Article

Twisting Theory: A New Artificial Adaptive System for Landslide Prediction

Paolo Massimo Buscema [1,2,*] , **Weldon A. Lodwick** [2] **, Masoud Asadi-Zeydabadi** [2] **, Francis Newman** [2] **, Marco Breda** [1] **, Riccardo Petritoli** [1] **, Giulia Massini** [1] **, David Buscema** [1] **, Donatella Dominici** [3] **and Fabio Radicioni** [4]

1   Semeion Research Center of Sciences of Communication, 00128 Rome, Italy
2   Department of Mathematical and Statistical Sciences, University of Colorado, Denver, CO 80204, USA
3   Department of Civil, Construction-Architectural and Environmental Engineering, University of L'Aquila, 67100 L'Aquila, Italy
4   Department of Engineering, University of Perugia, 06123 Perugia, Italy
*   Correspondence: m.buscema@semeion.it; Tel.: +39-0650652350

Abstract: Landslides pose a significant risk to human life. The Twisting Theory (TWT) and Crown Clustering Algorithm (CCA) are innovative adaptive algorithms that can determine the shape of a landslide and predict its future evolution based on the movement of position sensors located in the affected area. In the first part of this study, the TWT and CCA will be thoroughly explained from a mathematical and theoretical perspective. In the second part, these algorithms will be applied to real-life cases, the Assisi landslide (1995–2008) and the Corvara landslide (2000–2008). A correlation of 0.9997 was attained between the model estimates and the expert's posterior measurements at both examined sites. The results of these applications reveal that the TWT can accurately identify the overall shape of the landslides and predict their progression, while the CCA identifies complex cause-and-effect relationships among the sensors and represents them in a clear, weighted graph. To apply this model to a wider area and secure regions at risk of landslides, it is important to emphasize its operational feasibility as it only requires the installation of GNSS sensors in a predetermined grid in the target area.

Keywords: landslide; Deep Neural Networks (DNN); Twisting Theory (TWT); artificial intelligence; artificial adaptive systems

1. Introduction

The analysis of landslide susceptibility is complicated by the presence of environmental factors that are either non-correlated or exhibit nonlinear correlations (Huang et al., 2020 [1]). These factors make it difficult to approach the study through physically or mechanically based models. Data-driven models based on advanced learning algorithms, fuzzy logic, and Machine Learning, such as Support Vector Machines (SVM), random forest (RF), boosting methods, and especially Deep Neural Nets (DNNs), have achieved superior performance compared to statistics that, thanks to advanced modeling of complex connections of interlayer interdependencies, obtain satisfactory internal representations of the data (Conforti et al., 2014 [2]; Gomez & Kavzoglu, 2005 [3]; Bui et al., 2020 [4]; Van Dao et al., 2020 [5]; Rudin, 2019 [6]; Gunning et al., 2019 [7]; Adadi & Berrada, 2018 [8]; Wang et al., 2020 [9]; Luti et al., 2020 [10]). However, the advanced data driven approaches utilized to date (especially Deep Neural Network models, given the high number of layers and hyperparameters used) have a significant drawback—not in terms of performance, but in terms of interpretability. The extreme difficulty in understanding the precise relationship between inputs and outputs often hinders experts from comprehending the underlying dynamics, resulting in a lack of trust in the results (Rudin, 2019 [6]; Gunning et al., 2019 [7];

Adadi & Berrada, 2018 [8]). Moreover, local territory and geomorphologies are often not adequately taken into account (Xiao et al. 2020 [11]). This implies that in landslide mitigation application forecasts generated by these systems are frequently minimized or disregarded, particularly if the decision-making process has significant consequences for human lives and involves high insurance and reconstruction costs (Cui et al., 2019 [12]; Froude & Petley, 2018 [13]; Huang & Fan, [14]).

One of the most commonly used approaches in the field of Landslide Susceptibility Assessment (LSA) is to employ Convolutional Neural Networks (CNNs) to create zoning maps in order to identify areas susceptible to slope failures. CNNs are particularly effective in analyzing geospatial data and are capable of automatically learning the features of the data, without the need for explicit knowledge of the physics or geology underlying landslide formation.

In Azarafza et al., 2021 [15] a deep convolutional neural network (CNN-DNN) was developed for mapping landslide susceptibility in the Isfahan province, Iran. The model was trained and validated using historical landslide data, field records, remote sensing images, and geomorphological and environmental factors. Upon evaluation, the model was found to be the most accurate among other machine learning techniques, with a prediction accuracy of 90.9%. The resulting map showed high-susceptibility areas in the western and southwestern regions of the province, which can aid in managing landslide risk and shaping land-use planning in the Isfahan province.

Nikoobakht et al., 2022 [16] used a GIS-based approach to assess landslide susceptibility in the Gorzineh-khil region of northeastern Iran using a convolutional neural network (CNN). The 15-layer CNN was programmed in Python and was trained using data on the main triggering factors for landslides including geomorphologic/topographic and water condition parameters. The results were cross-validated using loss functions and other classifiers like SVM, k-NN, and DT, and the CNN was found to have the best accuracy with 79.0% accuracy.

Nanehkaran et al., 2021 [17] used a fuzzy, logic-based, multicriteria decision-making method with five factors affecting landslides (climate, geomorphology, tectonic and seismic parameters, geological and hydrological conditions, and human activities) to create a hazard map. Through the identification of landslide influencing factors using satellite images, DEM data, and field survey, a Convolutional Neural Network (CNN) was trained; the model was used to classify the hazard potential of the region into five susceptibility classes, showing better performance compared to other machine learning classifiers (SVM, k-NN, and DT).

Therefore, it's crucial to continuously monitor landslide hazards. Typically, GNSS sensors are deployed at various points in the risk zone to track their movement over time with millimeter or centimeter accuracy. During observations, data on land movement is collected. In this study, utilizing data from two recorded observation efforts, we aim to demonstrate how to achieve two main objectives.

1. Reconstruct the cause-and-effect relationships that evolve over time between the monitored points. In other words, determine "which displacement" is the most probable cause of "which other displacement," thereby identifying the points that are the center of the landslide and those that are its by-products. To achieve this objective, we will utilize a patented and published algorithm named the **Crowd Clustering Algorithm (CCA)** [18,19];

2. Determine the shape of the entire landslide area based on the monitored points in a risk zone. To attain this goal, we will utilize an updated version of a previously patented and published algorithm named **Twisting Theory** (TWT) [18,19]. After reconstructing the landslide shape at time T(n), we can predict its new shape at a subsequent time $T(n + 1)$. To accomplish this forecasting task, we will utilize a novel **Deep Neural Network** (DNN) [20–32]. In other words, this innovative DNN has the capability to learn from the images of the complete landslide grids produced by TWT

at every observation stage, and predict the location of the sensors and the shape of the entire landslide at time $T(n + 1)$.

To accomplish these two objectives, we only require the information of the GNSS coordinates over the duration of the recorded observations. Now, let us mathematically represent this challenge.

$$Dataset = \left\{ \left\{ x_i^t, y_i^t \right\}_{i=1}^P \right\}_{t=1}^T;$$
$where:$
$x, y =$ Coordinates of GNSS;
$P =$ Number og GNSS points;
$T =$ Number of observations.

Reconstructing the form/shape of the landslide is based on the information collected from the monitored points, which is a crucial yet complex task. Merely monitoring the displacement of these points in millimeters and/or centimeters over time does not provide much insight, as the Earth's surface naturally shifts. For example, Italy moves imperceptibly towards Albania and Croatia each year. Thus, it is essential to go beyond this information to accurately determine the existence of a landslide.

The challenge in this study is to go beyond just monitoring the movement of a few GNSS sensor-monitored points, but to understand the movement of the entire terrain. The scientific challenge is to determine the behavior of all unmonitored points in relation to the few points under observation. The goal of this research is to reconstruct the overall shape of the ground movement to determine the presence, size, and evolution of a landslide.

2. Materials and Methods

This paper will describe the theory and benefits of two adaptive algorithms: Twisting Theory and Crowd Clustering Algorithm. These algorithms belong to the field of machine learning as they are based purely on data, and they evolve and learn patterns present in the data.

"Twisting Theory—TWT" is an innovative algorithm patented by Semeion (Twisting Theory (TWT): a new theory and a new class of algorithms able to model the global deformations of the space, considering the trajectories of only a little sample of points along the time flow. Applicant: Semeion Research Center & CSI. Inventor: M Buscema. USA Patent: 12/969,887. Deposited 16 December 2010) and published in international scientific journals [18,19]. The publication has shown how, starting from the data produced over time from a few points monitored by GNSS sensors, the TWT model was able to reliably reconstruct the behaviour of all "invisible" areas. It has a completely different architecture from classic neural networks: instead of reasoning on record of features, these algorithms directly process the movements of all the position sensors over time. Table 1 shows a summary comparison of advantages and disadvantages between the TWT-CCA method and the Classic LSA ANNs. The TWT-CCA solution expresses the dynamics of the landslide for the entire convex hull box in which the sensors are positioned, through a grid deformation. The widening grid shows a greater speed of the landslide; i.e., it indicates a greater susceptibility, reveals a critical point, and develops greater kinetic energy. Conversely, a narrowing grid shows the places where the landslide stops spontaneously and acquires potential energy. Therefore, the TWT–CCA algorithms are able to simultaneously display multiple susceptibility points for the entire area.

The model has already been applied to several landslides and has undergone independent validation. TWT recently was updated with a special kind of ANNs able to make predictions about the next steps of the landslide analysed.

This paper introduces the fundamental ideas behind two algorithms for landslide analysis: the Crowd Clustering Algorithm (CCA) and Twisting Theory (TWT). The main focus of the paper is to present the concepts of CCA for determining cause-and-effect

relationships and TWT for reconstructing the shape of the landslide. The Deep Neural Network (DNN) is not covered in this exposition, as the emphasis is on CCA and TWT.

Table 1. Comparison of advantages and disadvantages between the TWT–CCA method and Classic LSA ANNs.

Feature	TWT–CCA	Classic LSA ANNs
Monitoring	Can monitor slowly moving landslides.	Cannot provide specific information about monitoring.
Modeling	Can model the dynamics of a landslide and identify its most energetic location.	Can define a global landslide risk map through macro-division into regions.
Prediction	Can provide specific information on the evolution of landslides over time.	Can estimate the levels of territory risk aimed at land use planning.
Achieving Goals	Finding cause-and-effect relationships and determining the shape of the landslide.	Definition of a "global landslide risk map through macro-division into regions".
Application	Can be applied to large areas with the requirement of GNSS sensors only.	Needs specific measures regarding topographic, geologic, climatologic, seismic, human activity-related factors.

2.1. Crowd Clustering Algorithm (CCA)

We introduce a novel algorithm called Crowd Clustering (CC) (Patent: An algorithm to model the causation process of a discrete process. Applicant: Semeion Research Centre & CSI. Inventor: P.M. Buscema. USA Patent: 8,521,668/B2. Filed 24 March 2011), which aims to understand the mechanics of the cause-and-effect relationships between entities, in our case sensors, over time in a two-dimensional discrete process. This algorithm provides a new perspective to analyze the interactions between these entities in a dynamic environment. CC is a generalization for data georeferenced in the two-dimensional space of a previous algorithm, capable of managing multiple one-dimensional temporal signals simultaneously [18,19]. The main goal of the Crowd Clustering algorithm is to construct a plausible model that clearly shows the most probable cause-and-effect process, in which each entity at each time step affects other entities in the following time step.

It is assumed that the dataset to be processed by CC has the following:

1. Space Position of each entity (Space—"*S*").
2. Amount, quantity, of displacement of each entity (Quantity—"*Q*")
3. These data (space position, *S*, and the amount of displacement, *Q*) are present for each entity at any time step (Time "*T*").

The Causation Model (*M*) that CC tries to approximate has the general form:

$$(0.1) \quad M = \psi(S, T, Q).$$

Specified in greater detail, a local model (*M*[*t*, *t* + 1]), describing the likelihood of transition of all the entities (i, j) from time (*t*) to time (*t* + 1) has the following representation:

$$(0.2) \quad M_{i,j}^{[t,t+1]} = \psi\left(dx_i^{[t,t+1]}, dy_i^{[t,t+1]}, dx_j^{[t,t+1]}, dy_j^{[t,t+1]}, \sigma m_i^{[t,t+1]}, \sigma m_j^{[t,t+1]}, \Delta\phi_i^{[t,t+1]}, \Delta\phi_j^{[t,t+1]}\right)$$

where:

$dx_i^{[t,t+1]}, dx_j^{[t,t+1]}$ are the *x* displacement of the i − th and the j − th points from time *t* to *t* + 1;

$dy_i^{[t,t+1]}, dy_j^{[t,t+1]}$ are the *y* displacement of the i − th and the j − th points from time *t* to *t* + 1;

$\sigma m_i^{[t,t+1]}, \sigma m_j^{[t,t+1]}$ are the summation metric displacement of the i − th and the j − th points in time *t* and *t* + 1;

$\Delta\phi_i^{[t,t+1]}, \Delta\phi_j^{[t,t+1]}$ are the difference of phase displacements of the i − th and the j − th points from time *t* to *t* + 1.

CC needs data coming from a set of points placed in a two-dimensional space (like GNSS sensors), whose position is dynamically sampled in different temporal steps, see Table 2.

Table 2. Example of dataset to be processed by CC.

Process	Sensor 1		Sensor 2		Sensor . . .		Sensor M	
	Lat.	**Long.**	**Lat.**	**Long.**	**Lat.**	**Long.**	**Lat.**	**Long.**
t(1)	x1(1)	y1(1)	x2(1)	y2(1)	. . .	. . .	xM(1)	yM(1)
t(2)	x1(2)	y1(2)	x2(2)	y2(2)	. . .	. . .	xM(2)	yM(2)
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
t(N)	x1(N)	y1(N)	x2(N)	y2(N)	. . .	. . .	xM(N)	yM(N)

The outputs of the CC algorithm applied to any dataset of this type are the following:

a.   A table of the most likely and prevalent cause-and-effect relationship among the given sensors (points) during and at the end of the temporal flow.

b.   A sparse and a directed graph of the most effective excitatory relationship among the sensors during the whole process.

c.   A sparse and a directed graph of the most effective inhibitory relationship among the sensors during the whole process.

The cause-and-effect relationship table plays a crucial role in comprehending the dynamics of a landslide. It highlights the areas that are key to the expansion of the landslide from a local to a larger region. Understanding the distinction between excitatory and inhibitory relationships is vital in determining the direction of the landslide's development. Excitatory relationships occur when the movement of a sensor, and the area it represents, influences the movement of other areas. In contrast, inhibitory relationships occur when an area impedes the movement of the landslide. Furthermore, the indication of the strength of the excitatory or inhibitory relationship provides insights into the stability of the ongoing landslide phenomenon.

