UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

Earthquakes Prediction using Artificial Intelligence

Trabajo de integración curricular presentado como
requisito para la obtención
del título de Ingeniero en Tecnologías de la
Información

Autor:

Guamán Guerra Cruz Elias

Tutor:

Ph.D. Chang Tortolero Oscar Guillermo

Urcuquí, diciembre de 2021

Urcuquí, 20 de diciembre de 2021

SECRETARÍA GENERAL
(Vicerrectorado Académico/Cancillería)
ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
ACTA DE DEFENSA No. UITEY-ITE-2021-00038-AD

A los 20 días del mes de diciembre de 2021, a las 13:00 horas, de manera virtual mediante videoconferencia, y ante el Tribunal Calificador, integrado por los docentes:

Presidente Tribunal de Defensa	Dr. ARMAS ARCINIEGA, JULIO JOAQUIN , Ph.D.
Miembro No Tutor	Dr. CUENCA LUCERO, FREDY ENRIQUE , Ph.D.
Tutor	Dr. CHANG TORTOLERO, OSCAR GUILLERMO , Ph.D.

El(la) señor(ita) estudiante **GUAMAN GUERRA, CRUZ ELIAS**, con cédula de identidad No. **1002082335**, de la **ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES**, de la Carrera de **TECNOLOGÍAS DE LA INFORMACIÓN**, aprobada por el Consejo de Educación Superior (CES), mediante Resolución **RPC-SO-43-No.496-2014**, realiza a través de videoconferencia, la sustentación de su trabajo de titulación denominado: **EARTHQUAKES PREDICTION USING ARTIFICIAL INTELLIGENCE**, previa a la obtención del título de **INGENIERO/A EN TECNOLOGÍAS DE LA INFORMACIÓN**.

El citado trabajo de titulación, fue debidamente aprobado por el(los) docente(s):

Tutor	Dr. CHANG TORTOLERO, OSCAR GUILLERMO , Ph.D.

Y recibió las observaciones de los otros miembros del Tribunal Calificador, las mismas que han sido incorporadas por el(la) estudiante.

Previamente cumplidos los requisitos legales y reglamentarios, el trabajo de titulación fue sustentado por el(la) estudiante y examinado por los miembros del Tribunal Calificador. Escuchada la sustentación del trabajo de titulación a través de videoconferencia, que integró la exposición de el(la) estudiante sobre el contenido de la misma y las preguntas formuladas por los miembros del Tribunal, se califica la sustentación del trabajo de titulación con las siguientes calificaciones:

Tipo	**Docente**	**Calificación**
Tutor	Dr. CHANG TORTOLERO, OSCAR GUILLERMO , Ph.D.	10,0
Miembro Tribunal De Defensa	Dr. CUENCA LUCERO, FREDY ENRIQUE , Ph.D.	9,0
Presidente Tribunal De Defensa	Dr. ARMAS ARCINIEGA, JULIO JOAQUIN , Ph.D.	9,0

Lo que da un promedio de: **9.3 (Nueve punto Tres)**, sobre 10 (diez), equivalente a: **APROBADO**

Para constancia de lo actuado, firman los miembros del Tribunal Calificador, el/la estudiante y el/la secretario ad-hoc.

GUAMAN GUERRA, CRUZ ELIAS
Estudiante

CRUZ ELIAS GUAMAN GUERRA
Digitally signed by CRUZ ELIAS GUAMAN GUERRA
Date: 2021.12.21 20:58:03 -05'00'

Dr. ARMAS ARCINIEGA, JULIO JOAQUIN , Ph.D.
Presidente Tribunal de Defensa

Firmado electrónicamente por:
JULIO JOAQUIN ARMAS ARCINIEGA

Dr. CHANG TORTOLERO, OSCAR GUILLERMO , Ph.D.
Tutor

Firmado electrónicamente por:
OSCAR GUILLERMO CHANG TORTOLERO

Dr. CUENCA LUCERO, FREDY ENRIQUE , Ph.D.
Miembro No Tutor

FREDY ENRIQUE CUENCA LUCERO
Firmado digitalmente por FREDY ENRIQUE CUENCA LUCERO
Fecha: 2021.12.21 13:15:40 -05'00'

DAYSY MARGARITA MEDINA BRITO

Firmado digitalmente por DAYSY MARGARITA MEDINA BRITO
Fecha: 2021.12.21 08:24:29 -05'00'

MEDINA BRITO, DAYSY MARGARITA
Secretario Ad-hoc

Autoría

Yo, **Cruz Elias Guamán Guerra**, con cédula de identidad **100208233-5**, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el autor del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, Octubre del 2021.

CRUZ ELIAS GUAMAN GUERRA

Digitally signed by CRUZ ELIAS GUAMAN GUERRA
Date: 2022.01.05 13:36:12 -05'00'

Cruz Elias Guamán Guerra
CI: 100208233-5

Autorización de publicación

Yo, **Cruz Elias Guamán Guerra**, con cédula de identidad **100208233-5**, cedo a la Universidad de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Urcuquí, Octubre del 2021.

CRUZ ELIAS GUAMAN GUERRA

Digitally signed by CRUZ ELIAS GUAMAN GUERRA
Date: 2022.01.05 13:37:22 -05'00'

Cruz Elias Guamán Guerra
CI: 100208233-5

Dedication

Este trabajo está dedicado a mis padres: María Antonia Guerra y Narcizo Tadeo Guamán quienes fuerón mi inspiración para lograr este objetivo. También esta dedicado a mis hijos: Anderson Isaac Guamán Víveros y Shirley Dayana Guamán Víveros a quienes en los últimos 6 años me brindarón todo su apoyo y comprensión para culminar con éxito esta meta. Finalmente, una dedicación especial a mi amada compañera de vida, Magola Viveros Montenegro, quién supo apoyar y comprender en una manera exepcional este reto de volver a estudiar y conseguir el título que hoy también es suyo.

Cruz Elias Guamán Guerra

Acknowledgments

Quiero agradecer a todas las personas que formarón parte de este proceso de aprendizaje. En primer lugar a todos los profesores quienes me guiaron de una manera adecuada con sus conocimientos y experiencias. Un agradecimeinto especial a mi tutor Oscar Chang por ser un gran profesional y sobre todo un excelente ser humano.

También agradezco a todo el personal administrativo de la Universidad por su valiosa colaboración y asesoramiento para resolver los inconvenientes surgidos durante los 6 años de estudio.

A mis compañeros de aula y convivencia un gracias enorme por formar parte de una etapa dura y a la vez maravillosa de mí vida, decirles que siempre estarán presentes en mis mejores recuerdos ya que se vivierón experiencias únicas en el día a día durante 6 preciados años de crecimiento académico.

Finalmente, Decir gracias a todas las personas e instituciones quienes de una u otra forma colaborarón para hacer de este tiempo un período extraordinario de vivencias espectaculares. Gracias totales.

Cruz Elias Guamán Guerra

Resumen

A través del tiempo, los terremotos son y serán un fenómeno natural de fatales consecuencias para los seres humanos, debido a que estos causan cientos de miles de muertes, destrucción de edificaciones y enormes pérdidas económicas. Por otro parte, la ciencia computacional crece a pasos gigantes en el área de estudio llamada inteligencia artificial (AI), esta ciencia permite realizar predicciones de eventos de series de tiempo mediante el uso de técnicas como aprendizaje profundo, aprendizaje automático, clasificación, agrupación de datos, etc. El objetivo principal de esta tesis es desarrollar procedimientos con herramientas computacionales para predecir la magnitud de los terremotos con un grado de precisión razonable, se conoce que predecir terremotos es una tarea difícil pero este estudio pretende hacer y mejorar los resultados de investigaciones anteriores al combinar herramientas de AI. Por otro lado, los datos utilizados en este trabajo para la predicción de sismos fueron proporcionados por el Instituto Geofísico de la Escuela Politécnica del Ecuador (IGEPN); estos datos comprenden información digital sobre los fenómenos sísmicos del Ecuador recopilados durante el periodo comprendido entre el año 1901 y el año 2020. Para procesar estos datos se utilizaron técnicas de aprendizaje automático basadas en Máquinas de soporte de vectores (SVM) y análisis de componentes principales (PCA), estas técnicas permiten reducir la dimensionalidad de los datos para posteriormente alimentar una red neuronal artificial (ANN) que está diseñada con la técnica de autocodificador, el cual es capaz de auto reproducir los datos de la capa de entrada en la capa de salida. Finalmente, Los resultados de este trabajo muestran que podría ser posible obtener una predicción razonable de fenómenos naturales altamente peligrosos relacionados con los terremotos. Es de esperar que los resultados obtenidos en este estudio sean utilizados en la construcción de medidas preventivas para minimizar los terribles efectos de los simos de gran magnitud en Ecuador.

Palabras Clave: Terremotos, inteligencia artificial, Ecuador, red neuronal, autocodificador.

Abstract

Throughout time, Earthquakes have been an important issue to human beings because this phenomenon has caused thousands of deaths, destroying buildings, and enormous economic losses through history. On the other hand, computational science has a powerful study area called artificial intelligence (AI) which allows to make predictions of time series events, by using techniques such as deep learning, machine learning, classification and clustering. The main goal of this thesis is to study and develop good computational tools to predict earthquakes or their related events with a reasonable precision. It is well known that earthquake prediction is a difficult task, this study improves previous results by incorporating new tools like deep learning and artificial intelligence. The used data was obtained from The Geophysical Institute in Quito, at the Escuela Politécnica Nacional (IGEPN), and it comprises digital information about Ecuador earthquakes phenomena compiled during several years. For data processing purposes, machine learning techniques based on Support Vector Machine (SVM) and Principal Component Analysis (PCA) where used. This allow to reduce the dimensionality of the data to be fed in an artificial neural network (ANN) which comprises an autoencoder with data reconstructing capacities. The found results show that it could be possible to obtain reasonable prediction for dangerous earthquakes related events, thus contributing with an AI tool to the planning of preventive measures that minimize earthquakes effects in Ecuador.

Keywords: Earthquakes, artificial intelligence, Ecuador, neural networks, autoencoder.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Earthquake is a word to refer to the movements of the earth crust. They are usually measured on the Richter scale and earthquakes that exceed a magnitude of 6 degrees are important because they cause great economic and human losses. Paradoxically, they have a positive aspect that is to provide us with information about the interior of our planet [6]. Currently, through technique known as seismological or seismic tomography [7] the interior of our planet is known in great detail. However, The effects of an earthquake result in ground shaking, fires, seismic waves and landslides, as well as the interruption of vital services, panic and psychological shock. Damage depends on the time the earthquake occurs, the magnitude, the distance from the epicenter, the geology of the area, the type of construction of the various structures, population density and duration of the shaking.

The interaction between Tectonic Plates is the main cause of earthquakes. But, it is not the only one. Any process that can achieve large concentrations of energy of surface earth can generate earthquakes whose propagation will depend on how large the stress concentration zone.

Earthquakes are generally a recurring and periodic phenomenon in some cases. The accumulation of energy in the tectonic plates will have to be released by the occurrence of a new earthquake. Seismic events generally occur periodically in certain geographic regions. Then, the more time passes in a region where a strong earthquake has not occurred, the greater the probability that one will occur there. Generally, in regions where there are reports of earthquakes greater than 6 degrees they will reappear in the future. Ecuador has a great history of earthquake phenomena with catastrophic consequences. The phenomenons that occurred in 1979, 1987, and 2016 are some with major economic and human losses, these years are used in this work such as points of reference in the training and the testing phase in the implementation of an Autoencoder [8] [9] [10] .

Currently, the prediction of a process of nature is one of the goals of all science. Computing science with the technique of artificial intelligence and machine learning tools is not out of these aspirations. In fact, there is no effective technique to predict earthquakes, but the advances made and the knowledge acquired in this domain allow us to assert that the day will come soon when the possibility of anticipating the occurrence of an earthquake will be an awesome reality.

1.2 Objectives

1.2.1 General Objective

Develop an earthquake prediction system based in data compression and deep learning by using data preserved and supply by the Ecuadorian Government.

1.2.2 Specific Objectives

Learn how to acquire the seismic data from the government.

Process data by using SVM and PCA tools.

Develop C++ code to simulate and run reconstructing Autoencoder.

Develop C++ code to unify the PCA and autoencoder platforms Debug integrate software.

Combine the processing capacities of Support Vector Machine (SVM), Principal Component Analysis (PCA), and reconstructing Autoencoder to process the acquired data and produce useful predictive data.

