metric, and for synthetic test set, I reduce the number of false positives from 12 when using DBSCAN to just 4 when combining it with the semivariogram analysis. The Table 3.3 shows that combining DBSCAN and semivariogram analysis gives high overall accuracy, as the false positives from DBSCAN are mitigated by the incorporation of the semivariogram analysis.

3.6 Results and analysis

I evaluate all models with three testing scenarios, i.e. a normal test sequence with no deformation, the same normal test sequences but with synthetic anomalies added, and test data that contains a real earthquake (see Figure 3.3 c). The purpose of the normal test case is to show the comparison of reconstruction error between an anomaly that occurred within a normal sequence (as demonstrated in the Figure 3.6). All of these testing sequences are from a different location than the training data. Due to the fact that no ground truth is available for the epoch responses, the extent of the model's accuracy can be judged by the accuracy of interferogram reconstruction. The reconstruction error used for analysis is root-mean-squared-error $RMSE$ between input IFG i.e. ground truth and the reconstructed IFG. An accurate model should result in a reconstruction error near to or equal to zero when a normal test sequence is passed, whereas the error should increase by a large fraction when a synthetic deformation is added in that same normal test sequence. The detecting mechanism relies on the output (EP responses) of all three models, the overall accuracy of detection depends on TP (true positives), TN (true negatives), FP (false positives) and FN (false negatives), where positive refer to anomalous class and negative refers to normal class. I independently analyse the test results of all models first, then plug in the detection mechanism on top of it and independently investigate the overall accuracy of detection by splitting up the semi-variogram and DBSCAN mechanism and also by merging them together in an AND and OR the combination.

3.6.1 Synthetic 2D Gaussian test case

Ground truth for real-world deformation signals in InSAR data is rarely available, so in order to assess the accuracy of framework, I first simulate a simple deformation anomaly (Figure 3.5 g) that has the structure of a 2D Gaussian in space and is effectively instantaneous in time with respect to the temporal-frequency of the data

(i.e. the deformation event takes place completely in the time-period between two epochs, which in this dataset is 6 days). The spatial structure of this signal is given by:

$$Z(x, y) = A. \exp\left(-(x^2 + y^2)/r\right) \tag{3.5}$$

where the exponential length-scale $r = 10.5$ km, the amplitude $A = 4.34$ cm and x and y are spatial coordinates relative to the location of Gaussian peak. This is an ideal test signal as the amplitude and spatial size of this structure are similar to that of noise in the data, as shown in Figure 3.4 and 3.5 g. To enable a robust assessment of the detection accuracy and minimise the impact of the natural variability of noise throughout the dataset on this assessment, I create a synthetic test case with the same anomaly added to the data at multiple instances in time. Therefore the synthetic anomaly is added at 8 different time instances in the real interferogram dataset for a patch location that is separated from the train and validation data. Each interval occurs twice due to the overlap of successive sequences, so in total there are 16 anomalous synthetic deformation structures in the test case. This patch features a total of 10 sequences spanning July 2017 to April 2018, and contains 260 interferograms and 80 epoch intervals, out of which 16 are anomalous and 64 are normal.

The reconstruction error is plotted in Figure 3.6 (bottom), showing the minimum error recorded by the Bi-Deep model, although an elevation in error can be seen when in the same normal sequence the synthetic anomaly is added (Figure 3.6 (top)). A comparison of mean error values for all scenarios and all models (Compact, Deep and Bi-Deep) can be seen in Table 3.2. In order to further analyse the difference in error, the percentage increase in RMSE is also computed between the 'real' anomalous test set and the cleaned set (when the estimated structure of anomaly is removed). The model predicts the time and location of these anomalies with an overall accuracy of 91.25% (presented in the Table 3.3) and a true positive rate of 81.25%. The full confusion matrix is shown in Figure 3.5 (f). To compute the confusion matrix and the accuracy score, each overlapping interval is treated

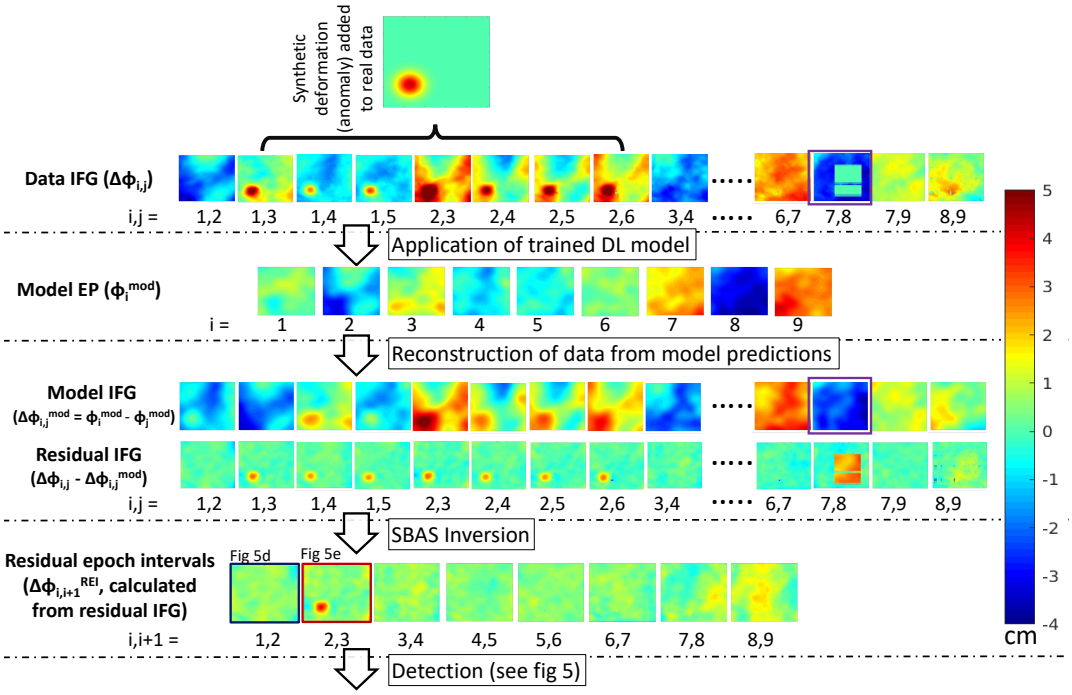

Figure 3.4: The results for the synthetic test case. Detailed discussion is given in section 3.6.1. The synthetic test signal - a 2D Gaussian with peak amplitude of 4.34 cm and exponential length-scale of 10.5 km; this signal was added to the interferogram time series. Following the synthetic input, are the *EP* predictions, using them the interferograms are reconstructed and residuals are computed. These residuals are used in the least square inversion to compute residual based epoch intervals.

as an individual anomaly. Figure 3.4 also shows the results from the 10th and final sequence, which is the worst constrained sequence in the data set as there is no overlapping sequence available for the last 4 epochs. Despite this, the images in the Figure 3.4 show how the anomaly is still accurately detected in the residual images. Figure 3.4 (purple box) also demonstrates how the method can accurately estimate the spatial structure of interferograms even when the ground truth images contain missing data (purple box in Figure 3.4 first and third row).

3.6.2 Case study: 2019 Earthquake in south-west Turkey

Finally, I also test the model's ability to detect a real Magnitude 5.7 earthquake that occurred in south west Turkey on 20th March 2019 (Elliott (2020)). Unlike

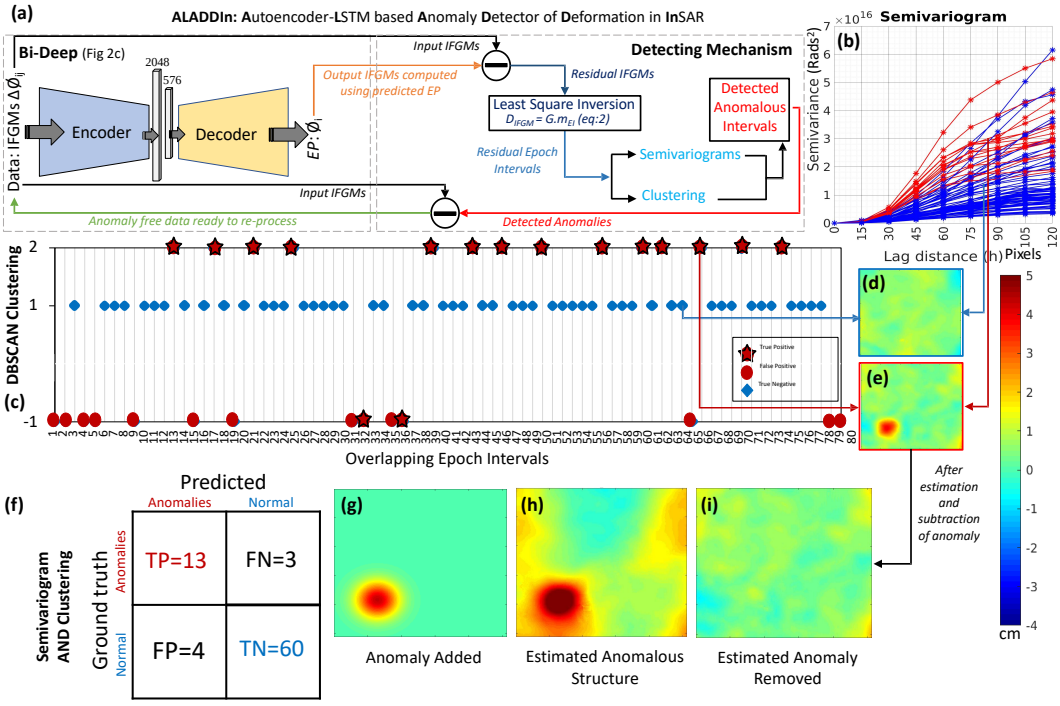

Figure 3.5: The ALADDIn pipeline. (a) shows the Bi-Deep model (sec 3.4.2.3) plugged in with the detecting mechanism that involves semivariogram analysis (sec 3.5.1.1) and DBSCAN clustering (sec 3.5.1.2), summing up the ALADDIn pipeline. (b) shows the semivariogram plot, red lines are the synthetic anomalies- all have high RMSE with respect to the majority of semivariograms corresponding to 'normal' epoch intervals (blues lines). (c) shows the results of clustering when it is applied to residual epoch intervals. The results for the synthetic test case (detailed discussion in section 3.6.1). (d) and (e) are one of the normal and anomalous residual epoch intervals (also shown in the Figure 3.4). (f) Shows the confusion matrix for the synthetic test results for the Bi-Deepmodel, where 13 out of a total of 16 are correctly detected. (g) The synthetic test signal - a 2D Gaussian with a peak amplitude of 4.34 cm and exponential length-scale of 10.5 km; this signal was added to 8 different time intervals in the interferogram time series, which makes a total of 16 anomalies due to the overlap between successive sequences. (h) shows the estimated spatial structure of synthetic anomaly for one of the intervals and (i) shows the undetected output of the same interval when the estimated structure is subtracted and the data is reprocessed.

many transient deformation signals of interest, both the time of this event and its location are known, which means I can verify whether the model can correctly identify the earthquake interval as anomalous. Sentinel-1 InSAR data for this test case includes 7 sequences starting from September 2018 to April 2019, and due to the overlap between successive sequences, both the 6th and 7th sequences include

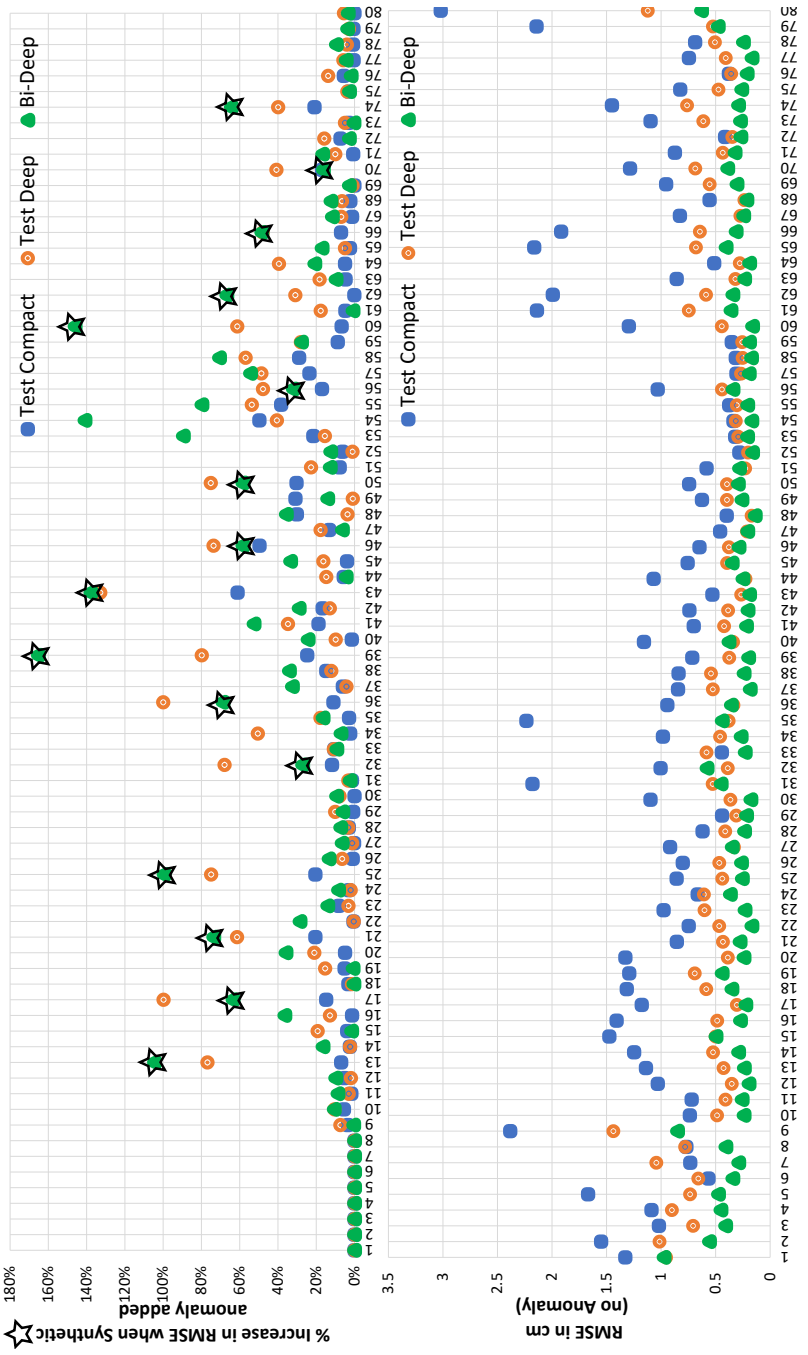

Figure 3.6: Comparison error plots of all models. (Bottom) The plot shows the comparison of RMSE i.e. the reconstruction error of all models. (TOP) The plot shows the percentage increase in the error when a synthetic anomaly (represented by a black star) is added in the same testing sequence. This plot indicates that the Bi-Deep has the lowest reconstruction error (bottom) and when an anomaly is added, it has the highest percentage increase in reconstruction error (top) that is required to detect them. The x-axis covers a total of 10 sequences each containing 8 epoch intervals, and the last 4 of every sequence overlaps with the first 4 of the next sequence.

the earthquake anomaly.

The 2 realisations of the earthquake interval are accurately detected as anomalous (shown in Figure 3.7 pink box), along with another anomaly which on further inspection is a feature of InSAR data known as an unwrapping error (Figure 3.7 pink box). These errors occur during the process of converting discrete cycles ('fringes') of $+/-\pi$ phase into continuous values and are particularly significant in areas of phase incoherence associated with steep topography, changes to surface scatterers between satellite image acquisitions or exceptionally high deformation rates. Phase unwrapping errors have magnitudes in multiples of 2π in individual interferograms (potentially several cm apparent displacement) and propagate through time series analysis to hinder the interpretation of tectonic or volcanic deformation. Deep learning approaches to phase unwrapping for InSAR have been proposed by (Zhou et al. (2021); Sica et al. (2020); Wang et al. (2021)). The identification of such errors is valuable in itself, as they often need to be fixed in order to improve a wide range of InSAR-derived products and results. The spatial structures of these three anomalies are then computed by taking the mean of all original data interferograms that contain the anomalous epoch interval (Figure 3.7 2nd row). In future, for more complex temporal patterns of deformation than those investigated here, where some interferograms may include contributions from more than one anomaly, I can use the same inversion approach as I applied to the interferogram residuals in Equation 3.2. This would enable to jointly estimate the spatial structure of each anomaly from the subset of original interferograms that have been identified as containing anomalies.

In order to examine the detection results and predicted estimate of anomaly, I remove these estimated anomaly signals from the original interferogram time-series and then re-process the analysis. The re-processed results (Figure 3.7 3rd row) show that the spatial and temporal patterns of earthquake deformation have been accurately predicted and largely removed because the intervals containing the earthquake are no longer identified as anomalous by the detecting mechanism. In contrast, the

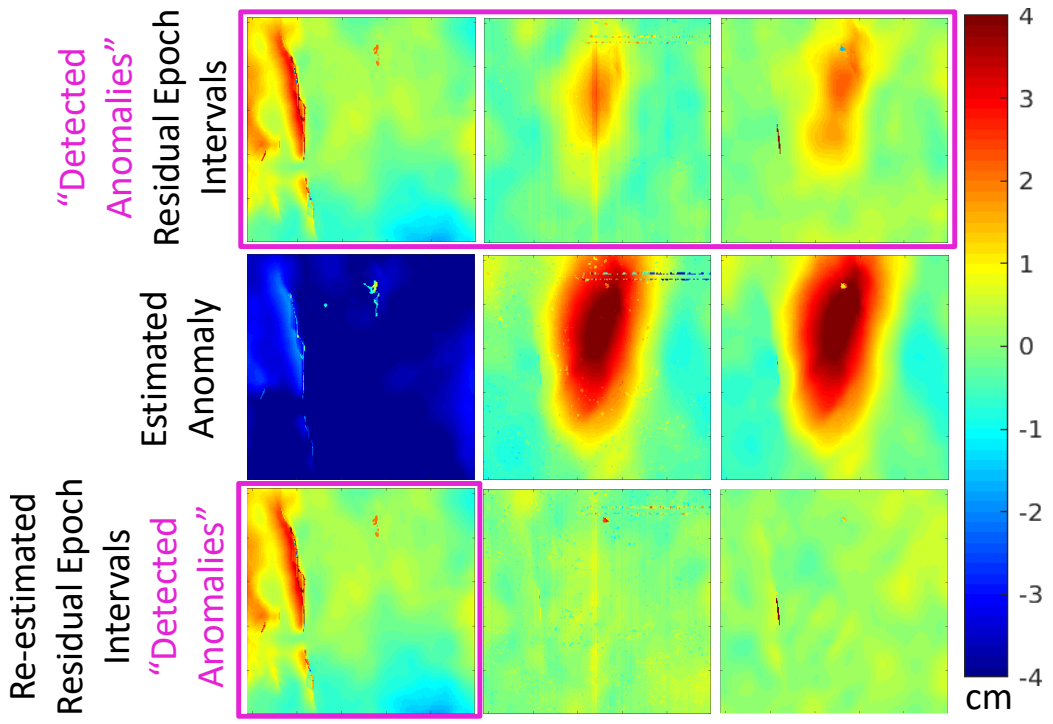

Figure 3.7: Anomaly detection results on a real earthquake of M_w 5.7. The top row shows the residual epoch intervals that are detected as anomalous. 2nd row shows the estimated spatial structures of detected anomalies, which are subtracted from the time series before this 'cleaned' data is then processed again. 3rd row shows the new detection output, after removing previously detected anomalies (pink box) and reprocessing it through the method. The bottom row, the pink box, is still identified as an anomaly, which is also unwrapping errors. In contrast, the earthquake intervals are now identified as normal

unwrapping error persists and is flagged again because it is a data error rather than a natural transient phenomenon. In all experiments, the first 4 and last 4 epoch intervals are ignored during the identification of anomalies because they are always poorly estimated, and are always separated by DBSCAN into the negative cluster with epoch intervals that are derived from interferograms with large amounts of missing data.

Table 3.2: Mean reconstruction error of interferograms, *EP*-Intervals

	Mean RMSE IFG						Mean RMSE Ep-Intervals					
	Normal	Synth Anomaly	% Increase (Synth)	Real	Estimated Anomaly Removed	% Increase (Real)	Normal	Synth Anomaly	% Increase (Synth)	Real	Estimated Anomaly Removed	% Increase (Real)
Test Compact	1.05	1.15	15.18	1.59	1.53	9.95	0.99	1.03	9.48	1.40	1.38	2.46
Test Deep	0.53	0.66	**33.7**	0.87	0.82	**18.8**	0.48	0.56	23.49	0.75	0.73	**7.67**
Bi-Deep (ALADDIn)	**0.40**	**0.47**	28.73	**0.79**	**0.77**	10.35	**0.30**	**0.36**	**29.42**	**0.66**	**0.65**	6.66

Table 3.3: Overall accuracy of models with all testing scenarios.

	Semivariogram Analysis			Clustering Analysis			**V AND C**			V OR C		
	Normal	Synth Anomaly	Real	Normal	Synth Anomaly	Real	Normal	Synth Anomaly	Real	Normal	Synth Anomaly	Real
Test Compact	0.825	0.76	0.78	0.45	0.57	0.60	0.91	0.78	0.92	0.36	0.55	0.46
Test Deep	0.77	**0.95**	0.82	0.40	0.42	0.58	0.81	0.83	0.82	0.36	0.53	0.58
Bi-Deep (AladdIn)	**0.88**	0.91	**0.83**	**0.96**	**0.90**	**0.73**	**0.96**	**0.91**	**0.85**	**0.88**	**0.90**	**0.71**

3.7 Discussion

3.7.1 Temporal dependency

Results displayed in this chapter show that the model has accurately learned the varying spatial patterns of the training data. But due to the unique structure of data and its overlapping behaviour explained in the section 3.4.3, the model lacks learning the temporal dependency for a consecutive set of sequences. For every sequence, there are 5 *EP* that occur more than once. The predictions for these overlapping epochs and their corresponding interferograms must be really similar to the ones which are predicted by the previous sequence. The model must learn to minimize this difference between the multiple predictions of similar epochs. This issue needs attention and a novel training scheme that I have proposed in the following chapter.

3.7.2 Event agnostic detection

One of the goals of this thesis is to develop a methodology that is both unsupervised and even agnostic. I plan to judge the model's accuracy on this criterion. This chapter provides a detailed proof-of-concept and presents the results of a real earthquake and a synthetic Gaussian-based anomaly. Transient phenomenons differ from each other not only on the basis of the spatial pattern but they have different temporal lengths as well, for example, an earthquake is instantaneous, a volcano's age depends on its activity over a range of time and a slow-slip event can last for years with a very small magnitude. I have addressed this by evaluating the methodology in the next chapter.

3.8 Conclusion

In this chapter, I have attempted to systematically automate the detection and extraction of transient episodes of crustal deformation applicable to global InSAR datasets, a goal which is valuable for a wide range of solid earth and natural hazard applications. I propose a new, state-of-the-art deep-learning based anomaly detection approach for the automatic identification of transient deformation events (anomalies) in noisy time-series of unwrapped InSAR images, without requiring supervision or labelling of known example events. The novel workflow learns patterns of the 'normal' non-tectonic signals in the InSAR dataset, leveraging the unique three-dimensional structure of the interferogram stack to estimate the unknown 2D fields that correspond to individual SAR acquisition dates (epochs). The method automatically flags intervals containing deformation and separates the deformation from the normal background time-series. The method can successfully identify synthetic deformation signals with peak line-of-sight displacements of 4.3 cm and of length scale 10 km, with high overall accuracy 91.25% and true positive rate 81.25%, and has also been used to successfully identify a Magnitude 5.7 earthquake and unwrapping errors within data from SW Turkey - a geographic region distinct from the location of the training dataset. I plan to further develop this method by incorporating joint analysis of data from multiple overlapping InSAR tracks, undertaking detailed testing on deformation events with varying temporal and spatial signatures, and employing domain adaptation so that the method can be applied to varied global regions beyond the training region.

Chapter 4

Improved detection of tectonic and volcanic deformation as anomalies in InSAR: deep-learning tailored to differential data

In this chapter, I present the development of an alternative approach based on anomaly detection and tailored specifically to the differential structure of InSAR data, where individual images (interferograms) are the difference in phase between two temporally separated SAR images. I have incorporated a unique model training system called 'Temporal self-attention', which is assisted by the previously trained Bi-Deep (ALADDIn) model in Chapter 3. Just like Bi-Deep, the deep learning model developed in this chapter is unsupervised and event agnostic but with the added advantage of improved temporal consistency.

Part of this work is in the review in the Journal of Geophysical Research (JGR) of the American Geophysical Union (AGU). The text and figures from this submission are reproduced under a creative commons license and changes have been made accordingly for the work to better fit the thesis. The trained models of ALADDin explained in previous Chapter 3 and model for temporal-self attention elabor-

ated in this chapter can be found on GitHub: `https://github.com/AnzaShakeel/Deep-Learning-for-InSAR.git` via DOI: `https://doi.org/10.5281/zenodo.7326911` (Anza Shakeel (2022)).

4.1 Introduction

The abundance of routinely acquires Synthetic Aperture Radar (SAR) imagery from missions such as the European Space Agency's Sentinel-1 (and anticipated for the NASA-ISRO SAR Mission - NISAR) has led to a surge in deep-learning based approaches for the detection of deformation (Anantrasirichai et al. (2018); Gaddes et al. (2019); Rouet-Leduc et al. (2021)). These efforts are critical for optimising the usefulness of large Interferometric SAR (InSAR) for monitoring deformation, given that the high volumes and rates of data (10TB/day, 1000-2000 images/day) precludes systematic manual analysis. Detecting deformation in InSAR data sets is critical for monitoring geohazards (especially volcanoes (Ebmeier et al. (2018)), slow landslides (Bekaert et al. (2020)) and anthropogenic deformation (Semple et al. (2017)) and for understanding of broader tectonic processes (Elliott (2020)). Deep learning approaches also have the potential to transform the emphasis of scientific research, allowing the automated discovery of signals in uniformly analysed regional or global datasets rather than studies focused on locations where deformation is already known.