CC algorithm works according to the following steps:

a.   The displacement of each sensor at each temporal observation is calculated (Equations (1) and (2)):

$$dx^{[n]} = x^{[n]} - x^{[n-1]};\tag{1}$$

$$dy^{[n]} = y^{[n]} - y^{[n-1]};\tag{2}$$

where $n = [t, t-1]$.

b.   The calculation of the phase and of the momentum between any couple of sequential displacement of all the sensors is represented by Equations (3) and (4):

$$\phi_i^{[n]} = \left\{ \begin{array}{ll} 0 & \left(dx_i^{[n]} > 0\right) \wedge \left(dy_i^{[n]} > 0\right) \\ \pi & \left(dx_i^{[n]} < 0\right) \wedge \left(dy_i^{[n]} < 0\right) \\ 2\pi & \left(dx_i^{[n]} > 0\right) \wedge \left(dy_i^{[n]} < 0\right) \\ \pi & \left(dx_i^{[n]} < 0\right) \wedge \left(dy_i^{[n]} > 0\right) \end{array} \right\};\tag{3}$$

$$q_i^{[n]} = arctan\left(\frac{dy_i^{[n]}}{dx_i^{[n]}}\right) + \phi_i^{[n]}; q_i^{[n]} \in [0, 2\pi];\tag{4}$$

c.   Then the calculation of the strength of each displacement of any sensor is calculated (Equation (5)):

$$m_i^{[n]} = \frac{1}{m^*} \sqrt[2]{\left(dx_i^{[n]}\right)^2 + \left(dy_i^{[n]}\right)^2}; m^* = \underset{i,n}{Max}\left\{m_i^{[n]}\right\}; m_i^{[n]} \in [0, 1].\tag{5}$$

d.　The calculation of the resonance of the phase, the momentum, and the strength of any pair of sensors are inferred by Equations (6)–(8):

$$q \in [0, 2\pi]; \text{ Phase}; m \in [0, 1]; \text{ Module}$$

$$\alpha_{i,j}^{[n,n+1]} = \begin{cases} 2\pi - \left| q_i^{[n]} - q_j^{[n+1]} \right| & \left| q_i^{[n]} - q_j^{[n+1]} \right| > \pi \\ \left| q_j^{[n]} - q_j^{[n+1]} \right| & \left| q_j^{[n]} - q_j^{[n+1]} \right| \leq \pi \end{cases}; \tag{6}$$

$$Q_{i,j}^{[n,n+1]} = 1 - \frac{2 \cdot \alpha_{i,j}^{[n,n+1]}}{\pi}; \tag{7}$$

$$M_{i,j}^{[n,n+1]} = m_i^{[n]} \cdot m_j^{[n+1]}; \tag{8}$$

e.　Finally, the global strength with which the displacement of each sensor acts on the displacement of any other, also considering their reciprocal geographical distance is calculated:

$$S_{i,j} = \sum_{n=1}^{N-1} Q_{i,j}^{[n,n+1]} \cdot M_{i,j}^{[n,n+1]} \cdot e^{-\frac{d_{i,j}^{[n]}}{\sigma}}; \tag{9}$$

At this point two other steps are computed:

f.　The calculation specified by Equations (10) and (11) is performed for each sensor in order to determine its maximum excitatory strength over time. This step enables the algorithm to identify the sensor that is most likely to induce the displacement of any other sensor during the temporal flow.

$$E_{Win^+, j} = \underset{i}{\text{ArgMax}} \{ S_{i,j} \}; \tag{10}$$

$$\begin{cases} C_{i,j}^+ = 1 & i = Win^+ \\ C_{i,j}^+ = 0 & i \neq Win^+ \end{cases}; \tag{11}$$

g.　The calculation, for each sensor, of the maximum inhibitory strength it exerts on other sensors in time is provided by Equations (12) and (13). This step enables the algorithm to identify which sensor is most likely to exhibit the greatest inhibitory strength among all the sensors during the temporal progression.

$$I_{Win^-, j} = \underset{i}{\text{ArgMin}} \{ S_{i,j} \}; \tag{12}$$

$$\begin{cases} C_{i,j}^- = 1 & i = Win^- \\ C_{i,j}^- = 0 & i \neq Win^- \end{cases}; \tag{13}$$

At the end of the process there are two square matrices representing the cause-and-effect relationship among sensors during the temporal flow. These two matrices are:

1.　A matrix of the excitatory relations, $C + (i, j)$, among the sensors indicating which sensor imposes the direction of its displacement on any other sensor;
2.　A matrix of inhibitory relations, $C - (i, j)$, among the sensors indicating which sensor constrains the direction of the displacement on any other sensor.

These matrices can be visualized as direct and weighted graphs, which cluster the sensors and their asymmetrical relationships. These graph representations of the CC algorithm hypotheses, regarding the dynamics of the process, are easily verifiable through experimentation.

2.2. Twisting Theory (TWT)

The second algorithm is TWIST. To understand TWT, we must assume that the displacement of each GNSS in the territory over time will be the entities (x, y) that the algorithm analyzes. The movement of these entities over time represents given data points [18,19], which serve as input to the algorithm. Imagine a set of entities located in a two-dimensional space that are moving due to the influence of some underlying force. This force can be interpreted as the quantity associated with each entity, which imparts momentum to the entity proportional to its magnitude. The motion of each entity results in a trajectory in the environmental space, which is assumed to be linear for simplicity. Additionally, the space is divided into a grid and the forces acting on the entities also affect the grid, causing a deformation of the original grid. As a result, the field of forces generated by the quantities assigned to the entities can be visualized as modifications to the grid. The set of all points on the grid is referred to as geometrical points. What is the best way to describe the actual effect of the forces acting on the entities on the grid?

The Twisting Theory (TWT) algorithm is a model that predicts how each point on the grid will change its coordinates at each step in time, as entities move to their new positions. TWT divides trajectories into N equal substeps, allowing each entity to be identified by its starting coordinates, the local target during each sub-step, and its final destination coordinates (Figure 1):
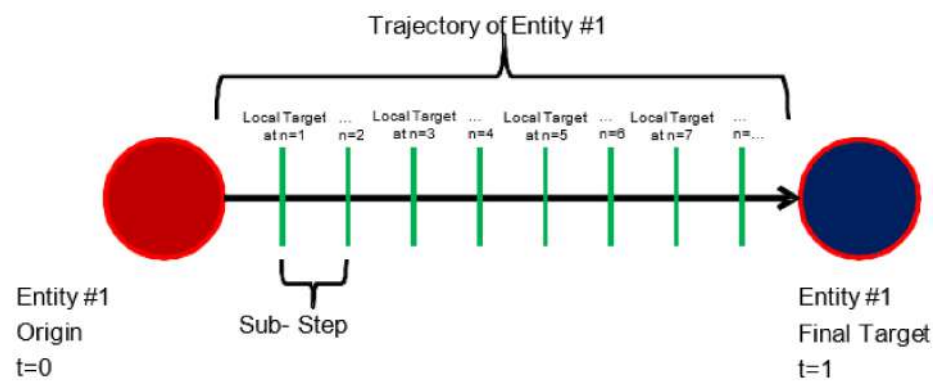

Figure 1. Splitting of the entity trajectory into substeps.

TWT focuses on the distances between a particular geometrical point on the grid and the positions of the entities as they move along their trajectory. At each substep, TWT measures how the distance between the geometrical point and the entity changes according to the following.

$$d^S_{pj}(n) = \sqrt{\left(x^G_p(n) - x^S_j\right)^2 + \left(y^G_p(n) - y^S_j\right)^2}$$
$$d^T_{pj}(n) = \sqrt{\left(x^G_p(n) - x^T_j(n)\right)^2 + \left(y^G_p(n) - y^T_j(n)\right)^2}.$$

where

$x^G_p(n), y^G_p(n)$ are the coordinates of a generic grid point (p) of the grid at substep (n); when $n = 0$, grid points all sit upon the regular grid.

x^S_j, y^S_j is the origin Coordinates of each entity (j), i.e., assigned point.

$x^T_j(n), y^T_j(n)$ are the local target of the coordinates of each entity (j) at any substep (n); when $n = 0$, the entity lies at its Origin, whereas when $n = $ N the entity has completed its trajectory.

$d^S_{pj}(n)$ is the distance of a generic grid point (p) from the Origin of any entity at substep (n);

$d^T_{pj}(n)$ is the istance of a generic grid point (p) from the Local Target (n) of any entity at substep (n);

The change in the distance between a given geometrical point and the entity's position as it moves along its trajectory can be interpreted as the buildup of potential energy that is then able to act on the grid, altering the position of the geometrical points. The more the entity's position changes relative to the given geometrical point from one step to the next, the more the grid is stretched or compressed in response (Figure 2).
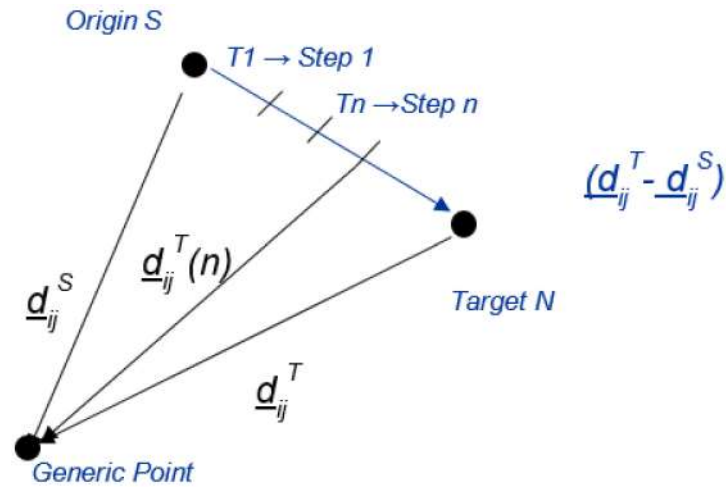

Figure 2. The subtrajectory of the entity and the accumulation of potential energy.

In mathematical terms, the variation of potential energy ΔE across the whole trajectory can be expressed as:

$$d^S_{p,j_{(n)}} = \sqrt{\left(x^G_{p_{(n)}} - x^S_j\right)^2 + \left(y^G_{p_{(n)}} - y^S_j\right)^2} \tag{14}$$

$$d^T_{p,j_{(n)}} = \sqrt{\left(x^G_{p_{(n)}} - x^T_{j_{(n)}}\right)^2 + \left(y^G_{p_{(n)}} - y^T_{j_{(n)}}\right)^2} \tag{15}$$

$$\Delta E_{p_{(n)}} = \sum_{j=1}^{N} exp\left(-\frac{\left(d^S_{p,j_{(n)}} + d^T_{j_{(n)}}\right)}{\alpha}\right) \cdot \left|d^S_{p,j_{(n)}} - d^T_{p,j_{(n)}}\right|. \tag{16}$$

However, we also need to consider the side effects, in Equations (14)–(16), of the grid points among themselves. The following Equations (17)–(20) take into account all the side effects of the energy that accumulates at each grid point, absorbed from its neighbour.

$$R = \text{ neighbour of } \pm \Delta E_{p_{(n)}} \text{ dimension of } p \text{ grid point;} \tag{17}$$

$$p_{(n)} + z_{(n)} = \text{a grid point belongs to the neighbour R.} \tag{18}$$

$$\Delta K_{p_{(n)}} = \sum_{z=1}^{R} exp\left(-\frac{\left(d^G_{p_{(n)},p+z_{(n)}} + d^G_{p,p+z_{(n-1)}}\right)}{\alpha}\right) \cdot \left|d^G_{p_{(n)},p+z_{(n)}} - d^G_{p_{(n)},p+z_{(n-1)}}\right|; \tag{19}$$

$$\Delta F_{p_{(n)}} = \Delta E_{p_{(n)}} + \Delta K_{p_{(n)}}. \tag{20}$$

Thus, Equations (21)–(30) show how the potential energy is transformed into kinetic energy, that is, how it produces a deformation of the grid as determined by the movement of the source entities:

$$\delta E\left(x_{p_{(n)}}^{G}\right) = \sum_{j}^{N} exp\left(-\frac{d_{p,j_{(n)}}^{S} + d_{p,j_{(n)}}^{T}}{\alpha}\right) \cdot \left(x_{j}^{S} - x_{j_{(n)}}^{T}\right); \tag{21}$$

$$\delta K\left(x_{p_{(n)}}^{G}\right) = \sum_{z=1}^{R} exp\left(-\frac{d_{p,z_{(n)}}^{G} + d_{p,z_{(n-1)}}^{G}}{\alpha}\right) \cdot \left(x_{p_{(n-1)}}^{G} - x_{p_{(n)}}^{G}\right); \tag{22}$$

$$\delta E\left(y_{p_{(n)}}^{G}\right) = \sum_{j}^{N} exp\left(-\frac{d_{p,j_{(n)}}^{S} + d_{p,j_{(n)}}^{T}}{\alpha}\right) \cdot \left(y_{j}^{S} - y_{j_{(n)}}^{T}\right); \tag{23}$$

$$\delta K\left(y_{p_{(n)}}^{G}\right) = \sum_{z=1}^{R} exp\left(-\frac{d_{p,z_{(n)}}^{G} + d_{p,z_{(n-1)}}^{G}}{\alpha}\right) \cdot \left(y_{p_{(n-1)}}^{G} - y_{p_{(n)}}^{G}\right); \tag{24}$$

$$\delta F\left(x_{p_{(n)}}^{G}\right) = \delta E\left(x_{p_{(n)}}\right) + \delta K\left(x_{p_{(n)}}\right); \tag{25}$$

$$\delta F\left(y_{p_{(n)}}^{G}\right) = \delta E\left(y_{p_{(n)}}\right) + \delta K\left(y_{p_{(n)}}\right); \tag{26}$$

$$x_{p_{(n+1)}}^{G} = x_{p_{(n)}}^{G} + \Delta F_{p_{(n)}} \, \delta F\left(x_{p_{(n)}}\right) < 0; \tag{27}$$

$$x_{p_{(n+1)}}^{G} = x_{p_{(n)}}^{G} - \Delta F_{p_{(n)}} \, \delta F\left(x_{p_{(n)}}\right) \geq 0; \tag{28}$$

$$y_{p_{(n+1)}}^{G} = y_{p_{(n)}}^{G} + \Delta F_{p_{(n)}} \, \delta F\left(y_{p_{(n)}}\right) < 0; \tag{29}$$

$$y_{p_{(n+1)}}^{G} = y_{p_{(n)}}^{G} - \Delta F_{p_{(n)}} \, \delta F\left(y_{p_{(n)}}\right) \geq 0. \tag{30}$$

TWT may be considered as a special kind of ANN whose weights are manifested by the grid deformation during its learning phase, where the dynamics implemented by the equations reaches a point of convergence. A contraction of a grid point indicates an inhibitory force, while an expansion of a grid point indicates an excitatory force. The number of iterations in this type of artificial neural network is determined beforehand by the number of observations of the movement of the entities and the number of substeps in each observation.

It is possible to analogize TWT with another type of neural network: the well-known Self Organizing Map, SOM, designed by Kohonen around the 1970s [22]. The SOM is an Unsupervised Neural Network whose termination criterion is defined a priori before to begin the training phase. This ANN generates a special kind of projection of the input space into a two-dimensional space, based on the multiple similarities of the patterns of the training set. At the end of the training phase, the resulting weights (codebook) define the optimal parameters through which the SOM classifies, in a blind way, the patterns of the testing set, taking into account the global similarity that each one of them has with the already trained codebooks. Under this point of view, TWT works in a very similar way to the SOM:

a. Both the algorithms are unsupervised.
b. The number of the epochs for TWT is also defined a priori, according to the number of observation campaigns sampled to monitor the landslide.
c. During the training phase the TWT codebooks are represented by the progressive twisting of the regular grid defined at the beginning. These codebook at the end

of the training represent the weights of TWT, able to provide a picture of how the GNSSs movement has deformed the entire space of the land slide.

d. These codebook are the weights that TWT show at the end of the training phase. They are locale and explainable because the deformation of the entire plane is defined only by the local weights that deform the landslide plane in each specific part of the space. So, TWT does not work as a black box, but is more like the ANNs whose learning law is vectorial quantization (Learning Vector Quantization [23], Adaptive Vector Quantization [24]).

e. TWT is different form a SOM because during its training phase it works only on temporal series (series of observational campaigns); under this profile TWT is a special kind of recurrent ANN, because any updating of its weights considers all the previous states of the algorithm.

f. While most classic ANNs start their training with a random initialization of their weight matrices, TWT initializes its weights with the same values (regular grid—see point "c"), in the same way in which a special type of ANN, the Auto Contractive Map [6], used to do.

g. The only big difference that the TWT algorithm has with the classic ANNs is that TWT has only the training phase, because TWT modifies its weights (codebook) every time that a new observation campaign is added to the old data set. The prediction TWT was add the TWT in order to allow to the TWT to make predictions about the next step of any landslide, after the training phase. The prediction TWT will be explained in the next paragraphers, where TWT will need the support of the classic supervised ANNs to complete its predictive job.

Taking account all these considerations, we are able to assert that TWT belongs to the Artificial Adaptive System family and that, in particular, TWT is a new and unusual kind of ANN.