Use the predictive data of this work so that the Ecuadorian organizations in charge of planning preventive measures to minimize the effects of earthquakes.

Chapter 2

Theoretical Framework

This chapter is designed to explain some important definitions about the terminology, tools, and processes of computer science used to accomplish the goal. These definitions begin with an explanation of the kinds and consequences of earthquakes, data source, neural networks approach, and finally the mathematical and computational techniques of Autoencoder.

2.1 Kinds of earthquakes

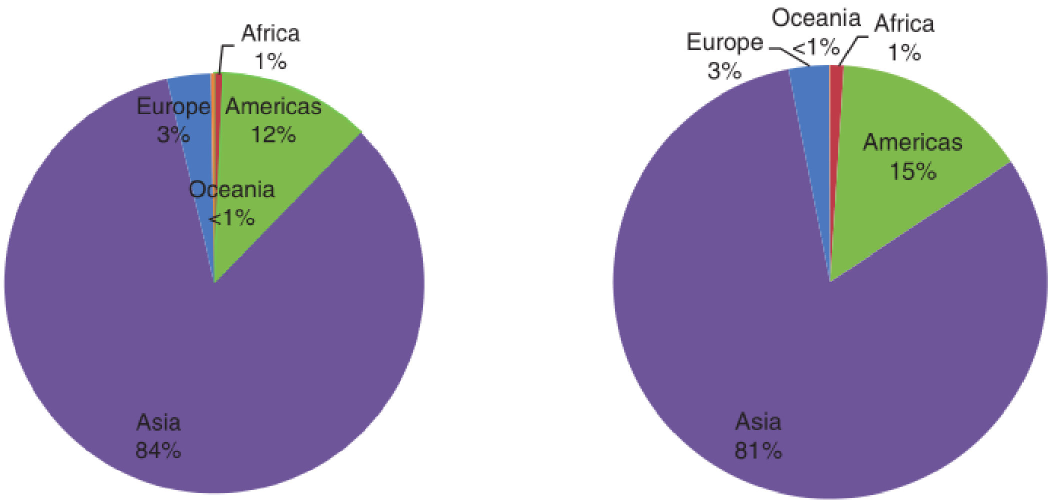
An earthquake has an underground focus or point of origin, known as the hypocenter, and a point on the surface directly above the focus, called the epicenter, where the movement is usually most intense.

Three different types of earthquakes are considered, depending on the region of the crust in which its hypocenter is located:

- **Superficial**.- They have a focus no greater than 70 kilometers deep, so they have a greater impact on the surface. This makes them the most devastating earthquakes [11].
- **Intermediate**.- Its focus ranges between 70 and 300 kilometers deep [11].
- **Deep**.- events that occur deep within the Earth, generally outside the lithosphere, more than 300 kilometers from the surface. Called bathysisms, they are usually imperceptible [11].

2.2 Earthquake consequences

Earthquakes around the world have great consequences in economic and human losses. In addition, these also cause psychological and physical illnesses. For these reasons, a reasonable prediction in time and magnitude of these natural events would mitigate large undesirable effects for humans in a high percentage.



(a) Earthquake fatalities per continent (%) (b) Earthquake victims per continent (%)

Figure 2.1: Humans Impacts 1970-2008 [1]

2.2.1 Humans Impacts

Earthquakes greater than 6 degrees on the Rircher scale cause the collapse of the vast majority of buildings, the destruction of roads, tsunamis, etc. According to Guha-Sapir and his colleague "*Earthquakes have claimed an average of 27,000 lives a year since 1990*" This high amount in terms of annual mortality allows to see the importance of a correct and adequate earthquake prediction process [1].

Figure 2.1 shows the percentages of victims and fatalities by continent. An important fact is the American continent fill second place with a percentage of 12 % of fatalities of earthquakes. It is that ± 3240 people die per year in sinister related to earthquakes [1]. In addition, table 2.1 shows that Peru and Guatemala are in the top ten of the most destructive earthquakes with the most negative impacts over humans.

2.2.2 Economic Impact

Throughout history, economic losses as a result of seismic movements have had a great impact on the countries gross domestic product (GDP). From 1900 to 2010, some 6500 earthquakes worldwide have caused huge costs in the reconstruction of roads, houses and basic services. For example, the earthquake that occurred on March 6, 1987 in the northern part of Ecuador caused an approximate cost of 1.500 billion dollars, which is equivalent to 16.48% of Nominal GDP [12] .

Estimates of economic and human losses as a result of earthquakes are enormous, Then, it is necessary to use tools such as artificial intelligence and its approaches to predict these natural phenomena and thus minimize the consequences.

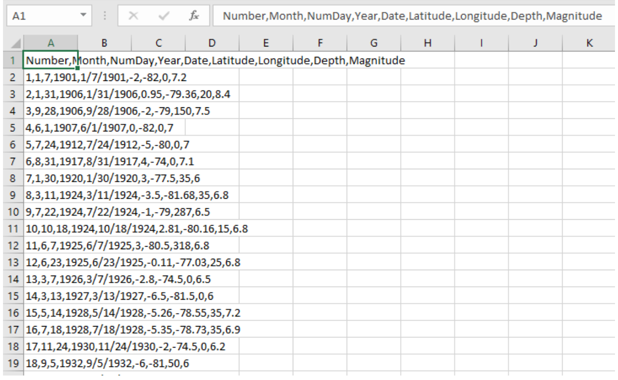
Table 2.1: Top ten of most destructive earthquakes in terms of human impact (1970–2008) [1].

Date	Country	Richter	Killer (x1,000)	Total affected (x1,000)
27 Jul 1976	China	7.8	242	164
26 Dec 2004	Indian Ocean tsunami	9.0	226	2,432
12 May 2008	China	7.9	88	45,977
08 Oct 2005	Pakistan, India	7.6	75	5,285
31 May 1970	Peru	7.8	67	3,216
21 Jun 1990	Iran	7.3	40	710
26 Dec 2003	Iran	6.6	27	268
07 Dec 1988	Armenia	6.9	25	1,642
16 Sep 1978	Iran	7.7	25	40
04 Feb 1976	Guatemala	7.5	23	4,993

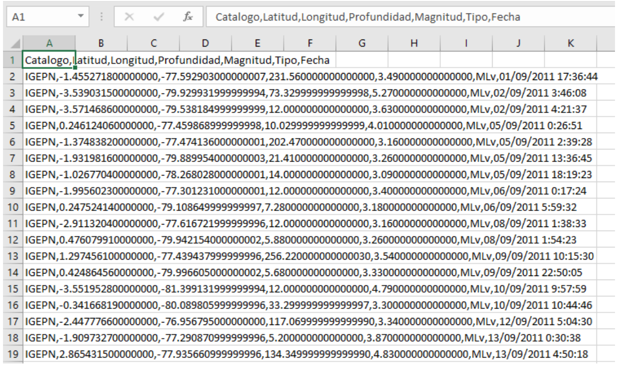
2.3 Data Source and Selected Country

I am an Ecuadorian citizen. Therefore, the country chosen for earthquake prediction is focus on Ecuador. This country is located in latitude from - 7° to + 4° and in longitude from 84° to 74° (roughly 1100 × 1000 km) . It is located in South America and is part of the southern hemisphere. In addition, the tectonic plate (Nazca) of the convergent type on which this country is located pushes and slides under the continental plate (South American plate) [13]; when the resistance of the materials of the crust is exceeded the rocks that make up the plates break then produce large seismic movements.

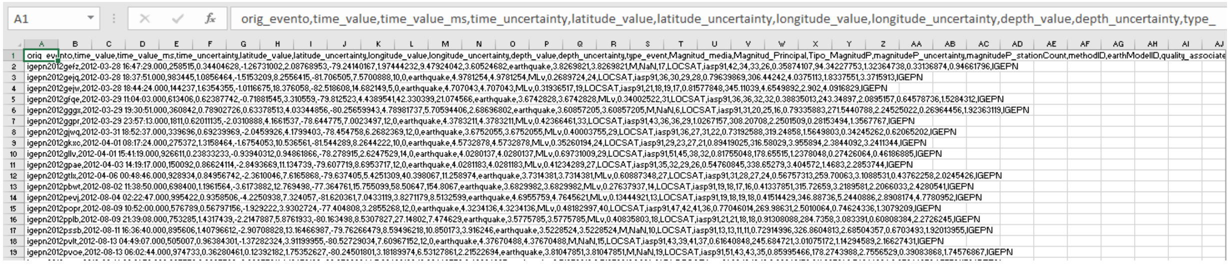
The Geophysical Institute in Quito, at the Escuela Politécnica Nacional (IGEPN) is the governmental entity that registers the seismic data that occurs in the country. To carry out the seismic information survey the IGEPN has 61 seismic stations and 89 acellometric stations distributed in strategic points of the country. Fig.2.2 shows how the data was collected. 3 zip files with different data and format were provided by IGEPN. The data that this institution give us is the main tool used for the purpose of this work.



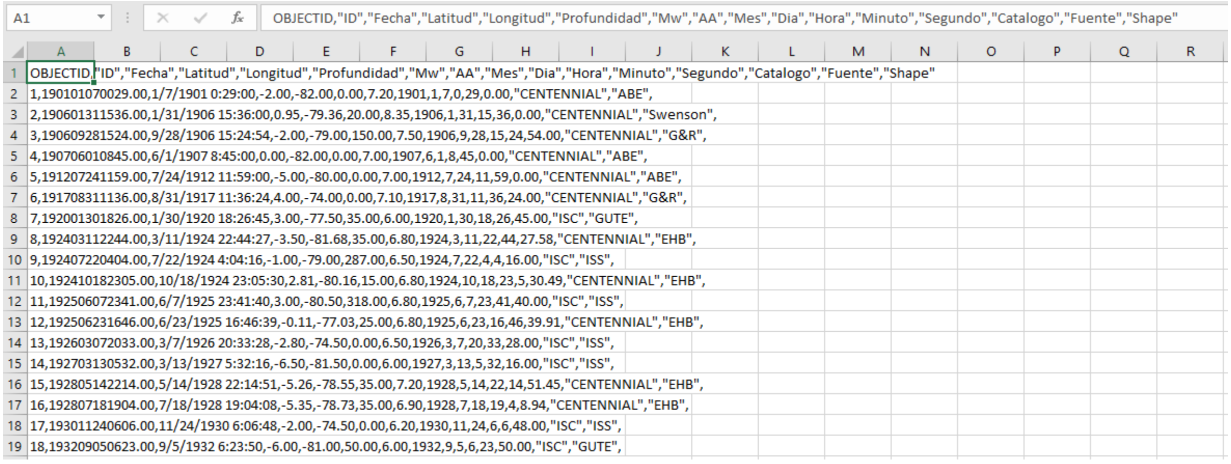
(a) complete dataset processes



(b) IGEPN.ene.2010_dic.2011)



(c) ev_2012-2020



(d) homogenized catalog 1901-2009

Figure 2.2: Intensity database for Ecuador

2.4 Biologic and Artificial Neural Networks

A successful criteria to distinguish something natural and artificial was advanced by Aristotle. The various nuances to distinguish the natural from artificiality. Thus, it makes of the natural that which has the principle of motion and rest in itself, while the artificial has such a principle in the human being [14]. And the Philosopher clarifies that when we try to imitate the natural through the artificial, it is important to differentiate the arts that reproduce nature, from others that only try to perfect, complete or modify it for the better results.

2.4.1 Biological Neural Network

Neurons are the fundamental units of the brain and nervous system [2]. They are in charge of receiving sensory stimulus from the outside world as well as sending orders to different parts of the body to transform and transmit the electrical signals necessary to carry out any action.

Neurons have different shapes and sizes, but they all consist of three elemental parts as shown in Fig. 2.3: the cell body, the axon, and the dendrites.

- The cell body is the nucleus (which contains DNA) and is where proteins are formed.
- The axon is a wire-like part of the cell that transmits electrochemical messages. A single axon can have multiple branches, allowing it to synapse with multiple postsynaptic cells. Similarly, a single neuron can receive thousands of synaptic inputs from many different pre-synaptic or emitter neurons.
- The dendrites or nerve branches are short projections of the cell, like branches, that establish connections with other cells. Dendrites receive messages through neurotransmitters released by axons of other nerve cells. In the initial part of the axon of a neuron (where it joins the neuronal body) an action potential is generated, a short electrical impulse that travels along the axon and causes the release of neurotransmitters (they are like messengers) at the synapse, the point where this release occurs and the reception of the message by another neuron, thus allowing communication between them.