Deep learning has been widely applied to the field of remote sensing, (e.g., (Sharma et al. (2020); Ren et al. (2021); Shakeel et al. (2019)), mostly to satellite datasets that comprise time sequences of images. In contrast, InSAR has a unique spatio-temporal structure, as interferograms provide information about changes between two dates (differential data). However, the majority of applications of deep learning to InSAR so far have used 2D spatial patterns of phase in individual interferograms (Anantrasirichai et al. (2018)). This has allowed the application of off-the-shelf models like AlexNet (Krizhevsky et al. (2012)) or U-Net (Ronneber-

ger et al. (2015)), modified by (Chen et al. (2022)) for semantic segmentation of active landslides. A disadvantage of this is that off-the-shelf methods are rigid in terms of input size (e.g., the input size of VGG (Simonyan and Zisserman (2015)) is $224 \times 224 \times 3$, where 224 is the size of an image in X and Y dimensions and 3 represents the R(red), G(green) and B(blue) channels of a digital image) because they are built on existing models that were initially trained on RGB images data. An alternative approach is to use time-series derived from interferograms to obtain time sequences of images for input (Gaddes et al. (2019)), although this requires an additional processing step that also has the potential to introduce errors. Most deep learning approaches applied so far for InSAR data are supervised and tailored to detect specific deformation events (Anantrasirichai et al. (2018); Sun et al. (2020); Rouet-Leduc et al. (2021)). These approaches are either trained on event-specific labelled interferograms (Anantrasirichai et al. (2018)) or on synthetic data specifically designed for a particular task, for example, to detect volcanic deformation (Anantrasirichai et al. (2019)), landslides (Zhang et al. (2022); Chen et al. (2022)), anthropogenic signals (Radman et al. (2021); Anantrasirichai et al. (2020)) or tectonic deformation (Rouet-Leduc et al. (2021)).

InSAR data are very different to the video time-series commonly used to develop machine learning analysis methods, where an individual image instead captures information at a particular instant in time (for example surveillance video time-series (Nawaratne et al. (2019)). A particular challenge presented by InSAR data is the often very low signal-to-noise ratio, because the contribution of deformation to the phase in an individual interferogram may be an order of magnitude lower than contributions from changes in atmospheric properties. I aim to develop an approach that is event-agnostic, sensitive to both low-rate and transient deformation, and insensitive to errors associated with higher level InSAR processing (e.g., time-series smoothing, fading signal in time series constructed from short-timespan interferograms).

In the previous Chapter (ALADDIn: Autoencoder-LSTM based Anomaly Detector

of Deformation in InSAR (Shakeel et al. (2022)), was trained on sequences of unwrapped interferograms taken from northern Turkey. The model is a fully convolutional network (FCN) (Long et al. (2015)) that comprises a CNN-LSTM-based encoder and decoder, separated by a neural network. Although the model is capable of detecting deformation as an anomaly, I observed a lack of temporal dependency in some results, and qualitative analysis showed that estimations of deformation varied more than expected for independent estimations of the same epoch. For ALADDIn, a group of 9 epochs (making 26 interferograms, see Figure 4.1), made up a single 'data batch', and is treated independently from the next batch. A comparison of epochs that were estimated by sequential batches (green circles in Figure 4.1) showed that ALADDIn sometimes estimated different spatial patterns of phase for the same epoch due to poor perception of the temporal connection between batches.

In this chapter, I present improvements to the ALADDIn approach comprising an improved training method that makes use of the redundancy in interferogram networks to incorporate information about the temporal structure of signals from multiple data batches. I take a transfer learning (Torrey and Shavlik (2010)) approach by re-purposing the pre-trained model from ALADDIn (Shakeel et al. (2022)) with new model for longer interferogram sequences. In addition, I assess the performance of method using three different scenarios. I evaluate and compare models on an extensive synthetic test data set consisting of multiple variations of magnitude and wavelength of an anomaly representing deformation (Figure 4.6 and Figure 4.7). I use the magnitude 5.7 earthquake from southwestern Turkey previously used as a test for ALADDIn (Chapter 3) (Shakeel et al. (2022)) to illustrate the impact of my method improvements (Figure 4.8). I then assess the impact of the choice of 'patch' size on anomalous deformation retrieved while exploring the potential for reproducing long-lived variations in displacement rate using a volcanic test case from Domuyo, Argentina (Figure 4.9).

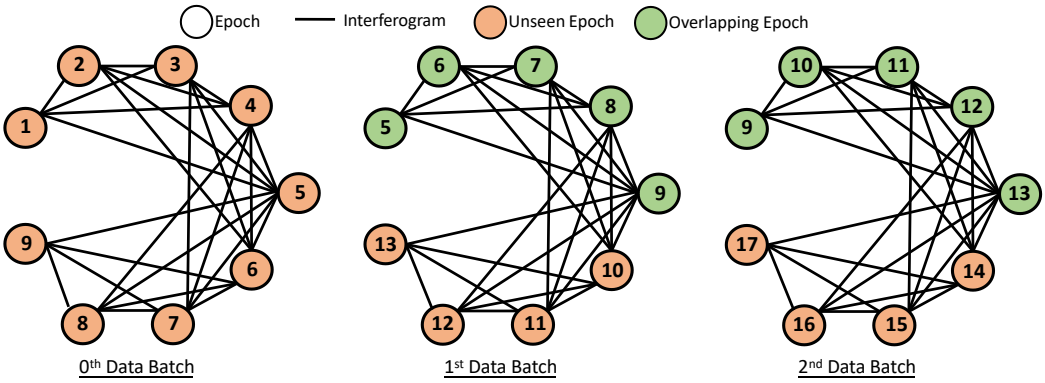

Figure 4.1: Cartoon of the data set structure. Each line represents an interferogram made by subtracting corresponding epochs (single date images) represented by circles. Signals in the epochs contribute either positively or negatively to the corresponding interferograms, and these linkages are learned by the model. I consider a combination of 9 epochs (Time t^1 to t^9) that constructs 26 interferograms, to be a single 'data batch'. Moving along in time with a temporal overlap, the next data batch consists of 9 epochs but from t^5 to t^{13} and so on. For illustration purposes, only three consecutive data batches are shown here. The interferograms (input) constructed from the relevant epochs (output) are shown in orange when they are first passed to the model and green when they have been passed twice due to the overlapping nature of input data. The data network constructed within each batch is such that epochs from t^1 to t^5 are connected to the following four epochs but for the last 4 epochs from t^6 to t^9, each epoch connects only with the following available epochs in the data batch. For example, at epoch t^8, it is used to construct interferograms with t^9 only, these connections can be visualized by the 'lines' in each data batch. Only one line is going forward from t^8 to t^9.

4.1.1 Chapter overview

In this chapter, I first provide an introduction to InSAR data structure used in this chapter and how it can be used with deep learning models in Section (4.1). Then I present the methodology explaining the Temporal self-attention technique 4.2. Finally, as proof of efficiency and accuracy, I put forward a detailed experimental analysis of test results for two real and a synthetic test cases when passed through the model in Section (4.3). Discussion about how these results can be further improved and how should it be tested to judge its diversity is given in Section (4.4). In conclusion, I discuss the key contributions of this Chapter in Section (4.5).

4.2 Methodology

I aspire to provide a method for learning from very large, unlabelled InSAR datasets without the need for manual interpretation. The process of labelling interfeorgrams to act as training data is labour intensive, potentially subjective and requires a priori choices about the characteristics of deformation considered interesting. In principle, more diverse data results in more accurate outputs for deep learning methods (Marcus (2018)), but for unlabelled datasets, this relies on the model architectures being intelligent enough to focus on useful information as there is no 'target' or ground truth (set of actual input interferogram as shown in Figure 4.4(b), referred to as 'GT') available. This solution is to treat the analysis of large, unlabelled InSAR datasets as an anomaly detection problem, where anomalies correspond to any phenomena that deviate from the 'normal' spatio-temporal patterns in the data set. For InSAR, I consider 'normal' phenomena to arise from any contributions to phase not caused by changes to the Earth's surface. These are generally dominated by atmospheric phase contributions but may also include errors in estimations of satellite orbitals and 'nuisance' signals associated with processing such as unwrapping errors, e.g., (Emardson et al. (2003); Simons and Rosen (2007)).

Autoencoders (Baldi (2012)) and fully convolutional networks (FCN) (Long et al. (2015)) are types of network architectures commonly deployed to perform unsupervised tasks (Bengio et al. (2012)). The input and output of such models are identical, so the models learn the underlying distribution of the data and represent them in the form of low-dimensional feature embedding. These embedding act as a bridge between an encoder and a decoder (the main components of an autoencoder), that encrypts and de-crypts useful information about multiple attributes of the data. For the task of anomaly detection, these models are trained on 'normal' data so that they learn the distribution of 'normality' (Gong et al. (2019)). After training, when these models are tested on anomalous data, they predict the out-

put with high reconstruction loss as they are unable to accurately reconstruct the anomaly (as they are rare and never seen by the model). Different combinations of layers can be added to these architectures to meet the objectives of the task and to suit the particular data properties. The wide applications of autoencoders for anomaly detection include (Zhao et al. (2017a); Gong et al. (2019)).

Both ALADDIn (Shakeel et al. (2022)) and the developments presented here take an anomaly-detection approach based on the use of autoencoders and thus avoid both time-consuming pixel-wise labelling for training data and are agnostic in terms of deformation detected. By using networks of interferograms as inputs I treat spatial and temporal patterns co-dependently and do not rely on the derivation of time-series from interferograms that could introduce further artefacts. I exploit the fact that 'normal' signals associated with individual SAR acquisition dates ('epochs') contribute to related interferograms with a temporal pattern that is quite distinct from deformation, which appears as 'anomalous'. The model is trained to predict background epoch time-series from the noise in a redundant network of interferograms. Because deformation has a distinct temporal structure, I can therefore separate it out from the predicted baseline signals.

4.2.1 Data set details - temporal dependency

I use Sentinel-1 InSAR data for training and testing the model. The input data are networks of unwrapped interferograms in radar coordinates generated automatically by the COMET LiCSAR processing system (Lazecký et al. (2020)) *. This system constructs interferograms with the 4 shortest possible timespans both forwards and backwards from each epoch (as illustrated in the Figure 4.1). For a satellite repeat time of 6 days, this results in each epoch contributing to 8 interferograms (6,12,18,24 days). ALADDIn was trained on data from Turkey (Shakeel et al. (2022)) and here, I use the same training data set (LiCSAR Frame name:

*The unwrapped radar-coordinate data format with which the interferograms were saved until the year 2019, is no longer saved. However, the exact data format on which this model is trained can be reconstructed from the LiCSAR intermediate products that are preserved

014*A*_04939_131313, data spanning from the year 2017 to 2019) that was first passed through ALADDIn and its predictions are used to initiate the training of this new temporal self-attention model. This data set was selected on the basis that they were not expected to contain any known examples of deformation, but were dominated by atmospheric signals.

to manage the complexity of the model and the memory required to train a large number of parameters, the frame is divided into cubes of size $256 \times 256 \times 26$ pixels (a spatial extent of approximately 20.5 km $\times$ 20.5 km) with a fifty percent spatial overlap (in both E-W and N-S directions) and instead of passing the whole time-series in every training iteration, a set of 26 interferograms (abbreviated to IFGM) that cover 9 epochs (abbreviated to *EP*) is passed (the set is called a data batch and is shown in the Figure 4.1, where circles represent *EP* and the lines connecting each circle i.e. *EP* represents the IFGMs). The temporal sliding window is 9 *EP* in length and moves with a stride of 4 ensuring a temporal overlap of $> 50\%$ between successive input data batches. The 26 IFGM link each *EP* with all successive and preceding *EP* within the sequence, up to a maximum distance of 4 forwards and backwards in time. For example, the central *EP* is linked by 8 IFGM to all other *EP*, but all other *EP* in the data batch are linked with fewer than 8, to a maximum of 4 IFGM for the first and last *EP* in the batch (as illustrated in the Figure 4.1, where only four lines/IFGMs can be seen linking *EP* at t^1 and t^9).

The term temporal dependency is directly related to the temporal overlap that exists within the data set. This can be seen in the Figure 4.1 when the colour of *EPs* changes from orange to green with successive input data batches. For every *EP* that occurs in overlapping data batches, the model must predict outputs that are similar realistically. So, the output of each data batch should be temporally dependent on the output of all batches. despite a high detection rate (91.25% overall performance accuracy on a synthetic deformation test case) ALADDIn (Shakeel et al. (2022)), produced a different set of solutions for the five *EP* that overlap between data batches (Figure 4.1 green circles represent overlapping *EP*s in the

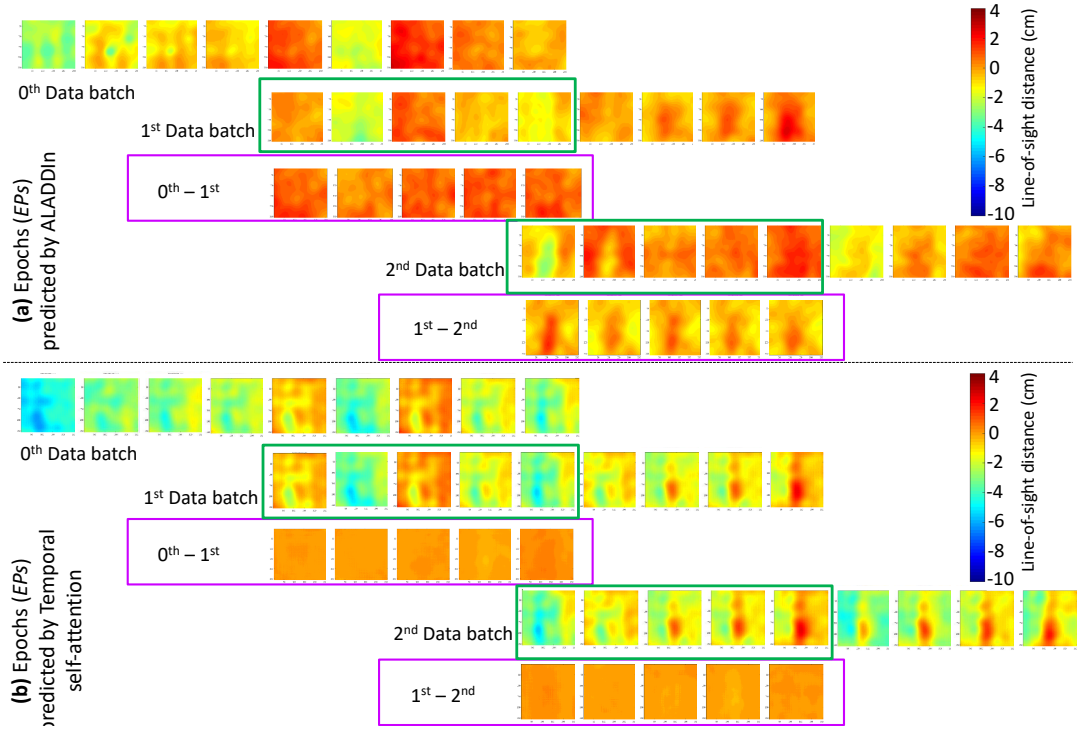

Figure 4.2: The comparison between the predicted epochs of both models. ALADDIn (a) and the Temporal self-attention model (b). The green box covers the *EPs* that are overlapping (as in the Figure 4.1). The purple box shows the difference between the *EPs* that are occurring in multiple data batches. Comparing both the boxes for (a) and (b), it can be seen that the difference computed for Temporal self-attention is approximately zero. That is not the case for ALADDIn, the predictions made by ALADDIn (a) are not similar. Hence its output is not temporally dependent.

data network). The evidence of this is shown in the Figure 4.2, where the output of ALADDIn (4.2 (a)) is evaluated based on the difference between successive data batches. I, therefore, aim and have successfully designed a system that predicts realistically similar/temporally dependent spatio-temporal patterns for *EP*s in the overlap between data batches (as shown in Figure 4.2 (b)).

4.2.2 Transfer learning - Pre-trained ALADDIn

I re-purpose the model pre-trained for ALADDIn here, using a transfer learning approach. In the new network architecture, the same number of layers are used as in the encoder and decoder of ALADDIn (Shakeel et al. (2022)). In fact, instead of

initializing the weights of these layers from scratch, transfer learning (incorporation of previously learned knowledge (Torrey and Shavlik (2010)) is applied and the previously trained weights are utilized to begin training.

4.2.3 New network architecture using temporal self-attention

I designed the network architecture of this training system to exploit redundancy in the input interferogram network, that is, that information about each epoch may appear in multiple interferograms (Figure 4.1). The base of the deep learning model is a fully convolutional network, including an autoencoder combined with a neural network as shown in Figure 4.3(b).

The encoder translates the hidden features/distributions of the input data batch, which is a batch of interferograms (26x256x256) (Figure 4.1). The neural network then converts the features that are learned from IFGMs into the form of *EP*s (9x256x256). This converted feature space is then decoded and interpreted to predict the unknown *EP* time-series.

The model consists of a set of convolutional and convolutional Long Short Term Memory (LSTM) with pooling layers in the encoder. The purpose of pooling layers is to down-sample the input, by taking, for example, a minimum, maximum or average value. In this model, I have used maximum pooling (referred to as max pool, to gather the maximum value of features within a window size. Likewise, transposed convolutions and convolutional LSTM with upsampling layers are combined in the decoder. In between the encoder and decoder, the neural network consists of three 1D fully connected (FC) layers. The 3D convolution layer spans the input spatially in all dimensions, whereas the Long shot-term memory (LSTM) layer (originally 1D) is capable of maintaining memory with the help of learnable 'forget', 'input' and 'output' gates. This, when combined with the convolutional operation, serves for any multi-dimensional input. This layer is tailored for the specific task of learning both spatial and temporal patterns co-dependently. Every convo-

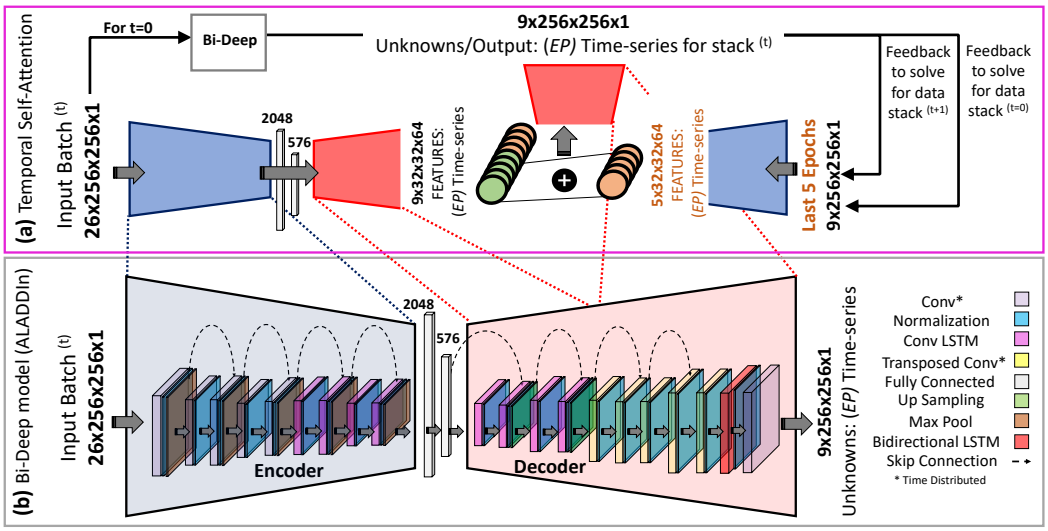

Figure 4.3: Illustration of network architecture and self-attention training scheme. (b) Bi-Deep model from ALADDIn pipeline (Shakeel et al. (2022)) is used for pre-trained weights and to initiate predictions for self-attention. The layers used to create the architecture are: time-distributed 2D convolutional, max-pooling, normalization, 2D convolutional LSTM, fully connected layers, transpose convolutions, upsampling and a bi-directional LSTM layer. Skip connections are also in place to merge features. For temporal self-attention in (a) the decoder is disconnected to fuse features from the overlapping epochs from the previous data batch$^{(t-1)}$. The pre-predictions from the previous data batch$^{(t-1)}$ is passed through a mini-encoder consisting of a pair of time-distributed conv and two convolutional LSTMs

lution layer is followed by a normalization layer and max-pool for downsampling in the encoder. Similarly, in the decoder, every convolutional LSTM is followed by a normalization and upsampling layer. Transposed convolutions (often called deconvolutions) perform similar operations but with broadcasting the feature map instead of downsizing. The mini-encoder used for the attention of overlapping 5 epochs consists of 2 pairs of convolutions, convolutional LSTM following normalization and max-pooling layers.

The neurons used in the fully connected layers are 2048 and 576 (see Figure 4.3). These number of neurons are of immense importance, as they are used for converting the feature maps from 26 interferograms (at the encoder side) to 9 epochs (at the decoder side). For example, the size of data after being processed by the encoder is downsized from $26 \times 256 \times 256 \times 1$ to $26 \times 8 \times 8 \times 1$ then the decoder should receive an input of $9 \times 8 \times 8 \times 1$. Hence, the neurons in the 2nd FC layer are

computed by multiplying the dimensions $9 \times 8 \times 8 \times 1 = 576$. So, it can be unrolled back into the dimension $= 9 \times 8 \times 8 \times 1$ and used by the multi-dimensional layers in the decoder. Tanh activation functions are used for every layer. This function ranges from [-1 to 1] which is ideal for this model, as negative values are equally important as positive values.

Furthermore, skip connections are added for feature reusability and to avoid the problem of vanishing gradients (Hochreiter (1998)), where the weights (calculated by each layer in a 'deep' model) gradually decrease to zero and backpropagation fails. In deep convolutional models, this problem often occurs and hinders learning. Skip connections are represented by dotted lines in Figure 4.3(b), their purpose is to re-use the output of layers and feed to deeper layers, to merge the information and process t adding more features to help to dodge the vanishing gradient problem. Finally, a bidirectional convolutional LSTM layer is added, that spans the output both forwards and backwards and combines features to refine the predictions.

4.2.3.1 Training protocol - feature merging

The predicted *EP* time-series by the model should be spatially and temporally consistent (in a sequential manner), regardless of the overlapping nature of the data batches. To restore continuity in the data, that is lost when I divided the time-series in to multiple *data batches* for computational purposes, each *data batch* is computed with a temporal stride of 4 epochs that ensure 5 overlapping epochs. The value for temporal stride is selected after considering computational capacity as decreasing this value would increase the total number of *data batches* and vice versa. The features learned for every ($data\ batch^{(t-1)}$) batch should therefore facilitate learning for its proceeding ($data\ batch^{(t)}$), especially for overlapping epochs (which are 5, as illustrated in Figure 4.1) represented as green circles, as they are already been computed in the previous iteration when ($data\ batch^{(t-1)}$) was processed. This form of attentive learning (a mechanism that focuses on specific temporal regions in a sequence to create a representation of it, for example, here

it focuses specifically on overlapping *EP*s.) is introduced in the decoder part of the model. The continuity of the decoder is interrupted and predictions of the last 5 epochs ($data\ batch^{(t-1)}$) are fed back by passing them through a mini encoder. These features are combined with the first 5 epochs ($data\ batch^{(t)}$). The merged features are passed to the rest of the decoding layer to make refined epoch predictions. In this way, the cyclic nature of deep learning model training and back-propagation is not affected. Because no previous prediction is available in the case of the very first $data\ batch^{(t=0)}$, the 5 epochs are constrained only by features using the Bi-deep model of ALADDIn (Shakeel et al. (2022)).

Instead of one, two loss functions are used to constrain the model:

$$\begin{aligned} Loss_{IFG} &= \sum_{i=1}^{n}(Output_{IFGM} - Input_{IFGM})^2 \\ Loss_{EP} &= \sum_{i=1}^{n}(5_{epochs^t} - 5_{epochs^{t-1}})^2 \\ Loss &= Loss_{IFG} + Loss_{EP} \end{aligned} \tag{4.1}$$

where n is the number of interferograms in the case of $Loss_{IFG}$, it refers to the loss computed between reconstructed interferograms by the model and the input interferograms which is also the ground truth. $Loss_{EP}$ is the difference between the current predictions of overlapping epochs and the previous ones, so here n is 5, and the accumulated loss is then back-propagated. The model is trained using Keras (Chollet et al. (2015)) (a deep learning API) with the TensorFlow (*Abadiet al.* (2015)) backend. Due to the large size of the images in memory the batch size was set to 1. Adam optimizer (Kingma and Ba (2014)) was used with a learning rate of 0.00001. A lower learning rate gives the model a chance to learn features through steady changes in a loss instead of rapid fluctuations.

4.2.4 Framework for Anomaly Detection - Shuttling

When the model is tested against anomalous data, I expect to find high spatial residuals between input data and predicted signals and/or a high overall recon-

struction error. Due to the event-agnostic nature of this approach, I do not rely on identifying specific patterns in the reconstruction error. Because I have no preconceptions about the spatial size, intensity and temporal structure of an anomaly, I design a novel detection framework to pick up all kinds of anomalies with a minimum possible rate of false positives. The prediction for every input data batch is refined based on the estimations of findings from previous data batches. I take full advantage of this capability and introduce "*shuttling*" during test time. "*Shuttling*" as its name suggests, completes multiple passes in forward and backwards directions for all of the data batches, as shown in Figure 4.4(a), here for illustration purposes the data contains three batches only. During the backward pass, the data is flipped spatial as well because now the pre-*EP* image serves as the post-*EP* image to compute the interferogram which is: $post_{EP} - pre_{EP}$. The residuals (RES) shown in Figure 4.4(b) reveal that reconstruction is more accurate after shuttling is implemented and the residuals are near zero after the second forward pass. Input IFGMs that have spatial data gaps or pixels with missing values ('NaN') are a data error and one such IFGM is shown in Figure 4.4(b) black boxes. If 'NaN' is passed through the model, it will propagate through the model due to back-propagation and diminish the learning to a 'NaN'. To avoid this, I identify these missing values and replace them with a zero. This introduces box-like patterns of zeros in the input, but as I am enforcing the model to learn both spatial and temporal patterns. This helps the model to predict even when the input is zero (as shown in the black boxes of Figure 4.4 (b) 'PRED'). Even though the model makes estimates for missing pixels, these box-like patterns are passed in the 'RES' through subtraction of PRED and GT.