2.3. Predictive Twisting Theory (P-TWT)

TWT generates a new grid at each time step of its processing task. If the assigned dataset is composed of T observations along the time, TWT generates a series of grids, where each one of them will show the estimated deformation of the entire plane at that time step, according to the displacements (shifts) recorded in the assigned data set. These grids can be used as a new dataset to enable TWT to make predictions about the future shifts of the GNSS at the next steps.

To achieve this new prediction task TWT needs two new methodological operations:

• A specifying coding of the data generated by each grid;

• An ANN (deep or shallow) able to learn from the data already generated from the grids, in order to make estimations about the new positions of the GNSSs at the next observation. In this study, classic neural networks (both deep and shallow) as well as non-classical networks such as SVCM (deep) are utilized. Additionally, the MLP with multiple hidden layers was preinitialized with autoencoders and RBMs to prepare the weights of each layer (see [25,26]).

Further, to control the accuracy of these estimations we need to use the first grids generated by TWT as Training set, and the last grids as Testing set, in order to make a blind verification of the accuracy of the predictions of the ANN already trained.

The coding strategy: The organization of the data of each grid generated by TWT in a new data set is quite simple: a pattern will consist in the latitude and longitude of each GNSS in each grid concatenated with the coordinates of its first eight neighbors.

Let us imagine a dataset of 51 GNSS (N = 51) and 13 observations (T = 13). TWT will generate 13 grids (t1, t2, . . . , t13). In each grid there is the actual position (latitude and longitude) for each GNSS, and the longitude and the latitude of each of the eight first neighbors to each GNSS. Thus, each input pattern will consist of a vector of 18 components: x and y of the considered GNSS, and x and y of each of its first eight neighbors. The target to be learnt of each input pattern will be the coordinates (x and y) of that GNSS in the next

TWT grid. Consequently, the topology of the ANN to be trained will present 18 inputs and 2 outputs. Figure 3 shows how the topology of each GNSS of a grid must be coded in a pattern representing that GNSS, and the target that the ANN has to learn to associate at such patterns.
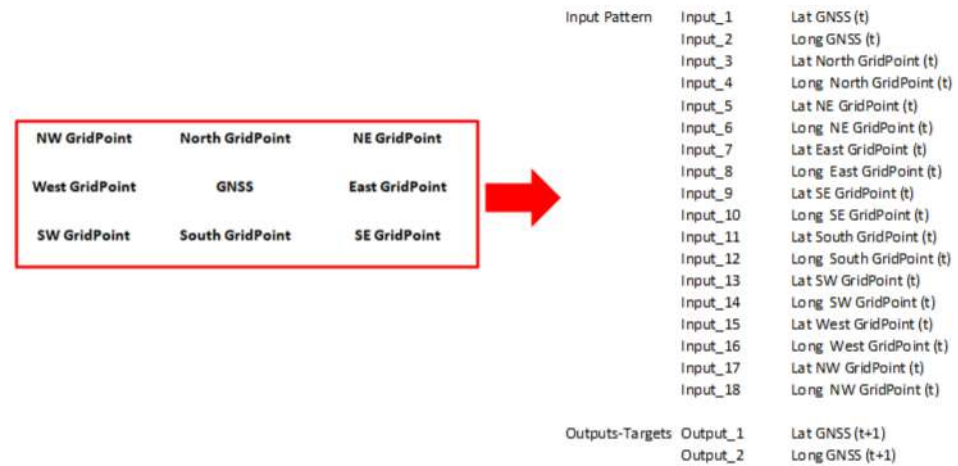

Figure 3. How to code each GNSS of each TWT grid into a pattern to be learnt by an ANN.

If the number of GNSS of the assigned dataset is N = 51, each grid generated by TWT will be composed of 51 couples of input = 18 and output = 2 patterns.

Consequently, if the assigned dataset is composed of T = 13 observations, TWT will generate 13 grids; twelve of these will be used as training set for ANN learning process, while the last one (grid) will be used as testing set in order to estimate the ANNs prediction accuracy. This means that the training set in this exemplificative case will present 612 patterns (12 × 51), while the testing set will be composed of the 51 patterns of the thirteenth grid.

The ANN learning strategy: Once the training set and the testing set have been prepared, a specific ANN has to be selected for the learning phase. Many different ANNs topologies and Learning Laws were compared to optimize this regression task; classic Multilayer Perceptron, with different learning gradient strategies and topologies (number of hidden layers), has shown many problems of convergence (time and loss function). The best results were represented by a new supervised ANN, not based on gradient descend methodology: the Supervised Contractive Map, SVCm for short [27]. This ANN has already been successfully used in several application areas (earthquake prediction [28], atrial fibrillation detection [29]) where it has demonstrated not only superior performance compared to classical models, but also a great ability to converge in a few epochs with large and highly nonlinear data sets. We have reported the main equations of SVCm, able to suggest the differences and the similarities that this specific ANN has with the more classic and known supervised ANNs.

Legend:

$[l]$ = number or name of the ANN layer;

$u_i^{[l]}$ = values of the all $i - th$ nodes of the $l - th$ layer;

$w_{i,j}^{[l]}$ = weight matrix connecting the layer $[l-1]$ to the layer $[l]$;

$C^{[l]}$ = number of nodes of $l - th$ layer; t_i = value of $i - th$ of the dependent variable;

$LCoef$ = ANN learning rate.

Signal Transfer from Input layer to Output layer (passing through each hidden layer):

$$CNet_i^{[l]} = \sum_{j}^{C^{[l-1]}} u_j^{[l-1]} \cdot \left(1 - \frac{w_{ij}^{[l]}}{C^{[l-1]}}\right); \tag{31}$$

$$INet_i^{[l]} = \sum_j^{C^{[l-1]}} u_j^{[l-1]} \cdot w_{ij}^{[l]}; \tag{32}$$

$$u_i^{[l]} = \sin\left(INet_i^{[l]} \cdot \left(1 - \frac{\sin(CNet_i^{[l]})}{C^{[l-1]}} \right) \right). \tag{33}$$

Weights update:

$$\delta_i^{[out]} = \left(t_i - u_i^{[out]} \right) \cdot \cos\left(INet_i^{[out]} \cdot \left(1 - \frac{\sin(CNet_i^{[out]})}{C^{[out-1]}} \right) \right); \tag{34}$$

$$\delta_i^{[hid]} = \sum_k^{Num^{[hid+1]}} \left(\delta_k^{[hid+1]} \cdot w_{ki}^{[hid+1]} \right) \cdot \cos\left(INet_i^{[hid]} \cdot \left(1 - \frac{\sin(CNet_i^{[hid]})}{C^{[hid-1]}} \right) \right); \tag{35}$$

$$\Delta w_{ij}^{[l]} = LCoef \cdot \delta_i^{[l]} \cdot u_j^{[l-1]} \cdot \left(1 - \frac{w_{ij}^{[l]}}{C^{[l-1]}} \right). \tag{36}$$

The supervised Contractive Map calculates two net inputs for each destination node: a classic weighted input, see Equation (32); and a contractive input, see Equation (31). The first net input tends to decay or to increase when the positive or negative value of the weight (w) becomes close to a specific constant (C).

Equation (33) activates each node according to a sine function of the two net inputs (the contractive input works as a harmonic modulation of the weighted input). The advantages and the disadvantages of sine transfer function to work properly into the topology of Multilayer Perceptron was already analyzed in the scientific literature (Le Cun et al., 1991 [30]; Le Cun et al.,1998 [31]).

Equation (34) shows a typical error calculation using the distance between the desiderate output and the estimated output, times the first derivative of sine transfer function.

Equation (35) works in the same way as Equation (4), but using the chain rule to calculate the local error of each hidden unit.

Equation (36) updates the weight matrices, using the chain rules, with a contractive factor useful to limit an extreme growth of each weight value.

In the next paragraphs (Section 3.3 and 4.2) we show two different applications of Predictive TWT (P-TWT): the Assisi and the Corvara landslide. In both cases, a Supervised Contractive Map was applied to the grid data after the coding processing, already described. In this section we show the topology implemented of SVCm for these two applications. Below are the topology and the starting parameters for the Assisi application (Table 3a), and the same information for the Corvara application (Table 3b) are shown.

Table 3. Supervised Contractive Map: (**a**) Assisi Training Architecture; (**b**) Corvara Training Architecture.

(a) SVCm Architecture (Assisi)		(b) SVCm Architecture (Corvara)	
Input: 18		Input: 18	
Hidden 1: 48	LCoef: 0.0100 Function: 24 Contractives	Hidden 1: 24	LCoef: 0.0100 Function: 24 Contractives
Hidden 2: 48	LCoef: 0.0100 Function: 24 Contractives	Hidden 2: 24	LCoef: 0.0100 Function: 24 Contractives
Output: 2	LCoef: 0.0100 Function: 2 Contractives	Output: 2	LCoef: 0.0100 Function: 2 Contractives
Training Patterns: 612 Testing Patterns: 51 Epochs (Train): 2771 RMSE (Train): 0.0051103		Training Patterns: 65 Testing Patterns: 13 Epochs (Train): 49,572 RMSE (Train): 0.00117701	

3. Application 1: The Assisi Landslide

The Assisi landslide (see map in Figure 4) is described in Bovenga et al., 2013 [32]. It states: *"This case refers to a landslide affecting an urban expansion area of Assisi, built starting from the years 1950–1960. The area is located completely outside the historical town centre, at a distance of some hundreds of metres from the ancient city wall. . . . Since 1995, the Perugia University . . . has established a GNSS control network over the area, integrated from 1999 on a levelling network. GNSS and levelling monitoring have been performed since then, with an approximately annual data collections up to 2008."* See Figure 5.

Figure 4. Map showing the location of the two study areas within Italy: Assisi and Corvara.
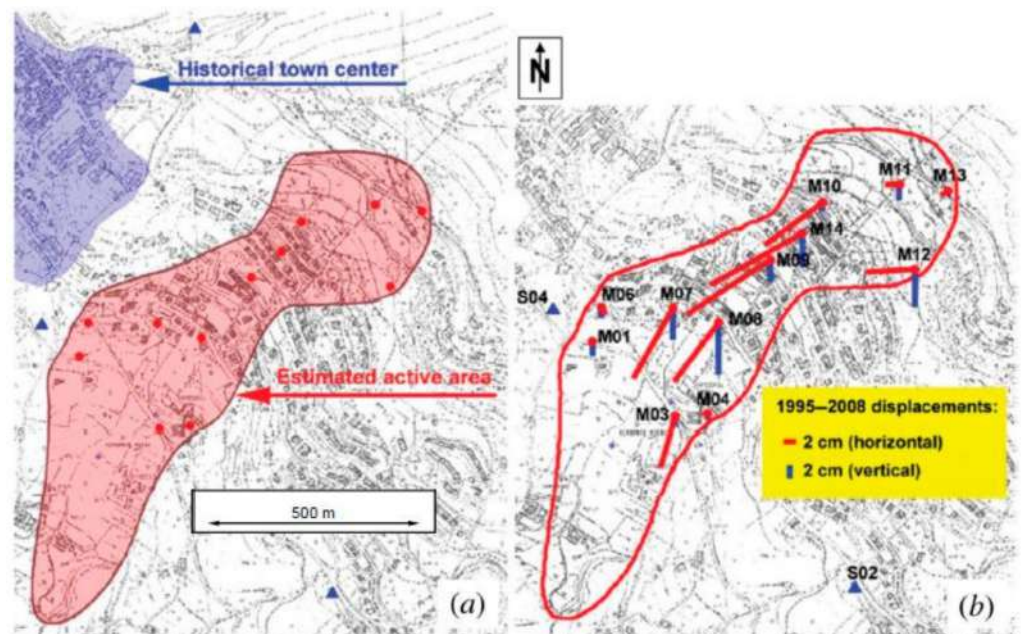

Figure 5. Taken from Bovenga et al., 2013 [32]: "(**a**) The Assisi landslide and its location with respect to the town center; triangles and dots represent GNSS fiducial and control points, respectively; (**b**) The Assisi GNSS network and the 1995–2008 displacement vectors".

The final data set of the Assisi landslide is so far composed of 13 GNSS points collected during 8 observational times. In Figure 6 we have renamed the GNSS points to make clearer the following application.
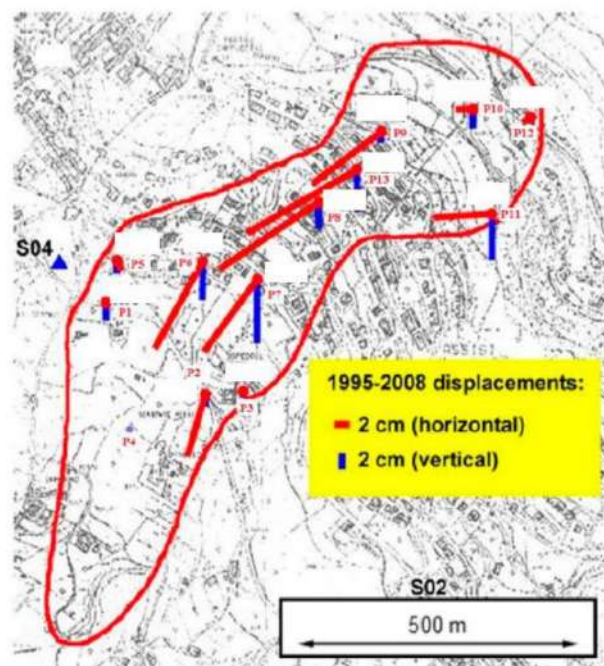
Figure 6. The GNSS points renamed from P1 to P13.

The vertical displacements (1995–2008) of the points from the reference are shown in blue, while the horizontal displacements by the sensors in the landslide are shown in red.

3.1. CCA Algorithm

The CCA algorithm generates 6 direct and weighted graphs (N-2), considering 8 time steps of observation. The algorithm analyzes the delta of delta displacements of the GNSS points. The strength, displacement, and angle degree that each GNSS exercises on each of the others is made explicit in each graph. Table 4 shows this information after the first two observations, and Figure 7 displays the corresponding cause-and-effect relationship graph at this step.

Table 4. From step 1 to step 2. The points with"++" symbol are the more active GNSS and the points with "−" symbol are the most resilient to the displacement.

| Excitations [0,1]							
Cause		**Effect**	**Strenght**	**Cause Module T_1**	**Effect Module T_2**	**Cause Degree T_1**	**Effect Degree T_2**
P2	→	P3–	0.015217	0.230837	0.455262	247.948959°	312.468597°
	→	P4	0.043036	0.230837	1	247.948959°	254.608795°
	→	P7++	0.004337	0.230837	0.322421	247.948959°	268.315308°
P3–	→	P1	0.001568	0.27031	0.134086	52.326408°	51.972771°
	→	P5	0.002237	0.27031	0.385002	52.326408°	32.795769°
P4	→	P2	0.008367	0.216745	0.215642	268.388977°	258.407837°
P7++	→	P6	0.095422	0.492819	0.869689	239.630249°	268.394012°
	→	P10	0.000144	0.492819	0.42529	239.630249°	316.548187°
P8	→	P13	0.06098	0.407436	0.507571	192.280960°	207.658875°
P9	→	P12	0.000244	0.209448	0.411627	220.279877°	304.003754°
P12	→	P11	0.019294	0.201505	0.66613	216.253845°	239.995087°
P13	→	P8	0.095164	0.681295	0.701071	209.010361°	248.589050°
	→	P9	0.3411	0.681295	0.193786	209.010361°	243.434967°
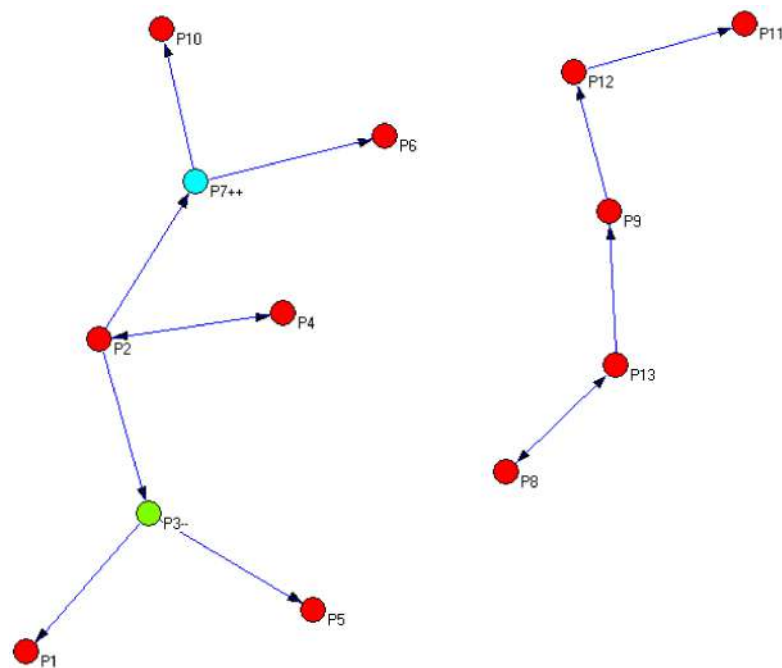
Figure 7. Graph of cause-and-effect relationships after steps 1 and 2. The points with"++" symbol are the more active GNSS and the points with "−" symbol are the most resilient to the displacement. Cause-and-effect relationship graph after the first two steps of the landslide shows two partially independent clusters. The cluster on the left appears to be the dominant one, with two prominent points, GNSS P7 and GNSS P3. GNSS P7 shows the highest level of activity in this phase, while GNSS P3 displays the highest level of resilience.