2.4.2 Artificial Neural Network

Neural Networks are a very important field within Artificial Intelligence. This approach is inspired by the known behavior of the human brain referring mainly to neurons and their connections, They try to create artificial models that solve problems that are difficult to solve using conventional algorithmic techniques.A beautiful biologically-inspired programming paradigm which enables a computer to learn from observational data [15].

Since the 1940s, in which computing was born and began to develop, the neural model has accompanied it. In fact, the emergence of digital computers and the development of modern theories about learning and neural processing occurred around the same time, in the late 1940s.

Nowadays, neurophysiological research and the study of Artificial Neural Systems (ANS) have gone hand in hand. However, the ANS models do not focus on neurological research, but rather take concepts and ideas from the field of natural sciences to apply them to solving problems belonging to other branches of science and engineering.

The first examples of these systems appear in the late 1950s. The most common historical reference is the one that alludes to the work carried out by Frank Rosenblatt in

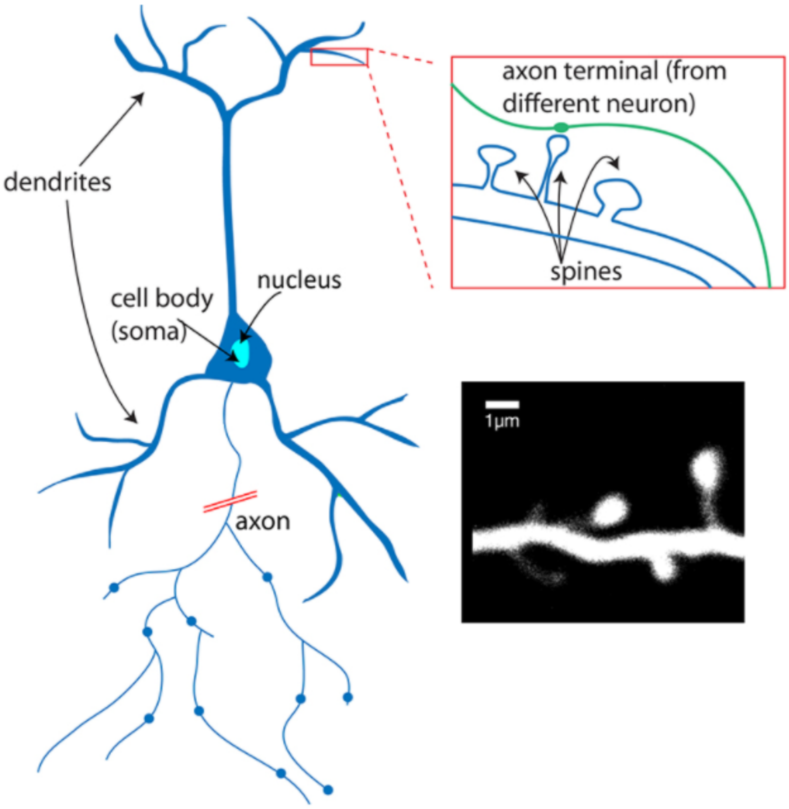


Figure 2.3: The tree-like structure of a neuron. Dendritic spines are small structures that receive inputs from the axons of other neurons. Bottom-right image: a segment of dendrite from which spines branch off, like leaves off a tree branch. Note the very small size (0.001mm) [2].

a device called the perceptron [1] [16]. There are other examples, such as the development of the Adaline by Professor Bernard Widrow.

2.5 Layers of ANNs

ANNs are a computational model based on the behavior observed in their biological equivalents. In addition, these structures are a set of processing units called artificial neurons and are generally linked to each other which transform the input data that reaches them into other output data. This process causes the information to proceed through the entire neural network while it is transformed into each artificial neuron crossed. Finally, it reaches the end of the network where the resulting value is returned.

As shown in figure 2.4, the simplest version of artificial neuron works as follows:

The neuron has as inputs a vector $X(x_1, x_2, x_3...x_n)$ and with this values the weighted

[1]In machine learning, the perceptron is an algorithm for supervised learning of binary classifiers. A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class.

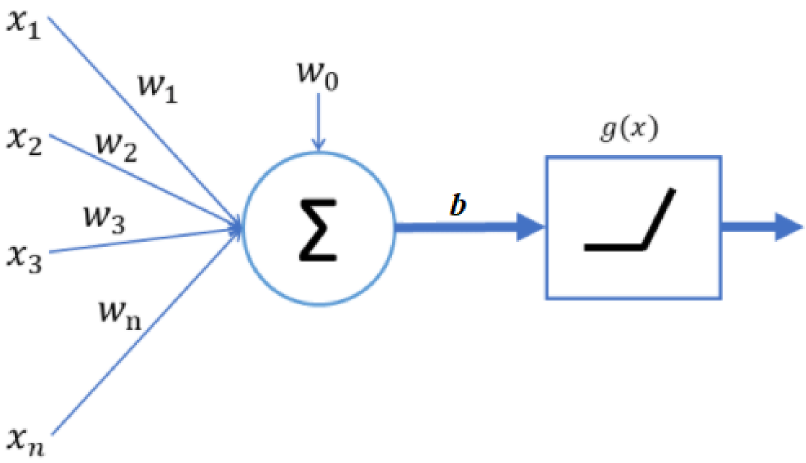


Figure 2.4: Diagram of the basic structure of a perceptron.

weight is calculated. These weights are denoted by the abbreviations $w_1, w_2, w_3...w_n$. These values an additional value w_0 or b better known by the name of bias is added. Then, all the weights and the value of w_0 are calculated. That is, they are chosen during the learning of the neuron. The result of applying this linear function is shown in the previous graph with the label b:

$$b = w_o + x_1w_1 + x_2w_2... + x_nw_n$$

Once the value of b has been calculated a function non-linear $g(x)$ is then applied to it which is called the activation function. A classic example is the function called binary step. This function returns 1 if the independent variable is zero or greater than zero, and returns 0 otherwise.In fact, there are notable difficulties when training the model because the output of the activation function is the same for any value above 0. For this reason, today other activation functions such as sigmoid, ReLU, Tanh, and more are used.

There are typically three parts to a neural network (see Figure 2.5): an input layer with units representing the input fields; one or more hidden layers without direct connection with the environment this can be preceded by other hidden layers or by the input layer; and an output layer with a neuron or several neurons generally representing solutions for any problem or determinate input. Layers are connected with varying connection forces or weights. The input data is presented in the first layer, and the values propagate from each neuron to each neuron in the next layer.

2.6 Backpropagation

According to Karazi et al., 2019 "Back-propagation algorithm is the most common supervised learning algorithm. The concept of this algorithm is to adjust the weights minimizing the error between the actual output and the predicted output of the ANN" and "the network training consists of three stages: (a) feed-forward of the input training pattern; (b) calculation and back-propagation of the associated error; and (c) the adjustment of the

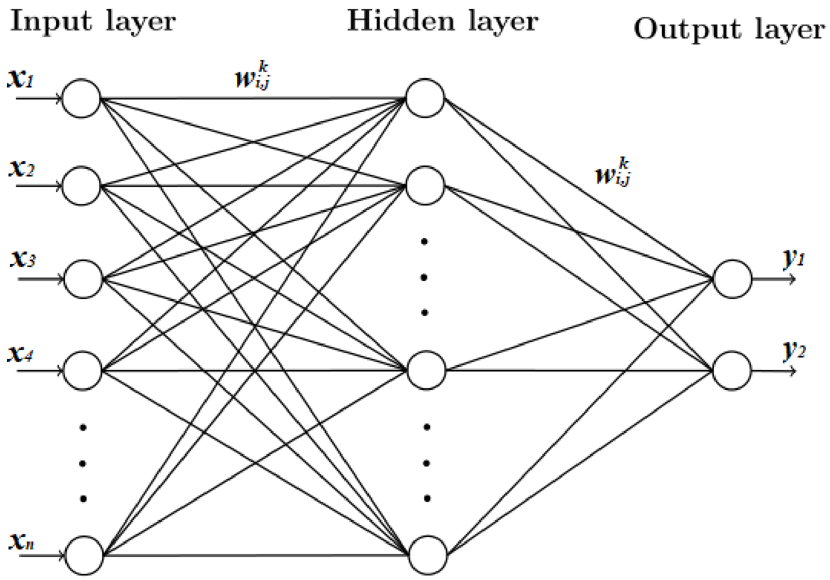


Figure 2.5: General structure of an artificial neural network

weights. By starting from the output layer, backwards pass propagates the error". Once the first iteration is done, the error outputs are propagated backwards. This process begins in the output layer and propagates to all the neurons in the hidden layer that directly contribute to the output. However, the hidden layer neurons only calculate a small fraction of the total amount of error based on the relative contribution that each neuron made to the original output. This process is repeated on all the layers as long as the relative contribution of the total error is minimal in each neuron of the hidden layer. The connection weights of each neuron are calculated and updated in each backpropagation cycle until the entire network converges and it allows all training patterns to be correctly classified.

The backpropagation algorithm has a main characteristic the ability to self-adapt the weights of the intermediate neurons to learn the relationship between a set of patterns given as input in the first network layer and their corresponding outputs. After a full training phase, if the network is presented with a similar or noisy input, the neurons in the hidden layer of the neural network produce an active output once the new input has resemble individual neurons learned. On the other hand, the neurons of the hidden layers tend to inhibit their output if the input values do not have resemble inputs for which they have been trained.

The backpropagation algorithm generally develops internal relationships between neurons for the purpose of organizing training data into classes. This trend can be extrapolated to make the consistent hypothesis that all neurons in the hidden layer are associated in some way with specific features of the input pattern as a consequence of training [17]. This association may not be obvious to the human observer. The main idea is to make the network has an internal representation that allows it to generate the desired outputs when any inputs are provided. Finally, once the internal representation was obtained in the training phase is likely apply to inputs that the network has not seen before and the network will classify these inputs according to the features that they share with the training examples.

2.7 Support Vector Machine

Support vector machines are a set of supervised statistical learning algorithms that belong to the set of linear classifiers developed by Vladimir Vapnik and his collaborators at AT&T laboratories in 1995 [18]. A SVM builds a hyperplane in a very high or even infinite dimensional space. The hyperplane generally most optimally separates the points or required information into two classes defined by a kernel. The concept of optimal separation is the fundamental characteristic of SVMs and one of its main advantages and uses in cases of binary selections. Also, This is why SVMs are sometimes referred to as maximum margin classifiers [19].

2.7.1 Radial basis functions

These types of functions return a value that depends only on the distance from the origin to some central point. Let $\phi(r) = \Phi(\| x \|)$ and x_k any central point such that $\phi(r) = \Phi_k(x) = \Phi(\| x - x_k \|)$. Then, any function that satisfies $\phi(r) = \Phi(\| x \|)$ is called as radial basis function.

taking in consideration that $r = \| x - x_k \|$ and let ε shape parameter. Then, the most common types of radial basis functions are [20] [21]:

- Gaussian function
$$\phi(r) = e^{-(\varepsilon r)^2}$$
- Multiquadratic function
$$\phi(r) = \sqrt{1 + (\varepsilon r)^2}$$
- Inverse multi quadratic function
$$\phi(r) = \frac{1}{\sqrt{1 + (\varepsilon r)^2}}$$
- Polyharmonic Spline function
$$\phi(r) = r^k, k = 1, 3, 5, ...$$
$$\phi(r) = r^k \ln(r), k = 2, 4, 6, ...$$
- Thin plate spline function
$$\phi(r) = r^2 \ln(r)$$

2.7.2 One-class

One class support vector machine (OCSVM) is a variation on the original SVM. This new algorithm can clean or find outliers from the input database. Figure 2.6 shows how OCSVM classifies in negative and positive values (outliers and inliers), it classifies in a correct way the inliers, but the others will be taken into account as outliers. [3]. However, in some cases the data does not allow finding hyperplanes in an optimal and fast way.