Once the data are shuttled completely (terminated at the third pass, when no improvement results are observed), I expect anomalies to appear as a residue in residual (RES) IFGM as shown in Figure 4.7. An interferogram captures the changes that occurred between two dates, so potentially spans multiple *EP*s. Hence, for the data structure, an anomaly will always appear in multiple interferograms.

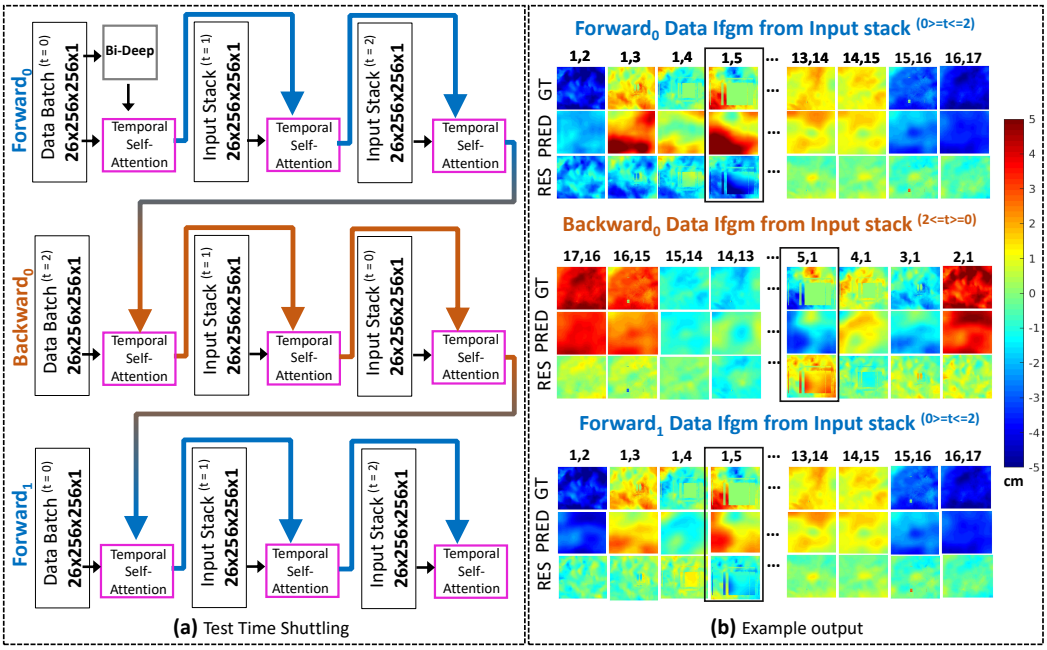

Figure 4.4: The Shuttling scheme. (a) Temporal self-attentive training and test-time shuttling procedure is illustrated here. Features of overlapping *EP* (green circles in Figure 4.1) from *data batch*$^{t-1}$ are fed in the model (pink box) for every *batch*^{t} as moved across time from *batch*$^{t=0}$ to *batch*$^{t=2}$. While testing the same procedure is repeated for backward interferograms (both in space and time). The last IFGMs of the forwarding pass (e.g., 15,16 and 16,17) are now the first IFGMs of the backward pass (e.g., 17,16 and 16 and 16,15). This process is called shuttling, it is repeated for another forward pass and so on until no change in the output is observed (only three passes are shown here for illustration purposes). (b) Shows an example output from the process, where 'GT' refers to ground truth IFGMs which is the input to the model (presented as lines in the Figure 4.1), 'PRED' is predicted IFGMs (output of the model) and 'RES' are the residual IFGMs computed by subtracting PRED from GT to measure what is missed by the model. Shuttling helps to achieve model predictions that are close to the input 'GT'. In a comparison of 'RES' of $forward_0$ (top row) and with 'RES' of $forward_1$ (bottom row), it is clear that the 'RES' is near zero. Black boxes enclosing IFGM 1,5 in all shuttling iterations display an example of spatial data gaps or pixels with missing or 'NaN' values, that are replaced with a zero before passing through the model.

To accurately detect the temporal window of an anomaly, I first reduce the residuals down to a mutually exclusive set of N_{EI_r}) "residual epoch intervals". An *EP* interval is a different image that spans two successive *EP*s, so, therefore, N_{EI_r} is equal to one less than the number of *EP*s. These intervals are essentially equivalent to the shortest spanning set of residual 'daisy-chain' interferograms but are instead estimated from the full set of residual (RES) interferograms so are more robust to

noise than any residual image. In order to estimate this set of N_{EI_r} residual epoch intervals, I perform a linear least squares inversion on a pixel-by-pixel basis of the N_{IFGM_r} residuals as follows (based on the SBAS approach from (Berardino et al. (2002))):

$$d_{IFGM_r} = G.m, \tag{4.2}$$

where d_{IFGM} is a $N_{IFGM_r} \times 1$ array containing pixel values of all N_{IFGM_r} residual interferograms, m is the $N_{EI_r} \times 1$ vector of residual EP intervals that I wish to solve for, and G is the $N_{IFGM_r} \times N_{EI_r}$ sized design matrix for this system of equations, containing 1s and 0s only. Eq 4.3 shows an example of matrix G for a set of six residual interferograms ($IFGM_{12_r}$, $IFGM_{13_r}$, $IFGM_{14_r}$, $IFGM_{23_r}$, $IFGM_{24_r}$, $IFGM_{34_r}$), constructed from four epochs (EP_{1_r}, EP_{2_r}, EP_{3_r}, EP_{4_r}), which will output three epoch intervals (EI_{12_r}, EI_{23_r}, EI_{34_r}) based on the residuals.

$$\begin{pmatrix} IFGM_{12_r} \\ IFGM_{13_r} \\ IFGM_{14_r} \\ IFGM_{23_r} \\ IFGM_{24_r} \\ IFGM_{34_r} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} EI_{12_r} \\ EI_{23_r} \\ EI_{34_r} \end{pmatrix} \tag{4.3}$$

Instead of using all residuals of the overlapping epochs, I only use the latest ones predicted by the model, which should be the most reliable. Also, I perform the linear least square inversion for both, forward progressing data and backward progression data, to create two independent sets of residuals EP intervals for detection. These intervals are then automatically analysed for the presence of spatial anomalies using two complementary analysis methods: semivariogram analysis (Wackernagel (2013)) and density-based clustering (DBSCAN) (Kriegel et al. (2011)).

In the absence of deformation, the residual epoch intervals are expected to take values near to zero (as they will be accurately reconstructed by the model, e.g. see Figure 4.4(b)), but in case of an anomaly or multiple anomalies within a sequence, the spatial structure of that anomaly will appear in at least one epoch residual. My goal is thus to separate 'normal' intervals from the anomalous ones, without any prior knowledge of where anomalies appear in a sequence. I use a clustering algorithm (DBSCAN) (Kriegel et al. (2011)) that does not require a-priori specification of the number of clusters to locate anomalies. To ensure I detect all anomalies, using both, forward and backwards independent sets of RES *EP* intervals and perform DBSCAN combining them, setting the minimum points in a cluster to be two.

I incorporate the spatial variability measured by the semivariogram, to separate anomalies with certain spatial frequencies (e.g. deformation that is spread over small regions) from larger areas that are normal. These localised but significant changes are less likely to be detected by clustering or simply computing bulk differences between actual and reconstructed images (e.g. by a Mean Squared Error). I expect that residual epoch intervals (epoch intervals constructed using residual interferograms i.e. ground truth - prediction) containing no anomaly will all have similar spatial structure and therefore will also have similar empirical semivariograms, whilst epoch intervals containing anomalies will have semivariograms that differ substantially from this normal structure. A semivariogram is calculated for each residual epoch interval, and the root-mean-squared-error is computed between each semivariogram and all others in the entire set of residual epoch intervals across all sequences. Finally, the classified anomalies from the semivariogram and clustering analysis of epoch interval time series are combined using an AND operation in order to reduce the number of false positives. The variables and parameters used for both these analysis methods are the same as ALADDIn (Shakeel et al. (2022)).

4.3 Results

I evaluate the performance of improved models on the basis of (1) temporally consistency in overlapping data batches, (2) the range of anomalies detected in terms of spatial and temporal scale, and (3) the model's ability. to process large areas using sliding spatial and temporal windows and (4) its ability to detect a range of deformation types.

4.3.1 Performance evaluation - 2D Gaussian synthetic test data set

Before testing the models on real scenarios, I use a synthetic data set to test the model. I design a synthetic test to compute the accuracy and assess the capacity of the models. It is important to analyse the extent in terms of the wavelength and magnitude of the deformation signal that is accurately captured by the model. For simplicity, I use a 2D Gaussian spatial pattern with varying magnitude and wavelength and add it at eight random instances in a test patch location of the Turkey data frame (014A_04939_131313 i.e. same as the training frame but the southern region of this frame is separated for evaluation purposes) that has never been seen by the model during training. The structure of this anomaly is computed using the equation given below:

$$Z(x, y) = A. \exp\left(-(x^2 + y^2)/r\right) \tag{4.4}$$

where r is the exponential length-scale or wavelength, that varies from 10 m to 12 km. A is the scaling parameter that is directly proportional to the magnitude or peak-value that varies from 1 cm to 11 cm (almost doubling each time to cover maximum range with fewer test variations). x and y are spatial coordinates relative to the location of Gaussian peak. This means that the signal-to-noise ratios for this test dataset range from 0.0003 to 70 (1 cm peak displacement) to 0.0000003 to 1.7 (11 cm peak displacement). These anomalies are shown in Figure 4.5.

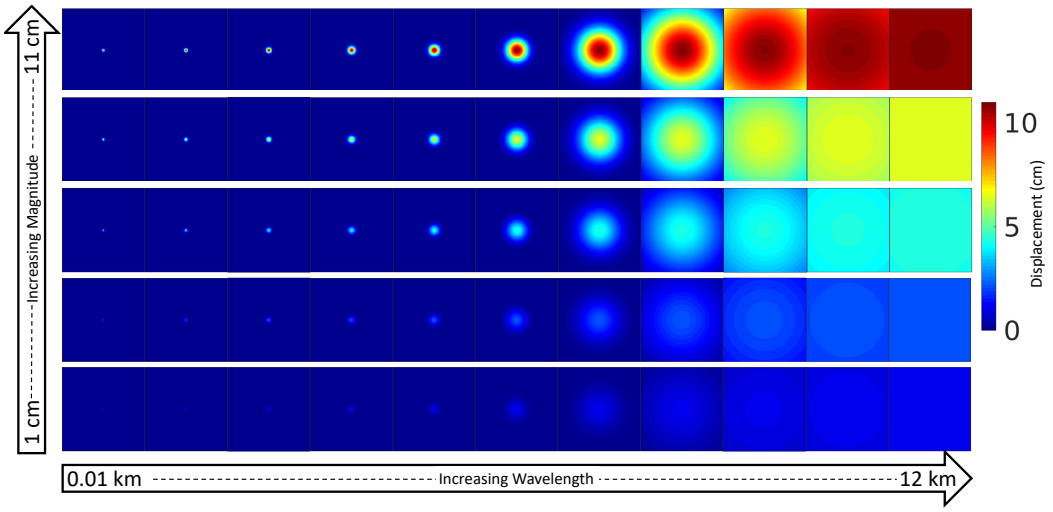

Figure 4.5: The synthetic test data set. It is created using the equation 4.4 with varying wavelength (10 m to 12 km) and magnitude (1 cm to 11 cm). Each anomaly is augmented with same wavelength and magnitude but with different center location, creating 105 anomalies altogether. These synthetic anomalies are added into 8 different time intervals in the test data.

I examine a total of 105 scenarios, each consisting of 10 data batches (spanning from to $28^{th} July 2017$ to $18^{th} April 2018$), containing 8 anomalies. The test set is passed through the model and analysis is done on a quantitative as well as a qualitative basis. Figure 4.6 (a) shows a heat-plot of overall accuracy (OA) and Figure 4.6 (b) shows a heat-plot of true positive rate (TPR) and Figure 4.6 (c) shows the average OA, TPR and FPR of this model for each magnitude. Where sample outputs of two synthetic cases are shown in Figure 4.7. As it is a synthetic data set, ground truth is known beforehand and one-hot encodings of ground truth and predictions are computed, where a label '0' is for normal data and '1' is for an anomaly. TPR and OA are then plotted for each test sample. The temporal self-attention model performs with a higher true positive rate (>60%) for lower magnitudes (2 cm) and higher wavelengths. The mean true positive rate is then plotted in comparison with the mean false positive rate and mean overall accuracy for each magnitude.

The equations of which are given below:

$$\begin{aligned} TPR &= TP/(TP+FN) \\ FPR &= FP/(FP+TN) \\ OA &= (TP+TN)/(TP+TN+FP+FN) \end{aligned} \tag{4.5}$$

where TP is truly positive, FP is a false positive, TN is a true negative and FN is a false negative. Negative here corresponds to the 'normal' data or 0's and positive corresponds to anomalies or 1's. OA is the overall accuracy, and TPR and FPR are true and false positive rates respectively. The plot in Figure 4.6 (c) shows that the average overall accuracy of the model for magnitudes greater than and equal to 4 cm is greater than 70%, this is due to the true negatives that are much more in number and carry equal importance, as it shows the architecture's specificity. Whereas a fairly high false positive rate is majorly due to the fact that the model is fitting on 'normality', leaving greater residuals resulting in false flags.

The false positives include unwrapping errors as well as missing data. The model does a better job of predicting 'normal' interferogram patterns and therefore flagging more of the errors in the input data, which are classified as anomalies according to the tests. The temporal self-attention model has proven to be accurate by detecting anomalies with magnitudes of 4 cm or peak line-of-sight displacement of a few cm (1-2 fringes), and of a length-scale greater than a few hundred metres, it displays an overall accuracy of 80 to 90% - this is the region in the pink polygon in Figure 4.6 (a) and (b). The Figure 4.7 displays the results of temporal self-attention for two synthetic test scenarios both for low magnitude and greater wavelength (bottom) and for higher magnitude and lower wavelength (top). The model accurately estimates the centimetre scale (4 cm and 7 cm) spatial structure of the flagged anomaly as shown in the Figure 4.7.

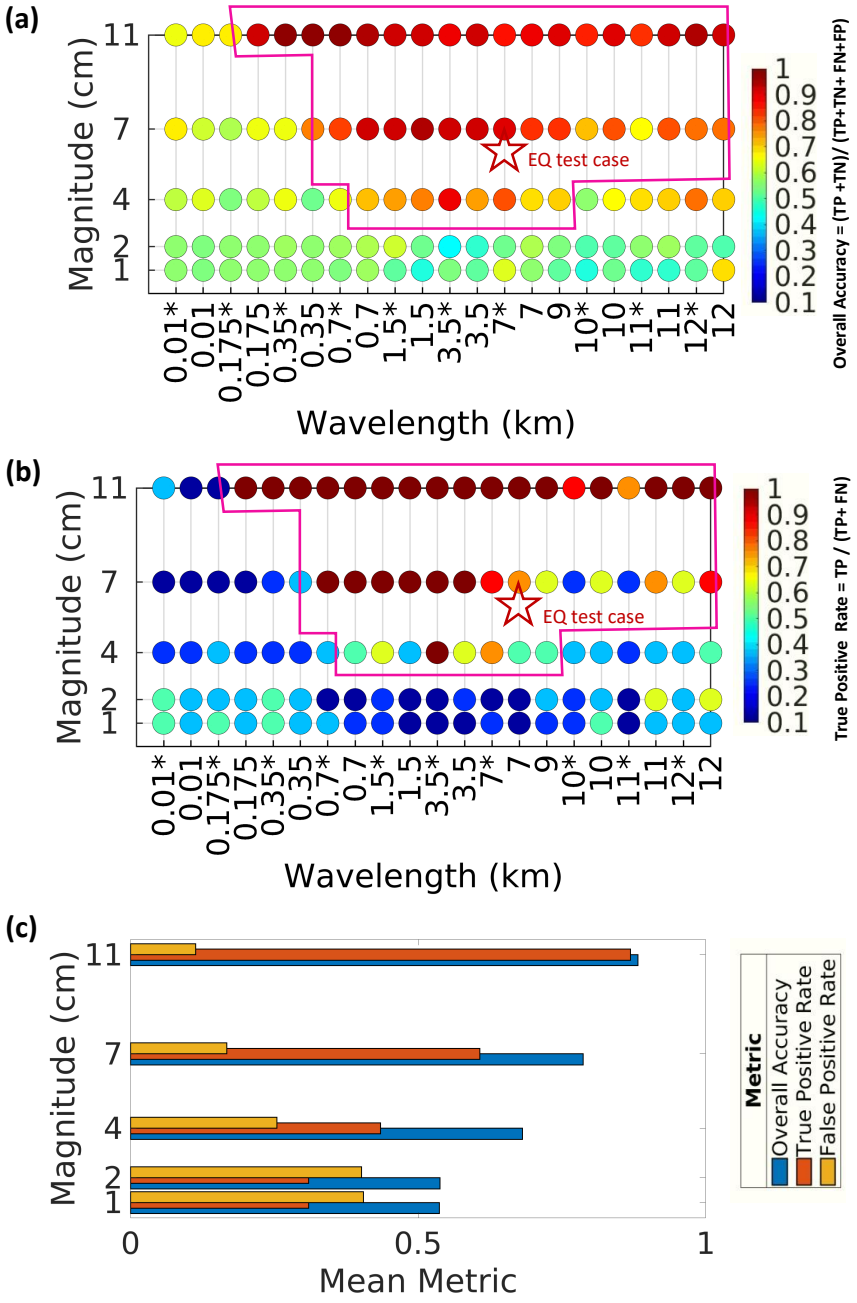

Figure 4.6: Performance evaluation of the model on synthetic test data. (a) Shows a heat plot of overall accuracy for each magnitude and wavelength of anomalies. (b) Similarly, the same is plotted for a true positive rate. (c) Shows the bar plots for mean true positive rate (orange), mean false positive rate (yellow) and mean overall accuracy (blue). The pink polygon covers the predicting region or spatial scale of detection of the model. The true positive rate is high for magnitudes as low as 4 cm and length scale few m to km. The red star shows the real earthquake (in terms of its magnitude and wavelength) that occurred in Turkey and is detected by the model.

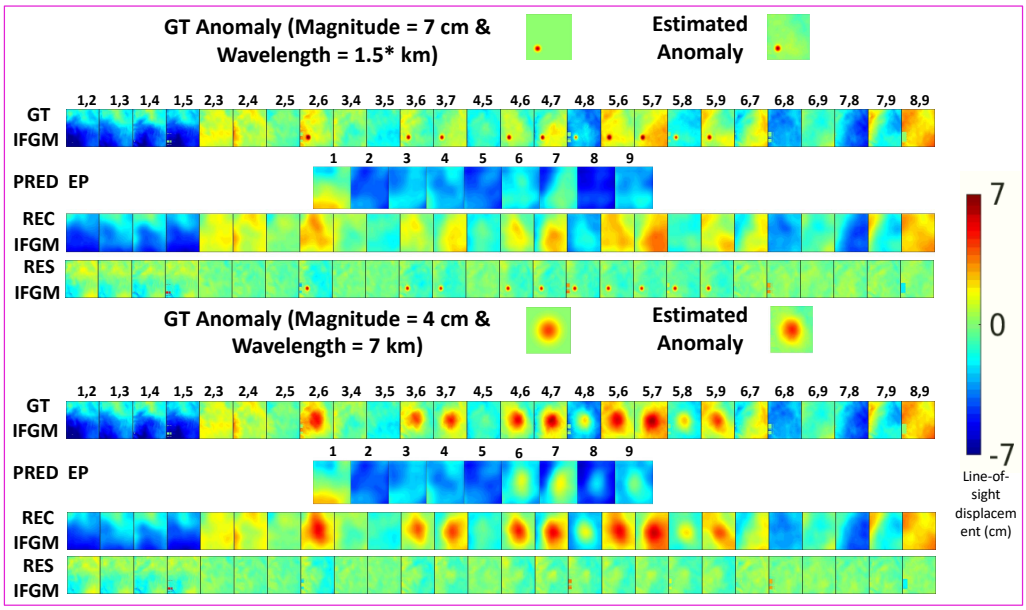

Figure 4.7: The figure shows two cases of synthetic test data. (Top) Magnitude 7 cm and wavelength 1.5 km and (bottom) magnitude 4 cm and wavelength 7km. GT IFGM is the ground truth interferogram, PRED *EPs* is the predicted epoch, REC IFGM is the reconstructed interferograms made using PRED *EPs* and RES IFGM is the residual interferogram. The RES IFGM is of importance as it carries the anomalous signal missed by the predictions.

4.3.2 Case study: 2019 Earthquake in south-west Turkey

To compare the accuracy of models I revisit a real earthquake of magnitude 5.7 that occurred in southern Turkey on 20^{th} March 2019 (Elliott (2020)), and was previously used to test ALADDIn (Shakeel et al. (2022)). This south-western region of Turkey has experienced major earthquakes in the past (M_w 7.0 in 1914 (Ambraseys (1988)), M_w 6.2 in 1971 (Taymaz and Price (1992)), M_w 6.2 in 1995 (Wright et al. (1999)) and M_w 6.6 in year 2017 (Karasözen et al. (2018))). InSAR data that has been analysed by the model, estimates deformation of approximately 4 cm (as reported by (Elliott (2020))), shown in Figure 4.8 (c). The data for this test case is processed from the time-period 18^{th} September 2018 to 10^{th} April 2019. I divide the data into 7 data batches (according to the Figure 4.1), comprising 31 epoch intervals in total. This test region is never seen by any of the models during training. The method ALADDIn successfully detected this earthquake and

estimated its spatial structure, but due to the overlapping data structure, two variations of the same instantaneous anomaly were retrieved, as shown in Figure 3.7. It can be seen that the Temporal self-attention model constructs one accurate estimate of epoch interval whereas the two variations estimated by ALADDIn are different in terms of the spatial pattern (shown in Figure 4.8 (a)). In a comparison of residual *EP* intervals with the estimated earthquake structure (Figure 4.8 (c)), it can be seen that the proposed method produces the pattern of an anomaly in the residuals similar to the actual structure (as reported by (Elliott (2020))). All model variations prove capable of flagging the earthquake, but I am still able to compare their performance on the basis of the spatial structure of the anomaly leaked in the residual i.e. the foreground for the detection framework. Each of the predicted epoch intervals depicts the behaviour of background noise which is negatively and positively correlated to one another and the interval capturing the earthquake is spatially different from other intervals. The estimated spatial structure of the flagged anomaly by the Temporal Self-attention model is shown in Figure 4.8 (c).