We have only displayed the Excitatory Cause-and-Effect table, not the Inhibitory Cause-and-Effect table. Let us take the first row as an example. The influence of P2 on P3 is shown in the first row, with a strength of approximately 0.015, as indicated in the "Strength" column. The "Cause Module T_1" column shows the displacement of P2 from its previous delta position, which in this case is the beginning of the landslide. This value is constant for each row of P2, i.e., row 1, 2, and 3. The "Effect Module T_2" column displays the displacement of P3 due to P2's influence in the previous delta. "Cause Degree T_1" indicates the direction of P2's displacement in degrees, while "Effect Degree T_2" indicates the direction of P3's displacement in degrees.

We created a topological graph from Table 4 (Figure 7) by analyzing only the causes and effects of the points. In the table, P2 acts as a force on P3, represented in the graph as a connection from P2 to P3 with an outgoing arrow. P2 also imparts a force on P4, but P4 reciprocates with a force on P2, depicted as a connection between P2 and P4 with a double-headed arrow. This results in the formation of two graphs and clusters. The first graph on the left shows that the interaction between P2 and P4 gave rise to all other cause-and-effect relationships, as indicated by the direction of the arrows. The second graph on the right demonstrates that it was the interaction between P13 and P8 that formed all other cause-and-effect relationships.

CCA cumulates the effects of the previous steps. Consequently, Table 5 and Figure 8 show respectably the effects of step 3 and step 4, taking account the steps 1 and 2.

Table 5. GNSS (P13) becomes in this phase the most active point and GNSS (P2) works as the new resilient point of the entire landslide.

| Excitations [0,1]							
Cause		**Effect**	**Strenght**	**Cause Module T_3**	**Effect Module T_4**	**Cause Degree T_3**	**Effect Degree T_4**
P1	→	P3–	0.001854	0.04297	0.069856	178.106628°	232.431427°
	→	P10	0.00001	0.04297	0.087575	178.106628°	109.903755°
	→	P12	0.000005	0.04297	0.071665	178.106628°	123.690071°
P2	→	P4	0.012232	0.050972	0.329537	257.125000°	238.073135°
P6	→	P2	0.003544	0.17583	0.470611	196.898651°	257.011871°
	→	P5	0.001302	0.17583	0.245087	196.898651°	324.192261°
	→	P7	0.018873	0.17583	0.550277	196.898651°	227.771057°
P7	→	P6	0.030652	0.101252	0.499697	167.035507°	243.871765°
P9++	→	P11	0.006346	0.157026	0.511014	194.664459°	193.495728°
	→	P13	0.069017	0.157026	1	194.664459°	206.565063°
P11	→	P1	-0.000001	0.168732	0.05679	104.620872°	269.641907°
P13	→	P8	0.046129	0.046223	0.939036	222.510452°	223.345932°
	→	P9++	0.032914	0.046223	0.510582	222.510452°	210.772507°
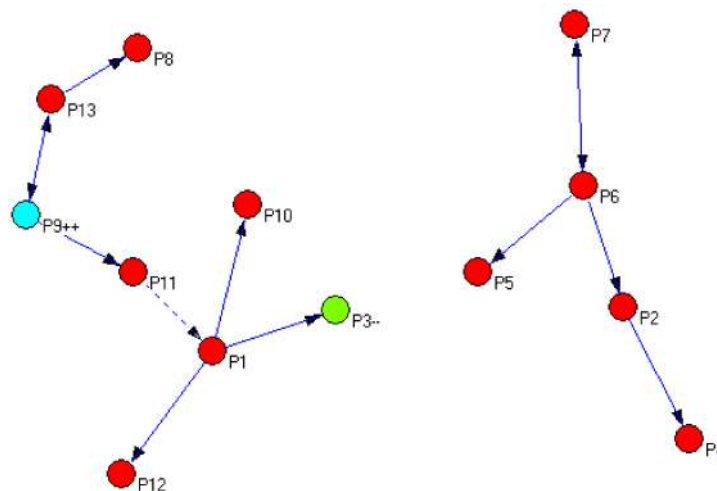
Figure 8. Graph of cause-and-effect relationships after the steps 3 and 4. The points with "++" symbol are the more active GNSS and the points with "–" symbol are the most resilient to the displacement. The landslide still displays two partially independent clusters, which differ slightly from the previous graph. The cluster on the left continues to be the dominant one, with two key points belonging to it. This time, GNSS (P9) is the most active, while GNSS (P3) remains the most resilient.

Accordingly, Table 5 depicts the delta following the one analyzed above. Here, it can be noted that P9 exerts more force than all other points and also experiences a significant amount of force from P13, while P3 experiences minimal force compared to the others. As a result, P9 is marked with the symbol "++", while P3 is marked with the symbol "–" representing the point of resilience.

In Figure 8, P9 is positioned in the center of the left graph and the arrows indicate that the interaction between P9 and P13 resulted in all other cause-and-effect interactions.

From step 4 to step 5 the landslide exhibits a distinct structure. Figure 9 illustrates the change in the cause-and-effect relationships graph and Table 6 provides additional details about this transformation. It is important to remember that this new configuration of the

landslide is a result of the cumulative effects of steps 1, 2, 3, and 4, and the CCA outputs consider the overall dynamics of the landslide.
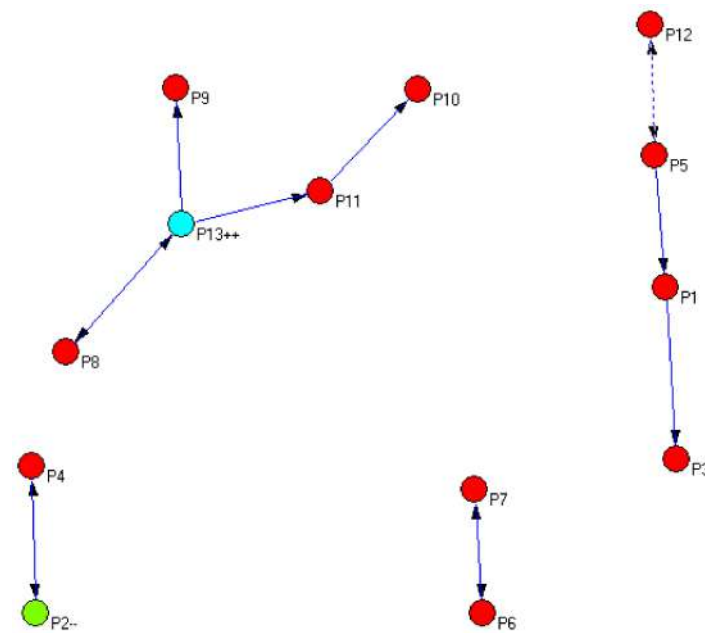

Figure 9. Graph of cause-and-effect relationships after the steps 4 and 5. The points with "++" symbol are the more active GNSS and the points with "−" symbol are the most resilient to the displacement. The landslide exhibits a singular change of structure, with the newly formed clusters still being unstable.

Table 6. GNSS (P9) becomes in this phase the most active point and GNSS (P3) remains the resilient point of the entire landslide.

Excitations [0,1]							
Cause		**Effect**	**Strenght**	**Cause Module T_4**	**Effect Module T_5**	**Cause Degree T_4**	**Effect Degree T_5**
P1	→	P3	0.000882	0.043026	0.15963	269.641907°	124.707367°
P2−	→	P4	0.0428	0.35655	0.510164	257.011871°	254.938263°
P4	→	P2−	0.22346	0.249668	0.485026	238.073135°	241.161392°
P5	→	P1	0.007402	0.185686	0.128173	324.192261°	315.850037°
	→	P12	0.000004	0.185686	0.099819	324.192261°	307.116852°
P6	→	P7	0.091283	0.378586	0.764968	243.871765°	230.276962°
P7	→	P6	0.105379	0.416908	0.710315	227.771057°	236.237747°
P8	→	P13++	0.233903	0.711443	0.856142	223.345932°	221.077576°
P11	→	P10	0.00448	0.387161	0.126509	193.495728°	174.143997°
P12	→	P5	−0.000001	0.054295	0.217289	123.690071°	80.094704°
P13++	→	P8	0.190976	0.757632	0.723519	206.565063°	220.629013°
	→	P9	0.311705	0.757632	1	206.565063°	213.416580°
	→	P11	0.014745	0.757632	0.410018	206.565063°	181.804001°

Step 5 and step 6 represent the final structure of this landslide. Table 7 and Figure 10 display the cumulative information gathered throughout the process and the resulting directed and weighted graph.

Table 7. GNSS (13) continues to be the most active point, and GNSS (P3) reverts to its role as the resilient point for the entire landslide.

| Excitations [0,1]							
Cause		**Effect**	**Strenght**	**Cause Module T_5**	**Effect Module T_6**	**Cause Degree T_5**	**Effect Degree T_6**
P1	→	P3–	0.001652	0.128173	0.152114	315.850037°	324.061981°
P2	→	P4	0.050661	0.485026	0.092368	241.161392	252.725571°
P4	→	P2	0.021211	0.510164	0.255459	254.938263	348.832031°
P5	→	P1	0.010875	0.217289	0.133274	80.094704	17.364714°
P6	→	P7	0.110596	0.710315	0.12913	236.237747	268.329376°
P7	→	P6	0.136612	0.764968	0.144331	230.276962°	242.480103°
P8	→	P13++	0.2668	0.723519	0.221791	220.629013°	182.501633°
P9	→	P10	0.016019	1	0.089762	213.416580°	224.028992°
	→	P12	0.004555	1	0.140562	213.416580°	204.407669°
P12	→	P5	-0.000001	0.099819	0.031462	307.116852°	0.000000°
P13++	→	P8	0.225587	0.856142	0.163312	221.077576°	193.715912°
	→	P9	0.340571	0.856142	0.128086	221.077576°	187.721832°
	→	P11	0.016386	0.856142	0.136656	221.077576°	154.846893°
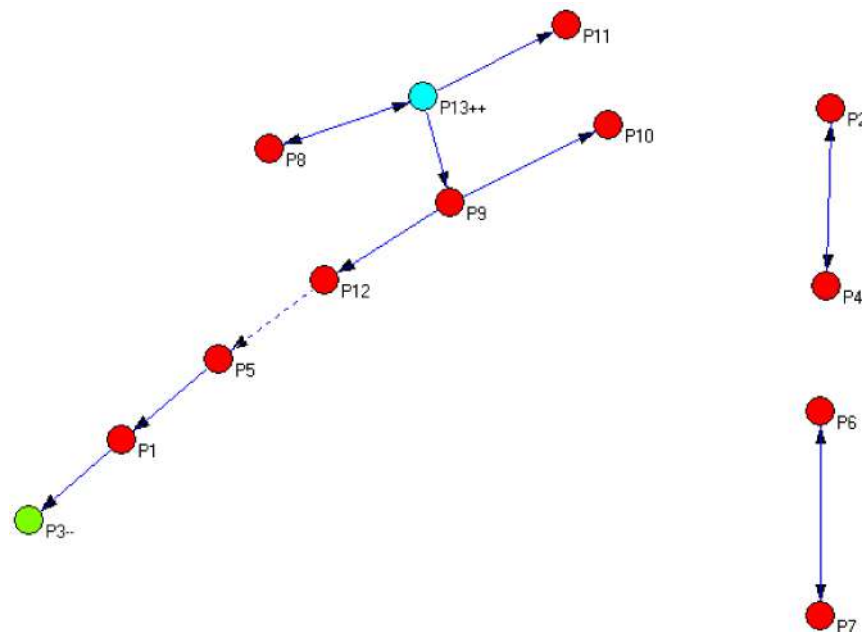
Figure 10. Graph of cause-and-effect relationships after the steps 5 and 6. The points with "++" symbol are the more active GNSS and the points with "−" symbol are the most resilient to the displacement. The landslide displays its final configuration regarding the cause-and-effect relations among the GNSS points.

At this final stage, P3 is identified as the resilient point marked "−". P1 exerts very little force on P3, while P5 experiences little force from P12 but exerts force on P1. P3 does not exert any force on any other point. P13 exerts significantly more force than the other points and is also subjected to a large amount of force from P8, therefore, marked with the symbol "++".

The left graph of Figure 10 shows that the interaction between P8 and P13 generated all other interactions, with P3 being the furthest point from P13. The arrow connecting P12 and P5 is depicted as dashed, as the force exerted by P12 on P5 is close to zero, as previously mentioned.

3.2. The Twisting Theory Algorithm (TWT)

The measurement data is processed to quantitatively determine the kinematics of the landslide using classical computation. Then, qualitative hypotheses regarding average yearly velocities are formulated, typically using a statistical approach [33]. Applying the Twisting Algorithm to slow landslides means analyzing the measurements whilst taking into account not only the values of displacements, but also the density of monitored vertices, the mutual relationships between each measured point, and the interaction between each measured point and the geometrical space where they are located. This approach provides additional information compared to analyzing displacement values alone and improves the description of landslide phenomena, as it enables the creation of a continuous model using only a few discrete points. Additionally, TWT can be used to predict the future positions of GNSS points and the global shape of the landslide at the next time step by incorporating classic Artificial Neural Networks. However, to implement TWT, several parameters must be defined:

a. The number of observational times (T) and the number of GNSS points (P), that are inherent in the data set.

b. The number of substeps (S) in which each time step must be divided (see Figures 1 and 2). The more the substeps in a simulation, the more detailed the time flow of TWT will be.

c. The number of grid cells (L) that represent the entire region where the landslide can occur. The higher the density, the more spatial details will be defined by TWT.

d. A matrix of parameters (F) representing the rigidity/elasticity of each cell of the analyzed area. In this application, it will be assumed that each grid cell has the same elasticity.

The label "Assisi (T8-P13-S10-F25-L2)" indicates that the TWT analysis of the Assisi landslide involved 8 observational time steps (T), monitored by 13 GNSS points (P). Each time step was divided into 10 substeps (S), and each grid cell is square, has a fixed elasticity of 25 (F), and the side length is equal to 2 pixels (L). The parameters T and P are determined by the data set, while variations in the parameters S, L, and F alter the temporal or spatial resolution of the simulation or the strength of deformation in the landslide. However, the overall shape of the landslide remains relatively unchanged. Figure 11 displays the shape of the Assisi landslide as defined by geologists at the end of the observational period and the positions and displacements of the 13 GNSS points, which are the only information available for the TWT analysis.

Consequently, the data set processed by TWT has the following format:

$$AssisiData = \left\{ \left\{ x_i^t, y_i^t \right\}_{i=1}^{P=13} \right\}_{t=1}^{T=8}. \tag{37}$$

Figure 12a,b display the results of the TWT, where different grid densities are used. The larger cells indicate areas where the landslide kinematics is faster, while deformed cells indicate areas with a higher concentration of potential energy. Figure 13 uses a color code to present the same study area, with a higher intensity of red indicating a greater degree of deformation caused by the landslide.