Steps to do One-class SVMs framework

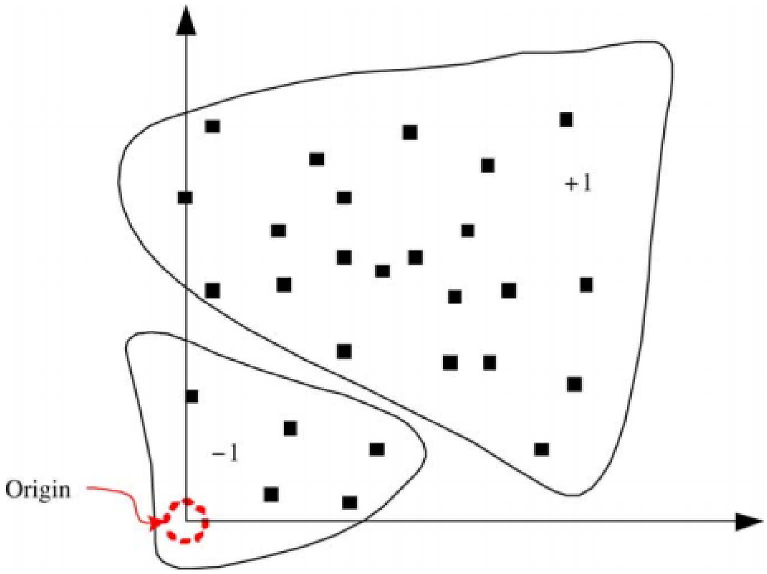


Figure 2.6: Representation to classifier OCSVM [3]

- Transform an input space into a feature space

$$x = (x_1, ..., x_m) \mapsto \varphi(x) = (\varphi_1(x), ..., \varphi_m(x))$$

- Maximize the distance of the hyperplane

$$\min_{w \in F} \frac{1}{2} \parallel w \parallel^2$$

Where w is a weight vector, and new space $F = \{\phi(x_i) \mid x \in X\}$

- Calculate the trade-off between the margin and the outliers

$$\min_{w,\xi_i,\rho} \frac{1}{2} \parallel w \parallel^2 + \sum_{i=1}^{m} (\xi_i - \nu\rho)$$

Where ξ is a slack variable, $\rho > 0$, and ν is $\in$ (0,1]. Subject to: $\langle w \cdot \phi(x_i) \rangle \geq \rho - \xi_i,\ \xi_i \geq 0,\ i = 1, ..., m$

- Chose an appropriate kernel method
  - linear
  - radial basis function (RBF)
  - Polyharmonic Spline
- Computing a dual problem

$$\min \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)$$

Where α is Lagrange multipliers and $K(x, y) = \phi(x) \cdot \phi(y)$. Subject to: $0 \leq \alpha_1 \leq 1$

2.8 Principal component analysis

Principal Component Analysis (PCA) is a statistical method that simplifies the complexity of sample spaces with many dimensions while preserving their information. PCA is one of the most popular multivariate statistical techniques used in the areas of signal processing and pattern recognition. The goal is to take out the most relevant information from the data and to extract this information into a set of new orthogonal variables called principal components.

Suppose there is a data with m samples each with n variables. Then, the sample space has mxn dimensions. PCA allows finding a number of underlying factors ($l < m$) that explain approximately equal features as the original m variables without loss of information. Therefore, the PCA method allows condensing the information provided by n variables into l principal components.

2.8.1 Linear Transformation

Let a dataset X with m as number of features and n as number of variables. Then, to achieve a new representation data Y is necessary an orthogonal transformation P. The rows of a new orthogonal matrix are called principal components. In addition, rows and columns of P matrix are orthogonal vectors with unit norm. This is showed in the below representation.

$$\underset{(mxm)}{P}\underset{(mxn)}{X} = \underset{(mxm)}{Y}$$

$$\begin{bmatrix} p_1 \\ . \\ . \\ . \\ p_m \end{bmatrix} \begin{bmatrix} x_1 & . & . & . & x_n \end{bmatrix} = \begin{bmatrix} p_1.x_1 & . & . & . & p_1.x_n \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ p_m.x_1 & . & . & . & p_m.x_n \end{bmatrix}$$

The columns of a new matrix Y carries elements that are a dot product between the *jth* row of P and the *ith* column of X. Each element into Y represents a projection of a point from feature space onto a principal component. Therefore, The matrix Y is a projections of the original features onto the space spanned by orthogonal matrix P, which are unit vectors. In other words, the multiplication between P and X means projecting X onto the space spanned by the rows of P [22].

2.8.2 How to compute Matrix P

The main goal to calculate orthogonal matrix P from matrix X is decrease redundancy to retain signal and filter out noise of the dataset. Hence, the first step to find matrix P is finding square symmetric covariance matrix whose diagonal terms are variances and whose off-diagonal terms are covariances.

$$C = \frac{1}{n}XX^T$$

Let D diagonal matrix with no covariance terms and a function of C and P to minimize redundancy. Then, D can be express as:

$$D = PCP^T$$

How Matrix C is symmetric. Then, it can compute using its eigenvectors that is denoted by E and its eigenvalues diagonal matrix denoted by V.

$$C = EVE^T$$

Replacing C into D

$$D = P(EVE^T)P^T$$

Chosen $P = E$, $E = E^T$ because E is a diagonal matrix, and by definition $EE^T = I$

$$D = (E^T E)V(E^T E) = V$$

Finally, the eigenvectors of the covariance matrix of X are in matrix V and they are called PCA.

2.9 Autoencoder

An Autoencoder is a kind of perceptron, basic structure of ANN, where the input and output layers have the same number of neurons and it is trained to reconstruct its own inputs. The training algorithm of an Autoencoder tries to reconstruct the input into de output layer from unlabeled data [23].

2.9.1 The architecture of Autoencoders

Figure 2.7 shows the basic architecture an Autoencoder. Encoder, code and decoder are components of it.

- Encoder

  This module is formed by a set of convolutional blocks followed by pooling modules that squeeze the input data into an encoded representation that is generally maps the input into the code module.

- Code

  The code phase is the most important in the autoencoder, and the smallest one. It contains the compressed knowledge representations of the input data. Also, this module is called bottleneck. If bottleneck is small, there are the risk of overfitting. Therefore, smaller bottlenecks likely loss important information slipping out through the pooling layers of the encoder.

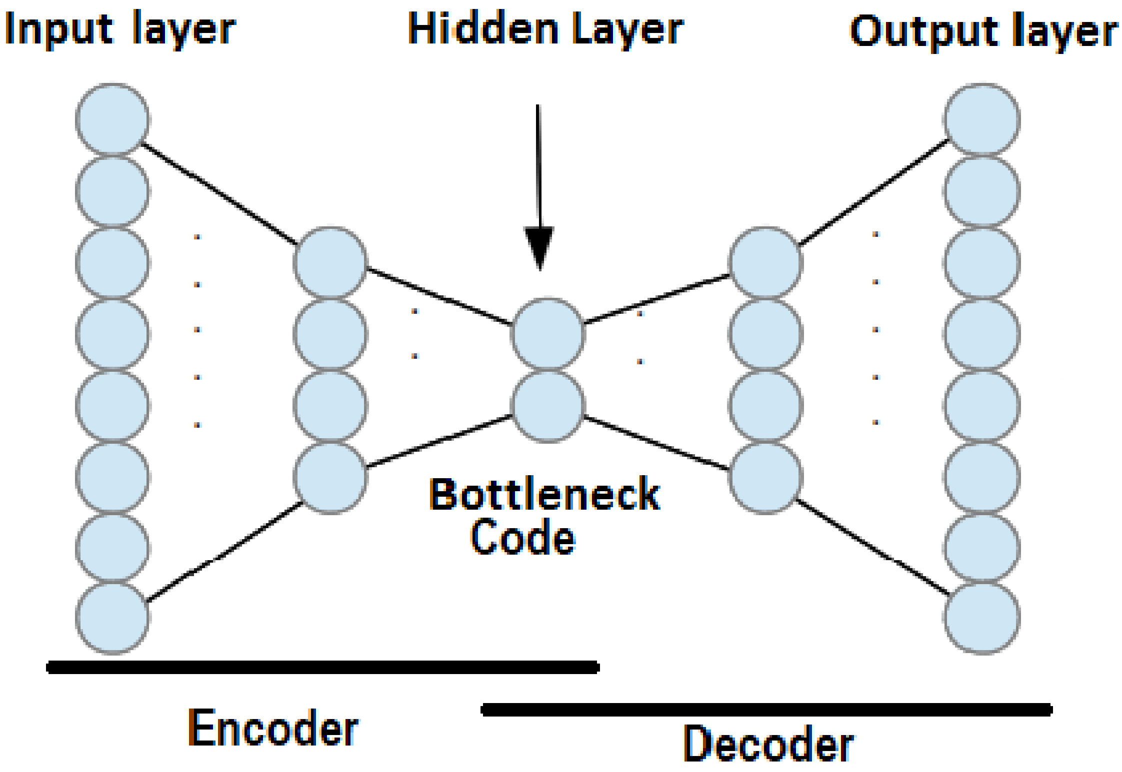


Figure 2.7: General architecture of autoencoder [4]

- Decoder

  This module of the network allows decompress the knowledge representations that was compress in the encoder module. Hence, the decoder module reconstructs the data back from its encoded form. It reconstructs a input data through a set of resampling and convolutional blocks from compressed data in bottleneck module.

The mathematics used to building the Autoencoder is not complex. Basically, 2 functions are related to encoder and decoder calculation.

$$\boldsymbol{\phi} : \boldsymbol{X} \rightarrow \boldsymbol{F}$$

$$\boldsymbol{\psi} : \boldsymbol{F} \rightarrow \boldsymbol{X}$$

$$\boldsymbol{\phi}, \boldsymbol{\psi} = \underset{\phi,\psi}{argmin} \parallel X - (\psi \circ \phi)X \parallel^2$$

Where ϕ is the function of the encoder that maps the input data representing the matrix X into a latent space F, which is stored in code module. ψ is the decoder function that maps the latent space F from the code module to the output. In this case, the output function must represents input data. In general, the goal is trying to recreate the original data after generalized non-linear compression.

Now, let z a latent dimension or the standard neural network function that to allow to find bottleneck through an activation function. Hence, the mathematical representation is:

$$z = \sigma(Wx + b)$$

Where σ is a lineal or non-linear function.

In the same way, the decoder module can be represented with a similar function, but with different weight, bias, and potential activation functions.

$$x' = \sigma'(W'z + b')$$

With the last 2 functions can be written the loss function, This function is used to train the neural network through the standard backpropagation procedure.

$$\mathcal{L}(x, x') = \parallel x - x' \parallel^2 = \parallel x - \sigma'(W'(\sigma(Wx + b)) + b') \parallel^2$$

Since an autoencoder has in the input and output the same data, this is not really supervised or unsupervised learning, this is typically called self-supervised learning. The main goal of an autoencoder is finding encoder and decoder functions in such a way that with the minimal information to regenerated the original data into output layer of the encoder module.

According to Baldi P., [5] there is a simple overall classification of autoencoders that is showed in Figure 2.8. This classification was presented in his Workshop and Conference Proceedings in 2012 [24].

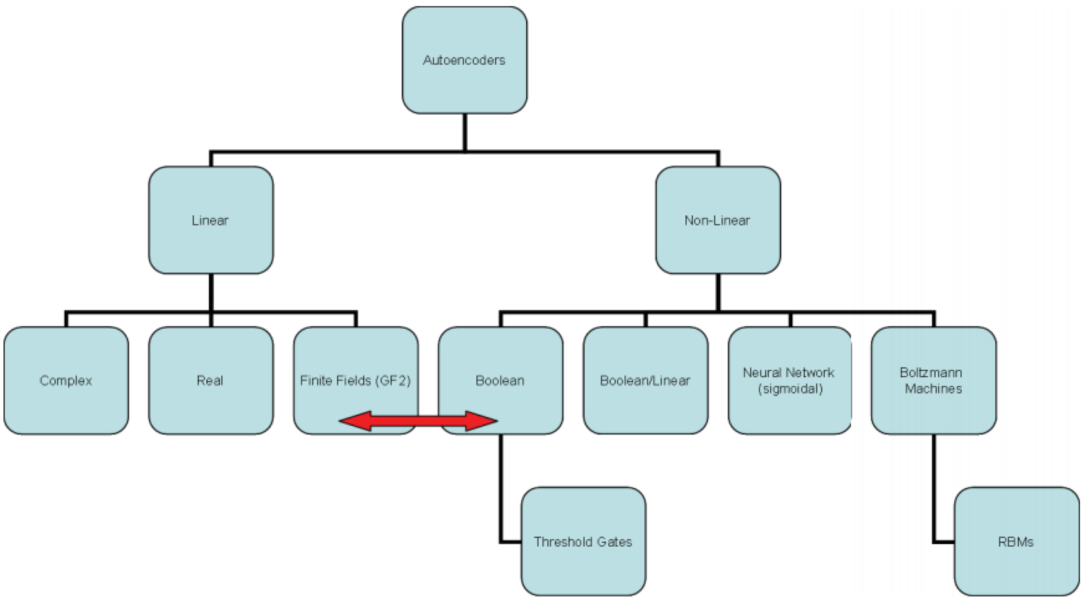


Figure 2.8: Simple Autoencoder classification [5]

Chapter 3

Related Works

3.1 Earthquake prediction model using support vector regressor and hybrid neural networks

This work makes predictions of earthquakes of 5.0 and above in three different places in the world. The regions selected by the researchers are Hindukush, Chile, and Southern California. The data are collected from the United States Geological Survey (USGS) and are found in the period of time from January 1980 to December 2016. To represent 60 parameters of earthquakes the authors apply seismology concepts such as Gutenberg-Richter law, seismic rate changes, foreshock frequency, seismic energy release, total recurrence time. For prediction, they use two concepts of artificial neural networks [25]. The first approach used is the supervised learning algorithm called Support Vector Regressor (SVR) which is a variant of the Support Vector Machine model and is used as a value prediction model. The second is a prediction model algorithm called Hybrid Neural Network (HNN) and is used for the training phase. HNN is a result of the combination of three different Artificial Neural Networks to achieve the optimization of the ANN weights [25].