4.3.3 Case study: The Domuyo Volcanic eruption

I apply the most accurate model - Temporal Self-attention trained on real data - to an additional real test case. I select a well-documented period of uplift at Domuyo volcano, Argentina (Lundgren et al. (2020); Astort et al. (2019); Derauw et al. (2020)), because this allows to examine (1) how the model trained on real InSAR data from northern Turkey performs in a location with completely different atmospheric conditions and topography and (2) how well it can detect persistent rather than transient deformation, and (3) how the model performs when the deformation signal (in this case - 64 $\times$ 40 km) exceeds the patch size (20.5 $\times$ 20.5 km). Domuyo stratovolcano (4702 m elevation), in northern Patagonia, is thought to be late Pleistocene (but possibly Holocene) age. It has no record of historical activity, but a major hydrothermal field centred southwest of the volcano's flanks has a very high thermal energy release and recent gas-driven explosions, which imply

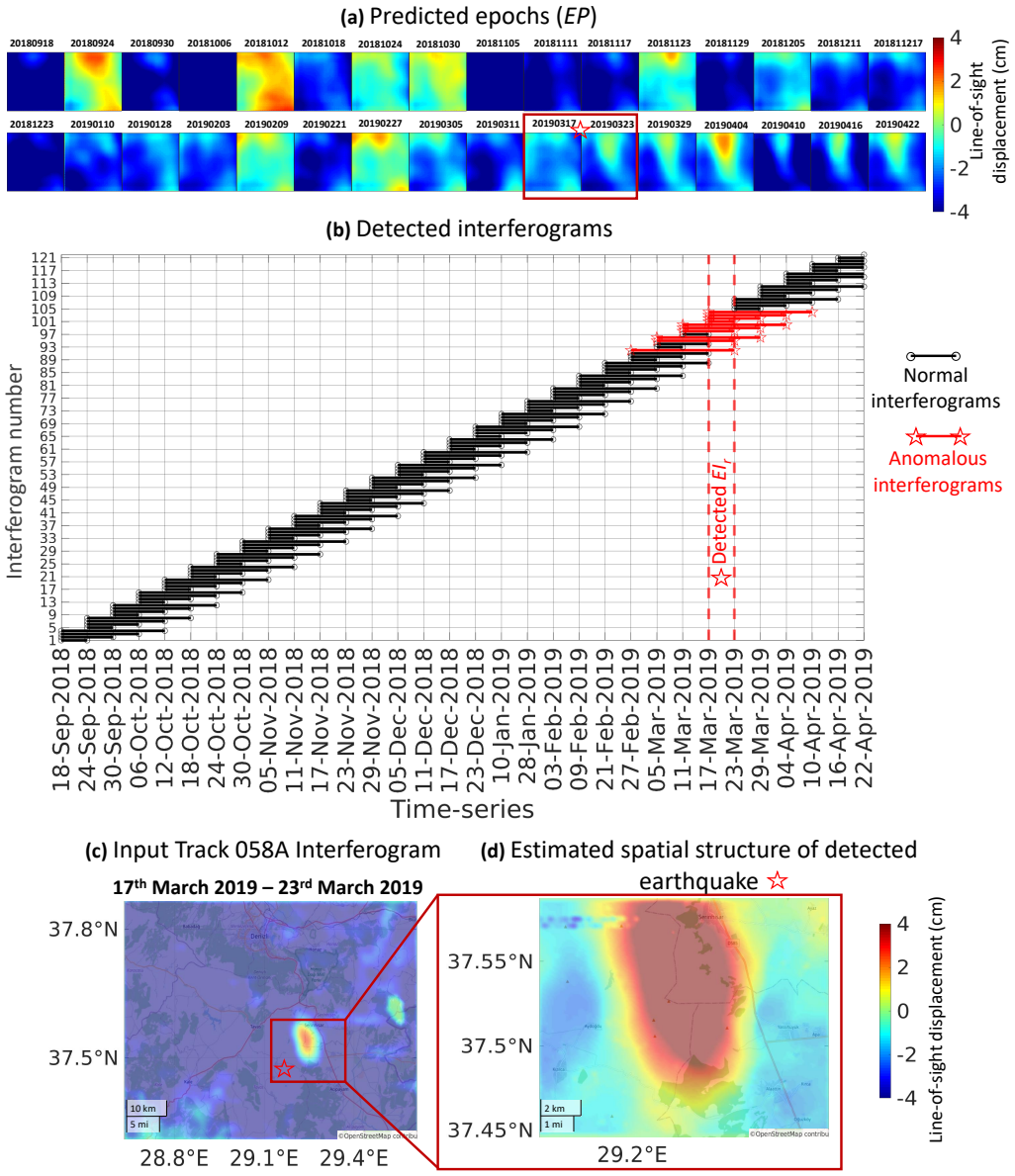

Figure 4.8: The results of the real earthquake test case. (a) Shows the predicted *EPs*. Red box is the time period 17^{th} March 2019 to 23^{rd} March 2019 that cover the event date, i.e. 20^{th} March 2019. (b) Shows the time-series plot of detected interferograms encompassing the detected epoch interval which is marked by a red dotted line. Interferograms marked with red colour and represented by the red 'pentagon' are anomalous. (c) Shows the shortest (6-day) ground truth (GT) interferogram capturing the earthquake anomalous signal. The red box covers the exact region of size 256 × 256 pixels that have been processed by the model. (d) Shows the estimated spatial structure of the flagged anomaly by the model. The deformations shown here are measured in cm.

the presence of an active magmatic system (Chiodini et al. (2014); Lundgren et al. (2020)). Further evidence for this comes from uplift, which has been attributed to the intrusion of volatile-rich magma at 6.5-7 km depth (Astort et al. (2019)), and has occurred in lagged correlation with edifice-wide warming (Lundgren et al. (2020)). Domuyo subsided between 2008 and approximately 2013, before entering a phase of uplift in 2014 with a maximum rate of 15 cm/yr. Uplift slowed until early 2021, when the volcano began to subside. I selected a period of relatively high-rate uplift between May 2017 and December 2018 for the method test (Figure 4.9a). Over this interval deformation was relatively constant, so every interferogram to be flagged is expected.

This case study also provides the opportunity to assess the implications of applying deep learning to automatically processed, noisy InSAR data sets with significant data gaps. The network design for standard LiCSAR processing relies on short timespan interferograms: 4 forward connections for each epoch, maximum interferogram length of 48 days for 12 day acquisition intervals, as at Domuyo. This means that even for the relatively high rate persistent deformation at Domuyo, displacements in individual interferograms are commonly < 1 cm, well below the level of atmospheric contributions (Figure 4.9b). Furthermore, the standard LiCSAR network design is not optimised for regions with major seasonal variations in phase coherence (e.g., snow cover). This results in loss of coherence in the test dataset, relative to a network tailored to include only summer-summer interferograms (Lundgren et al. (2020)).

The interferograms I analysed also had a minimum 12-day interval as compared to the training data which had a minimum of a 6-day gap. In order to create the data network with the structure shown in the Figure 4.1, I require each epoch to be connected by 4 interferograms forwards and backwards. Where there were large gaps between useable interferograms in the automatically processed data, epochs were skipped. The sample interferograms shown in Figure 4.9 (e) illustrate typical data gaps (circle), low coherence with unwrapping errors (square), low signal-to-

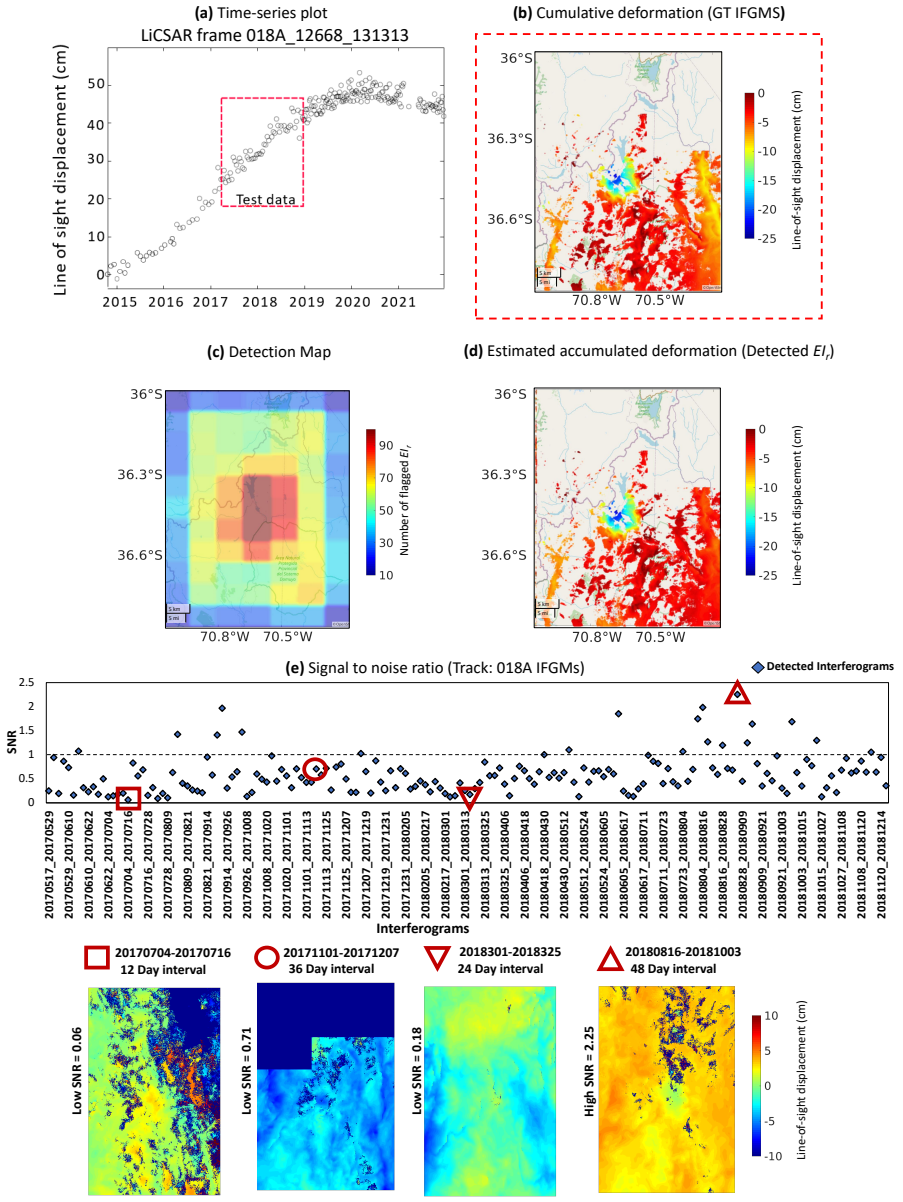

Figure 4.9: Result of anomaly detection when tested on a volcanic eruption in Domuyo. (a) Time series of displacements at Domuyo Volcano, made from LiCSBAS time series algorithm Morishita et al. (2020)) using LiCSAR interferograms (b) Cumulative deformation estimated from a stack of the displacement rates for all GT IFGMs, assuming a linear displacement rate. (c) Detection map showing flagged EI_r on a patch by patch basis (spatial and temporal sliding window for processing). (d) Final estimation of the spatial structure of deformation of all detected epoch intervals. (e) (Top) Signal to noise ratio, as estimated as the ratio of peak displacement to interferogram variance with deforming area masked, for the processed region of interest for all interferograms. (Bottom) Interferograms with lower SNR are displayed in comparison to the one with higher SNR (red triangle). All of which are detected by the model.

noise ratio (inverted triangle and square) and an example of a high signal-to-noise ratio image where the Domuyo displacements are visible (triangle). Although the training data set did include regions of poor coherence, it did not include data gaps as shown in Figure 4.9a (circle), so these are flagged as anomalies, leading to a high false positive rate.

The model not only accurately predicts the temporal extent of the volcano but also results show that it can be used to process larger regions and stitching the results together co-relates the spatio-temporal pattern of the large ground truth interferogram. The Figure 4.9 shows the visualizations of results when the volcano is tested. The bottom left corner of the interferogram of size 1408×896 pixels are processed, the centre of which is covered by the volcano, covering $\sim 13000km \times 11000km$ area on the ground. This region is divided into 60 overlapping patches of size 256×256 and each patch location is independently passed through the model and detection framework. The data temporally spans from 17^{th} May 2017 to 14^{th} December 2018 ($\sim$ 2 years), comprising 10 data batches with 44 epoch intervals for each patch location. The results show that the predictions are spatio-temporally consistent, despite the fact that the whole area is processed on a patch-by-patch basis in a sliding window scheme and stitched together in the end. The key benefit of the convolution neural network is translation invariance, which has proved to be beneficial in this case and has resulted in consistent output. This property of CNN is prompted by the spatial overlap in the training data set. The test set is full of missing data points resulting in a greater number of false positives. All the flagged epoch intervals of each patch are visualized on a heat map called the detection map (shown in Figure 4.9 (c)). The total number of epoch intervals is 44 but due to the overlapping nature of patches, each location is covered by at least 4 patches, except for the boundaries, hence epoch intervals are flagged by multiple patches. The most flagged region is enclosed by the orange-red intensity with most EI_r flagged in Figure 4.9 (c)). It performs well both for high and low signal to noise ratio interferograms (Figure 4.9 (e)), although large data gaps and unwrapping errors

result in false positives. These results demonstrate the transferability of training using data from Northern Turkey to a completely different geographic setting, with very different topography, vegetation, patterns and therefore interferogram noise. The estimation of the accumulated deformation from the automatic detection is very consistent with displacements estimated using conventional analysis methods such as 'stacking' (compare Figure 4.9 b and d).

4.4 Discussion

4.4.1 Training on the synthetic data set (background noise)

For anomaly detection framework, especially unsupervised, it is vital to create a training data set that entirely focuses on 'normality'. The probability of raw data containing errors is high. These errors can be due to multiple factors, for example, pixel-wise gaps, temporal errors/gaps, artefacts due to noise, low signal-to-noise ratio, etc. To further improve the understanding of normal data, so that anomalies can be flagged with high accuracy. It is important to train the models on a synthetic 'normal' data set to evaluate all models based on test results. For this, the synthetic data needs to be defined and created, so that the model can be trained on them.

4.4.2 Testing on descending frame

So far, I have evaluated my model on the basis of temporal dependency, area of anomaly and its type. The model accurately separates the anomaly from the atmospheric background noise that has been captured by the Sentinel-1 InSAR data. As elaborated in Section 1.2.2, the SAR acquisitions are made twice (day and night time). The model is trained on the data from Turkey that actually belongs to the data acquired at night time (Ascending frame). The look angle i.e. viewpoint and perspective of the satellite at night and daytime track are different because they travel in opposite directions (S to N for Ascending and N to S for

Descending). This makes the spatial patterns captured by the satellite appear at different pixel locations on the images, with different sizes and structures. Plus the participation of atmospheric noise is comparatively different for day and night times acquisitions. The transformation matrix between the two tracks can be defined by: Translation, Scale, Rotation, Shear and Reflection.

CNNs are expected to be both invariant and equivariant, invariance corresponds to ignoring the fact that a transformation has occurred, whereas equivariance relates to re-producing the same transformation (Mumuni and Mumuni (2021)). Invariance plays a major role in tasks like classification (but not limited to), where a score or label is to be produced for a whole image instead of pixel-wise output. Whereas equivariance majorly assists for tasks like dense segmentation or object reconstruction. Though these traits exist in the current layers of CNNs, research has been done to further improve them, for example (Dai et al. (2017) , Kayhan and Gemert (2020) ,Mumuni and Mumuni (2021)). The straightforward solution for this problem is training the models with augmented data, and different techniques for data augmentation have been introduced (Chen et al. (2020a)). The model understands the transformation while training and generalizes the whole data helping to predict the invariant output.

Due to the fact that the models are trained on Ascending data only, In order to check the 'generalization' of the model, it is important to test the anomalies captured during Descending tracks. The translation invariance property of CNN has already been proved by testing the Volcano at Domoyu (see Figure 4.9), which covered a greater area as compared to the size of patches I trained the model on. I seek to evaluate the impacts of processing an area from a Descending track in the following chapter.

4.5 Conclusion

In this chapter, I present a novel deep learning architecture and a unique, iterative training scheme of my model that successfully incorporates temporal consistency to learn the spatio-temporal patterns in InSAR data. The unsupervised and event agnostic property of the model is demonstrated in this chapter. Where both a tectonic ($\approx$ 5 to 6 cm) and a volcanic deformation ($\approx$ -25 to - 30 cm) is not only detected but its accurate spatial structure is also estimated by a single pipeline, that is trained on generic unwrapped interferograms, mostly consisting of background noise. Test results clear all doubts regarding the limitations of the model with respect to the size of patches it is trained on. In fact, the sliding window technique of processing large areas is proven to be advantageous. Because anomalous intervals can be flagged by multiple patch locations, or if missed by one can be flagged by others. This can be extended and refined by developing a voting system instead of taking a simple union for gathering the epoch intervals. The quality of data plays an important role both in the detection as well as in prediction of the spatial structure of the detected anomaly. For instance, if the output of the Turkey earthquake is compared with the Domuyo volcano, it can be seen that the spatial structure of the earthquake anomaly is clearer than the volcanic anomaly, despite the fact that the temporal signature of the volcano is much longer (i.e. accurately detected by the model). High atmospheric noise together with a greater number of missing pixel-wise information in the Domuyo data is reflected in the accumulated spatial structure of the anomaly. Signal to noise ratio (SNR) of the Domuyo data is plotted in Figure 4.9 (e). This ratio represents the variance of only the processed region of the frame (Figure 4.9(b)) verses the suspected volcanic region. More than 90% of the data have low SNR (below the dotted line plotted at SNR 1). The model successfully detects the anomaly even with SNR as low as 0.06, the red square (shown in the bottom of Figure 4.9 (e)) encapsulates this interferogram. The extent of missing pixel wise values can also be seen here, that exist consistently

around the anomalous region.

The proof-of-concept study in ALADDIn (Shakeel et al. (2022)) Chapter 3, demonstrated the efficient and accurate automatic analysis of unwrapped interferograms. Here I have investigated and eliminated the limitations of temporal dependency of my proof-of-concept. By introducing Temporal Self-attention, I have not only initiated temporal consistency in the results but also improved the capacity of detection (range of magnitude $\approx$ 4 cm and wavelength $\approx$ 700 m) and its overall accuracy. This has been evaluated by a synthetic test data set. Similarly, the results of 2 real scenarios (tectonic and volcanic) depict that it is capable of detecting anomalous deformation of different temporal length, spatial extent and magnitude. The output of the model is not only restricted to detection of anomalies, the predicted epoch time-series estimated by the model can be used for studying atmospheric and distinctive spatial-temporal patterns of crustal activities. The unsupervised efficient analysis is also useful for automatically labelling/categorizing anomalies. The nine year long data set from the start of Sentinal-1 mission till date (2014 to 2022) carries useful global information, that can be processed to categorize this data for future work.

Chapter 5

A performance evaluation and analysis of anomaly detection methods across training sets, architectures and events

In this chapter, I seek to improve detection accuracy by introducing training with a synthetic dataset in both deep learning models discussed in Chapter 3 and Chapter 4. I present my analysis by evaluating and comparing all models, based on the consistency of predicted spatio-temporal patterns; accuracy of epoch prediction (unknown before as no ground truth of epochs is available); rate of anomaly detection and the capacity of detection. Here, I also test my model's performance on a synthetic slow-slip event and a real tectonic activity (small earthquake and aseismic creeping) in Descending frames around the North Anatolian Fault in Turkey.

Some portion of the work presented in this chapter has been submitted to the Journal of Geophysical Research (JGR) of the American Geophysical Union (AGU). The text and figures from this submission are reproduced under a creative commons license and changes have been made accordingly for the work to better fit the thesis.

5.1 Introduction

The InSAR data creates unlabelled maps of surface deformation. This motion can be due to the 'normal' atmospheric signals the satellite is capturing or it can be a consequence of a tectonic activity like an 'anomaly' earthquake or an 'anomaly' volcanic eruption. Deformation occurred by both anomalous tectonic activity and the normal atmospheric signal may occur at the same time and location in the interferogram, overpowering the signal of interest (anomaly) and occurring with the vast majority in all of the InSAR data. The Sentinal-1 InSAR data provides interferograms with a minimum of a 6-day interval, which is the shortest interferogam, maximum can be of at least 24 days or more, as I create InSAR data networks of each epoch (*EP* single-acquisition) going four times forwards (as shown in Figure 3.1 and Figure 4.1). The anomalous deformation is best captured by the shortest interferogram spanning the event. Generally for an anomaly occurring on an instance of time, for example, the deformation resulted from the Turkey earthquake of M_w 5.7 that occurred on 20^{th} of March 2019 (Elliott (2020)) was captured only by 10 interferograms as compared to 125 normal atmospheric images (Shakeel et al. (2022)), as shown in Figure 4.8 (b). One of the challenges of this dataset is an imbalance in the number of data samples between normal and anomalous scenarios.

One of the major difficulties while finding a solution for anomaly detection lies in the availability of balanced data. An anomaly is an irregularity/abnormality that appears in the regular pattern of normal behaviour. The frequency of occurrence of these irregularities is by definition low as compared to the normality. For example, globally, 552 events (anomaly) that are classified as earthquakes occurred for the time span of 20 years from 2000 to 2019 (CRED). These 552 anomalous cases are very few in number (and are also unlabelled) as compared to the 20 years worth of data. This introduces a large bias in the dataset and makes it difficult to train deep learning models for classification purposes (Pang et al. (2021)). Deep learning models on the other hand need a balanced dataset for training; otherwise,

the convolutional layers may over-fit to cases in the dataset that are in majority (Yan et al. (2015)). Datasets for deep learning models are prepared to keep in mind the following: (a) the task it is being created to solve, (b) the total number of samples, (c) the total number of classes/categories, (d) the total number of samples per category and (d) diversity in the samples of each category (a variety of different examples belonging to a category). Points (c) and (d) do not apply in the case of anomaly detection, as there is only one category the model is trained on, which is the 'normal' class, so I aim to have as many cases of diverse atmospheric background noise as possible.

To get around this data imbalance, the classical way of training such models is used, i.e. to train on normal data only and understand the normal pattern only (Ye et al. (2021)), as implemented in Chapter 3 and Chapter 4. Other ways to train supervised anomaly detection models are to augment the anomalous data by synthesizing/faking labelled similar cases for validation or testing purposes (Lim et al. (2018)). But, these supervised models may not be able to generalise to unseen real anomalies that exhibit different aberrant traits from the labelled synthetic anomalies because they are only designed to detect the few identified anomalies.

Most of the available literature incorporates CNNs with InSAR data, train supervised models (Anantrasirichai et al. (2020); Sun et al. (2020); Rouet-Leduc et al. (2021)) or evaluate (Rouet-Leduc et al. (2021); Sun et al. (2020)) on labelled synthetic datasets. All these methods pose limitations like (I) model supervision, (II) model generalization on specific events and (III) extra data processing steps before testing through the model (Gaddes et al. (2019)) or for creating pixel-wise labeling (Anantrasirichai et al. (2018)). Model generalization here is referred to in terms of the extent of diversity in the training data, as the model while training will generalize only over samples that are in the training set. The general principle is to train such a model that can be accurately applied to various scenarios.

Instead of using synthetic anomalous samples either for training or for validation, I aim to design a synthetic training set particularly based on a 'normal' atmospheric

signal. The goal is to synthesize atmospheric behaviour that is solely established on the phase (ϕ) contributions from a planar ramp (ϕ_{ramp}, representing residual errors in the estimation of satellite orbits), the stratified troposphere (ϕ_{strat_atm}) and turbulent troposphere (ϕ_{turb_atm}, described in terms of maximum phase variance, $maxvar$ and characteristic length scale exponent, α). The objective behind this is to improve the understanding of atmospheric patterns by training my deep learning network architectures on this data, improving overall detection accuracy (OA), true positive rate (TPR) and accurately separating atmospheric signals from anomalous deformation.

5.1.1 Chapter overview

In this chapter, I introduce the issue of training data imbalance that exist in anomaly detection models in Section (5.1). Then I present the methodology to design synthetic training data in Section 5.2. Finally, I provide a comparative performance evaluation of all models and their synthetically trained counterparts in Section (5.3). Discussion in regards to the spatial extent and type (synthetic or real) of the training data is elaborated in detail in Section (5.5). In conclusion, I discuss the key contributions of this chapter in Section (5.6).

5.2 Training on synthetic data

The anomaly detection model is built by understanding the continuous background atmospheric noise (normality). So when an anomalous event (earthquake, volcano, etc) occurs, the model detects it with high error - as the model fails to understand it due to its anomalous nature. The data used to train the model is real InSAR data from a region of Turkey, that do not contain any anomalous activity but does contain data errors like unwrapping errors that introduce anomaly-like patterns in the IFGMs. Deep learning models have been presented in the past to pick unwrapping errors (Zhou et al. (2021); Sica et al. (2020); Wang et al. (2021)), ALADDIn

(Shakeel et al. (2022)) also detects these as an anomaly. The spatial data gaps and missing values for pixels also introduce artefacts in the data. In an attempt to further improve the detection accuracy of the method, a synthetic training data set is designed based on realistic background atmospheric noise. Synthetic datasets are commonly used to train deep learning models, as employing them overcomes the problem of data imbalance (Anantrasirichai et al. (2019)). I test the impact of using synthetic training data on the performance of the model. I expected that the addition of synthetic training data should reduce any impact of unwrapping and other processing-related errors.

5.2.1 Synthetic data preparation

I construct the synthetic interferograms by first generating synthetic epoch images from which to build them. The reason behind generating synthetic *EP*s instead of interferograms is to obtain ground truth *EP*s, so that the models can be evaluated based on quantitative judgements of *EP* prediction. This is not possible before as no ground truth of *EP*s are available for raw unwrapped interferograms. So, before the model's *EP* predictions are indirectly judged by the reconstruction of interferograms by them.

Our simple synthetic 'normal' (non-deforming) data set is made up from phase (ϕ) contributions from a planar ramp (ϕ_{ramp}, representing residual errors in estimation of satellite orbits), stratified troposphere (ϕ_{strat_atm}) and turbulent troposphere (ϕ_{turb_atm}, described in terms of maximum phase variance, $maxvar$ and characteristic length scale exponent α, similar to (Ebmeier (2016)), described as:

$$\begin{aligned} \phi_{ramp} &= aX + bY + c \\ \phi_{strat\_atm} &= kH \\ \phi_{turb\_atm} &= \sqrt{maxvar} * \exp(-r * \alpha) \\ \phi_{EP} &= \phi_{ramp} + \phi_{strat\_atm} + \phi_{turb\_atm} \end{aligned} \tag{5.1}$$

where X and Y are pixel locations and H is the elevation from the digital elevation model (DEM). The appropriate parameters are estimated using linear least square inversions (e.g., eq 4.2), where matrix d_{IFG} is the interferogram patch, reshaped as 65536 $\times$ 1 array, G is the design matrix (size: 65536 $\times$ 4) containing horizontal pixel location (X), vertical pixel location (Y), elevation (H) and ones (for constant c) and m is the desired output of 4 parameters (a,b,c,k). These values are computed for each patch location for all available time acquisitions of the Turkey data frame used for training. I estimate the parameters ($maxvar$,α) for (ϕ_{turb_atm}) using the residual interferogram after removal of $\phi_{strat\_atm} + \phi_{ramp}$:

$$R = d_{IFG} - G.m, \tag{5.2}$$

To generate the synthetic *EP*, a, b, k, c, max var and α are drawn randomly from the distribution of each of these parameters with mean = 0 and standard_deviation estimated their distribution in the training dataset. Because the variables are computed using interferograms, their sigma values are divided by $\sqrt{2}$ so that it can be used to draw a distribution for *EP* images. The created *EP* and interferograms are displayed in Figure 5.1.

We use the same network architecture described in Chapter 3 and Chapter 4, but synthetic data replaces the real data for training both the Bi-Deep model of ALADDIn called Synth-ALLADIn and the Temporal self-attention model called Synth-Temporal self-attention. The one main difference when using synthetic training data is in the estimation of the loss function. Previously, I have been using the predicted epochs to reconstruct interferograms to compute loss functions. Now, I use the aggregated loss of interferograms and epochs, to take advantage of having synthetic 'ground truth' of *EP* themselves (given in equation 5.3).