Figure 12 detects the local deformations of the area caused by the landslide, indicating the nonuniform directions of the landslide and variations in its kinematics. Figure 13 depicts the overall shape of the landslide, which is similar to the shape of the manually drawn landslide by geologists after a thorough investigation on the site (see Figure 11a).
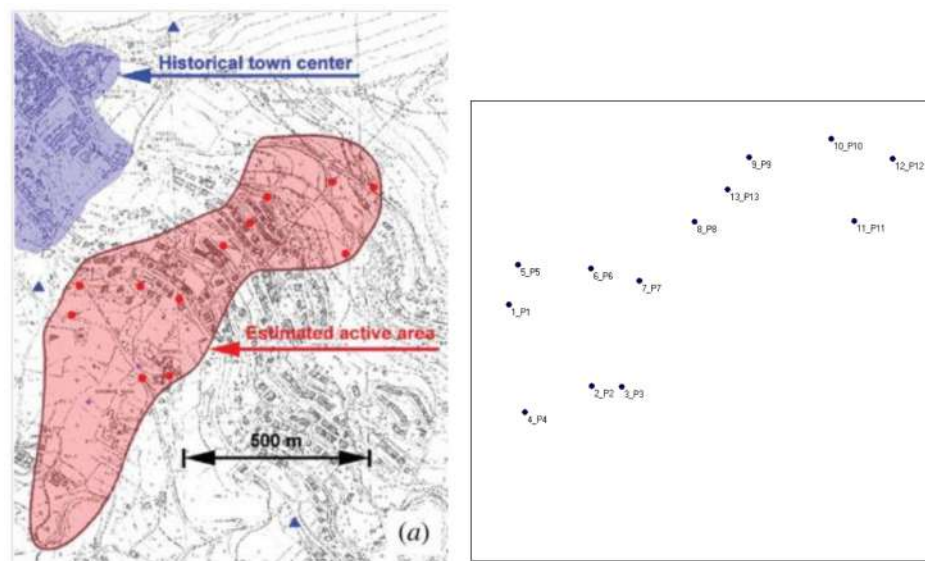
Figure 11. (**a**) The shape of the Assisi landslide at the end of the observational period (2008); (**b**) The positions of the 13 GNSS points at the start of the observational period (1995), serving as the unique input for the TWT.
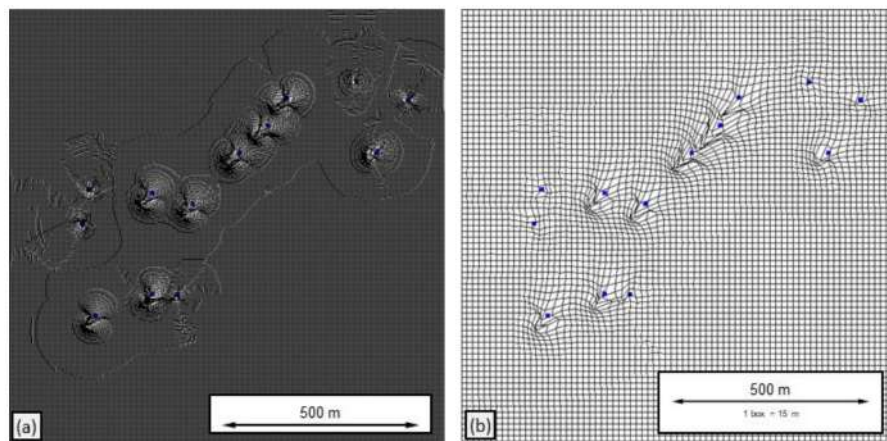

Figure 12. (**a**) The output of TWT on the Assisi landslide (T8-P13-S10-F10-L2); (**b**) The output of TWT on the Assisi landslide (T8-P13-S10-F30-L8).
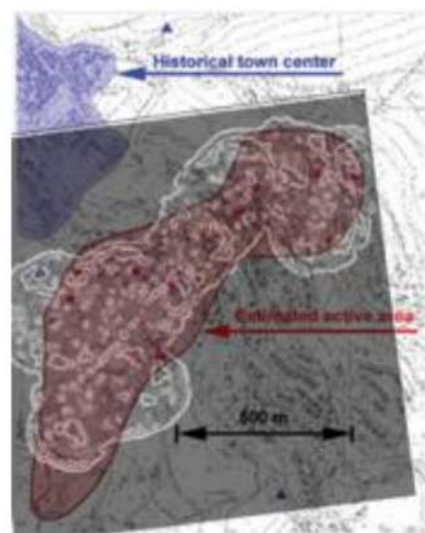

Figure 13. The output of TWT on the Assisi landslide (T8-P13-S20-F20-L2) using a color code.

3.3. The Predictive TWT

The preprocessing of the TWT grids is necessary in order to prepare the data for ANNs learning and subsequent prediction (as outlined in Section 2.3). As the Assisi landslide consists of 13 GNSSs monitored over 8 observation campaigns, TWT will generate 6 grids in this case. Five of these grids will be coded to form the training set for learning with the SVCm ANN (to associate each input time (t) with the position of the GNSSs at time (t + 1)). The last grid (the sixth) will be used as the blind testing set, to predict the position of the GNSSs at step 7 and assess the accuracy of the prediction phase. Table 8 illustrates how the data of a typical grid is coded

Once the SVCm has estimated the new positions of the GNSSs at time step #7, these estimations will be added to the dataset. TWT will then be run again to generate the entire grid at step #7. Two comparisons can be made at this point: (a) a correlation between the real positions and the predicted positions of GNSSs at step #7, and (b) a comparison between the estimated grid and the grid generated by TWT using the real positions of the GNSSs at step #7. A simple linear correlation between the real grid at step #7 and the estimated grid at the same step will be used to measure the accuracy of the prediction.

Figure 14a shows the resulting state generated at the time step #7 by TWT using all available data, while Figure 14b displays the TWT state generated using the predicted positions of the 13 GNSS points at the same time step. The linear correlation between the longitude and latitude of the two maps is 0.999712 for longitude and 0.999747 for latitude, respectively, resulting in an average correlation of 0.9997295.

Figure 15 shows instead the correlation between the real positions of the GNSSs at step #7, and the estimation of the SVCm at the same step.

This application showcases the effectiveness of crowd clustering and twisting theory algorithms in modeling the dynamics of slow-moving landslides. The predicted movement of the landslide in the next time period is found to correspond closely with its actual movement.
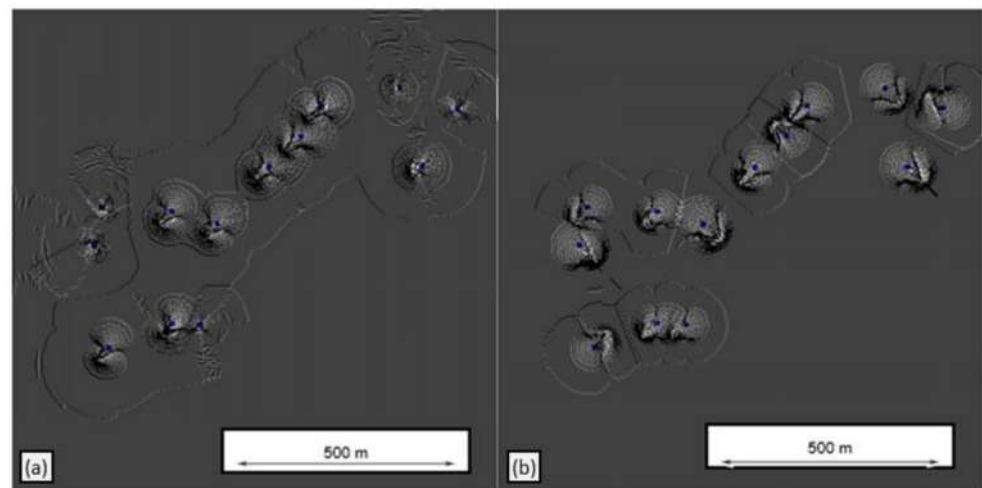

Figure 14. (**a**) TWT with real data; (**b**) TWT using predicted data of the last step.

Table 8. How all the GNSS points (yellow) must be coded with the grid points around them (columns).

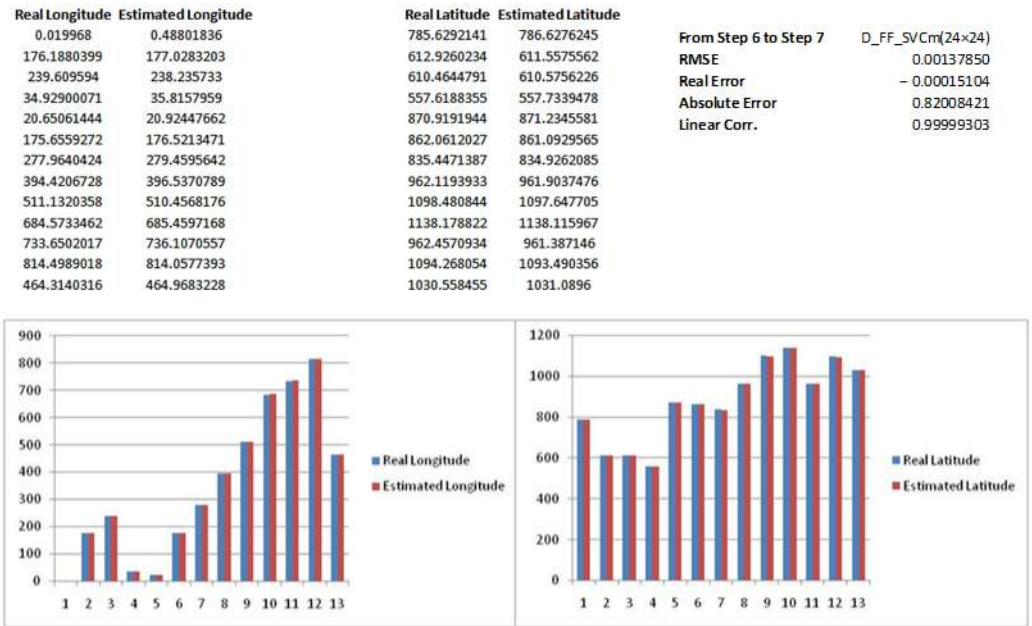| Time Step	Coord.	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6	Point 7	Point 8	Point 9	Point 10	Point 11	Point 12	Point 13
	x	0.004253	176.194458	239.602814	34.952839	20.625507	175.710831	278.025146	394.52475	511.235687	684.584656	733.727356	814.494507	464.439484
	y	785.632751	612.993774	610.464661	557.697205	870.910645	862.164673	835.528748	962.20166	1098.542847	1138.179443	962.467896	1094.271606	1030.632324
	x	−34.371132	141.171097	199.122299	−5.431697	−5.462948	141.161331	258.783783	376.433807	492.280426	669.680725	698.658875	787.339905	434.339417
	y	792.45105	615.077881	615.067688	557.165955	879.317871	850.369507	821.381409	967.991699	1085.648315	1143.585815	968.04895	1085.674194	1025.897339
	x	−34.384129	141.164474	199.124237	−5.438146	−5.461166	141.169312	258.800354	376.446411	492.265625	669.694092	698.664124	787.345581	434.34549
	y	761.693726	586.110962	586.105591	528.199158	850.359436	821.416809	792.437683	939.044006	1056.673218	1114.640137	939.082825	1056.719604	996.943298
	x	−5.461305	170.12471	229.892838	23.511894	23.525455	170.128159	287.734192	405.402313	523.063049	698.660522	727.617371	816.303833	463.325378
	y	821.398987	645.840332	645.877686	586.109497	908.306274	879.337219	850.331848	996.960266	1114.620972	1172.566772	997.010193	1114.637451	1056.693359
Time N	x	0.004253	176.194458	239.602814	34.952839	20.625507	175.710831	278.025146	394.52475	511.235687	684.584656	733.727356	814.494507	464.439484
	y	785.632751	612.993774	610.464661	557.697205	870.910645	862.164673	835.528748	962.20166	1098.542847	1138.179443	962.467896	1094.271606	1030.632324
	x	−5.401855	170.115128	229.896591	23.512184	23.514975	170.134537	287.76767	405.423584	523.061218	698.653931	727.620056	816.310181	463.323578
	y	761.671997	586.100281	586.151428	528.18927	850.375305	821.420898	792.444702	939.060913	1056.698608	1114.63855	939.086243	1056.723267	996.960876
	x	23.515982	199.119385	258.854736	52.472984	52.476437	199.083801	316.723663	434.34549	552.03418	727.618286	756.582214	845.264038	492.265625
	y	821.415955	645.840942	645.880127	586.110352	908.296997	879.332581	850.361023	996.943298	1114.631836	1172.564453	997.005554	1114.63855	1056.673218
	x	23.537193	199.122299	258.857361	52.487679	52.46249	199.076218	316.743805	434.35611	552.0354	727.611877	756.579895	845.268799	492.268951
	y	792.463623	615.067688	615.112	557.164734	879.322754	850.364746	821.420837	967.993408	1085.672729	1143.597168	968.047791	1085.682251	1025.90625
	x	0.004253	176.194458	239.602814	34.952839	20.625507	175.710831	278.025146	394.52475	511.235687	684.584656	733.727356	814.494507	464.439484
	y	785.632751	612.993774	610.464661	557.697205	870.910645	862.164673	835.528748	962.20166	1098.542847	1138.179443	962.467896	1094.271606	1030.632324
Time N + 1	x	0.006601	176.193222	239.611572	34.945271	20.634718	175.710144	278.024872	394.517456	511.233154	684.593506	733.717896	814.501038	464.426666
	y	785.635742	612.987732	610.455139	557.669678	870.916565	862.139954	835.519531	962.183044	1098.537842	1138.171143	962.451477	1094.261841	1030.62561

Real Longitude	Estimated Longitude		Real Latitude	Estimated Latitude
0.019968	0.48801836		785.6292141	786.6276245
176.1880399	177.0283203		612.9260234	611.5575562
239.609594	238.235733		610.4644791	610.5756226
34.92900071	35.8157959		557.6188355	557.7339478
20.65061444	20.92447662		870.9191944	871.2345581
175.6559272	176.5213471		862.0612027	861.0929565
277.9640424	279.4595642		835.4471387	834.9262085
394.4206728	396.5370789		962.1193933	961.9037476
511.1320358	510.4568176		1098.480844	1097.647705
684.5733462	685.4597168		1138.178822	1138.115967
733.6502017	736.1070557		962.4570934	961.387146
814.4989018	814.0577393		1094.268054	1093.490356
464.3140316	464.9683228		1030.558455	1031.0896

From Step 6 to Step 7	D_FF_SVCm(24×24)
RMSE	0.00137850
Real Error	− 0.00015104
Absolute Error	0.82008421
Linear Corr.	0.99999303

Figure 15. Comparison between real latitude and longitude and predicted latitude and longitude of the 13 GNSSs at the step #7.