The training is done with 70% of the total dataset and 30% is used for testing. The output layer in the structure of the models results in a binary value 0 or 1, where 1 is the prediction of an event 5.0 and above, and 0 otherwise. The network is trained with the backpropagation technique and the output layer error is propagated in all layer weights in each epoch[1] [25]. The results are stored in a confusion matrix with the following values: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). the Matthews correlation coefficient (MCC) uses the values of the matrix to obtain a measure of the quality of the prediction. This paper uses MCC to compare the results separately from the SVR, HNN, and the combination of both SVR-HNN models. In the results shown in the work of researchers, the advantage of using the SVR-HNN technique is clearly noted and as an example, it is obtained for the region of Chile which values: with SVR = 0.44, with HNN = 0.52 and with SVR-HNN = 0.613 [25].

Finally, the approaches of SVR followed by HNN are encouraging to predict possible

[1] An epoch means training the neural network with all the training data for one cycle. In an epoch, we use all of the data exactly once

earthquakes. The backpropagation and support vector regression approaches are used in the planning of this thesis and it has a similar structure to achieve a good earthquake prediction in the present study.

3.2 Medium–large earthquake magnitude prediction in Tokyo with artificial neural networks

This article begins with a brief review of the place chosen for the study and prediction of earthquakes. The authors said that Tokyo is located on a large tectonic plate and it has approximately 5000 seismic activities per year of which 1000 have an intensity greater than 3.5 degrees on the Richter scale. Prediction aims to use the artificial neural network called perceptron which is the most basic in terms of structure. The result of this algorithm is compared with other machine learning tools to compare its efficiency. The prediction is based on knowing the site of a possible earthquake, the time on the next 7 days, and the magnitude above 5.0.

The dataset is taken from the catalog of the US Geological Survey (USGS) which is freely accessible on the internet. As in the work of Asim, et al. to extract the main characteristics of the data, the MCC technique is used, and then it is divided into 2 parts. The first is used for the training phase and the second is for the testing phase. The neural network is structured with 96 neurons in the input layer, 48 neurons in the hidden layer, and 2 neurons in the target layer [26]. Furthermore, the weights are calculated with the backpropagation technique which has a learning rate of 0.3. To obtain reasonable values in the hidden layer and to make a good prediction, the network is trained with 500 epochs.

The results obtained with the neural network proposal were compared with other machine learning techniques known as nearest neighbors (KNN), support vector machines, Bayesian networks(BN), and decision trees (J48). Although the SVM, KNN algorithms have an average rate greater than 50% in the prediction although the method proposed in this work obtained the best averages. In addition, a statistical comparison was made using the Kruskal Wallis with next numbers ($x^2 = 17.899$; p value = 0.0012, therefore p < 0:01).) and Mann – Whitney techniques showing significant differences between ANN and KNN-SVM-NB-J48 (p value = 0.0079; p <0.02; r = 0.82945) [26].

3.3 Spatial Analysis of Magnitude Distribution for Earthquake Prediction using Neural Network Based On Automatic Clustering in Indonesia

This scientific article is based on earthquake data taken from Indonesia provided by the Indonesian Agency for Meteorological, Climatological and Geophysics (BMKG) and USGS. The main objective of this work is to find an optimal number of clusters and the prediction of earthquakes greater than 5.5 and 6.0. The study is focused on Hierarchical K-means techniques to find the clusters and the prediction is made for the next 5 days after the earthquake occurrence. ANN is made up of 4 layers. The input layer has 7 neurons that are connected to 2 hidden layers of 32 neurons each, and finally the output layer has a single node.

The data obtained from BMKG and USGS has a total of 82,580 of which was reduced to 9233 after applying Gutenberg-Richter law. The optimal global number of clusters is calculated using the Valley Tracing and Hill Climbing algorithms and to classify within the clusters the Hierarchical and K-means algorithms are used [27]. The seven parameters of the output layer are calculated as follows: 5 are obtained with Gutenberg-Richter law, the sixth is the maximum magnitude of the data subset and the seventh value is the threshold wish to predict. To calculate the weights of the neural network, the backpropagation algorithm and the sigmoid activation function are used with a learning rate of 0.1. Additionally, 1000 epochs are used for network training.

The results of the study are based on 2 values for the prediction in input layer seven with values of 5.5 and 6.0. For the first value, the results show that 76% belong to negative prediction values and 0% to positive values. With the entry of 6.0, there are 60% negative values and 97% positive values for the prediction. Finally, the average of each entry is 44% and 75% respectively. In a conclusion, it is shown that the prediction is radically improved when it comes to aftershocks of greater than 6.0 magnitude [27].

Chapter 4

Methodology

The main objective of this project is to make earthquake predictions focusing on machine learning methods and artificial neural networks such as SVM, PCA, and Autoencoders. The data provided by IGEPN were selected as the main input for this study. The data set was partitioned into training and testing data sets through hyperparameter values. In this work, 4 main phases are identified:

- To process dataset
- To process one class SVM
- To process PCA
- To process Autoencoder

The software and hardware used to do this research are:

- Software
  - Windows 10 Pro
  - Microsoft office excel 2016
  - Orange 3.27.1
  - MATLAB R2018A
  - Borlan C++ 5.5
- Hardware
  - Laptop Acer ACPIx64-based PC
  - SSD disk of 128 GB
  - 4 Processors Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz
  - Memory RAM of 4 GB DDR4

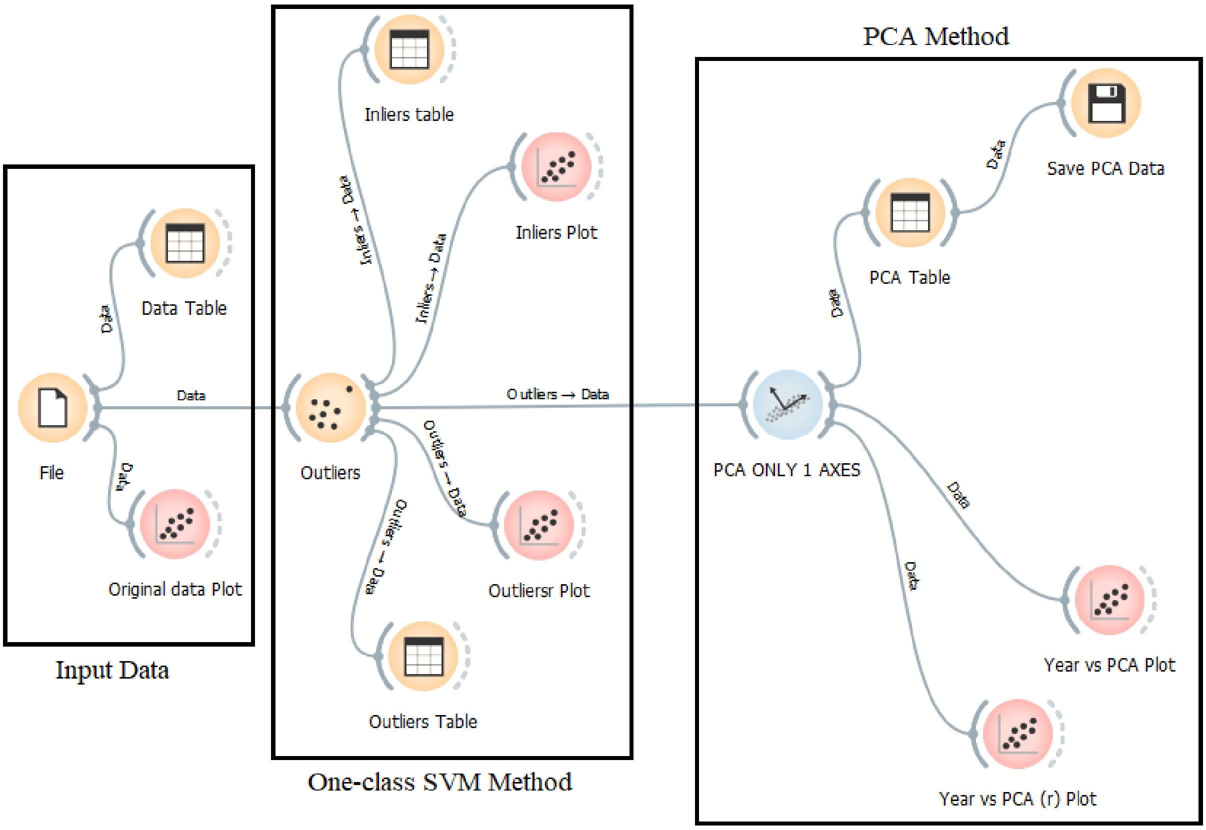


Figure 4.1: Overall scheme of Machine Learning approach applied on Orange App.

4.1 Dataset

The first step in this project is data collection. For this, the IGEPN was asked for the data collected from the earthquakes registered in Ecuador, since this government institution stores the necessary information on each earthquake event. This institution has different ways of collecting the earthquake data from the year 1901 to 2020. These data are divided into 3 zip files that have different items regarding their information (see Figures 2.2b, 2.2c, and 2.2d).

For this stage, the Microsoft office excel tool was used due to its versatility in data management and the ability to open files in different formats. From the combination of the three files, a total of 13,766 rows and 9 columns were obtained; the columns have 8 features and one meta attribute.

The eight features are:

- Number.- This is an ordinal number.
- NumDay.- This is a number between 1 to 31.
- Month .- This is a number between 1 to 12.
- Year.- This a number between 1901 until 2020.
- Latitude.- This is the number of latitudes when the epicenter is located.

- Longitude.- This is the number of longitudes when the epicenter is located.
- Depth.- This a number of depth to the epicenter.
- Magnitude.- This is a number of seismic magnitudes measured on the Richter scale.

The meta attribute is:

- Date.- This is a date when an earthquake happened and has the format mm/dd/yyyy.

A few samples of this data is shown in Figure 2.2. The data contained in the eight columns are the most representative of a seismic events.

4.2 Identification of Outliers

Outliers are values that are outside of a range in a sample and have the characteristic that if they are eliminated, they can change the values of some parameter of interest by over 10% [28]. Moreover, these values can be calculated and grouped with statistical or machine learning methods. Generally, outliers are processed in three ways:

- keep as genuine value in the database.
- modify its value with some statistical method (e.g., the average).
- eliminate it.