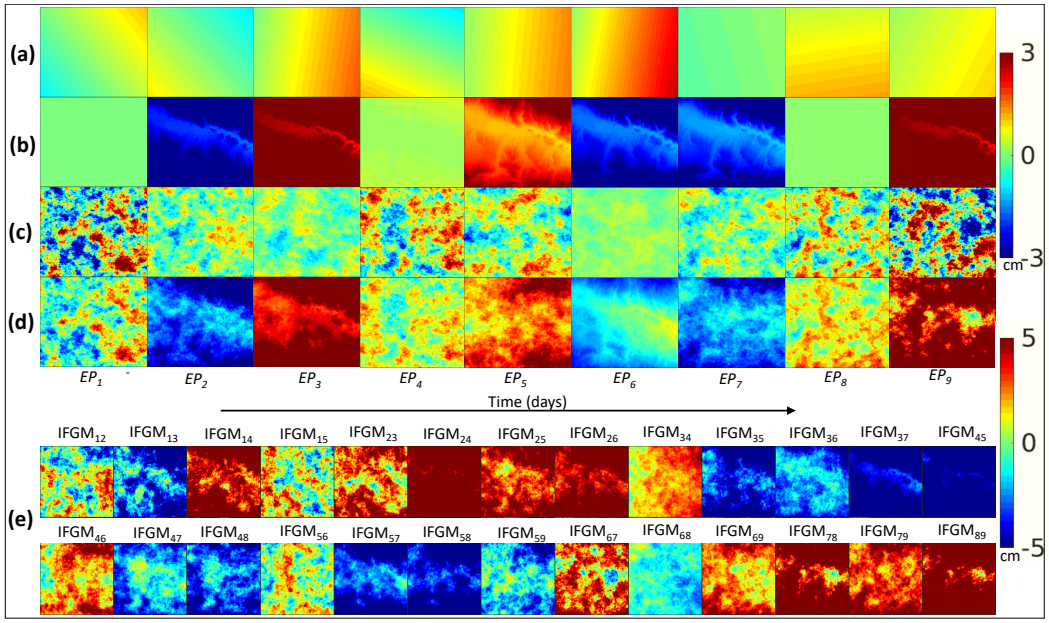

Figure 5.1: Visualization of the synthetic training data. (a) Planar ramp ($Component_A$), (b) Stratified tropospheric ($Component_B$), (c) turbulent tropospheric ($Component_C$), (d) synthetic 9 epoch generated by aggregating all components (eq 5.1). (e) 26 synthetic interferograms, following the data structure presented in the Figure,4.1 are made using the generated epochs shown in (d). The line-of-sight displacement is measured in cm on a scale of -5 to 5 cm for (d) and (e), whereas (a), (b) and (c) are measured on a scale of -3 to 3 cm.

$$
\begin{aligned}
Loss_{IFG} &= \sum_{i=1}^{n}(Output_{IFGM_i} - Input_{IFGM_i})^2 \\
Loss_{EP_1} &= \sum_{i=1}^{n}(5_{epochs_i^t} - 5_{epochs_i^{t-1}})^2 \\
Loss_{EP_2} &= \sum_{i=1}^{n}(Output_{EP_i} - Input_{EP_i})^2 \\
Loss_{ALADDIn} &= Loss_{IFG} + Loss_{EP_2} \\
Loss_{TSA} &= Loss_{IFG} + Loss_{EP_1} + Loss_{EP_2}
\end{aligned} \tag{5.3}
$$

where n is the number of interferograms in the case of $Loss_{IFG}$, it refers to the loss computed between reconstructed interferograms by the model and the input interferograms which is also the ground truth. $Loss_{EP_2}$ is the error computed between ground truth EPs and predicted EPs. $Loss_{EP_1}$ is the difference between the current predictions of overlapping epochs and the previous ones, so here n is 5, and

the accumulated loss is then backpropagated for Synth-Temporal self-attention network architecture. Whereas for Synth-ALADDIn network architecture the aggregated loss backpropagated is $Loss_{ALADDIn}$ for architecture. The model is trained using Keras (Chollet et al. (2015)) (a deep learning API) with the TensorFlow (*Abadiet al.* (2015)) backend. Due to the large size of the images in memory the batch size was set to 1. Adam optimizer (Kingma and Ba (2014)) was used with a learning rate of 0.00001. A lower learning rate gives the model a chance to learn features through steady changes in a loss instead of rapid fluctuations.

5.3 Performance evaluation across models

In order to select the best model out of (1) ALADDIn, (2) Temporal self-attention, (3) ALADDIn trained on synthetic data and (4) Temporal self-attention trained on synthetic data. I evaluate the performance of all four of them on the basis of (a) temporal consistency in overlapping data batches, and (b) the range of anomalies detectable in terms of spatial and temporal scale. It has already been established in Chapter 4, that the network architecture that is trained through the 'Temporal self-attention' technique is temporally consistent as compared to ALADDIn which estimates different spatial patterns for similar overlapping *EP*s. For the quantitative analysis, I evaluate all four models on a synthetic dataset based on 2D Gaussian test data (see Figure 4.5 in Section 4.3.1). And for qualitative analysis I use a real earthquake test case of M_w 5.7 in southern Turkey (Elliott (2020)).

5.3.1 Synthetic case study: 2D Gaussian-based synthetic test data

The 105 synthetic test cases are added to the atmospheric noise of Turkey test data, i.e. a testing patch from the frame: 014*A*_04939_131313, that is separated for testing before training. So, the noise in this data plus the synthetic anomaly added to it has never been seen before by any of the models. This test data spans from

28^{th} *July* 2017 to 18^{th} *April* 2018, with 170 interferograms that are divided into 10 data batches as per the required data structure of all four models as depicted in Figure 4.1. The anomalies are added in this data at 8 different temporal instances, totalling a ground truth (GT) true positives (TP) of 16 anomalies for each case when tested through (1) ALADDIn and (3) ALADDIn trained on synthetic. Where the GT (TP) totals 8 for models: (2) Temporal self-attention and (4) its synthetic counterpart. This change in values is due to the overlapping nature of input data and temporal consistency in the output of the models.

The anomalies created in this dataset vary from 1 cm to 11 cm for peak value that is represented as magnitude and the size of the anomaly varies from 10 m to 12 km represented as wavelength. This synthetic test data induces a strict competition between the peak value/magnitude of the anomaly and its size (wavelength of the anomaly), as shown in Figure 4.5. Though peak values much lower than $\approx$ 4 cm are successfully detected by the Temporal self-attention model while it was applied to the data containing volcanic eruption in Domoyu, Argentina (see Figure 4.9), here similar peak-values/magnitudes are difficult to get detected by the model due to low signal-to-noise ratio because of the extremely small size of the anomaly. The patch covers the area of 20.5 km $\times$ 20.5 km, for the cases of anomaly of peak-values $<$ 4 cm and wavelength $<$ $\approx$ 7 km, these cover $\approx$ 30% or less area on the patch and often dwarfed by low signal-to-noise ratio in the data.

The results displayed in Figure 5.2 (a) True positive rate (TPR) also called the Recall, (b) Mean accuracy, Mean False positive rate (FPR) and (c) Comparison of all models, proves that the Temporal self-attention model outperforms ALADDIn and its synthetically trained counterpart. Firstly, a similar pattern of output performance is followed when a change in technique takes place and when training data is replaced. A drastic increase in TPR from 0.34 (ALADDIn) to 0.51 (Temporal self-attention) is observed when the technique is changed. Similarly, for the synthetic counterparts, the TPR increases from 0.24 (Synth-ALADDIn) to 0.50 (Synthetic-Temporal self-attention). This trend can be seen in the comparison plot

of Figure 5.2 (c).

Here, the evaluation goals are to look for an increase in overall TPR (Recall) on this difficult synthetic dataset and lower FPR as much as possible without compromising the TPR ((Recall)). I have focused on the metric that is, Recall, which is a metric that measures the proportion of accurate positive predictions among all possible positive predictions. Overall accuracy (OA) computed using eq 4.5 is dependent not only on the true positives but also on the true negatives. Negative here corresponds to the atmospheric noise in the normal InSAR data. Greater false positives mean a lower number of true negatives, these can result due to any artefacts in the test data, extremely low signal-to-noise ratio and unwrapping errors. Hence, mathematically decreasing the overall accuracy. The extent of the model's capacity in terms of magnitude/peak-value of the anomaly is the poorest for ALADDIn, where it fails to detect anomalies the majority of the time for magnitudes $<$ than 4 cm. Whereas for the same magnitudes, it is greatest for the Temporal self-attention model where it competitively detects anomalies of varying sizes ($\approx$ 10 m to 12 km) with a mean TPR of $\approx$ 0.30, this is shown in plot (c) of Figure 5.2 where mean TPR (orange), mean FPR (yellow) and mean OA (blue) is plotted for all magnitudes.

The capacity of all models is presented by a pink polygon in Figure 5.2 (a) that covers the range of magnitude versus the range of wavelengths for which the model performs better ($>$ than 50%) in terms of TPR (Recall). Finally, the results prove that the technique of Temporal self-attention is best with a much higher TPR, but when the same technique is trained on synthetic data, it increases the FPR from 0.27 to 0.29 and decreases the TPR from 0.51 to 0.50. Similar trend can be seen in Table 5.1, where F_β score is considered to balance the affect of minimizing false positives (Precision) or minimizing false negative (maximizing true positive) (Recall). These two factors are combined with a β values of (1, 0.5 and 2) to compute the F_β scores. It is calculated using eq 5.4. When β is 1, where recall and precision are given equal importance, Temporal self-attention has the maximum

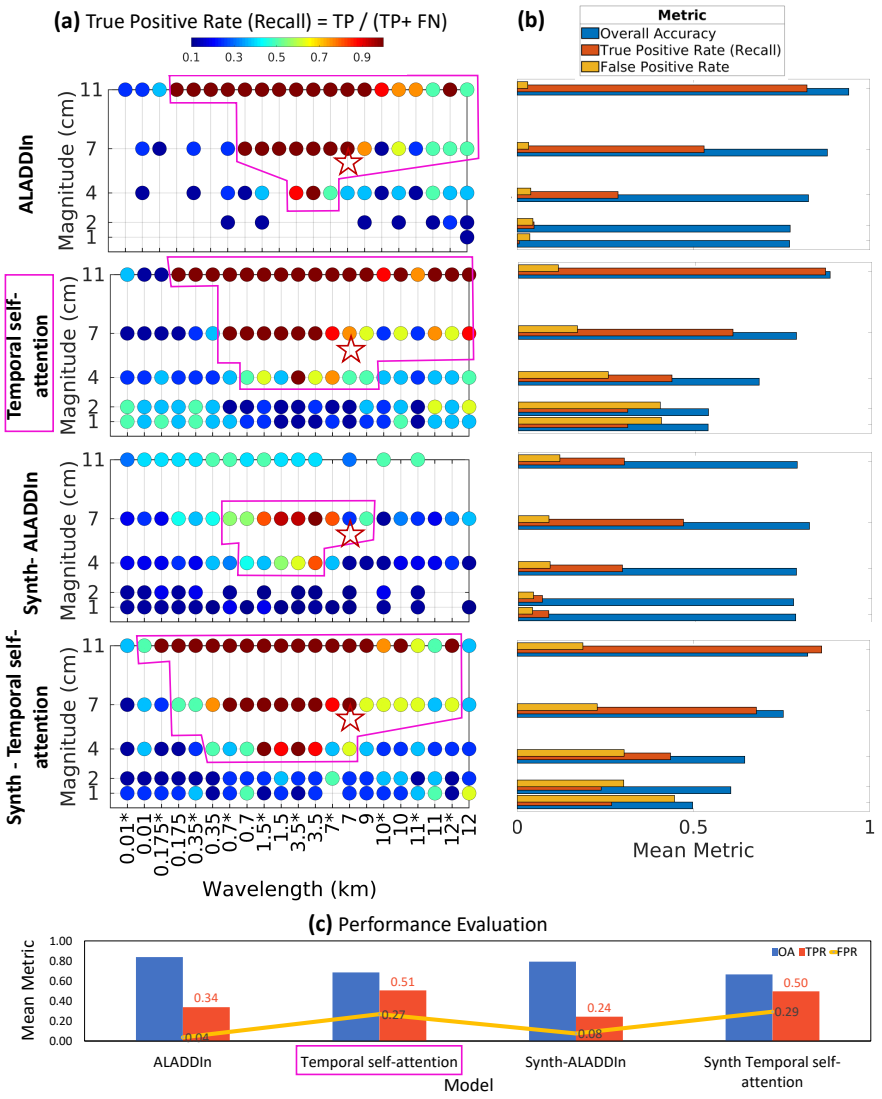

Figure 5.2: Performance evaluation and comparative analysis of all models. (a) Heat map plot of true positive rates (Recall) for all four models, ALADDIn (Shakeel et al. (2022)), Temporal self-attention, synthetic-ALADDIn (trained on synthetic data) and synthetic-Temporal self-attention (trained on synthetic data). SNR ranges from (0.0003 to 70) for 11 cm peak value and (0.0000003 to 1.7) for 1 cm. For 4 cm SNR ranges from (0.00004 to 11). Pink polygon on each of these plots display the region of accurate detection of each model. The x-axis corresponds to wavelengths starting from 10 m to 12 km, '*' represents the same wavelength but with different location (bottom left corner) on the patch. Y-axis corresponds to varying magnitudes from 1 cm to 11 cm. The red star marks the size of the Turkey earthquake shown in Figure 5.3. (b) Bar plots illustrating mean true positive rate also called Recall (orange), mean false positive rate (yellow) and mean overall accuracy (blue) for all four models. Note that the detectable displacement magnitudes quoted here are peak values in a dataset with a background of variance of 0.6 m, and are therefore very conservative.

Table 5.1: F_β Measure

Models	**F_β Score**		
	F_1	**$F_{0.5}$**	**F_2**
ALADDIn	0.36	**0.39**	0.34
Temporal self-attention	**0.41**	0.37	**0.45**
Synth-ALADDIn	0.26	0.30	0.24
Synth-Temporal self-attention	0.37	0.33	0.43

score of 0.41. But when β is decreased to 0.5, this mean that the measure is focusing more towards minimizing false positive then the score of ALADDIn is greatest with 0.39. Where as, when β is 2, that is giving more weight to the Recall then again Temporal self-attention has the maximum score of 0.45. The analysis proves that the Temporal self-attention model is best as it outperforms all others models in terms of balancing a high TP. I base my analysis emphasising on the true positives, because I don't want the model to miss any anomaly. This might increase the false positives but they can always be reviewed through human intervention. But missing an event that might be anomalous must not happen. Even though the change is slight, it nonetheless exists. For this reason, based on quantitative analysis, I decided to test the Temporal self-attention model in actual situations.

$$F_\beta = \frac{((1+\beta^2) \times Precision \times Recall)}{(\beta^2 \times Precision + Recall)}, \tag{5.4}$$

5.3.2 Real Case study: 2019 Earthquake in south-west Turkey

After the quantitative analysis, I test and evaluate all four models on the real earthquake of M_w 5.7 that occurred in Southern Turkey on 20^{th} $March$ 2019. In terms of temporal accuracy, meaning the flagging of correct epoch interval (EI) that captures the earthquake signal, all four models accurately detect the anomalous interval, which is the shortest interval/interferogram in the data 20190317 (17^{th} $March$ 2019) to 20190323 (23^{rd} $March$ 2019). Due to the overlapping nature

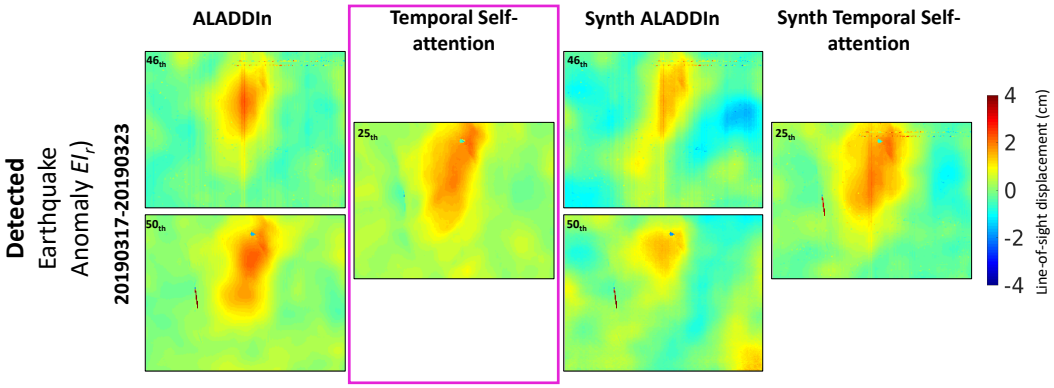

Figure 5.3: Test results of the real earthquake of all four models. The spatial structure of real earthquake anomaly captured in the detected residual-based epoch intervals (EI_r) of all four models are displayed.

of data, ALADDIn and its synthetically trained counterpart estimate two different realizations of the same earthquake signal. This is shown in Figure 5.3 (1^{st} and 2^{nd} column). The temporally independent model makes different predictions for the same overlapping input data and so the anomaly with different spatial structures is leaked in the residuals (RES) and similarly in the estimated residual epoch intervals (EI_r) that are passed on to the detection framework. Whereas the temporally consistent technique of Temporal self-attention and its synthetic counterpart accurately estimates a single realistic interval.

The remaining anomalous signal that is left in the RES interferograms, plays a vital role in the anomaly detection framework. As the more, the signal is captured in RES the stronger chance it has to be distinctive as compared to the 'normal' RES and then detected as an anomaly. Both, the Temporal self-attention and its synthetic counterpart release a much greater signal in terms of area and magnitude in the RES and so in the estimated EI_r. This is shown in Figure 5.3.

5.4 Results

After a competitive qualitative and quantitative evaluation and analysis of test results of all four models, I pick the Temporal self-attention model as the most

accurate one. So, this model is further tested and its output is analysed for a synthetic slow slip event and for data created from Descending frame that captures deformation around the North Anatolian Fault in Turkey.

5.4.1 Synthetic case study: Synthetic slow slip event

When a fault ruptures at a high velocity (several km/s), an earthquake occurs and seismic waves are generated. Faults typically experience repeated semi-periodic failure in earthquakes as part of the earthquake cycle. The earthquake cycle consists of three phases, the inter-seismic, co-seismic, and post-seismic phases. During the inter-seismic phase, elastic strain accumulates in the shallow crust as the fault is 'locked' by friction. This accumulated strain is then released in the co-seismic phase when the fault ruptures seismically. Finally, stress concentrations caused by the earthquake are alleviated by slow, transient slip on the fault and by other aseismic viscous processes during the post-seismic phase. However, in some instances, certain faults may also release strain slowly, by rupturing over hours-weeks rather than seconds during a normal earthquake, this type of event is called a silent or slow earthquake and the energy released during these events is hard to detect by a seismometer (Dragert et al. (2001)). However, satellite radar (InSAR) observations can be used instead to detect these slow slip events, by instead measuring the small-magnitude ground deformation they cause. (Nissen et al. (2011); Nippress et al. (2017)).

5.4.1.1 Synthetic test data preparation

One such slow slip event is synthesized using Okada's model (Okada (1992)). Deformation maps of patch size 256 × 256 pixels and 20.5 km × 20.5 km are generated due to strike-slipping fault and converted to displacements in the satellite's line-of-sight with parameters similar to (Rousset et al. (2016); Rouet-Leduc et al. (2021)), as given in the Table 5.2. The maximum slip value of 0.0002 m is exponentially

varied to create a slow-deforming event that slips gradually over a period of seventeen months with a minimum difference of 6 days between each *EP*. The data covers a total of 350 interferograms, made up of 85 *EP*s starting from 5^{th} January 2017 to 30^{th} May 2018. For each *EP*, the slip value is computed using the equation 5.5:

$$slip = \frac{S}{1 + e^{-a \cdot (EP - C)}}, \tag{5.5}$$

where S is the maximum slip value, i.e, 0.0002 m, a is 0.2 is a constant value of the rate of change or the size of gradient, it decides how long the function will take to reach peak value. *EP* is the time for which the slip needs to be calculated (1 to 85) and C is a constant the value of which is 40, it measures the mid-point of the exponential curve. The graphs that plot equation 5.5 along with the line-of-sight (LOS) displacement in mm is shown on the secondary axis in Figure 5.4 (a). For background atmospheric noise, interferograms from the patch of Turkey test data are used (see Figure 3.3 (b)), which are also employed to synthesize 2D gaussian test data. Each LOS displacement map is then added to the interferogram that covers its temporal time-stamp.

Table 5.2: Parameters for synthetic right-lateral on North Anatolian Fault, using Okada's model (Okada (1992))

Parameter	Value (Units)
Depth	3 (km)
Strike	90 (degrees)
Dip	0 (degrees)
Length	8 (km)
Width	3 (km)
Rake	170 (degrees)
Slip	0.0002 (m)
Open	0

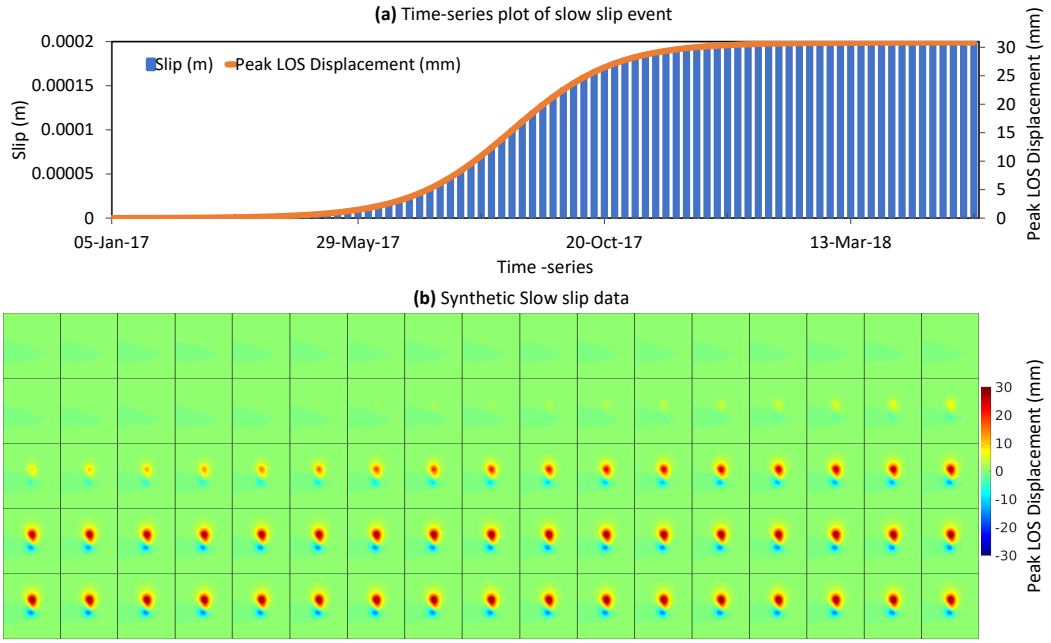

Figure 5.4: Visualization of the synthetic slow slip data. (a) Time-series plot of slip values and corresponding peak line-of-sight of displacement in mm is plotted together over a period of 17 months from 5^{th} January 2017 to 30^{th} May 2018. (b) Deformation maps for each time are shown here on a scale of -30 to 30 mm. These deformation maps are then added to relative interferograms.

5.4.1.2 Results

The data is passed through the Temporal self-attention model and testing procedures explained in Chapter 4 for the detection of anomalies. 84 EI_r are created and variogram analysis and DBSCAN clustering is applied. The attributes of this synthetic test data are known beforehand so, the ground truth temporal and spatial pattern of this anomaly is noted. I have marked EPs starting from 30^{th} March 2017 as anomalous as can be seen from the plot in Figure 5.4 (b), the deformation at this point is 0.2 mm. This makes 70 continuously anomalous out of 84 EI. The model successfully detects 66 EIs out of which 56 are true positive (TP) and 10 false positive (FP). The model is able to detect anomalous deformation starting from LOS displacement of 11 mm without considering the FPs. The model achieves an overall accuracy (OA) of 71% and a true positive rate (TPR) of 80%. The confusion matrix can be seen in Table 5.3. The estimated spatial structure of this continuous

anomaly is displayed in Figure 5.5 (b). The anomalous signal estimated by the model's response is dwarfed by the surrounding atmospheric noise.

Table 5.3: Confusion Matrix for synthetic slow slip test data

	Actual Anomaly	Actual Normal
Predicted Anomaly	TP = 56	FP = 10
Predicted Normal	FN = 14	TN = 4

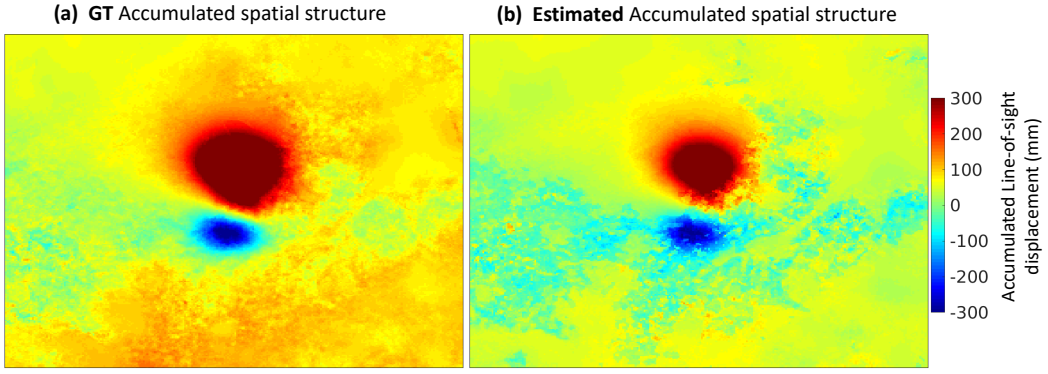

Figure 5.5: Comparison of estimated spatial structure. (b) By the model's detected 66 anomalous EI_r is shown in comparison to the actual/ground truth (GT) accumulated spatial structure (a) of the anomaly.