4. Application 2: The Corvara Landslide

Corvara in Badia is a town located in the Autonomous Province of Bolzano, and it is the largest town in Val Badia, at the heart of the Dolomites (see map in Figure 4). The town is impacted by a slow-moving landslide, which is causing repeated damage to the SS 244 highway in the Campolongo Pass area, as well as posing a threat to the town of Corvara. To study this landslide, a historical series of 51 GNSS stations monitored over 26 measurement campaigns conducted from 2001 to 2008 by the Province of Bolzano was collected, with an accuracy of less than one centimeter. Figure 16 shows the processing of the 51 GNSS stations using TWT.
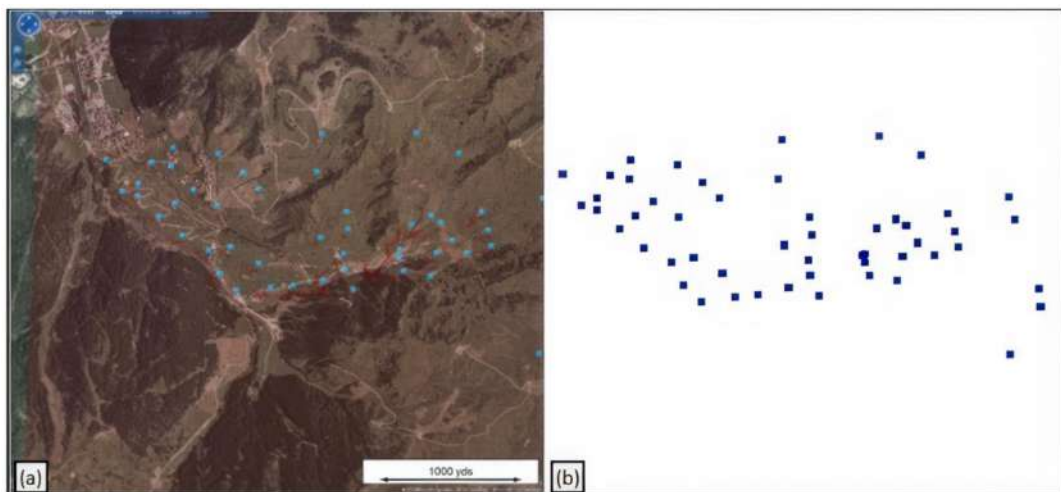

Figure 16. The 51 GNSS stations monitored over 26 time units from 2001 to 2008: (**a**) map that displays the locations of the GNSS stations; (**b**) digital format that is ready for the application of the TWT.

4.1. Corvara Landslide: TWT Application

The two objectives of TWT in this case are:

1. To reconstruct the movement of the entire landslide, including its edges, the varying sliding speeds in each region, and the potential subsidence areas;
2. To predict its evolution in the future based on the data collected from 2001 to 2005, in order to anticipate the shape of the landslide in 2008.

Figures 17–19 illustrate the reconstruction of the landslide shape after 2008 by a group of geologists [34,35].
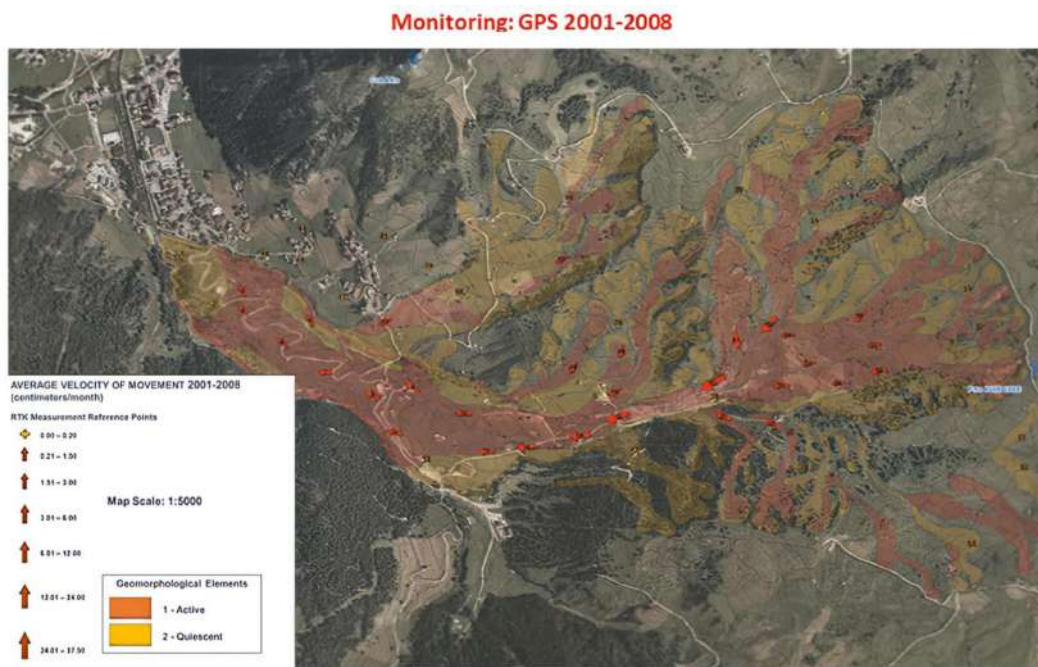
Figure 17. Reconstruction of the Corvara landslide after 2008. The size of the arrows indicates speed, while light brown represents inactive landslides and deep brown represents active landslides.
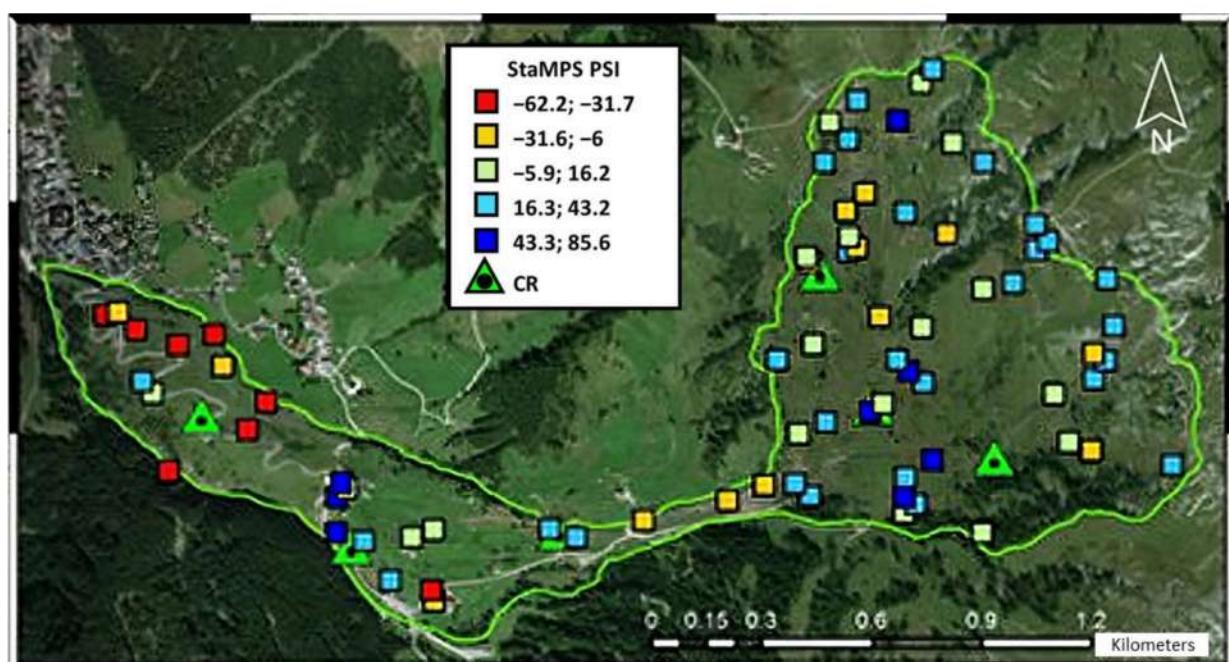
Figure 18. Monitoring system for the Corvara landslide in 2016.
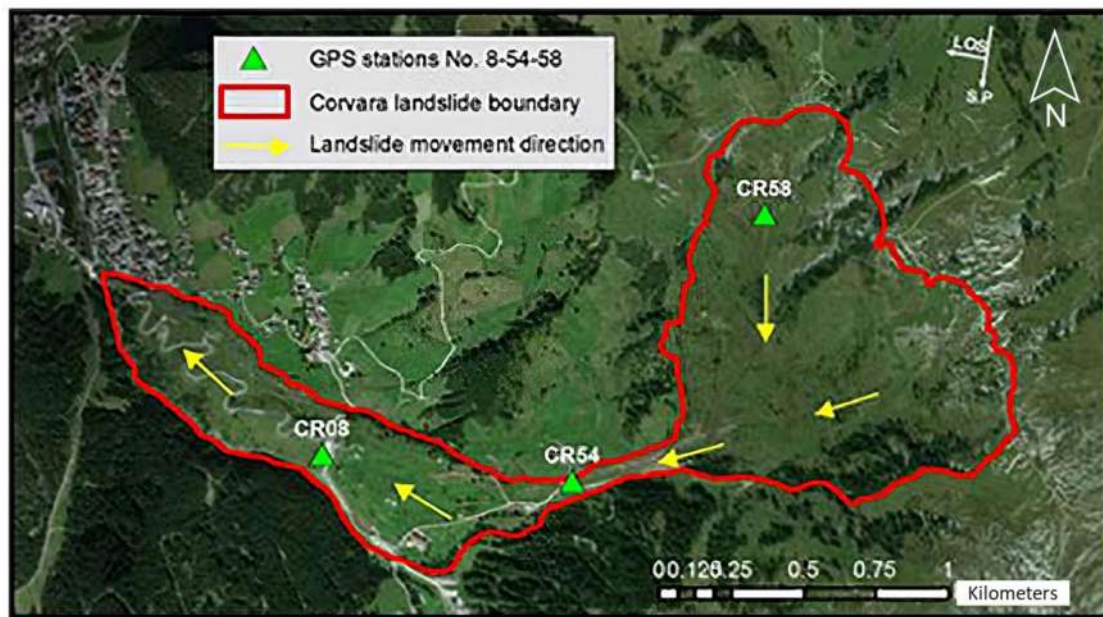
Figure 19. The active Corvara landslide up to 2016.

The TWT algorithm was applied to 51 points and was trained using the 26 observational monitoring collections (see Figure 16b) with the parameters T26-P51- S10-F10-L2. At the end of training phase TWT reconstructed the entire landslide pattern (Figure 20). Finally, in Figure 21, the reconstruction of the landslide perimeter is compared to the actual perimeter assessed post facto by local experts.
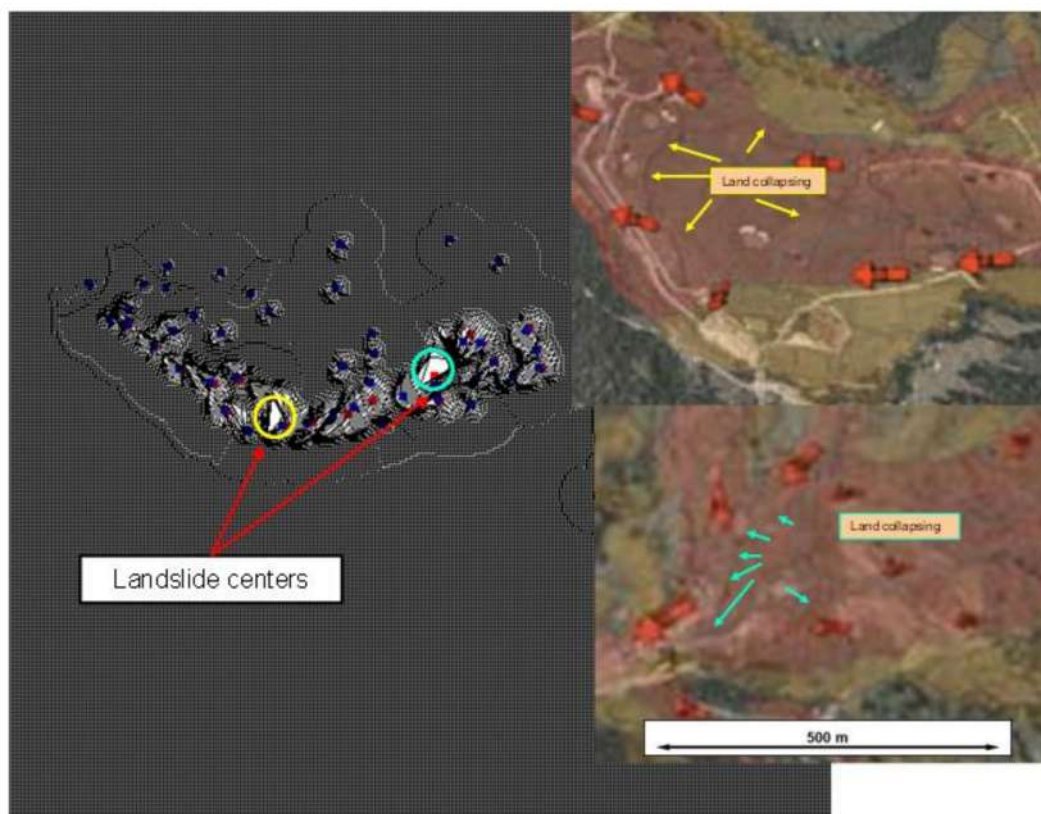

Figure 20. TWT final pattern and (on right) the projection on the geographical map of the two main location where the landslide occurred.
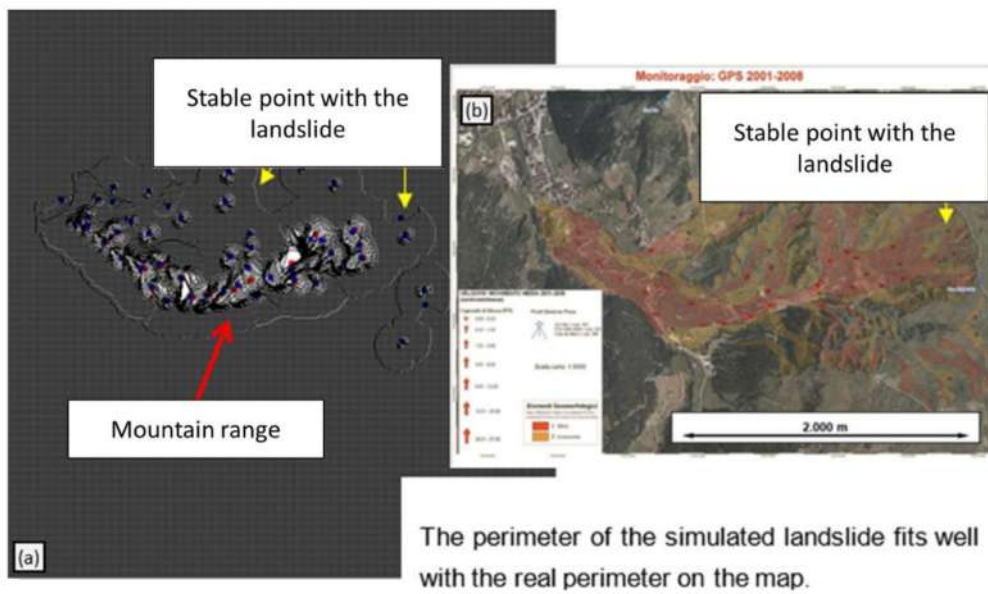
Figure 21. Comparison of the final perimeter of the landslide: (**a**) TWT's final pattern, (**b**) the real map.

4.2. Corvara Landslide: TWT Prediction

The second objective of this application was to predict the Corvara landslide. A deep ANN (SVCm 24x24) was applied to the first 12 grids generated by TWT (digital grid maps). Then, the trained ANN was tested using 51 GNSS locations from the 13th to the 22nd observational data sets, to assess the accuracy of the ANN predictions.

Table 9 illustrates the construction of the training set. Table 10 displays the test results from step 13 to step 22, while Table 11 compares the results of predicting the GNSS positions in the 22nd observational data set with the actual GNSS positions at that step. The accuracy of the ANN is quite good; so, an attempt was made to reconstruct the entire pattern of the landslide at step 22 using only the predictive estimations of the ANN, as was done previously with the Assisi application. A comparison with the pattern of the TWT map trained with all available data reveals that the two patterns are highly similar.

Table 9. Data for the ANN SVCm Training Set.

Imput Vector = 18 (x-y of each GNSS + the x-y of first 8 neighbours)	Imput Vector = 2 (x-y of the GNSS at the next step)	Number of Patterns
Step 1	Step 2	51
Step 2	Step 3	51
Step 3	Step 4	51
Step 4	Step 5	51
Step 5	Step 6	51
Step 6	Step 7	51
Step 7	Step 8	51
Step 8	Step 9	51
Step 9	Step 10	51
Step 10	Step 11	51
Step 11	Step 12	51
Step 12	Step 13	51
Number of Training Steps		12
Number of Training Patterns		612

Table 10. Main Cost Functions: Average Prediction Accuracy of the Trained ANN Tested on GNSS Positions from Step 13th to Step 22nd. Error is the difference between the actual position of the sensors in the next step and the predicted position of each sensor in the next step.