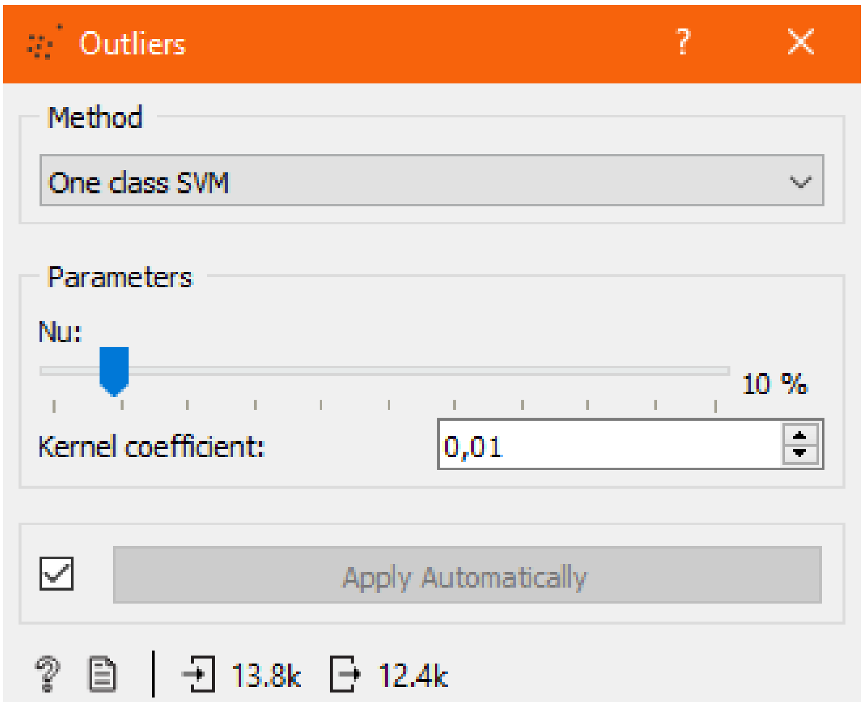


Figure 4.2: Parameters to compute Outliers on Orange package

The outliers, in this study, are taken in a different way. These will be taken as values that represent events that guarantee to visualize the peaks of the earthquakes data with a significant magnitude.

Orange is open source machine learning, and data mining, also it builds data analysis workflows visually, with a large, diverse toolbox [29]. In this research, Orange is used to calculate outliers with the One-class SVM method (see Figure 4.1). The toolbox shown in Figure 4.2 has the hyperparameters used to compute the outliers.

Hyperparameters were chosen after to prove different combinations between them. The new calculated database has 1378 rows with the original eight columns and these maintain all catastrophic earthquake data as shown in Figure 5.1.

4.3 PCA Method

Principal component analysis is a method that allows reducing the dimensionality of a data set, maintaining and minimizing the information in a lower dimensionality [30]. This machine learning method is used in this work to extract and combine the characteristics of the sample rows into a single vector. To calculate these values of the PCA vector, the Orange application is used with the parameters previously tested as shown in Figure 4.3.

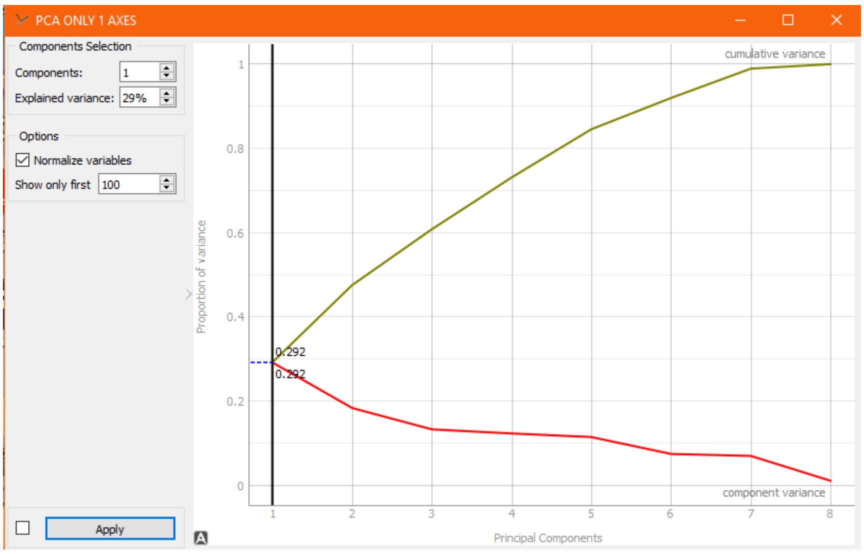


Figure 4.3: Parameters to compute PCA on Orange package

4.3.1 Normalization

One of the most successful ways to safeguard concise data for analysis is to normalize data and pre-process it. Normalizing data is a technique applied to the dataset reducing its redundancy in order to use it optimally. The main goal of this technique is to associate similar shapes to the same data in a single data shape. In fact, according to Shalabi L., Shaaban Z., & Kasasbeh B.,(2016) ”Normalization is the scaling technique or a mapping technique or a pre-processing stage”. On the other hand, for many ANN algorithms used

Table 4.1: Parameters and values to find a normalized vector to used in Autoencoder phase.

Num	Month	NDay	Year	Lat	Long	Depth	Mag	Date	PCA	Norm
1	1	7	1901	-2	-82	0	7,2	1/7/1901	4,01625	0,97378
2	1	31	1906	0,95	-79,36	20	8,4	1/31/1906	4,39361	1
3	9	28	1906	-2	-79	150	7,5	9/28/1906	4,32541	0,99526
4	6	1	1907	0	-82	0	7	6/1/1907	3,75460	0,95560
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
567	3	6	1987	-0,08	-77,81	12	7,1	3/6/1987	1,82857	0,82179
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1032	4	16	2016	0,31	-80,12	17,4	7,6	4/16/2016	-0,07462	0,68956
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1378	6	30	2020	-0,65	-81,19	4,08	4,2	6/30/2020	-2,01004	0,55510

in Data Science to work better the input variables to the algorithm must be normalized. However, a bad choice of the normalization method can create an unexpected result in the data and the analysis.

There are many methods to normalize such as min-max, z-score, decimal scaling, etc. Nevertheless, for this work these methods do not provide the required values to use as an input on the implementation of Autoencoder, the values required are between 0 to 1. For this reason, the method chosen is proper and particular, and it is called CEGG method. In fact, the values obtained within the PCA method are scattered in a range between 4.39361 to -2.89566. Therefore, using these values as input in the Autoencoder would cause sensitivity loss in the training and testing stage.

The CEGG method implemented here has elementary mathematician calculus. The first step is to sum a suitable value to eliminate negative numbers, in this case, the number 10 is which got a better normalization, the second step is to find the maximum value of the new PCA vector, once this value is finding, each value is divided for the maximum value and it is stored in a new vector that will use in encoder phase as an input.

Table 4.1 shows the normalized values between 1 and 0 in the last column. Also, it contains all the data in an orderly fashion on the time scale from 1901 to 2020.

4.4 Autoencoder implementation

This section illustrates the detailed structure and training method for the single-code encoder.

It is expected that in its eagerness to reconstruct binary data the autoencoder also reproduces the data to be predicted with high accuracy.

4.4.1 Design

The present Autoencoder implementation has an encoding and a decoding unit. The encoding part learns the feature representation of the input data sample under a sigmoid function that is a parameterized mapping [31]. The decoding counterpart reconstructs the input with the same parameterized mapping. The sigmoid function between input, hidden, and output layer ensures that hidden layer variables are valued between 0 and 1. Such as constraint is a reconstructed input that can take values between 0 to 1 and thus decoder can be interpreted as linear.

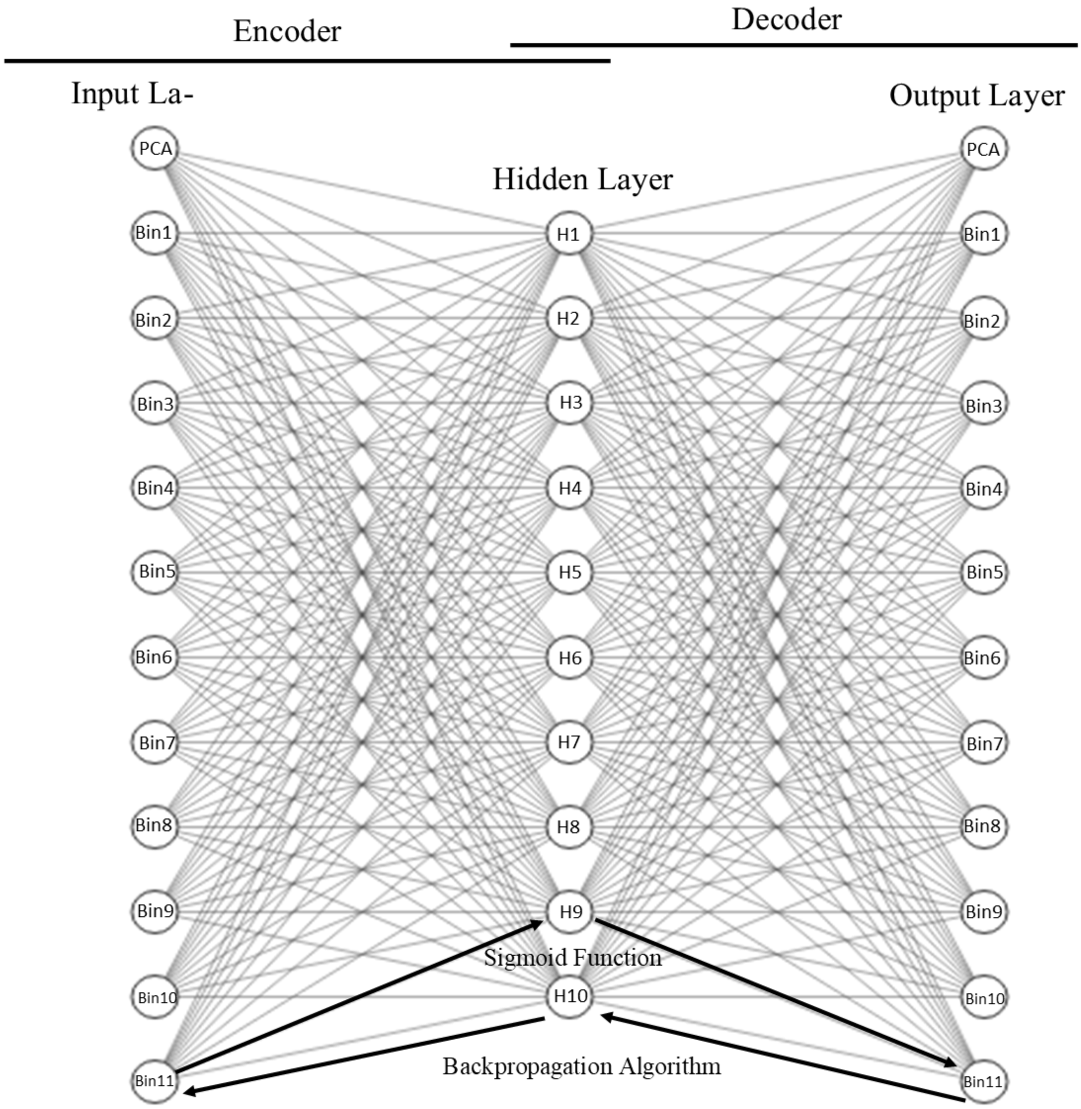


Figure 4.4: Autoencoder Structure

The design of the autoencoder is basic, but at the same time, it accomplishes the purpose of this study. Data processing with One-class SVM and PCA techniques ensures the correct performance of the autoencoder. Figure 4.4 shows the complete structure that consisting of 12 nodes in the input layer, 10 nodes in the hidden layer, and 12 nodes in the output layer. Additionally, this figure shows how the weights and errors between the autoencoder layers are computed.

The Autoencoder is designed to reconstruct its own inputs [23]. In this specific case, it is forced to reproduce binary data and it is expected that in this effort it will also carry the exact reproduction of the PCA values.

4.4.2 Sigmoid

In the implementation of artificial neural networks, the logarithmic sigmoidal function is the most used as the activation function of neurons between the layers. In this case, to calculate the neuron values on hidden and output layers, this function was used with a modified value of x that is equal to 1.5 times its value. As can be seen in figure 4.5, a major amplitude will be achieved in the values of the ordinate axis, these values are similar to the PCA1 vector with which a greater precision in the prediction is expected.