As the anomaly is small in magnitude (mm to few cm) and size (few km), it is difficult for the detection framework to distinguish noise covering the whole area and anomalous signal covering a few km on the patch. Hence, causing false positives. The false positives are basically those interferograms, that have a high range of missing pixel values (NaNs in raw data that are replaced by zero for processing) and have high atmospheric background noise. These artefacts are misinterpreted as anomalous signals and so, flagged as anomalies. Figure 5.6 and Figure 5.7 visualizes the half of 330 ground truth (GT) interferograms, predicted (PRED) interferograms and residual (RES) interferograms with corresponding signal-to-noise ratio SNR for 'Forward' and 'Backward' input data that is processed. It can be seen that the millimetre-scale deformation is captured by the RES interferograms. But, it is hard to quantify the small region of an anomaly as compared to the high atmospheric background that is too leaking into the residuals and overpowering it. The black box covering the rows of data in the forwarding pass in Figure

5.6 and similarly in the backward pass in Figure 5.7 displays the impact of large atmospheric noise (fairly low SNR) and a greater number of pixels with missing data, making it harder for the model to detect the anomalous signal. The RES interferograms exhibit this pattern, as for such cases no or very little anomalous signal leaks in the RES. The pink box shows the interferogram samples with a high signal-to-noise ratio and no pixels with missing data, for these cases, a significant quantity of anomalous signal is leaked in the RES and detected as anomalies for both forward and backward passes in Figure 5.6 and 5.7.

Despite all these factors, the Temporal Self-attention architectures successfully manage to capture millimetre-scale deformation ($\approx$ 11 mm and beyond) in an unsupervised manner using raw unwrapped interferograms. (Rouet-Leduc et al. (2021)) also extract similar millimetre-scale deformation but through a supervised model, that is specifically tasked to predict accumulated deformation of the input time-series and trained to labelled synthetic training data.

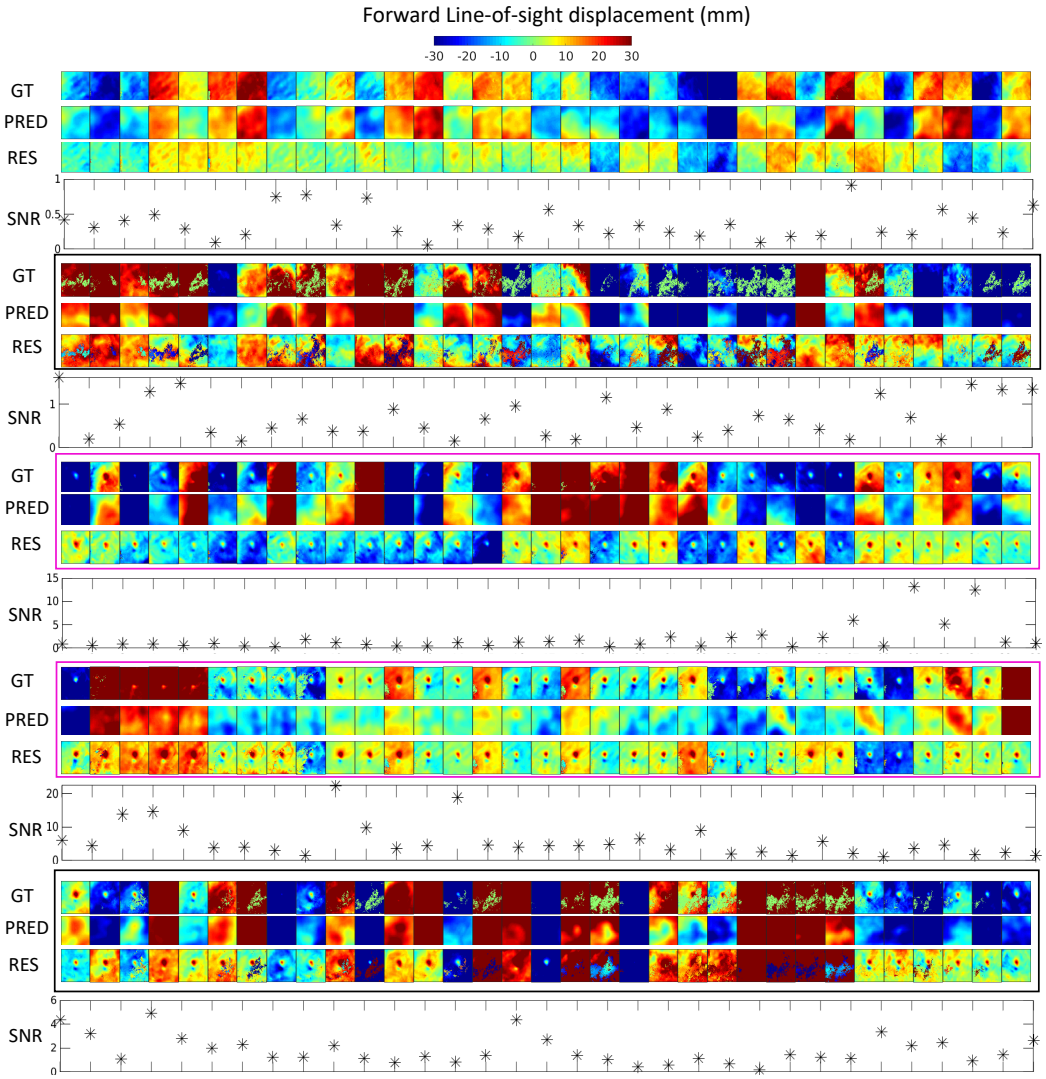

Figure 5.6: Model predictions for 2^{nd} shuttling iteration of **forward** slow-slip data. Half of the 330 interferograms are shown here. GT corresponds to model inputs or ground truth interferogram, PRED is the model predictions that are re-constructed using model predicted *EP*s and RES are the residual interferograms, computed by subtracting GT and PRED. These RES images are then processed by the detection framework. It can be seen that the millimetre-scale deformation is captured in Residual, though the size of the synthetic anomaly is small. Black boxes encapsulate the images that have greater noise, both atmospheric and in terms of missing pixel-wise data, depicted in the signal-to-noise ratio that is plotted for every interferogram. Whereas pink boxes cover the set of interferograms with a balanced signal-to-noise ratio.

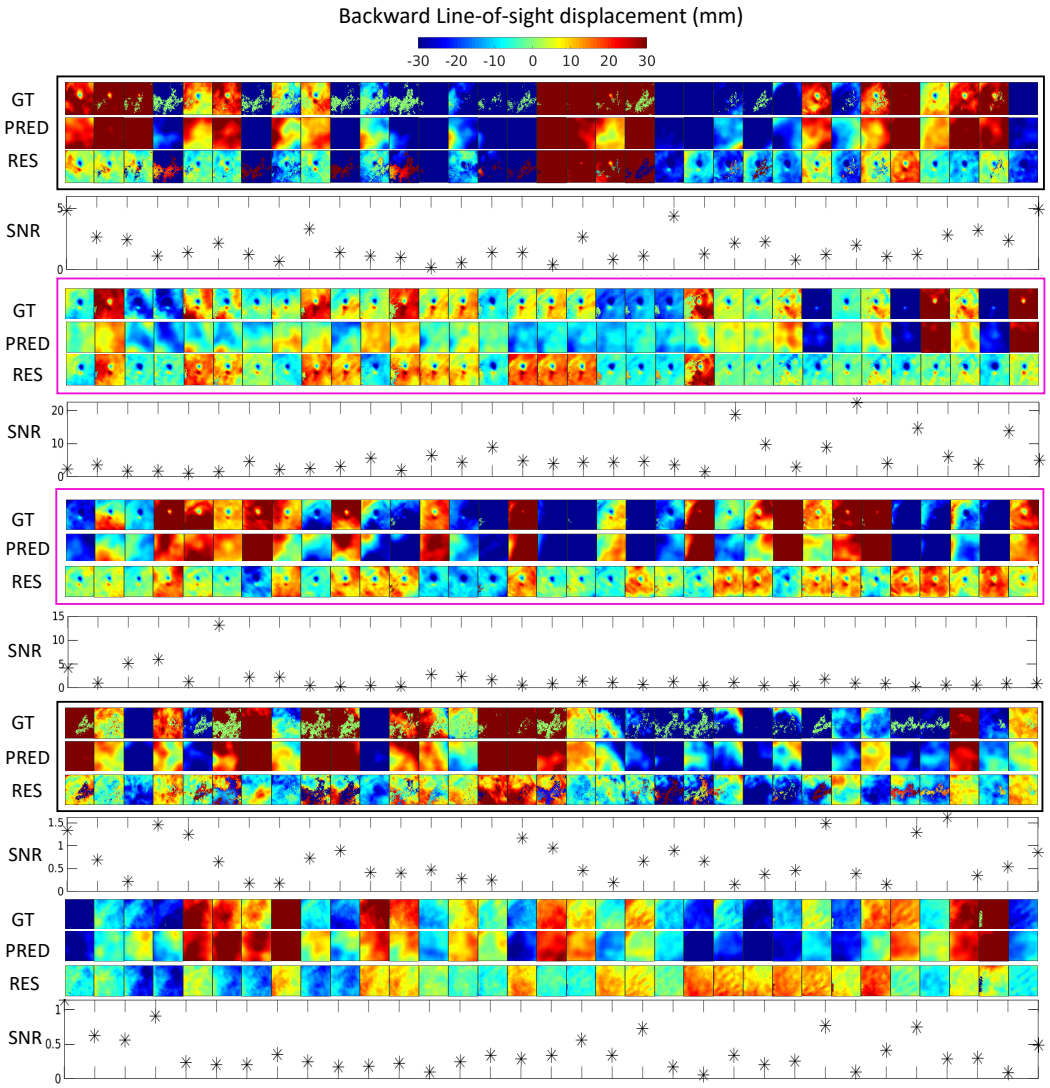

Figure 5.7: Model predictions for 1^{st} shuttling iteration of **backward** slow-slip data. Similar patterns in the output of the model and signal-to-noise ratio are followed when the Model processes for 1^{st} shuttling iteration of **backward** slow-slip data. Half of the 330 interferograms are shown here. GT corresponds to model inputs or ground truth interferogram, PRED is the model predictions that are reconstructed using model predicted *EP*s and RES are the residual interferograms, computed by subtracting GT and PRED. These RES images are then processed by the detection framework. It can be seen that the millimetre-scale deformation is captured in Residual, though the size of the synthetic anomaly is small. Black boxes encapsulate the images that have greater noise, both atmospheric and in terms of missing pixel-wise data. Whereas pink boxes cover the set of interferograms with a balanced signal-to-noise ratio.

5.4.2 Real Case study: Descending frames in Turkey, across the NAF

To evaluate the model's sensitivity on the basis of satellite look direction, I test the model on interferograms that are captured during a 'Descending' data (N to S, daytime pass). As the model is trained only on 'Ascending' data (S to N, nighttime pass), it is important to verify and interrogate its output on the former. For this purpose, I organized unwrapped interferograms from 4^{th} *January* 2018 to 8^{th} *August* 2018 (125 interferograms in total) around the North Anatolian Fault (NAF) in northern Turkey. This plate boundary experiences converging motion of the Anatolian and Eurasian plates and large earthquakes of M_w 7 and greater have occurred around it over the past century (Stein et al. (1997)). The fault experiences transient aseismic slip episodes and has been slipping aseismically since 1994, M_w 7.3 earthquake (Ambraseys (1970); Altay (1991); Bilham et al. (2016)). Similar regions around the NAF are analyzed (Rouet-Leduc et al. (2021)) and millimetre-scale ($\approx 5\ mm$) deformation is extracted during a slow earthquake in 2013. Similarly, I test $28,865\ km \times 39,360\ km$ region of this area that covers part of NAF. This region of interest and six to seven months long time-series also contains an earthquake of M_w 4.7 that occurred on 7^{th} of *April* 2018, 12 km North of Bolu, Turkey (USGS earthquake catalogue). The region around Bolu, Turkey has experienced similar slow earthquakes over the past years.

As per the model's requirement, each *EP* is connected 4 times forward and backwards with a minimum 6-day gap, with each data batch containing 26 interferograms (as shown in the Figure 4.1). If these *EP* connections are broken, for instance, instead of 4 forward and backward connections, there are three or two. Then, it would not be possible to construct a data batch of 26 interferograms from 9 corresponding *EP*s. The input and output nodes of the model, especially the number of interferograms and *EP*s cannot be changed. A similar scenario occurs here, the *EP*s in this data are strictly connected to 3 forward *EP*s only instead

of 4. I solve this problem by constructing the 4^{th} interferogram using the existing images in the data. The data is constructed by applying the following equation 5.6:

$$IFGM_{14} = IFGM_{12} + IFGM_{23} + IFGM_{34}, \tag{5.6}$$

All these missing interferograms are constructed and the dataset is organized with 32 *EP*s and 125 interferograms. These are further divided into 7 temporal data batches. The region is covered by 140 spatial patch locations out of which 17 patches are ignored that cover the Black Sea.

5.4.2.1 Results

The model accurately detects the small earthquake in the *EI*: 4^{th} to 16^{th} *April* 2018 in the descending frame. But due to missing pixel values (as shown in the Figure 5.8) across the earthquake region, the spatial structure of the earthquake anomaly is not estimated. The results from this case demonstrate the diversity of this model and the ability of transformation invariance that is possessed by the convolutional layers in the model. The black star in Figures 5.9 (a), (b) and (c) represents the location of the earthquake as per the USGS. The detection map in 5.9 (b) acts as a heatmap of deformation around the earthquake region, with most flagged EI_r. All flagged intervals from all of the patch locations are processed together to estimate the predicted accumulated spatial pattern of the deformation. This is shown in 5.9 (c).

This spatial pattern of earthquake anomaly is compared within patches of all of the ground truth interferograms capturing it. The comparison is displayed in Figure 5.8. It can be seen that the anomalous small signal is overpowered by high atmospheric noise and missing pixel values. High atmospheric noise is due to the fact that the area is very near to the sea. Despite the extremely low signal-to-noise ratio, the model accurately detects the interseismic, millimetre-scale deformation centred around the fault, that too in a descending frame, a look-direction never seen by the

model. A high false positive rate is observed in this test example mainly because of the strong edge contained in the images, because of the boundary between the sea and the land. Also, the high false positive is due to high atmospheric noise and unwrapping errors.

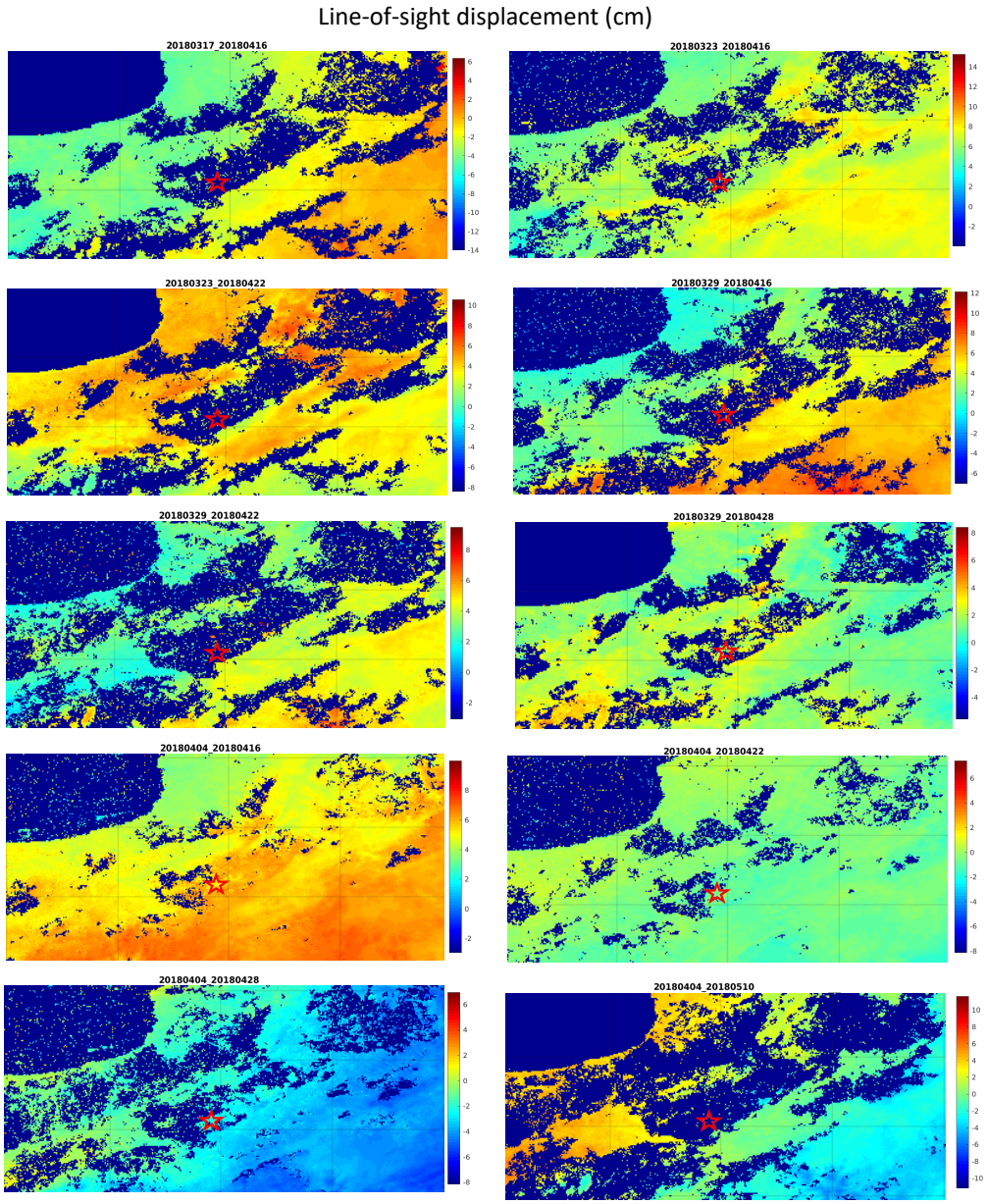

Figure 5.8: Ten interferograms covering the M_w 4.7 earthquake that occurred on 7^{th} of *April* 2018. The earthquake location is represented by the red star and the displacement is measured in cm. The earthquake signal is completely missed by the satellite and is covered by 'NaN' or missing pixel values.

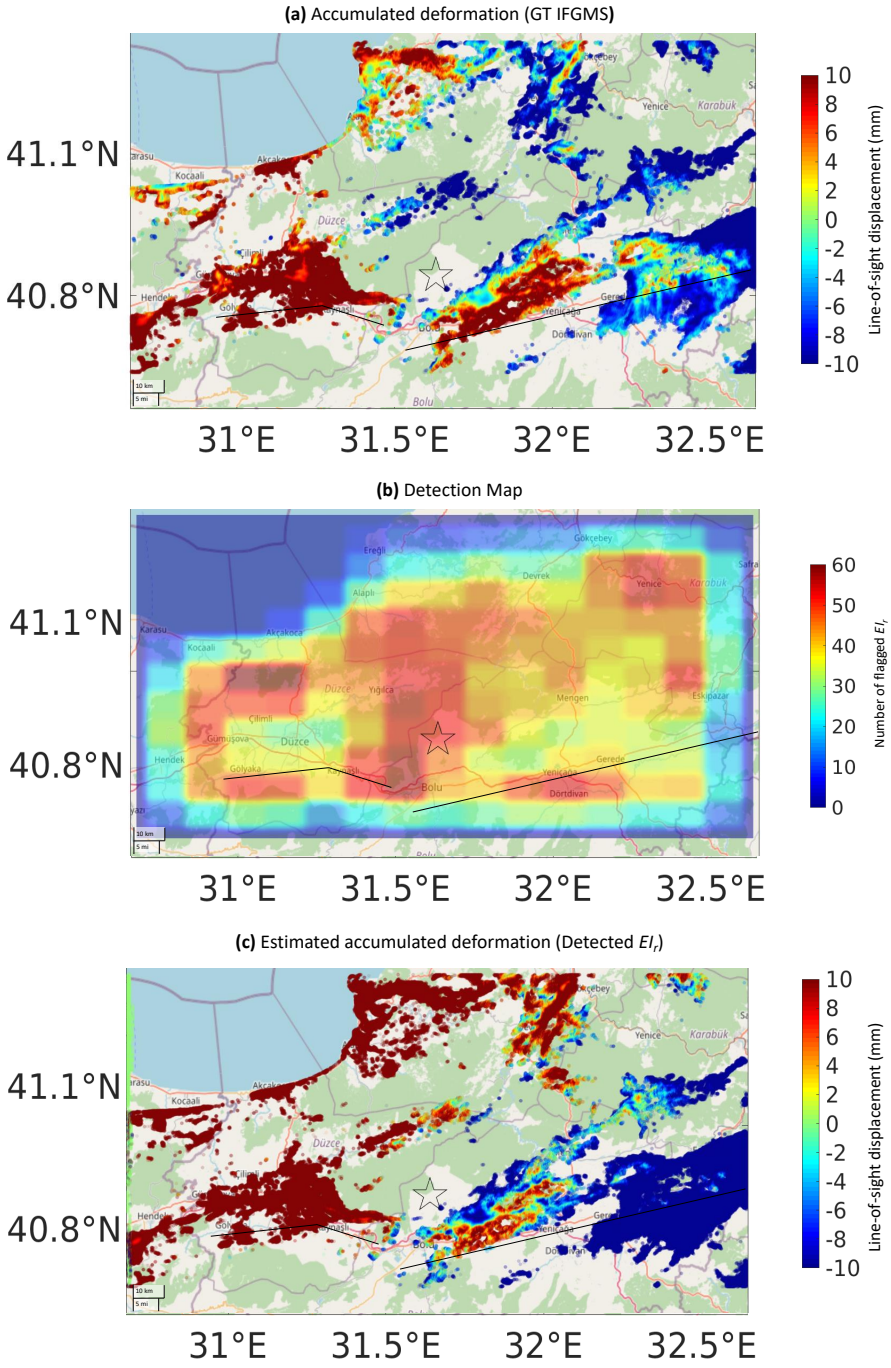

Figure 5.9: Results of descending frame around the NAF. (a) Shows the ground truth accumulated deformation. The accumulated deformation is computed by adding the rate of change captured in ground truth interferograms (b) Shows the detection map of the patch-wise anomaly detector. And (c) is the estimated deformation of all flagged intervals. The faults are shown in black lines plotted over (a), (b) and (c). The active faults are not drawn to scale.

5.5 Discussion

5.5.1 Selection of training data

The primary step for the application of deep learning models is the preparation of datasets. Which is further divided into three sets: training, validation and testing. Training data is the most important part of the whole learning process. As the distribution of samples in the training set is reflected in the training of the model. So, it is important to create a large, diverse, balanced dataset to be utilized for different tasks. For example, the ImageNet (Deng et al. (2009)), this data contains at least 1000 natural images for each category and subcategory, where there are 20000 categories, totalling up to 14 Million images for object recognition. Features learned by training the model on ImageNet are called 'general-purpose features that can be adapted later to solve other tasks through transfer learning (Huh et al. (2016)). And the end goal of training a hefty, deep learning model is that the features learned are transferable and could be repurposed for various tasks.

The preparation of labelled data is a laborious task. In most cases the labelled data is not readily available, for example, medical research data is not accessible for privacy issues and so, synthetic data is used (Benaim et al. (2020)). Similarly, for InSAR, it is strenuous to maintain pixel-wise labels for the global and temporal data sets.

5.5.1.1 Real or synthetic?

The gap that synthetic data bridges is massive, to collect, clean and mark real/raw or natural data is strenuous. On the other hand, it is easy to computerize the generation of large synthetic data and its labelling. Improving the training set both in numbers and quality directly affects the performance of the deep learning model. (Anantrasirichai et al. (2019)), improves the accuracy of detection of volcanic deformation by combining synthetic datasets in the training. Deep learning

models, specifically Generative adversarial nets (GANs) (Goodfellow et al. (2014)) are widely employed to generate realistic synthetic data, to mention a few, (Frid-Adar et al. (2018); Esteban et al. (2017)). In Chapter 4, I utilized the weights learned by ALADDIn (Shakeel et al. (2022)) and polished my dataset through its predictions, which are used for training Temporal self-attention. This removed the artefacts, data errors and missing pixel values in it, as the predictions made by ALADDIn are spatially relevant within a patch and only lacked temporal dependency.

In this thesis, the motive behind training on synthetic data is to generalize the model solely on background atmospheric noise. Real/raw interferograms across the globe are readily available, but raw data comes up with the cons of data errors, missing values, errors introduced due to noise corrections, etc. All of these may be detected as anomalies and in reality a false positive. So, to improve detection accuracy I too created synthetic data, as shown in section 5.2. However, this neither drastically improved nor worsened my results. It performed similarly to the models that are trained on raw unwrapped interferograms.

Initially, it was challenging to train similar network architectures on synthetic data. The problem of model over-fitting very early in training time hindered its learning. The network architectures are designed to cater for the raw unwrapped interferogram, the deformation values of which varied in a wide range. Whereas, the range of values in the synthetic dataset is tight and small-scale, not diverse as compared to raw data. To solve for over-fitting, (1) I lowered the learning rate, to give the model some time to converge before over-fitting, (2) introduced another loss function, so that the weights can be updated by a large number and (3) increased the number of training samples. The models learned the simplified version of background atmospheric noise and resulted in high FPR with lower TPR.

The end goal of a synthetically trained model is that it must accurately perform on real data. For, the case of supervised tasks, it is advantageous to generate large, labelled synthetic data for training. But, for unsupervised tasks, the images

in the data are the only source of information. The model is going to converge on the distribution of data, rather than the combined effect of categorizing it as well. So, I observed in this thesis, that training on raw data is useful and the features learned through this type of training are transferable and diverse. Transfer learning is applied using such features that are trained by ALADDIn (Shakeel et al. (2022)), this has significantly improved the detection. Though a mixed set of training data could have been generated by randomly selecting samples from both real and synthetic datasets. This was shadowed by the fact that I generated synthetic data using phase, stratified and turbulent tropospheric contributions from the real interferograms (as explained in Section 5.2.1), making them similar to the background atmospheric noise that is in real dataset. But actually the spatial patterns computed in the synthetic dataset are evenly spread (smoothed) with wavelength that is small in comparison to real data, leading the models to over-fit.