ANN	RMSE	Real Error	Absolute Error	Linear Corr.
D_FF_SVCm (48 × 48)(Step22)	0.00630936	0.00159118	11.36599193	0.99970293

Table 11. The prediction of the ANN regarding the 51 GNSS locations during the 22nd observation data collection effort.

Prediction at the Step 22nd	X		Y	
	Real	ANN	Real	ANN
GNSS_1	399.974	465.6641	8735.555	8738.035
GNSS_2	515.66	530.3192	8536.587	8536.059
GNSS_3	620.589	609.1868	8588.925	8586.828
GNSS_4	620.508	618.0858	8510.746	8510.3
GNSS_5	865.304	855.3544	8470.225	8471.014
GNSS_6	763.245	743.9014	8383.587	8385.679
GNSS_7	919.8909	922.0909	8261.178	8261.914
GNSS_8	1093.098	1091.929	8173.648	8174.809
GNSS_9	1238.316	1252.554	8202.036	8201.741
GNSS_10	1422.422	1424.259	8099.366	8098.508
GNSS_11	1171.875	1175.442	8023.99	8024.207
GNSS_12	1288.773	1288.494	7917.83	7922.718
GNSS_13	1505.48	1505.159	7948.988	7949.073
GNSS_14	1648.827	1642.184	7965.847	7967.287
GNSS_15	1842.655	1845.073	8007.324	8009.797
GNSS_16	1978.895	1971.305	8082.523	8082.572
GNSS_17	2045.661	2026.637	7955.034	7953.752
GNSS_18	1973.314	1964.695	8182.971	8184.664
GNSS_19	1821.737	1814.413	8281.33	8283.653
GNSS_20	1983.416	1971.528	8459.516	8461.534
GNSS_21	2366.539	2362.008	8087.827	8087.436
GNSS_22	2542.893	2549.193	8057.985	8057.367
GNSS_23	2576.425	2577.606	8210.325	8210.667
GNSS_24	2675.274	2677.164	8296.943	8297.728
GNSS_25	2932.199	2950.701	8269.991	8270.685
GNSS_26	2782.021	2803.382	8219.553	8220.042
GNSS_27	2911.671	2922.029	8370.89	8371.251
GNSS_28	2861.902	2852.833	8487.488	8484.992
GNSS_29	3255.407	3224.392	8594.464	8592.706
GNSS_30	3297.295	3282.94	8443.846	8443.185
GNSS_31	3451.029	3413.756	8004.191	8012.435
GNSS_32	3461.288	3421.679	7886.743	7885.536
GNSS_33	3268.818	3258.694	7581.099	7591.052
GNSS_34	2698.638	2697.54	8864.894	8886.041
GNSS_35	2422.933	2426.773	8989.194	8958.019
GNSS_36	1806.663	1793.816	8959.01	8940.632

Table 11. *Cont.*

Prediction at the Step 22nd	X		Y	
	Real	ANN	Real	ANN
GNSS_37	1779.911	1778.615	8711.944	8715.452
GNSS_38	2336.938	2331.204	8166.558	8164.779
GNSS_39	1297.579	1304.43	8686.978	8686.729
GNSS_40	1407.376	1409.538	8584.639	8583.33
GNSS_41	1140.48	1150.454	8464.103	8464.272
GNSS_42	1133.942	1139.68	8804.358	8813.357
GNSS_43	838.0848	829.4269	8834.597	8842.23
GNSS_44	829.9848	821.1478	8711.107	8712.023
GNSS_45	702.3591	693.5692	8735.002	8736.819
GNSS_46	982.203	988.3502	8563.483	8560.64
GNSS_47	1997.305	1984.407	8345.618	8348.143
GNSS_48	2599.636	2591.549	8410.392	8410.652
GNSS_49	2532.747	2521.965	8450.682	8452.008
GNSS_50	2414.119	2401.271	8389.094	8390.367
GNSS_51	2324.149	2302.229	8215.997	8218.747

In order to evaluate the quality of the reconstruction visually, Figure 22 presents a comparison of maps. The maps include the real map with annotations made by geologists, a reconstruction using all available data, a prediction map with the intensity of the landslide represented by colors, and a reconstruction using only the estimations of the ANN, which was trained and tested up to specific steps.
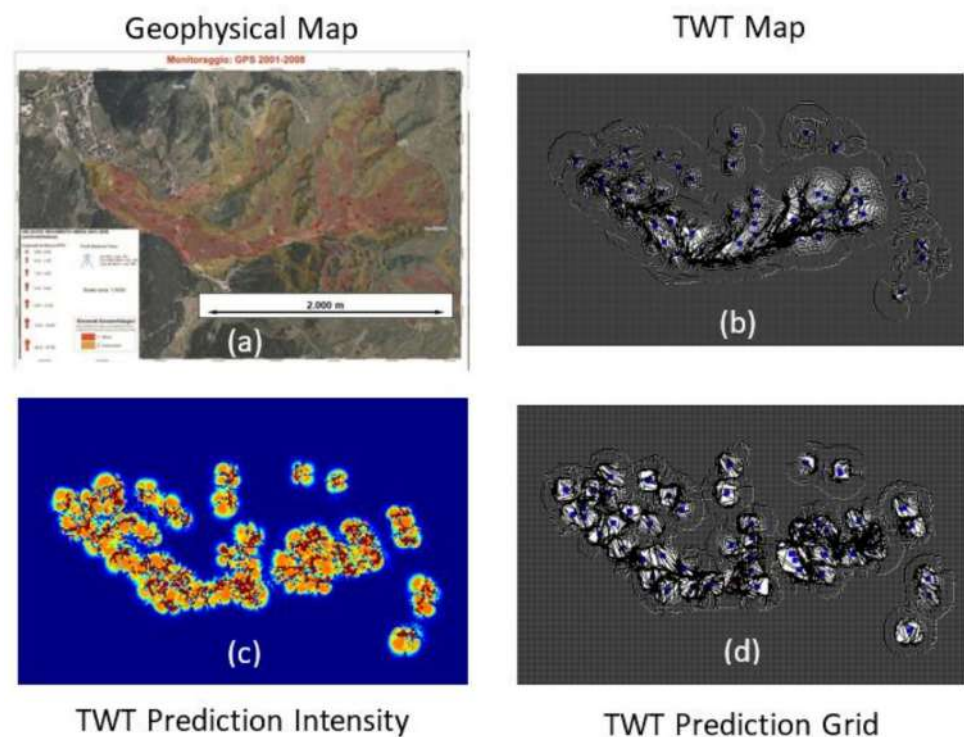
Figure 22. (**a**) Real map annotated by geologists; (**b**) TWT reconstruction using all available data; (**c**) Prediction map showing the intensity of the landslide through colors; (**d**) TWT reconstruction using only the estimations of the ANN trained up to the 12th step and tested up to the 22nd step.

4.3. Corvara Landslide: CCA Application

Figure 23 shows the 51 GNSS monitoring the Corvara landslide on the digital map with their labels. Figures 24 and 25 depict the graphs of cause-and-effect relationships among the GNSS, as identified by CCA at the end of 26 temporal intervals. Figure 24 illustrates the excitatory cause-and-effect relationships among the GNSS, while Figure 25 shows the inhibitory cause effect relations among the same GNSS.

Both in the excitatory graph (Figure 24) and in the inhibitory graph (Figure 25), the main cluster identifies GNSS#51 as the driving force of the entire landslide, as it activates most of the GNSSs around it and expands its effects through GNSS#16 to faraway GNSSs and to the border of GNSSs belonging to minor clusters. GNSS#51 also acts as the main cause of the inhibitory effects of the landslide, through GNSS#50 and GNSS#21 (See Figure 26).
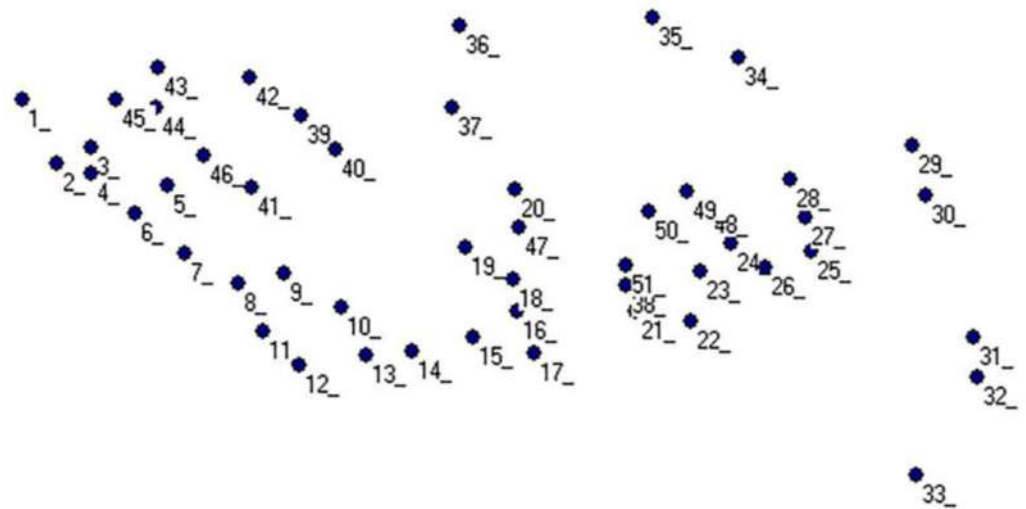
Figure 23. Digital map of the Corvara landslide with the positions of the 51 GNSS labeled.
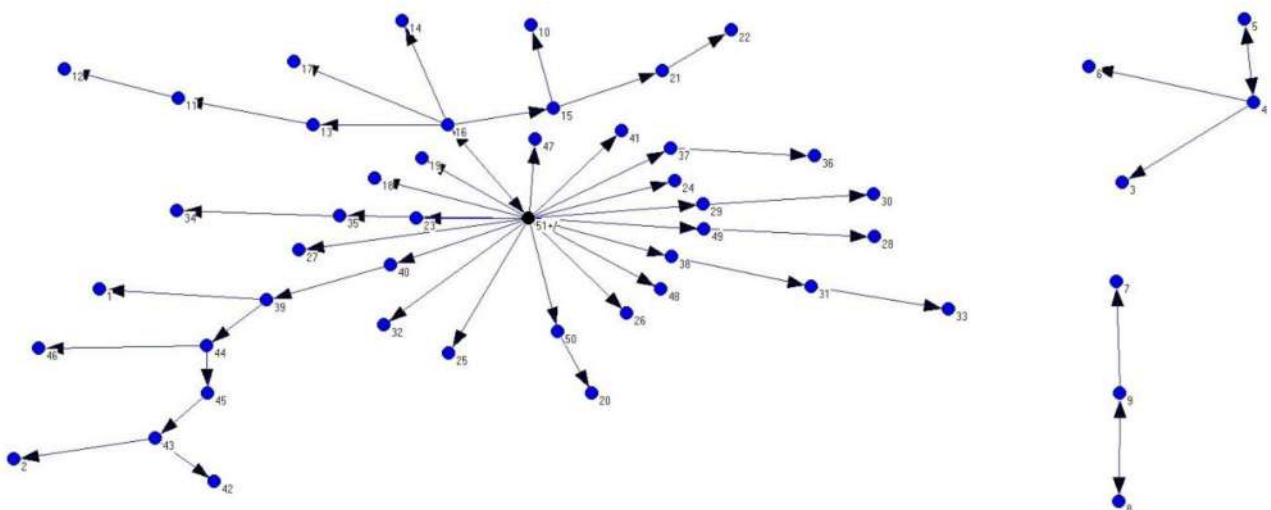
Figure 24. Excitatory cause-and-effect relationships among the 51 GNSSs.
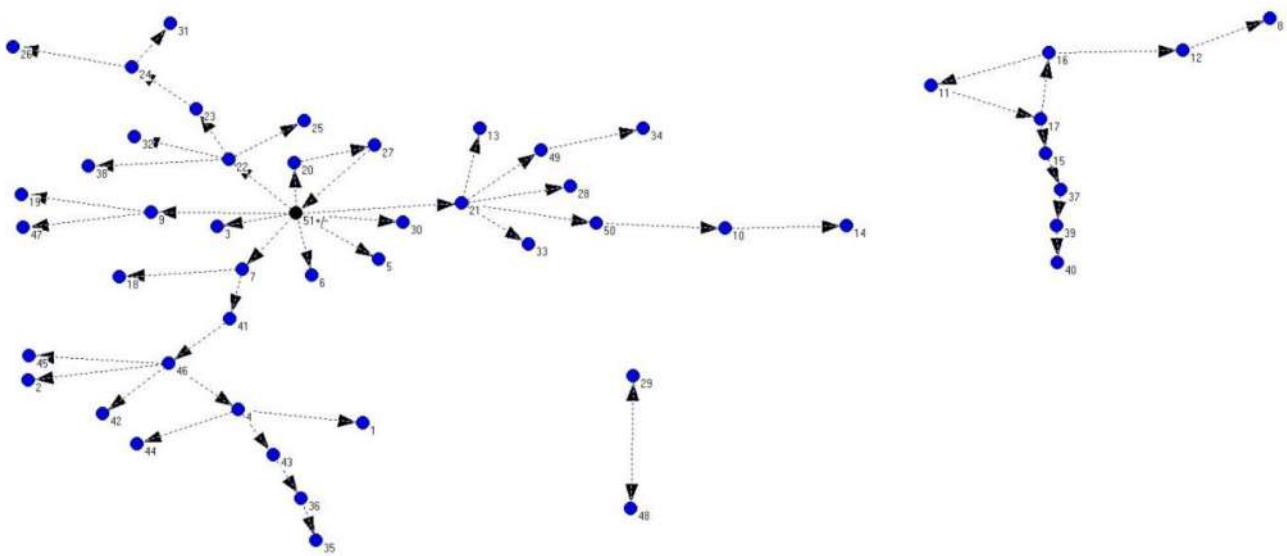
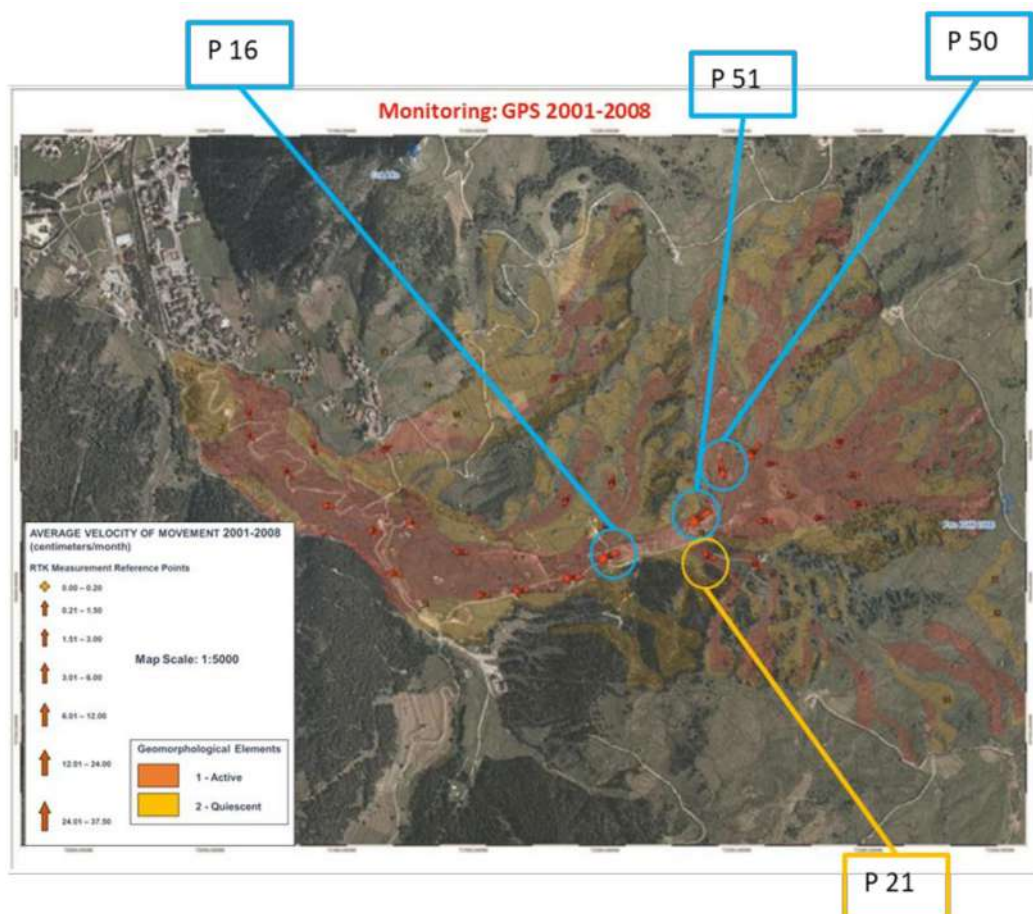**Figure 25.** Inhibitory casual relationships among the 51 GNSSs.