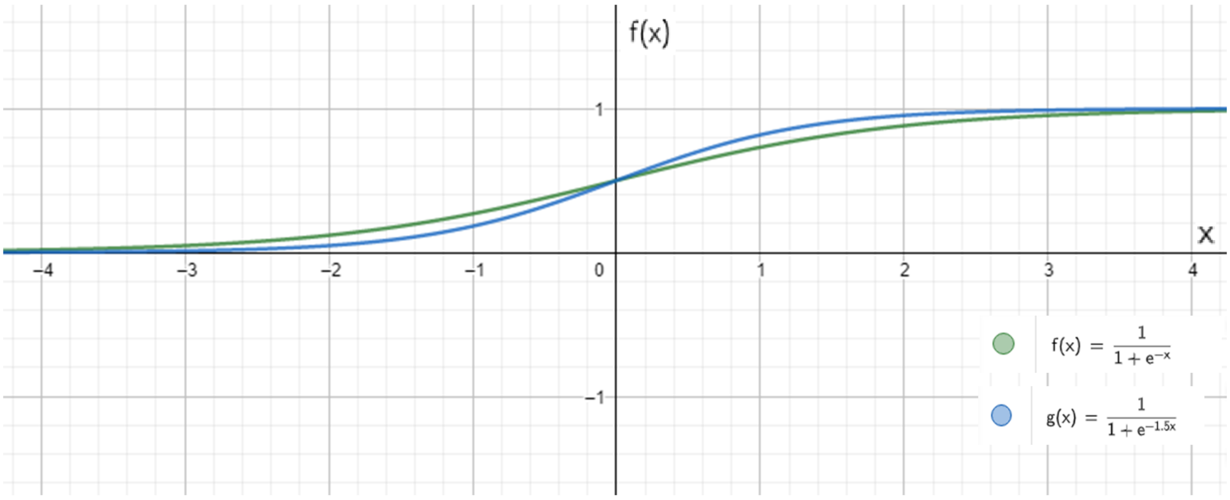


Figure 4.5: Sigmoid Function

4.4.3 Backpropagation

The backpropagation training algorithm is commonly employed to propagated the error into the weights of the hidden and output layers. The algorithm implemented have the following structure:

- Compute the network output from the input value sets.
- Compare with the correct output and compute the error according to the formulas:
  - $output\_error = CO \cdot (1 - CO) \cdot (target - CO)$
  - $hidden\_error = CH \cdot (1 - CH) \cdot \sum_{i=1}^{n}(output\_error - output\_weights)$
    Where CO are computed outputs values, and CH are computed hidden values.

- Compute and storage the partial derivatives of the error with respect to the weights that join the hidden layer with the output layer.
- Compute and storage the partial derivatives of the error with respect to the weights that join the input layer with the hidden layer.
- Adjust the weights of each neuron to reduce error.
- Repeat the process for each entry-exit iteration of the training phase.

4.4.4 Hyperparameter

To achieve the correct efficiency of neural networks, the selection of an optimal architecture to carry out certain tasks remains an open problem [32]. In fact, the implemented neural network here depends on some hyperparametric values, namely the number of hidden layers, the number of neurons in the hidden layer, gain factors, etc. The structure of the Autoencoder has been designed and tested with the trial error technique to evaluate and verify its efficiency. The hyperparameters are:

- Hidden layers = 1.
- $N\_HID = 10$; number of neurons in hidden layer.
- $Gain = 1.5$; value to calculate sigmoid function.
- $Eta = 0,025$; value to compute error in backpropagation.
- $alfa = 0,5$; value to correct the errors in the backpropagation.
- $Objetive\_Train = 85\%$; value to perform the training phase. Then,15% is used to perform the testing phase.

Chapter 5

Discussion and Results

This chapter shows all steps that had been carried out in order to achieve the better earthquake predictions. The results of three phase in this work are shown with figures and tables that describe the accuracy value obtained in each phase.

5.1 Debugged dataset

Selecting data, each dataset was manually and independently examined and depurated looking for the format, rows, columns, common features, obtaining unique file with 13766 items, each one has 8 features and 1 meta attribute. Three datasets were joined into a single file called earthquakeDataOk.csv to start the data analysis, see Table 5.1.

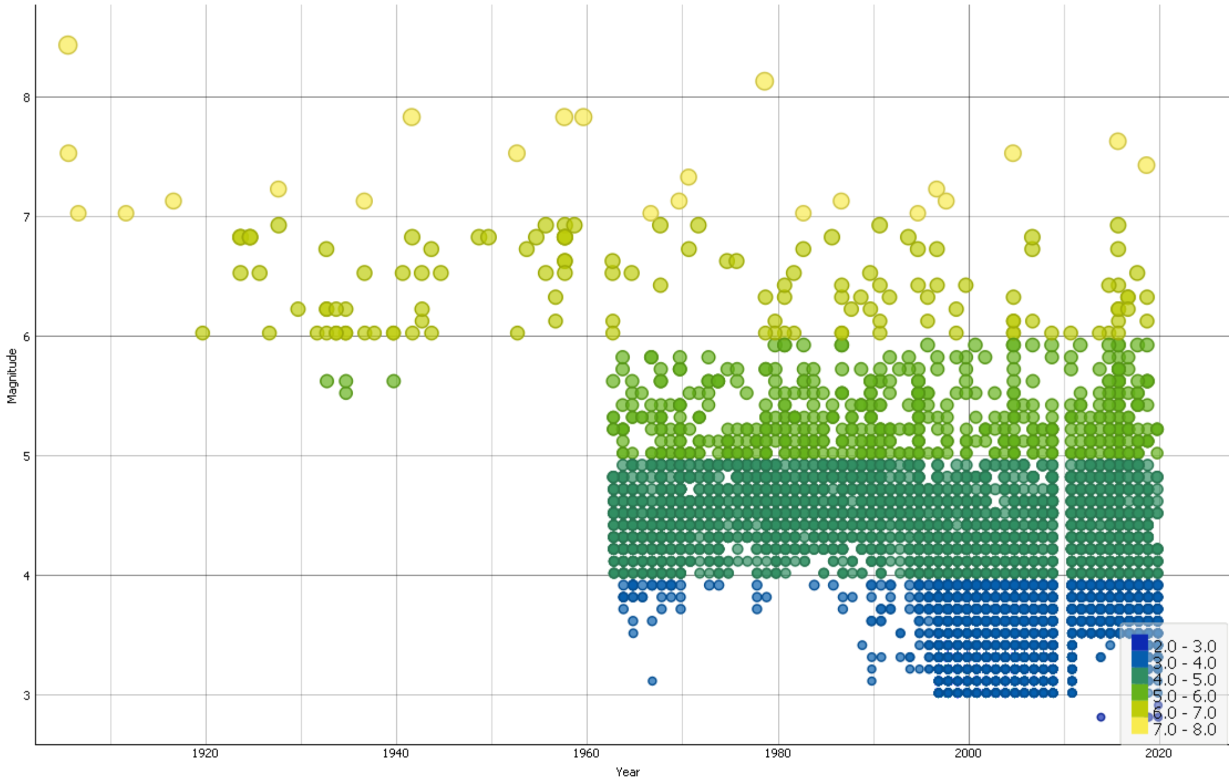
Table 5.1: Data provided by IGEPN

zip name	file name	time	# data
catalogo_csv.rar	catalogohomogenizado.txt	1901 - 2009	10823
sismos_2010_2011_csv.rar	IGEPN.ene.2010_dic.2011.csv	2010 - 2011	270
sismos_2012_may2020_csv.rar	ev_2012 2020.csv	2012 - 2020	2673
Total items	earthquakeDataOk.csv	1901 - 2020	13766

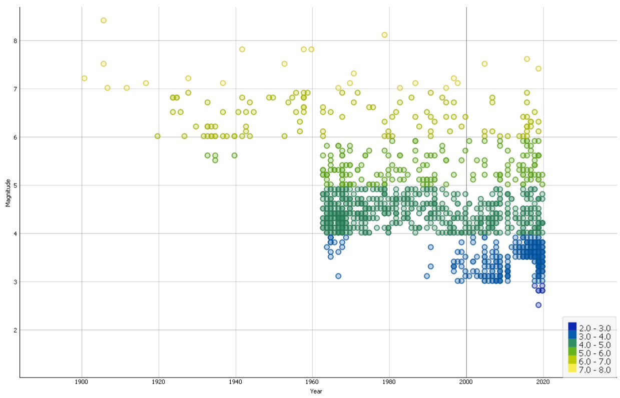
5.1.1 One_class SVM

To ensure the correct performance concerning earthquake predictions, the first step applied to the depurated file earhquekeDataOk.cvs was keeping the same characteristics but a reduced size of items. The results showed in Figure 5.1 show that once the file is passed by the One-class SVM method, the outliers measures are presents in the output of a new dataset that contains 1378 items.

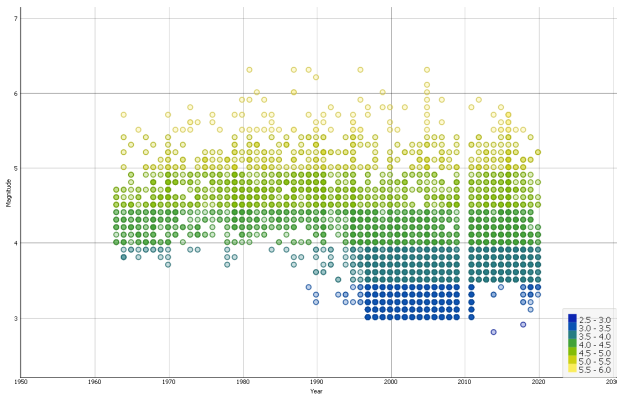
Regarding the main objective of this study, which is to predict earthquakes ≥ 6 degrees on the Richter scale, the reduction of the data In this first phase with One-class SVM only suffers a loss of 8.1632% , Table N shows that only 12 values are lost. In fact, earthquakes



(a) Input data



(b) Outliers



(c) Inliers

Figure 5.1: Plot to show how SVM works to maintain the most catastrophic earthquakes data with Outliers and Inliers Technique

≥ 7 degrees, which would the most dangerous [33], they keep in 24 values like a original dataset, this imply a loos of 0%.

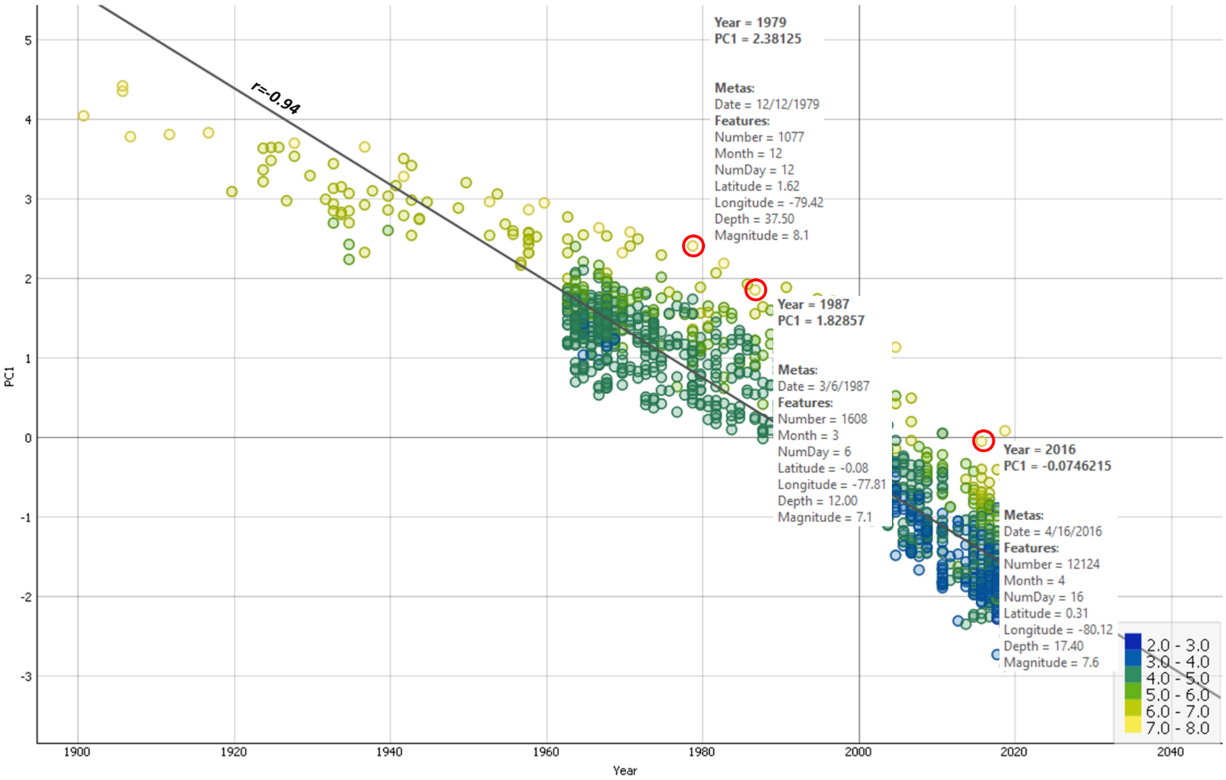


Figure 5.2: Values of PCA vector

For many researchers, outliers mean missing values because they may be unimportant evidence that they would rather ignore [34]. In this work, they are sources of interesting information. Previous knowledge about the consequences of earthquakes greater than 6 degrees and that these are atypical allows deciding what to do with them, Figures 5.1a and 5.1b are valuable in the sense that it is possible to visualize how the data is refined so that these values are not lost essential information. On the other hand, figure 5.1c shows how the inliers only contain values that can be considered redundant for this work.