5.5.1.2 The spatial extent of training data

The size of images in the training data is another important aspect to look into before designing and training deep learning networks. As the pixel width and height of an image coordinate with the convolutional layer in terms of the size of the filter and parameters like stride and padding, this finalizes the area covered in an image by the CNN while training. This concept is often referred to as the receptive field of CNN. Also, the greater the image, the more training time is required because the number of pixel-wise computations increases and most of the time, to cater for a larger image size the depth of the model is increased. Image resolution is also an important aspect to consider, especially when dealing with satellite imagery or images taken from above (aerial imagery). So, the optimum size of the input image is decided, based on available computation power, information that is captured within that size, and the depth of CNN and its receptive field respectively.

In this thesis, I have selected the image size of 256 $\times$ 256, the resolution of which is 20.5 km $\times$ 20.5 km. This size is reasonable in terms of capturing the most

anomalous signals. I am unable to increase this size due to computational reasons because here I am dealing with a 2D temporal sequence. Research has been done in the past and features from different levels and different convolutions are merged in a pyramid of features to make predictions for object detection (Lin et al. (2017)). Similarly, multiple convolutional filters are applied at the same level but with different parameters (kernel size, padding and stride) to capture and learn from various receptive fields (Liu et al. (2018)). The network architectures designed in this thesis can be extended for pyramid feature merging techniques, this will help in the prediction of high wavelength signals that are often missed by the models due to comparatively small image size.

5.5.2 Data augmentation

Due to insufficient training data, deep learning models find it difficult to converge and instead overfit the data. One of the solutions to this problem is data augmentation. Traditionally, this is done by applying multiple transformations (affine, elastic, etc) on each input image and augmenting transformed variants as individuals. This is the simplest way of scaling one's dataset. Deep learning, itself is employed to generate and augment the transformed data, using GANs (Goodfellow et al. (2014)). Also, the augmentation is learned during the training and generated samples are feedback as inputs (Perez and Wang (2017)). To generate variants of images, neural style transfer (Gatys et al. (2015)) can also be applied to output images containing similar content but different artistic features.

The data that is augmented, is not completely different from the original images. It is either enhanced or modified but carries a similar context. So, the modified images cannot be referred to as synthetic data. The InSAR data, I am working with does not contain an object but instead a pattern that can be re-created as synthetic data. In this thesis, I have created synthetic training data that realistically match the behaviours of background atmospheric noise. The fraction of this synthetic data set can be augmented with the raw interferograms for training and it can help

the model to detect anomalies with improved performance. With InSAR data, the descending and ascending frames can be augmented together to increase training samples. As both cover and carry similar geolocation and patterns but with a different angles.

5.6 Conclusion

In this chapter, I create a synthetic dataset containing background atmospheric noise. The data is created with an intention to improve performance. Here, I evaluate the models based on the techniques (ALADDIn and Temporal self-attention) the network architectures are designed on and the training data (real interferograms and synthetic data) they are trained on. The temporal self-attention model trained on real data is proved to be the most accurate, with greater capacity of detection (range of magnitude and wavelength) and highest TPR. The quantitative test results are based on a difficult 2D Gaussian based synthetic test dataset. I test all models on real earthquake anomaly, that occurred in Southern Turkey in 20^{th} $March$ 2019. The results in this case are judged on the basis of the intensity and size of anomalous signal that has been leaked in the residual epoch intervals EI_r, as this forms the basis of detection framework.

The performance-wise best model is then further evaluated on a synthetic slow-slip event added in noisy time-series of Turkey. The Temporal self-attention model accurately detects millimeter-scale deformation and flags anomalously deforming intervals with very low signal-to-noise ratio ($\approx$ 0 to 0.02). Ideally, it is expected that the spatial pattern of anomaly should be captured by InSAR. But due to unpredictable and uncertain data of the surrounding region of NAF, the model predicts false positives.

Chapter 6

Discussion

6.1 Synopsis

The aim of my thesis was to allow quick and accurate analysis of the large-scale InSAR data to understand ground deformation. Specifically:

- Design and development of an unsupervised deep learning model, that understands and learns from the spatio-temporal patterns.
- Training a generalized model, that is capable to be tested globally for all sorts of tectonic and volcanic processes.
- Detect in time and estimate the accurate spatial structure of anomaly.

Deep learning methods are rapidly being developed for InSAR data. When I began this project, there was only one method that employed deep learning for the classification of volcanoes Anantrasirichai et al. (2018). I wanted to develop an unsupervised method, specifically suited for the unlabelled and large InSAR data capable of detecting various deformation scenarios. In Chapter 3, I presented the proof-of-concept of the unsupervised model, which is trained on normal data, organized from unwrapped interferograms of an ascending frame of Turkey. This method is called 'ALADDIn' and it is tested on a 2D Gaussian based synthetic

data and demonstrated equally accurate results on a real earthquake of M_w 5.7 that occurred in Southern Turkey on 20^{th} of $March$ 2019. On further testing, I observed a lack of temporal dependence in the predictions of this model because it did not allow the possibility of processing a long time-series altogether. The computationally feasible way to overcome this is to create overlapping chunks and consecutively process them through the model. Despite the overlapping nature of the data batches, the model must predict similar spatial structures for overlapping temporal acquisitions. This is incorporated into the model through a technique called 'Temporal self-attention', presented in Chapter 4. This unique, state-of-the-art method, re-purposed the training and predictions of 'ALADDIn' and sequentially minimized the error between overlapping acquisitions while training. This new method demonstrated its accurate analysis of a volcanic deformation in Domoyu, Argentina from 17^{th} May 2017 to 14^{th} $December$ 2018 and the M_w 5.7 earthquake in Turkey that is tested before.

I then intended to further improve the detection of anomalies by training on synthetic data, that I created with only background atmospheric noise. I trained both of the models and a detailed comparative analysis is provided in Chapter 5. The best model is selected, based on TPR, FPR and OA of results on a complex 2D Gaussian based synthetic data with a signal-to-noise ratio < 0.1. 'Temporal self-attention' trained on raw unwrapped interferograms outperformed all other models for various reasons, which are elaborated in section 5.3. The choice is also made based on the computational time required for each technique. These processing times and test data details are given in Table 6.1. It showed, that the 'Temporal self-attention' method required $\approx$ half the processing time as compared to ALADDIn because here, there are no multiple predictions of overlapping data.

Finally, I test this model rigorously on a synthetic slow-slip event (millimeter-scale deformation) and on a descending LiCS frame capturing deformation around the NAF, Turkey from 4^{th} $January$ 2018 to 8^{th} $August$ 2018, where a small earthquake of M_w 4.7 occurred Bolu, Turkey on the 7^{th} of $March$ 2018.

Table 6.1: Dataset details and total time consumed

[HTML]DAE8FC**Dataset Details**

	Earthquake	**2D Gaussian Synth Data**	**Volcano**	**Synth Slow-Slip**	**NAF Descending Frame**
Number of patch location	1	105	60	1	123
Number of data batches	7	10	10	20	7

[HTML]FFFFC7**Time Required (minutes per patch location) *overhead of shuttling added**

Method	ALAD-DIn	Temp self-attention	ALAD-DIn	Temp self-attention	ALAD-DIn	Temp self-attention	ALAD-DIn	Temp self-attention	ALAD-DIn	Temp self-attention
Deep learning model	0.08	0.08	0.1	0.1	0.1	0.1	0.3	0.3	0.08	0.08
Least square inversion	3.5	2	5	2.75	5	2.75	5.25	10	3.5	2
Variogram analysis	39	22	55	30	55	30	110	58	39	22
Clustering analysis	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.02	0.02
Total processing time	42.6	**24.2***	6313	**3472***	3607	**1984***	116	**68.9***	5240	**2984***

6.2 Global application as a predictive model

The COMET's LiCS portal provides interferograms covering the globe. The model (Temporal self-attention network architecture) and a variety of different test results that I have processed and tested in this thesis prove that the model is diverse and so has a wide range of applications. The first and foremost application of the presented pipeline is to deploy the method online at the LiCS portal, to make real-time computations of the global dataset and separate suspected anomalies (earthquakes, volcanoes, creeping events and data errors like unwrapping errors). There are 923,989 interferograms available on the LiCS. If each interferogram covers 250 km × 250 km, then ≈ it will contain 700 overlapping patch locations or 150 non-overlapping patch locations with ≈ 60 data batches spanning 6 years 2017 to 2022. As per Table 6.1, this global dataset will just take 96 days for overlapping spatial patches and just 21 days for non-overlapping patches. This will assist to create a global catalogue of anomalies, initially unlabelled. But once the raw data is filtered, the detected anomalies can be further categorized by mapping the locations with already catalogued events such as by the USGS's earthquake portal and COMET's earthquake and volcanoes data provider.

6.2.1 Real-time detection

In recent times, automation is widely overlapping with real-time applications of artificial intelligence methods for a wide variety of tasks, for example, speech recognition Li et al. (2022), face recognition Parkhi et al. (2015), text analysis (Paolanti et al. (2019)), etc. I had similar motives for real-time application. So, when I began designing the model, I considered its deployability and real-time application in future. The primary step in this regard is to convert the methodology/model to a software. Some of the biggest hurdles in this step are (1) unsupported models, not all deep learning operations/layers are supported by the deployment framework, for example, developers face challenges in converting LSTM models for mobile ap-

plications (Chen et al. (2020b)) but not for browsers, (2) model specifications, my model that is presented in this thesis, requires a sequence of 26 raw unwrapped interferograms of size 256×256 (that are readily available), no normalization or data pre-processing steps are required. (3) Selection of API, I have trained my model in Tensorflow (*Abadiet al.* (2015)), Keras (Chollet et al. (2015)) and saved model files and trained weights in .json and .hdf5 file types, Tensorflow (*Abadiet al.* (2015)) alone provides 7 APIs for model conversion. And (4) model configuration and quantization, deep learning models have millions of parameters and are computationally expensive, before deploying on mobile phones they are first quantized, which sometimes reduces its accuracy. The weight file of my model ranges from $\approx$ 440 to 500 MB. The heavier the model, the more time it will take to be loaded online and the more space required for local storage on the browser.

I have designed and trained the models considering all these challenges and have avoided over-usage of complex deep learning operations/layers that might pose difficulty in conversion and deployment without compromising the overall performance of my model. Plus, the specifications of the model align with the data acquisition of LiCS. The event agnostic nature of the model makes it flexible and widens the range of its applications, real-time detection of anomalous events being the foremost. The labelled dataset can be created using this application and online competitions can be organised to engage researchers and developers to use, design and develop methods over it.

6.2.2 Past, present and future anomalies

Once the global data is automatically processed by my model and it has accurately separated anomalous signals from atmospheric noise. The past spatio-temporal patterns, that are already been established as irregular tectonic events in lengthy InSAR time-series can be thoroughly studied. The understanding of pre-patterns, patterns at the time of the event and post-patterns during an anomalous event can lead the scientific community to the forecasting of such events in future.

Deep learning and LSTM have been employed in the past to forecast short-term traffic (Zhao et al. (2017b); Yu et al. (2017)). The spatial temporal patterns in the traffic dataset are dynamic and random. Similarly, LSTMs are used for stock price prediction (Sunny et al. (2020)) and recently these methodologies are competing with the most accurate numerical weather models (Schultz et al. (2021); Hewage et al. (2020)).

I agree, that predicting or forecasting tectonic activities is a very challenging task, as the plate movement is random. But, trends in these movements in relation to ground deformation are already observed and studied. On top of that, the anomalous estimates made by my model after its global application on raw InSAR can be utilized to specifically learn the dynamic deformation and would be one step closer to training LSTM classifiers and predictors to forecast anomalous activity if any.

6.3 Further research

Another capability of the methodology presented here is that the deep learning component in it can solely serve as a building block for further development of various solutions. The *EP* time-series estimated by the model majorly represents signals from background atmospheric noise, this time-series in itself is of great significance. I use this time-series for anomaly detection, it can also be further developed for regional segmentation of anomalous activity or of different features by training another deep learning model on top of it. This can be done by using the residual images (RES) from my model as a mask or pixel-wise label for segmentation of anomaly.

The variogram component in my detection pipeline is computationally expensive as shown in the Table 6.1. This could be replaced by another deep learning component, that is trained on RES images. Most of the pixels in RES images are $\approx$ zero, except those areas which are anomalous. Convolutions applied to such area will

result in some values whereas zero will nullify the classifier, outputting zero. So, a discriminator-type mini-classifier can be trained, with labels initially randomly assigned 0 (normal) and 1 (anomalous). This will work accurately for anomalies with large spatial structures. For smaller anomalies, like small earthquakes or landslides, the same technique could be applied by dividing the images into non-overlapping smaller patches, for example, 64×64. For instance, considering this size, then each image will have 16 sub-patches and so 16 mini-classifiers can be trained on top and a voting system can be designed to gather results for flagging.

As discussed earlier in this section, the model tasked for segmentation can be involved indirectly in the flagging of each pixel. The RES images are used as a mask/pixel-wise label, that already holds pixel-wise spatial information of anomaly (small or large). The area covered by this region can act as an indicator for flagging. And so, deep learning components can take over the anomaly detection framework and that can be automated as well.

6.3.1 Integrating external features as input

The current model can further be extended and improved by adding in extra features like weather models, and digital elevation models (DEM). These features contain information that directly correlates with the background atmospheric noise, i.e. the topology of location (DEM) and weather models. The earthquake test case that is tested through all the models, depicts the role of DEM and related improvement. The lake can be seen in Figure 4.8 (c) just beside the earthquake location. The weather conditions and water in the lake, replicate the patterns of ground deformation, misleading the model. If, topology is added then pixel-wise information of topology with ground deformation information in the interferograms will assist in further understanding of tectonic activities. Similarly, the background atmospheric noise is directly related to the time of the year and weather. By adding this extra information, the understanding of atmospheric noise by the model can be further enhanced.

These features can be easily added to the model by adding another dimension to the network architecture. For example, currently the dimensions on which the model is trained are $256 \times 256 \times 26 \times 1$. The last dimension can be increased by adding DEM and weather model, this will make the size be $256 \times 256 \times 26 \times 3$, keeping all other convolutional layers of the same parameters. The operation of convolution is itself multidimensional by sliding window dot product in all dimensions.

6.3.2 Weakly supervised deep learning

In this thesis, I have repurposed the weights of the trained 'ALADDIn' model and employed transfer learning to train the 'Temporal self-attention' and also used its prediction (clean data) to train the latter. Similarly, my model can be used to automatically label/categorize data. And a weakly supervised model like this (Bilen and Vedaldi (2016)) can be trained on top of it. Weak supervision is referred to the deep learning models that are used for labelling the data and re-using the data with labels to re-train the model. All of this happens simultaneously. To initiate this type of training, a subset of data should be labelled. This could also be done by using the RES images as pixel-wise labels.

6.4 Concluding remarks

Automatic analysis of large unlabelled InSAR data is beneficial for understanding crustal activities and for various solid earth applications. In this thesis, I have presented a deep learning based anomaly detection framework that processes unlabelled unwrapped interferograms and predicts time-series and detects event agnostic anomalies in it. The model is capable to process different types of anomalies with varying spatial sizes (millimetre-scale $\approx$ 10 - 20 mm up to a few centimetres $\approx$ 11 and more) and temporal lengths (single temporal acquisition in case of an earthquake and a year-long temporal stamp for a volcano). As compared to the existing deep learning based InSAR applications (as shown in Table 6.2), the method

demonstrated here is unique and state-of-the-art. I have evaluated the model on data from different parts of the world with different topological and weather conditions (the global diversity of my test data is showcased in the Figure 6.1) that prove the model's capacity and diversity. Finally, I have discussed its applications that can be implemented efficiently. Also, I have made recommendations on the further development of this method and how future researchers can re-purpose this method in various ways to solve multiple tasks and further improve the results.

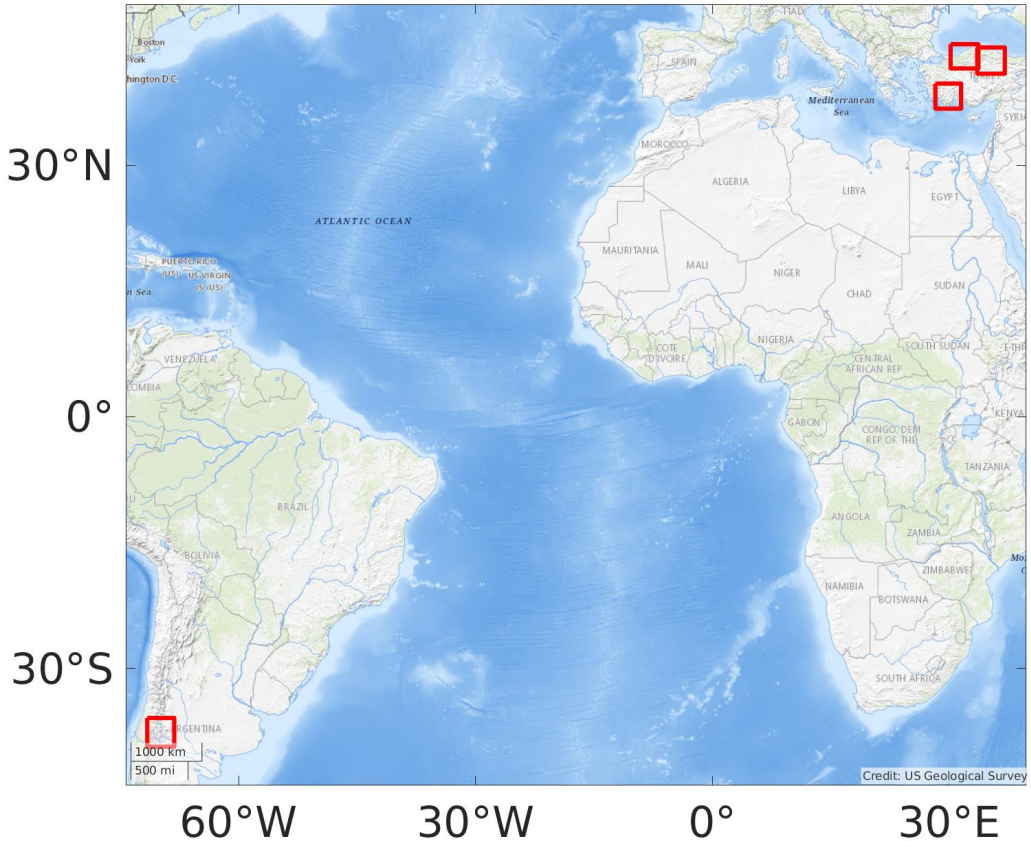

Figure 6.1: LiCS frame locations (marked in red) that have been processed in this thesis. Three frames from Turkey (2 ascending and 1 descending) and one ascending frame from Argentina.

Table 6.2: Comparison of existing deep learning based InSAR applications.

		Task	**Training data**	**No labels required**	**Event-agnostic**	**Testing event**
1	Anantrasirichai et al. (2018)	Classification	Wrapped interferograms	×	×	Volcanoes
2	Rouet-Leduc et al. (2021)	Accumalated ground deformation detection	Synthetic InSAR time-series + Ground elevation	×	×	Slow-earthquake and transient episode
3	Zhang et al. (2022)	Classification and Bounding box regression	Wrapped interferograms	×	×	Landslide
4	Sun et al. (2020)	Deformation detection	Synthetic unwrapped displacement maps (m)	×	×	Volcanoes
5	Bountos et al. (2022)	Classification	Synthetic unwrapped interferograms	×	×	Volcanoes
6	Anantrasirichai et al. (2019)	Classification	Synthetic interferograms	×	×	Volcanoes
7	Shakeel et al. (2022)	Anomaly detection	Unwrapped interferograms	✓	✓	Earthquake
8	This thesis	Anomaly detection	Unwrapped interferograms	✓	✓	Earthquake, volcano, slow-slip event and small earthquake

Appendix A : Detailed network architecture of ALADDIn

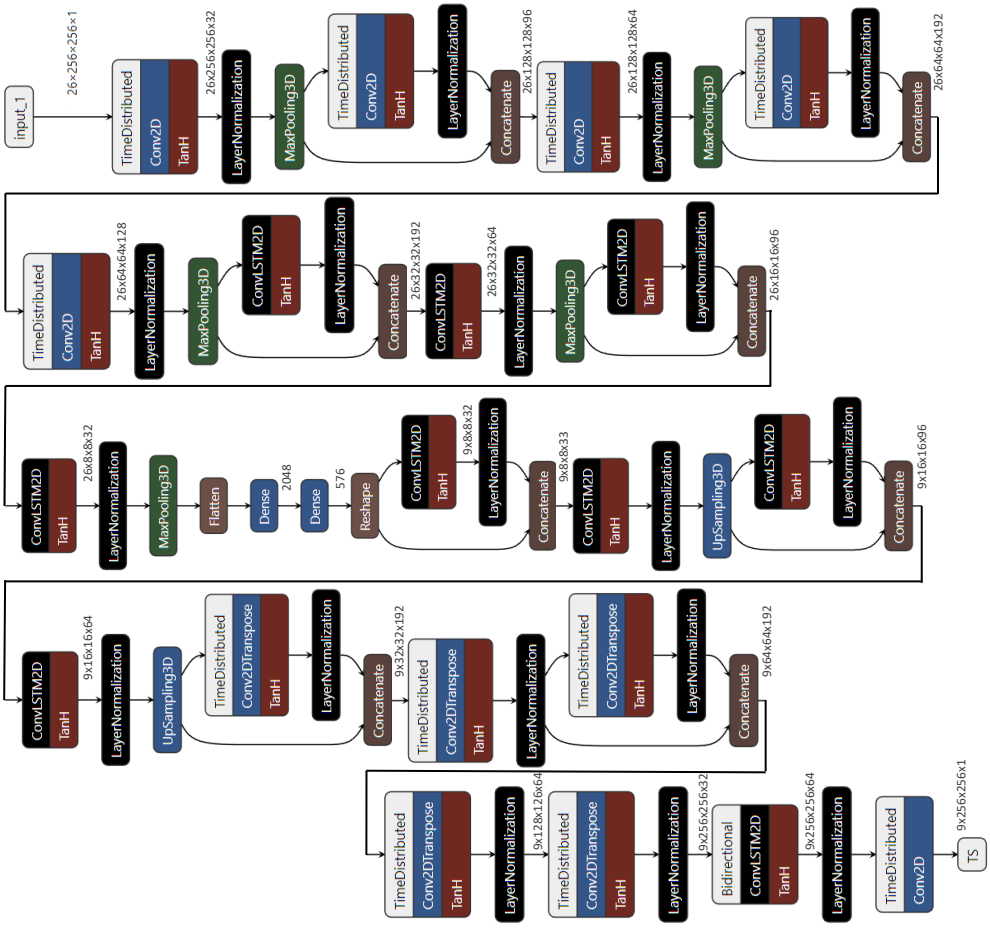

Figure A1: The detailed network architecture and connections of deep learning operations with feature sizes.

Appendix B: Detailed network architecture of Temporal self-attention

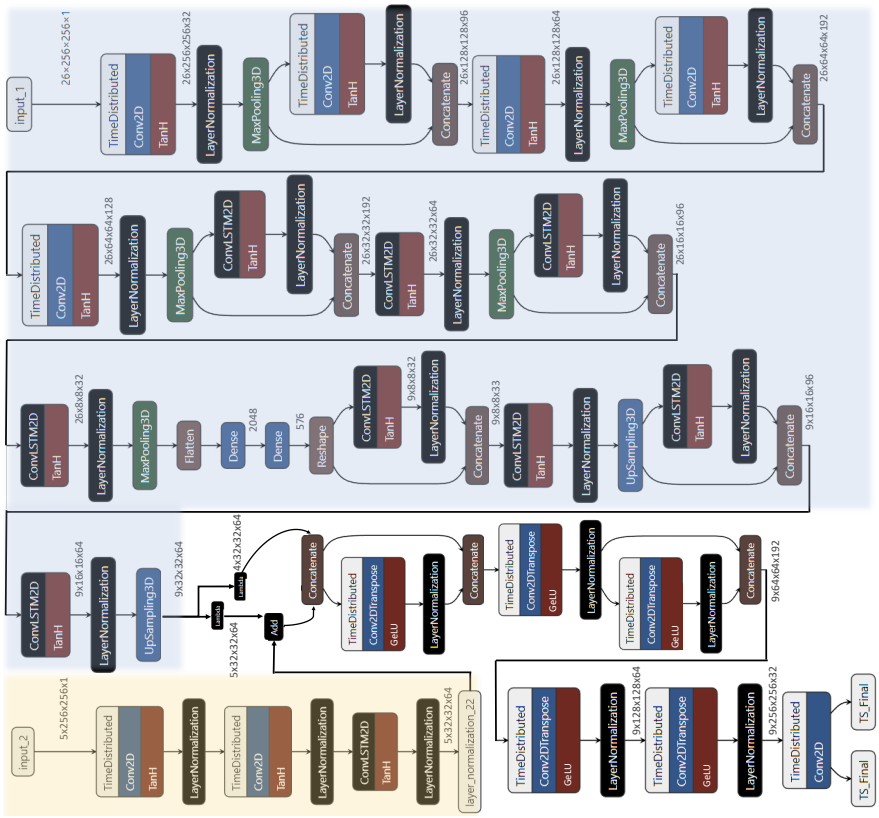

Figure B1: The detailed network architecture with feature sizes. Highlighted blue is same as ALADDIn and yellow one is new addition.

Bibliography

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

M. A. Ahmad, C. Eckert, and A. Teredesai. Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pages 559–560, 2018.

R. M. Allen and H. Kanamori. The potential for earthquake early warning in Southern California. *Science*, 300(5620):786–789, 2003.

C. Altay. Continuous creep measurement along the North Anatolian Fault zone. *Bull. Geolog. Congr. Turkey*, 6:77–84, 1991.

N. N. Ambraseys. Some characteristic features of the Anatolian fault zone. *Tectonophysics*, 9(2-3):143–165, 1970.

N. N. Ambraseys. Engineering seismology: part ii. *Earthquake engineering & structural dynamics*, 17(1):51–105, 1988.

N. Anantrasirichai, J. Biggs, F. Albino, P. Hill, and D. Bull. Application of machine learning to classification of volcanic deformation in routinely generated InSAR data. *Journal of Geophysical Research: Solid Earth*, 123(8):6592–6606, 2018.