Figure 26. Main excitatory and inhibitory cause-and-effect relationships among the 51 GNSSs.

In contrast to the Assisi landslide, we also reported the inhibitory cause-and-effect relationships here. Let us first analyze the graph of excitatory cause-and-effect relationships. We can see that Figure 24 is divided into three subgraphs: the largest one on the left and two smaller ones on the right. A similar division is present in Figure 25, which depicts the inhibitory cause-and-effect relationships. In both larger left subgraphs in Figures 24 and 25,

if we examine the direction of the arrows, we can observe that point 51 is the source of all the inhibitory and excitatory relationships in the two graphs. So, on the one hand, it has an important excitatory function in the landslide while, on the other hand, it also acts as a brake. For instance, in the inhibitory cause-and-effect relationship, point 51 is connected with point 3; however, in Figure 24, point 3 does not belong to the left subgraph, but to the right subgraph.

5. Discussion

Classical studies in various countries (see studies conducted in localized areas in Iran [15–17], South Korea [36], Saudi Arabia [37], Japan [38], India [39], and China [40–42]) often followed a typical approach: collecting expert-identified variables, gathering data from landslides and geomorphological risk situations in order to directly estimate the susceptibility levels of the examined area.

However, this procedure has some limitations and weaknesses: significant time and expertise is needed to select the significant features, preprocessing requires a lot of work, and the historical sample may not be representative enough of the phenomenon, leading to potentially incorrect classification of new geomorphological patterns. Additionally, the entire process lacks transparency in defining the cause and causal chain between the various parts of the territory and the introduced patterns, leading to a lack of clarity in defining the output.

From this perspective, the approach presented here aims to obtain specific answers about the geodynamics of locations by focusing on the singularities of the territory and their interconnections. TWT and CCA are adaptive systems that take into account the individual landslide and the positioned localization sensors, and provide a map of the local deformations that are occurring in real time throughout the territory. Neural networks are trained to predict the position of the sensors in the subsequent steps. These predictions are reintroduced into the two algorithms, which redraw the global shape of the landslide and estimate the cause-and-effect relationships between the sensors.

As a result, the identification of the landslide shape and its progression is explained locally and the various parts of the territory are localized through explicit and local cause-and-effect relationships. This supports analysts to understand, in advance, the path of the entire landslide and which parts of the territory need to be secured to halt the process. Compared to traditional systems or even deep-learning systems, TWT is proposed as a complementary method, as it aims to leverage the peculiarities of different types of predictive models capable of providing non-redundant information and perspectives. Using an ecosystem approach, multiple machine learning and evolutionary algorithms collaborate on the same database to extract a higher quantity of information, which can be advantageous for enhancing landslide susceptibility maps by offering an advanced understanding of on-site applications.

The main point we would like to highlight is that the current potential of applications of artificial intelligence tools and machine learning algorithms is not limited solely to applying a large and complex artificial neural network (ANN) to a problem with a large volume of data. To solve a practical problem, it can be more profitable to consider our machine learning algorithms as a complex ecosystem of algorithms whose cooperative architecture must be found in a new way every time. The more different algorithms we have, the more we can find an efficient solution to a new problem; in artificial intelligence, as in nature, diversity is the greatest asset of any evolution.

6. Conclusions

The presented theory is useful not only for monitoring slowly moving landslides, but it can also model the dynamics and identify the location of where the landslide is most "energetic" and its potential source. The theory, when combined with ANNs (deep and shallow [42]), it can also predict the movement of a landslide with high accuracy. The main objectives of the theory, to find cause-and-effect relationships within a landslide field over

time and determine the shape of the landslide, have been accomplished. The model can be applied to large areas of national territory, with the only requirement being the placement of GNSS sensors in the areas of interest. From the comparison between the model estimates and the expert's posterior measurements, a linear correlation of 0.9997 was obtained at both examined sites (Assisi and Corvara). Further study and additional monitoring instruments may improve the accuracy of the predicted movement of landslides.

However, a more general conclusion can be drawn about TWT. TWT is a new algorithm that expands the typology of algorithms within the Artificial Adaptive Systems family. In more detail, we propose considering TWT as a unique type of Artificial Neural Network (as discussed at the end of Section 2.2).

In this article, we aim to show an architecture in which three different adaptive algorithms (CCA, TWT, and classical ANNs) work together to detect the hidden information within a simple dataset: cause-and-effect relationships, the hidden forms of a landslide, and the prediction of the next forms of such a landslide. This is just a small example of how to approach a new problem more effectively, moving from a basic thought (what is the best algorithm to apply) to a complex architecture of thoughts (which is the most efficient artificial ecosystem capable of simulating the problem in a reasonable way). This research represents only a small step in this direction, but we believe that the methodological approach we propose is clearly outlined here: not a single machine learning algorithm to be applied everywhere, but many different learning algorithms capable of designing a specific artificial ecosystem in which the data can reveal most of its hidden information.

Data Availability Statement: The data are not publicly available due to legal agreements between the research institution and collaborating parties.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Huang, F.; Zhang, J.; Zhou, C.; Wang, Y.; Huang, J.; Zhu, L. A Deep Learning Algorithm Using a Fully Connected Sparse Autoencoder Neural Network for Landslide Susceptibility Prediction. *Landslides* **2020**, *17*, 217–229. [CrossRef]
2. Conforti, M.; Pascale, S.; Robustelli, G.; Sdao, F. Evaluation of Prediction Capability of the Artificial Neural Networks for Mapping Landslide Susceptibility in the Turbolo River Catchment (Northern Calabria, Italy). *Catena* **2014**, *113*, 236–250. [CrossRef]
3. Gomez, H.; Kavzoglu, T. Assessment of Shallow Landslide Susceptibility Using Artificial Neural Networks in Jabonosa River Basin, Venezuela. *Eng. Geol.* **2005**, *78*, 11–27. [CrossRef]
4. Bui, D.T.; Tsangaratos, P.; Nguyen, V.-T.; Liem, N.V.; Trinh, P.T. Comparing the Prediction Performance of a Deep Learning Neural Network Model with Conventional Machine Learning Models in Landslide Susceptibility Assessment. *Catena* **2020**, *188*, 104426. [CrossRef]
5. Dong, V.D.; Abolfazl, J.; Mahmoud, B.; Davood, M.-G.; Qi, C.; Hossein, M.; Tran, V.P.; Hai-Bang, L.; Tien-Thinh, L.; Phan, T.T.; et al. A Spatially Explicit Deep Learning Neural Network Model for the Prediction of Landslide Susceptibility. *Catena* **2020**, *188*, 104451. [CrossRef]
6. Rudin, C. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef]
7. Gunning, D.; Stefik, M.; Choi, J.; Miller, T.; Stumpf, S.; Yang, G.-Z. XAI-Explainable Artificial Intelligence. *Sci. Robot.* **2019**, *4*, eaay7120. [CrossRef]
8. Adadi, A.; Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. [CrossRef]

9.   Wang, H.; Zhang, L.; Yin, K.; Luo, H.; Li, J. Landslide Identification Using Machine Learning. *Geosci. Front.* **2021**, *12*, 351–364. [CrossRef]
10.   Luti, T.; Segoni, S.; Catani, F.; Munafò, M.; Casagli, N. Integration of Remotely Sensed Soil Sealing Data in Landslide Susceptibility Mapping. *Remote Sens.* **2020**, *12*, 1486. [CrossRef]
11.   Xiao, T.; Segoni, S.; Chen, L.; Yin, K.; Casagli, N. A Step beyond Landslide Susceptibility Maps: A Simple Method to Investigate and Explain the Different Outcomes Obtained by Different Approaches. *Landslides* **2020**, *17*, 627–640. [CrossRef]
12.   Cui, Y.; Cheng, D.; Choi, C.E.; Jin, W.; Lei, Y.; Kargel, J.S. The Cost of Rapid and Haphazard Urbanization: Lessons Learned from the Freetown Landslide Disaster. *Landslides* **2019**, *16*, 1167–1176. [CrossRef]
13.   Froude, M.J.; Petley, D.N. Global Fatal Landslide Occurrence from 2004 to 2016. *Nat. Hazards Earth Syst. Sci.* **2018**, *18*, 2161–2181. [CrossRef]
14.   Huang, R.; Fan, X. The Landslide Story. *Nat. Geosci.* **2013**, *6*, 325–326. [CrossRef]
15.   Azarafza, M.; Azarafza, M.; Akgün, H.; Atkinson, P.M.; Derakhshani, R. Deep Learning-Based Landslide Susceptibility Mapping. *Sci. Rep.* **2021**, *11*, 24112. [CrossRef]
16.   Nikoobakht, S.; Azarafza, M.; Akgün, H.; Derakhshani, R. Landslide Susceptibility Assessment by Using Convolutional Neural Network. *Appl. Sci.* **2022**, *12*, 5992. [CrossRef]
17.   Nanehkaran, Y.; Mao, Y.; Azarafza, M.; Kockar, M.; Zhu, H.-H. Fuzzy-Based Multiple Decision Method for Landslide Susceptibility and Hazard Assessment: A Case Study of Tabriz, Iran. *Geomech. Eng.* **2021**, *24*, 407–418. [CrossRef]
18.   Massimi, V.; Asadi-Zeydabady, M.; Buscema, M.; Dominici, D.; Lodwick, W.; Simeoni, L. The Contribution of Artificial Adaptive System to Limit the Influence of Systematic Errors in the Definition of the Kinematic Behavior of an Extremely-Slow Landslide. *Eng. Geol.* **2016**, *203*, 30–44. [CrossRef]
19.   Buscema, M.; Sacco, P.L.; Grossi, E.; Lodwick, W. Spatiotemporal Mining: A Systematic Approach to Discrete Diffusion Models for Time and Space Extrapolation. In *Data Mining Applications Using Artificial Adaptive Systems*; Springer: New York, NY, USA, 2013; pp. 231–275, ISBN 978-1-4614-4222-6.
20.   Bengio, Y. Learning Deep Architectures for AI. *Found. Trends®Mach. Learn.* **2009**, *2*, 1–127. [CrossRef]
21.   Buscema, P.M.; Massini, G.; Breda, M.; Lodwick, W.A.; Newman, F.; Asadi-Zeydabadi, M. *Artificial Adaptive Systems Using Auto Contractive Maps: Theory, Applications and Extensions*, 1st ed.; Springer: Berlin, Germany, 2018; ISBN 978–3-319–75048–4.
22.   Kohonen, T. Self-Organized Formation of Topologically Correct Feature Maps. *Biol. Cybern.* **1982**, *43*, 59–69. [CrossRef]
23.   Kohonen, T. Improved versions of learning vector quantization. In Proceedings of the 1990 IJCNN International Joint Conference on Neural Networks, San Diego, CA, USA, 17–21 June 1990. [CrossRef]
24.   Buscema, M.; Catzola, L. AVQ1 Basic and AVQ2 Advanced. In *Semeion Report*; Semeion: Rome, Italy, 2007.
25.   Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A Survey of Deep Neural Network Architectures and Their Applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
26.   Buscema, M. Recirculation Neural Networks. *Subst. Use Misuse* **1998**, *33*, 383–388. [CrossRef] [PubMed]
27.   Buscema, M.; Benzi, R. Quakes Prediction Using Highly Non Linear Systems and A Minimal Dataset. In *Advanced Networks, Algorithms and Modeling for Earthquake Prediction*; River Publishers: Aalborg, Denmark, 2011.
28.   Buscema, M.; Massini, G.; Maurelli, G. Artificial Adaptive Systems to Predict the Magnitude of Earthquakes. *Boll. Geofis. Teor. Ed Appl.* **2015**, *56*, 227–256.
29.   Buscema, P.M.; Grossi, E.; Massini, G.; Breda, M.; Della Torre, F. Computer Aided Diagnosis for Atrial Fibrillation Based on New Artificial Adaptive Systems. *Comput. Methods Programs Biomed.* **2020**, *191*, 105401. [CrossRef]
30.   Le Cun, Y.; Kanter, I.; Solla, S.A. Eigenvalues of Covariance Matrices: Application to Neural-Network Learning. *Phys. Rev. Lett.* **1991**, *66*, 2396–2399. [CrossRef] [PubMed]
31.   Le Cun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
32.   Bovenga, F.; Nitti, D.O.; Fornaro, G.; Radicioni, F.; Stoppini, A.; Brigante, R. Using C/X-Band SAR Interferometry and GNSS Measurements for the Assisi Landslide Analysis. *Int. J. Remote Sens.* **2013**, *34*, 4083–4104. [CrossRef]
33.   Raina, R.; Madhavan, A.; Ng, A.Y. Large-Scale Deep Unsupervised Learning Using Graphics Processors. In Proceedings of the Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14 June 2009; ACM: New York, NY, USA, 2009; pp. 873–880.
34.   Raiko, T.; Valpola, H.; Lecun, Y. Deep Learning Made Easier by Linear Transformations in Perceptrons. In Proceedings of the Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, Auckland, New Zealand, 21 March 2012; pp. 924–932.
35.   Schmidhuber, J. Deep Learning in Neural Networks: An Overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef]
36.   Sameen, M.I.; Pradhan, B.; Lee, S. Application of Convolutional Neural Networks Featuring Bayesian Optimization for Landslide Susceptibility Assessment. *Catena* **2020**, *186*, 104249. [CrossRef]
37.   Youssef, A.; Pradhan, B.; Dikshit, A.; Al-Katheeri, M.; Matar, S.; Mahdi, A. Landslide Susceptibility Mapping Using CNN-1D and 2D Deep Learning Algorithms: Comparison of Their Performance at Asir Region, KSA. *Bull. Eng. Geol. Environ.* **2022**, *81*, 165. [CrossRef]

38. Kikuchi, T.; Sakita, K.; Nishiyama, S.; Takahashi, K. Landslide Susceptibility Mapping Using Automatically Constructed CNN Architectures with Pre-Slide Topographic DEM of Deep-Seated Catastrophic Landslides Caused by Typhoon Talas. *Nat. Hazards* 2023, *in press*. [CrossRef]
39. Mandal, K.; Saha, S.; Mandal, S. Applying Deep Learning and Benchmark Machine Learning Algorithms for Landslide Susceptibility Modelling in Rorachu River Basin of Sikkim Himalaya, India. *Geosci. Front.* **2021**, *12*, 101203. [CrossRef]
40. Yi, Y.; Zhang, W.; Xu, X.; Zhang, Z.; Wu, X. Evaluation of Neural Network Models for Landslide Susceptibility Assessment. *Int. J. Digit. Earth* **2022**, *15*, 934–953. [CrossRef]
41. Chen, Y.; Ming, D.; Ling, X.; Lv, X.; Zhou, C. Landslide Susceptibility Mapping Using Feature Fusion Based CPCNN-ML in Lantau Island, Hong Kong. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 3625–3639. [CrossRef]
42. Suto, J.; Oniga, S. Efficiency Investigation from Shallow to Deep Neural Network Techniques in Human Activity Recognition. *Cogn. Syst. Res.* **2019**, *54*, 37–49. [CrossRef]