N. Anantrasirichai, J. Biggs, F. Albino, and D. Bull. A deep learning approach to detecting volcano deformation from satellite imagery using synthetic datasets. *Remote Sensing of Environment*, 230:111179, 2019.

N. Anantrasirichai, J. Biggs, K. Kelevitz, Z. Sadeghi, T. Wright, J. Thompson, A. M. Achim, and D. Bull. Detecting ground deformation in the built environment using sparse satellite InSAR data with a convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 59(4):2940–2950, 2020.

Anza Shakeel. Anzashakeel/deep-learning-for-insar: Temporal self-attention, 2022. URL `https://zenodo.org/record/7326911`.

A. Astort, T. R. Walter, F. Ruiz, L. Sagripanti, A. Nacif, G. Acosta, and A. Folguera. Unrest at Domuyo volcano, Argentina, detected by geophysical and geodetic data and morphometric analysis. *Remote Sensing*, 11(18):2175, 2019.

J. L. Babb, J. P. Kauahikaua, and R. I. Tilling. *The Story of the Hawaiian Volcano Observatory: A Remarkable First 100 Years of Tracking Eruptions and Earthquakes*. US Department of the Interior, US Geological Survey, 2011.

V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

P. Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49. JMLR Workshop and Conference Proceedings, 2012.

A. Barka, H. S. Akyüz, H. A. Cohen, and F. Watchorn. Tectonic evolution of the Niksar and Tasova–Erbaa pull-apart basins, North Anatolian Fault Zone: their significance for the motion of the Anatolian block. *Tectonophysics*, 322(3-4): 243–264, 2000.

D. P. Bekaert, A. L. Handwerger, P. Agram, and D. B. Kirschbaum. InSAR-based detection method for mapping and monitoring slow-moving landslides in remote regions with steep and mountainous terrain: An application to Nepal. *Remote Sensing of Environment*, 249:111983, 2020.

A. R. Benaim, R. Almog, Y. Gorelik, I. Hochberg, L. Nassar, T. Mashiach, M. Khamaisi, Y. Lurie, Z. S. Azzam, J. Khoury, et al. Analyzing medical research results based on synthetic data and their relation to real data results: systematic comparison from five observational studies. *JMIR medical informatics*, 8(2): e16492, 2020.

Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

Y. Bengio, Y. LeCun, et al. Scaling learning algorithms towards AI. *Large-scale kernel machines*, 34(5):1–41, 2007.

Y. Bengio, A. C. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR, abs/1206.5538*, 1(2665): 2012, 2012.

Y. Bengio, I. Goodfellow, and A. Courville. *Deep learning*, volume 1. MIT press Cambridge, MA, USA, 2017.

P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti. A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms. *IEEE Transactions on geoscience and remote sensing*, 40(11):2375–2383, 2002.

H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2846–2854, 2016.

R. Bilham, H. Ozener, D. Mencin, A. Dogru, S. Ergintav, Z. Cakir, A. Aytun, B. Aktug, O. Yilmaz, W. Johnson, et al. Surface creep on the North Anatolian fault at Ismetpasa, Turkey, 1944–2016. *Journal of Geophysical Research: Solid Earth*, 121(10):7409–7431, 2016.

H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta. Spatial anomaly detection in sensor networks using neighborhood information. *Information Fusion*, 33:41–56, 2017.

N. I. Bountos, D. Michail, and I. Papoutsis. Learning from Synthetic InSAR with Vision Transformers: The case of volcanic unrest detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

E. Bozkurt. Neotectonics of Turkey–a synthesis. *Geodinamica acta*, 14(1-3):3–30, 2001.

J. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, 2, 1989.

R. Bürgmann, P. A. Rosen, and E. J. Fielding. Synthetic aperture radar interferometry to measure Earth's surface topography and its deformation. *Annual review of earth and planetary sciences*, 28(1):169–209, 2000.

K. Burrows, R. J. Walters, D. Milledge, K. Spaans, and A. L. Densmore. A new method for large-scale landslide classification from satellite radar. *Remote Sensing*, 11(3):237, 2019.

E. Calais, A. Freed, G. Mattioli, F. Amelung, S. Jónsson, P. Jansma, S.-H. Hong, T. Dixon, C. Prépetit, and R. Momplaisir. Transpressional rupture of an un-

mapped fault during the 2010 Haiti earthquake. *Nature Geoscience*, 3(11):794–799, 2010.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

W. Cao, X. Wang, Z. Ming, and J. Gao. A review on neural networks with random weights. *Neurocomputing*, 275:278–287, 2018.

S. Chen, E. Dobriban, and J. H. Lee. A group-theoretic framework for data augmentation. *The Journal of Machine Learning Research*, 21(1):9885–9955, 2020a.

X. Chen, X. Yao, Z. Zhou, Y. Liu, C. Yao, and K. Ren. DRs-UNet: A Deep Semantic Segmentation Network for the Recognition of Active Landslides from InSAR Imagery in the Three Rivers Region of the Qinghai–Tibet Plateau. *Remote Sensing*, 14(8):1848, 2022.

Y. Chen, G. Zhang, X. Ding, and Z. Li. Monitoring earth surface deformations with insar technology: principles and some critical issues. *Journal of Geospatial Engineering*, 2(1):3–22, 2000.

Z. Chen, Y. Cao, Y. Liu, H. Wang, T. Xie, and X. Liu. A comprehensive study on challenges in deploying deep learning based software. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 750–762, 2020b.

G. Chiodini, C. Liccioli, O. Vaselli, S. Calabrese, F. Tassi, S. Caliro, A. Caselli, M. Agusto, and W. D'alessandro. The Domuyo volcanic system: an enormous geothermal resource in Argentine Patagonia. *Journal of volcanology and geothermal research*, 274:71–77, 2014.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

F. Chollet et al. Keras. `https://keras.io`, 2015.

U. CRED. Human cost of disasters: an overview of the last 20 years 2000-2019, 2020.

V. M. Cruz-Atienza, J. Tago, C. Villafuerte, M. Wei, R. Garza-Girón, L. A. Dominguez, V. Kostoglodov, T. Nishimura, S. Franco, J. Real, et al. Short-term interaction between silent and devastating earthquakes in Mexico. *Nature communications*, 12(1):1–14, 2021.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29, 2016.

J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

D. Derauw, M. Jaspard, A. Caselli, S. Samsonov, et al. Ongoing automated ground deformation monitoring of Domuyo-Laguna del Maule area (Argentina) using Sentinel-1 MSBAS time series: Methodology description and first observations for the period 2015–2020. *Journal of South American Earth Sciences*, 104:102850, 2020.

P. M. DeVries, F. Viégas, M. Wattenberg, and B. J. Meade. Deep learning of aftershock patterns following large earthquakes. *Nature*, 560(7720):632, 2018.

C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

H. Dragert, K. Wang, and T. S. James. A silent slip event on the deeper Cascadia subduction interface. *Science*, 292(5521):1525–1528, 2001.

A. Dukkipati, D. Ghoshdastidar, and J. Krishnan. Mixture modeling with compact support distributions for unsupervised learning. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 2706–2713. IEEE, 2016.

S. Ebmeier. Application of independent component analysis to multitemporal InSAR data with volcanic case studies. *Journal of Geophysical Research: Solid Earth*, 121(12):8970–8986, 2016.

S. Ebmeier, B. Andrews, M. Araya, D. Arnold, J. Biggs, C. Cooper, E. Cottrell, M. Furtney, J. Hickey, J. Jay, et al. Synthesis of global satellite observations of magmatic and volcanic deformation: implications for volcano monitoring & the lateral extent of magmatic domains. *Journal of Applied Volcanology*, 7(1):1–26, 2018.

J. Elliott. Earth observation for the assessment of earthquake hazard, risk and disaster management. *Surveys in geophysics*, 41(6):1323–1354, 2020.

J. Elliott, R. Walters, and T. Wright. The role of space-based observation in understanding and responding to active tectonics and earthquakes. *Nature communications*, 7:13844, 2016.

T. Emardson, M. Simons, and F. Webb. Neutral atmospheric delay in interferometric synthetic aperture radar applications: Statistical description and mitigation. *Journal of Geophysical Research: Solid Earth*, 108(B5), 2003.

S. Emerson, R. Kennedy, L. O'Shea, and J. O'Brien. Trends and applications of machine learning in quantitative finance. In *8th international conference on economics and finance research (ICEFR 2019)*, 2019.

E. Eskin. Anomaly detection over noisy data using learned probability distributions. 2000.

C. Esteban, S. L. Hyland, and G. Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

M. A. Floyd, R. J. Walters, J. R. Elliott, G. J. Funning, J. L. Svarc, J. R. Murray, A. J. Hooper, Y. Larsen, P. Marinkovic, R. Bürgmann, et al. Spatial variations in fault friction related to lithology from rupture and afterslip of the 2014 South Napa, California, earthquake. *Geophysical Research Letters*, 43(13):6808–6816, 2016.

M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 289–293. IEEE, 2018.

T. Fuhrmann and M. C. Garthwaite. Resolving three-dimensional surface motion with InSAR: Constraints from multi-geometry data fusion. *Remote Sensing*, 11 (3):241, 2019.

G. J. Funning and A. Garcia. A systematic study of earthquake detectability using Sentinel-1 Interferometric Wide-Swath data. *Geophysical Journal International*, 216(1):332–349, 2018.

M. Gaddes, A. Hooper, and M. Bagnardi. Using machine learning to automatically detect volcanic unrest in a time series of interferograms. *Journal of Geophysical Research: Solid Earth*, 124(11):12304–12322, 2019.

L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. 1999.

D. C. Ghiglia and M. D. Pritt. Two-dimensional phase unwrapping: theory, algorithms, and software. *A Wiley Interscience Publication*, 1998.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1705–1714, 2019.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

K. Gupta and A.-A. Sadiq. *Responses to mass-fatalities in the aftermath of 2010 Haiti earthquake*. Natural Hazards Center, 2010.

R. F. Hanssen. *Radar interferometry: data interpretation and error analysis*, volume 2. Springer Science & Business Media, 2001.

M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

P. Hewage, A. Behera, M. Trovati, E. Pereira, M. Ghahremani, F. Palmieri, and Y. Liu. Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24(21):16453–16482, 2020.

S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

J. Hu, Z. Li, X. Ding, J. Zhu, L. Zhang, and Q. Sun. Resolving three-dimensional surface displacements from InSAR measurements: A review. *Earth-Science Reviews*, 133:1–17, 2014.

M. Huh, P. Agrawal, and A. A. Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

R. T. Ionescu, F. S. Khan, M.-I. Georgescu, and L. Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7842–7851, 2019.

M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

B. L. Kalman and S. C. Kwasny. Why tanh: choosing a sigmoidal function. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 4, pages 578–581. IEEE, 1992.

E. Karasözen, E. Nissen, P. Büyükakpınar, M. D. Cambaz, M. Kahraman, E. Kalkan Ertan, B. Abgarmi, E. Bergman, A. Ghods, and A. A. Özacar. The 2017 July 20 mw 6.6 Bodrum–Kos earthquake illuminates active faulting in the

Gulf of Gökova, SW Turkey. *Geophysical Journal International*, 214(1):185–199, 2018.

O. S. Kayhan and J. C. v. Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14274–14285, 2020.

A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE international conference on computer vision*, pages 66–75, 2017.

S. Khalid, T. Khalil, and S. Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 science and information conference*, pages 372–378. IEEE, 2014.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

D. J. Lary, A. H. Alavi, A. H. Gandomi, and A. L. Walker. Machine learning in geosciences and remote sensing. *Geoscience Frontiers*, 7(1):3–10, 2016.

M. Lazecký, K. Spaans, P. J. González, Y. Maghsoudi, Y. Morishita, F. Albino, J. Elliott, N. Greenall, E. Hatton, A. Hooper, D. Juncu, A. McDougall, R. J. Walters, C. S. Watson, J. R. Weiss, and T. J. Wright. LiCSAR: An Automatic

InSAR Tool for Measuring and Monitoring Tectonic and Volcanic Activity. *Remote Sensing*, 12(15), 2020. ISSN 2072-4292. doi: 10.3390/rs12152430. URL `https://www.mdpi.com/2072-4292/12/15/2430`.

Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

J. Li et al. Recent advances in end-to-end automatic speech recognition. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

X. Li, X. Zhang, and B. Guo. Application of collocated GPS and seismic sensors to earthquake monitoring and early warning. *Sensors*, 13(11):14261–14276, 2013.

Z. Li, T. Wright, A. Hooper, P. Crippa, P. Gonzalez, R. Walters, J. Elliott, S. Ebmeier, E. Hatton, and B. Parsons. TOWARDS INSAR EVERYWHERE, ALL THE TIME, WITH SENTINEL-1. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41, 2016.

S. K. Lim, Y. Loo, N.-T. Tran, N.-M. Cheung, G. Roig, and Y. Elovici. Doping: Generative data augmentation for unsupervised anomaly detection with gan. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1122–1127. IEEE, 2018.

T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

M. L. Littman and C. Szepesvári. A generalized reinforcement-learning model: Convergence and applications. In *ICML*, volume 96, pages 310–318. Citeseer, 1996.

H. Liu, X. Xu, E. Li, S. Zhang, and X. Li. Anomaly detection with representative neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

S. Liu, D. Huang, et al. Receptive field block net for accurate and fast object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 385–400, 2018.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

W. Lu, Y. Cheng, C. Xiao, S. Chang, S. Huang, B. Liang, and T. Huang. Unsupervised sequential outlier detection with deep architectures. *IEEE transactions on image processing*, 26(9):4321–4330, 2017.

P. Lundgren, T. Girona, M. G. Bato, V. J. Realmuto, S. Samsonov, C. Cardona, L. Franco, E. Gurrola, and M. Aivazis. The dynamics of large silicic systems from satellite remote sensing observations: the intriguing case of Domuyo volcano, Argentina. *Scientific reports*, 10(1):1–15, 2020.

W. Luo, W. Liu, and S. Gao. Remembering history with convolutional lstm for anomaly detection. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 439–444. IEEE, 2017.

L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS journal of photogrammetry and remote sensing*, 152:166–177, 2019.

W. Malfliet and W. Hereman. The tanh method: I. exact solutions of nonlinear evolution and wave equations. *Physica Scripta*, 54(6):563, 1996.

P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

E. Marchi, F. Vesperini, F. Weninger, F. Eyben, S. Squartini, and B. Schuller. Non-linear prediction with LSTM recurrent neural networks for acoustic novelty detection. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2015.

G. Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.

R. Margesson and M. Taft-Morales. Haiti earthquake: Crisis and response. Library of Congress Washington DC Congressional Research Service, 2010.

D. Massonnet, M. Rossi, C. Carmona, F. Adragna, G. Peltzer, K. Feigl, and T. Rabaute. The displacement field of the landers earthquake mapped by radar interferometry. *Nature*, 364(6433):138, 1993.

A. E. Maxwell, T. A. Warner, and F. Fang. Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing*, 39(9):2784–2817, 2018.

S. McKlusky, S. Balassanian, A. Barka, C. Demir, S. Ergintav, I. Georgiev, O. Gurkan, M. Hamburger, K. Hurst, H. Kahle, et al. Global Positioning System constraints on plate kinematics and dynamics in the Eastern Mediterranean and Caucasus. *Journal of Geophysical Research*, 105(5):695–5, 2000.

J. R. Medel and A. Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016.

Y. Miao, M. Gowayyed, and F. Metze. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *2015 IEEE Workshop*

on Automatic Speech Recognition and Understanding (ASRU), pages 167–174. IEEE, 2015.

T. M. Mitchell et al. Machine learning, 1997.

Y. Morishita, M. Lazecky, T. J. Wright, J. R. Weiss, J. R. Elliott, and A. Hooper. LiCSBAS: An open-source InSAR time series analysis package integrated with the LiCSAR automated Sentinel-1 InSAR processor. *Remote Sensing*, 12(3):424, 2020.

A. Mumuni and F. Mumuni. CNN architectures for geometric transformation-invariant feature representation in computer vision: a review. *SN Computer Science*, 2(5):1–23, 2021.

R. Nawaratne, D. Alahakoon, D. De Silva, and X. Yu. Spatiotemporal anomaly detection using deep learning for real-time video surveillance. *IEEE Transactions on Industrial Informatics*, 16(1):393–402, 2019.

G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, and Z. He. Spatially supervised recurrent convolutional neural networks for visual object tracking. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4. IEEE, 2017.

S. E. Nippress, R. Heyburn, and R. Walters. The 2008 and 2012 Moosiyan Earthquake Sequences: Rare Insights into the Role of Strike Slip and Thrust Faulting within the Simply Folded Belt (Iran). *Bulletin of the Seismological Society of America*, 107(4):1625–1641, 2017.

E. Nissen, M. Tatar, J. A. Jackson, and M. B. Allen. New views on earthquake faulting in the zagros fold-and-thrust belt of Iran. *Geophysical Journal International*, 186(3):928–944, 2011.

K. Obara and A. Kato. Connecting slow earthquakes to huge earthquakes. *Science*, 353(6296):253–257, 2016.

Y. Okada. Internal deformation due to shear and tensile faults in a half-space. *Bulletin of the seismological society of America*, 82(2):1018–1040, 1992.

G. Pang, C. Yan, C. Shen, A. v. d. Hengel, and X. Bai. Self-trained deep ordinal regression for end-to-end video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12173–12182, 2020.

G. Pang, C. Shen, L. Cao, and A. V. D. Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021.

M. Paolanti, L. Romeo, M. Martini, A. Mancini, E. Frontoni, and P. Zingaretti. Robotic retail surveying by deep learning visual and textual data. *Robotics and Autonomous Systems*, 118:179–188, 2019.

O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. 2015.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

A. Qayyum, J. Qadir, M. Bilal, and A. Al-Fuqaha. Secure and robust machine learning for healthcare: A survey. *IEEE Reviews in Biomedical Engineering*, 14: 156–180, 2020.

A. Radman, M. Akhoondzadeh, and B. Hosseiny. Integrating InSAR and deep-learning for modeling and predicting subsidence over the adjacent area of Lake Urmia, Iran. *GIScience & Remote Sensing*, 58(8):1413–1433, 2021.

X. Ren, X. Li, K. Ren, J. Song, Z. Xu, K. Deng, and X. Wang. Deep learning-based weather prediction: a survey. *Big Data Research*, 23:100178, 2021.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

B. Rouet-Leduc, R. Jolivet, M. Dalaison, P. A. Johnson, and C. Hulbert. Autonomous extraction of millimeter-scale deformation in InSAR time series using deep learning. *Nature communications*, 12(1):1–11, 2021.

B. Rousset, R. Jolivet, M. Simons, C. Lasserre, B. Riel, P. Milillo, Z. Çakir, and F. Renard. An aseismic slip transient on the North Anatolian Fault. *Geophysical Research Letters*, 43(7):3254–3262, 2016.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin. Backpropagation: The basic theory. *Backpropagation: Theory, architectures and applications*, pages 1–34, 1995.

M. I. Sameen and B. Pradhan. A novel road segmentation technique from orthophotos using deep convolutional autoencoders. *Korean J. Remote Sens*, 33(4): 423–436, 2017.

A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng. On random weights and unsupervised feature learning. In *Icml*, 2011.

T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

M. G. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. H. Leufen, A. Mozaffari, and S. Stadtler. Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society A*, 379(2194):20200097, 2021.

P. Segall, E. K. Desmarais, D. Shelly, A. Miklius, and P. Cervelli. Earthquakes triggered by silent slip events on kīlauea volcano, Hawaii. *Nature*, 442(7098): 71–74, 2006.

A. G. Semple, M. E. Pritchard, and R. B. Lohman. An incomplete inventory of suspected human-induced surface deformation in North America detected by satellite interferometric synthetic-aperture radar. *Remote Sensing*, 9(12):1296, 2017.

A. Sengör, O. Tüysüz, C. Imren, M. Sakınç, H. Eyidogan, N. Görür, X. Le Pichon, and C. Rangin. The North Anatolian Fault: A new look: Annual Review of Earth and Planetary Science, v. 33. 2005.

A. Shah, J. B. Lamare, T. N. Anh, and A. Hauptmann. Accident Forecasting in CCTV Traffic Camera Videos. *arXiv preprint arXiv:1809.05782*, 2018.

A. Shakeel, W. Sultani, and M. Ali. Deep built-structure counting in satellite imagery using attention based re-weighting. *ISPRS journal of photogrammetry and remote sensing*, 151:313–321, 2019.

A. Shakeel, R. J. Walters, S. K. Ebmeier, and N. Al Moubayed. ALADDIn: Autoencoder-LSTM-Based Anomaly Detector of Deformation in InSAR. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2022.

M. Sharma, M. Dhanaraj, S. Karnam, D. G. Chachlakis, R. Ptucha, P. P. Markopoulos, and E. Saber. YOLOrs: Object detection in multimodal remote sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:1497–1508, 2020.

F. Sica, F. Calvanese, G. Scarpa, and P. Rizzoli. A CNN-based coherence-driven approach for InSAR phase unwrapping. *IEEE Geoscience and Remote Sensing Letters*, 2020.

M. Simons and P. Rosen. Interferometric synthetic aperture radar geodesy. 2007.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.

R. S. Stein, A. A. Barka, and J. H. Dieterich. Progressive failure on the North Anatolian fault since 1939 by earthquake stress triggering. *Geophysical Journal International*, 128(3):594–604, 1997.

J. Sun, C. Wauthier, K. Stephens, M. Gervais, G. Cervone, P. La Femina, and M. Higgins. Automatic Detection of Volcanic Surface Deformation Using Deep Learning. *Journal of Geophysical Research: Solid Earth*, 125(9):e2020JB019840, 2020.

X. Sun, P. Wu, and S. C. Hoi. Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing*, 299:42–50, 2018.

M. A. I. Sunny, M. M. S. Maswood, and A. G. Alharbi. Deep learning-based stock price prediction using LSTM and bi-directional LSTM model. In *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 87–92. IEEE, 2020.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

M. Talebian, E. J. Fielding, G. J. Funning, M. Ghorashi, J. Jackson, H. Nazari, B. Parsons, K. Priestley, P. A. Rosen, R. Walker, et al. The 2003 Bam (Iran)

earthquake: Rupture of a blind strike-slip fault. *Geophysical Research Letters*, 31(11), 2004.

T. Taymaz and S. Price. The 1971 May 12 Burdur earthquake sequence, SW Turkey: a synthesis of seismological and geological observations. *Geophysical Journal International*, 108(2):589–603, 1992.

I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

L. Torrey and J. Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

A. Uçar, Y. Demir, and C. Güzeliş. Object recognition and detection with deep learning for autonomous driving applications. *Simulation*, 93(9):759–769, 2017.

P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

H. Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media, 2013.

R. J. Walters. *Geodetic observation and modelling of continental deformation in Iran and Turkey*. PhD thesis, University of Oxford, 2012.

C. Wang, H. Yang, C. Bartz, and C. Meinel. Image captioning with deep bidirectional LSTMs. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 988–997, 2016.

H. Wang, J. Hu, H. Fu, C. Wang, and Z. Wang. A Novel Quality-Guided Two-Dimensional InSAR Phase Unwrapping Method via GAUNet. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:7840–7856, 2021.

H.-H. Wei, I. M. Shohet, M. J. Skibniewski, S. Shapira, R. Levy, T. Levi, A. Salamon, and M. Zohar. Assessment of casualty and economic losses from earthquakes using semi-empirical model. *Procedia Engineering*, 123:599–605, 2015.

T. Wright, B. Parsons, J. Jackson, M. Haynes, E. Fielding, P. England, and P. Clarke. Source parameters of the 1 October 1995 Dinar (Turkey) earthquake from SAR interferometry and seismic bodywave modelling. *Earth and Planetary Science Letters*, 172(1-2):23–37, 1999.

T. J. Wright, B. E. Parsons, and Z. Lu. Toward mapping surface deformation in three dimensions using InSAR. *Geophysical Research Letters*, 31(1), 2004.

S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*, 2015.

Y. Yan, M. Chen, M.-L. Shyu, and S.-C. Chen. Deep learning for imbalanced multimedia data classification. In *2015 IEEE international symposium on multimedia (ISM)*, pages 483–488. IEEE, 2015.

Z. Ye, Y. Chen, and H. Zheng. Understanding the effect of bias in deep anomaly detection. *arXiv preprint arXiv:2105.07346*, 2021.

R. Youssef, M. Aniss, and C. Jamal. Machine learning and deep learning in remote sensing and urban application: A systematic review and meta-analysis. In

Proceedings of the 4th Edition of International Conference on Geo-IT and Water Resources 2020, Geo-IT and Water Resources 2020, pages 1–5, 2020.

R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu. Deep learning: A generic approach for extreme condition traffic forecasting. In *Proceedings of the 2017 SIAM international Conference on Data Mining*, pages 777–785. SIAM, 2017.

H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.

C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1409–1416, 2019.

T. Zhang, W. Zhang, D. Cao, Y. Yi, and X. Wu. A New Deep Learning Neural Network Model for the Identification of InSAR Anomalous Deformation Areas. *Remote Sensing*, 14(11):2690, 2022.

Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, and X.-S. Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1933–1941, 2017a.

Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu. LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11 (2):68–75, 2017b.

L. Zhou, H. Yu, Y. Lan, S. Gong, and M. Xing. CANet: An unsupervised deep convolutional neural network for efficient cluster-analysis-based multibaseline InSAR phase unwrapping. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

Y. Zhou, J. R. Elliott, B. Parsons, and R. T. Walker. The 2013 Balochistan earthquake: An extraordinary or completely ordinary event? *Geophysical Research Letters*, 42(15):6236–6243, 2015.

Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai. Deep learning representation using autoencoder for 3d shape retrieval. *Neurocomputing*, 204:41–50, 2016.

Colophon

This thesis is based on a template developed by Matthew Townson and Andrew Reeves. It was typeset with LaTeX 2_ε. It was created using the *memoir* package, maintained by Lars Madsen, with the *madsen* chapter style. The font used is Latin Modern, derived from fonts designed by Donald E. Kunith.