5.1.2 PCA

The outlier identification method was the first step. The result of it was used to make up the dataset as an input to PCA method, the variables used (eight features and one meta attribute), for choose one PCA axis.

Figure 5.2 shows how the amount of information captured by the only main component decreases according to its number, that is, the main component number one represents more information than the two and so on. The values of the PCA vector are the projections of the 9 values of each row. In fact, the values of this vector are the eigenvalues which measure the amount of variance captured, that is, the information that each principal component represents. In addition, the calculated value of the correlation coefficient $r = -0.94$ shows that the correlation between the year variable and the PCA vector is inverse, which explains the shape of the graph, which has a descending form, but without losing the characteristics of the original seismic magnitudes.

5.1.3 Normalization

Normalization refers to the intention that the normalized values allow eliminating the downstream effects in the PCA vector. Some types of normalization involve only scaling, to arrive at values relative to some measurement variable.

Figure 5.3 plots the results of 2 known normalization techniques plus the CEGG method and their respective results, the CEGG method has the best-normalized values to implement the autoencoder phase. Moreover, Figure 5.5 shows how the normalization of the values has similar performance in terms of the earthquake magnitude values.

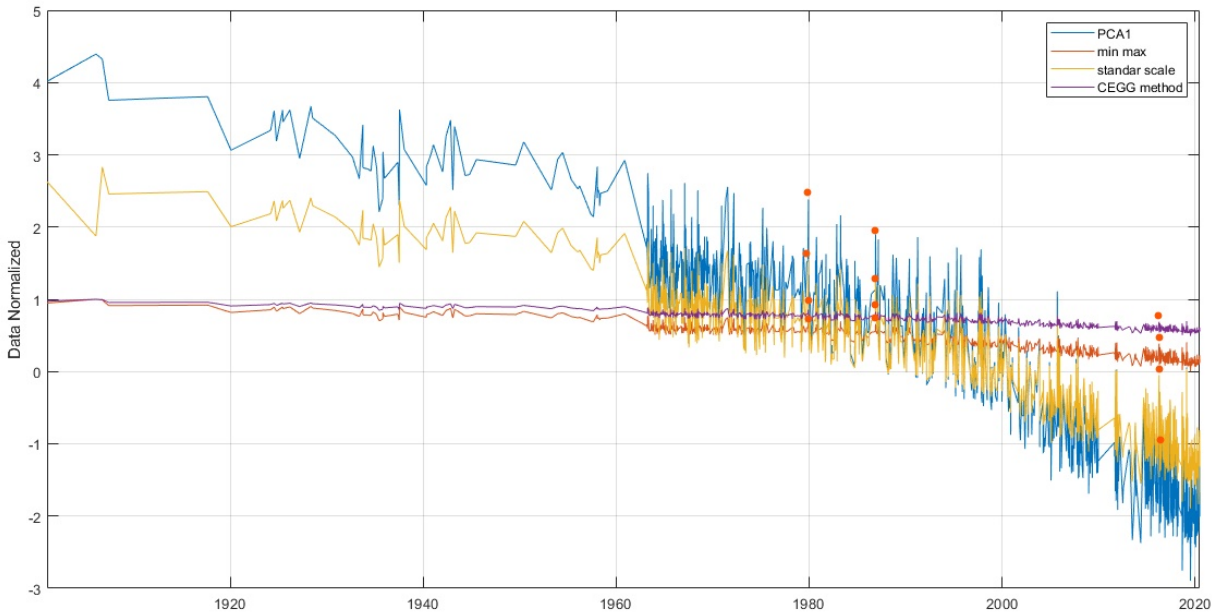


Figure 5.3: Three normalization methods and PCA1 values

5.2 Binary data and PCA

The last step to debug the data is linearize the time, which will be used as input in the autoencoder, this means, choose the highest seismic data of each month in each year starting from January of the year 1901 to June of 2020. It generates a total of 1434 data values, which is stored in binary structure as shown in table 5.2 . Figure 5.4 shows how the magnitude values and the PCA vector have the same tendency.

Figure 5.5 exactly shows how the peaks of the seismic data appear in the normalization of the data. To observe this trend, a period was chosen in which an earthquake of great magnitude and severe consequences occurred in Ecuador.

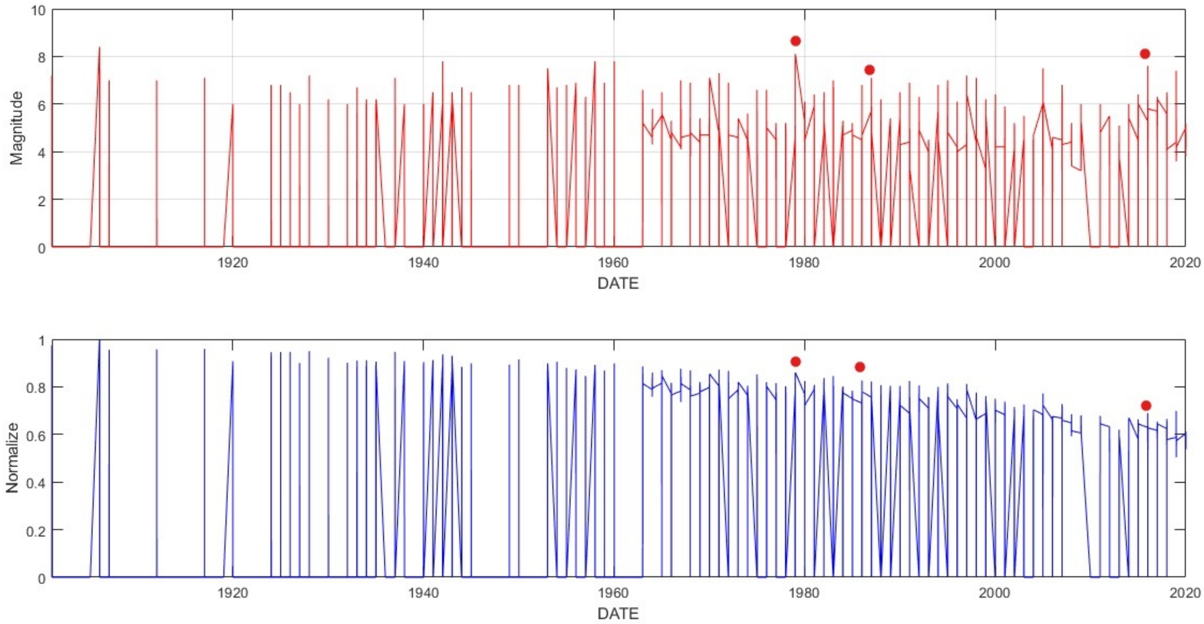


Figure 5.4: Magnitude and normalize data in sequential time

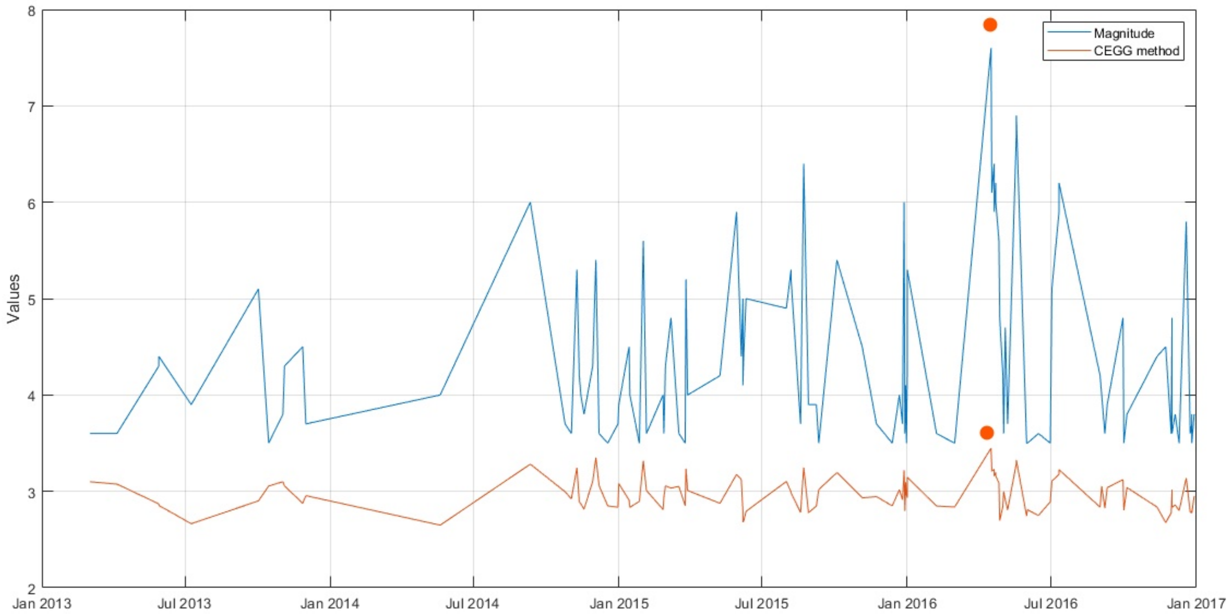


Figure 5.5: Real magnitude data vs normalize data multiply by 3, between 2013 to 2016

5.3 Autoencoder Results

The Autoencoder is designeg to find the best architecture to produce a large earthquake prediction. The Autoencoder proposed uses three steps: upload data, feed, and backpropagation. In addition, there are 3 seismic events chosen to show the accuracy of training and testing phase (points 948, 1035, and 1384 showed in table 5.2).A portion of the dataset (last 688 values) is used of which 85% is given for training and 15% for testing. In fact,

Table 5.2: Binary structure and PCA.

Number	Month/Year	Binary number	PCA value	IGEPN data
1	01/1901	0 0 0 0 0 0 0 0 0 0 1	0,973782946	yes
2	02/1901	0 0 0 0 0 0 0 0 0 1 0	0	no
3	03/1901	0 0 0 0 0 0 0 0 0 1 1	0	no
⋮	⋮	⋮	⋮	⋮
948	12/1979	0 1 1 1 0 1 1 0 1 0 0	0,860190358	yes
⋮	⋮	⋮	⋮	⋮
1035	03/1987	1 0 0 0 0 0 0 1 0 1 1	0,821793129	yes
⋮	⋮	⋮	⋮	⋮
1384	04/2016	1 0 1 0 1 1 0 1 0 0 0	0,689568257	yes
⋮	⋮	⋮	⋮	⋮
1434	06/2020	1 0 1 1 0 0 1 1 0 1 0	0,608615533	yes

this portion of dataset is which has the major consecutive data recolection, this is evident in Figure 5.4

Experimental results showed that the best architecture has 10 neurons in the hidden layer because the rebuilt error decreases quickly. Another important value is the number of epoch in the training phase which is 1000 because this produces the best accuracy in the testing phase. Other hyperparameters that yield better performance are variables called $Eta = 0,025$, and $alfa = 0,5$ that which are used to compute the error of hidden and output layers. Finally, the hyperparameter $Gain = 1.5$ was used to compute the sigmoid function. Figure 5.6 shows the features and results of this process. Letter **A** represents the first input in the Autoencoder; letter **B** is the representation of the training process; and letter **C** shows the prediction features in the output values.

The training phase is one of the most important parts in Autoencoder implementation to achieve good results in the prediction phase, specifically maintaining almost exactly the values of the PCA vector is the main objective of this phase. In a particular way, it is interesting to observe in Figure 5.7 how the peaks that are indicators of earthquakes with large magnitudes and catastrophic consequences are reproduced maintaining the values of the input data. The values that are calculated in the hidden layer of this phase are used later in the prediction of seismic events.

For the testing of the predictive model, 104 data were used that represent approximately 9 years of prediction, it should be noted that this period of time was never taken into account for any other process that is different from the prediction. Figure 5.8 shows the comparison between the PCA data and the prediction made by the model. A result to be highlighted is visible in the control point located at data 638, which shows with reasonable precision the time and the PCA value corresponding to the earthquake that occurred in April 2016. In addition, a positive sign that the model works properly is that the prediction loses precision as time advances into the future, this is approximately 25%

of the final stretch of the graph shown in Figure 5.8.

The method proposed in this work is new, the main objective of which is to reproduce binary data, and that in this reproduction attempt of data binary was achieved values of PCA with a reasonable prediction and small error margins.

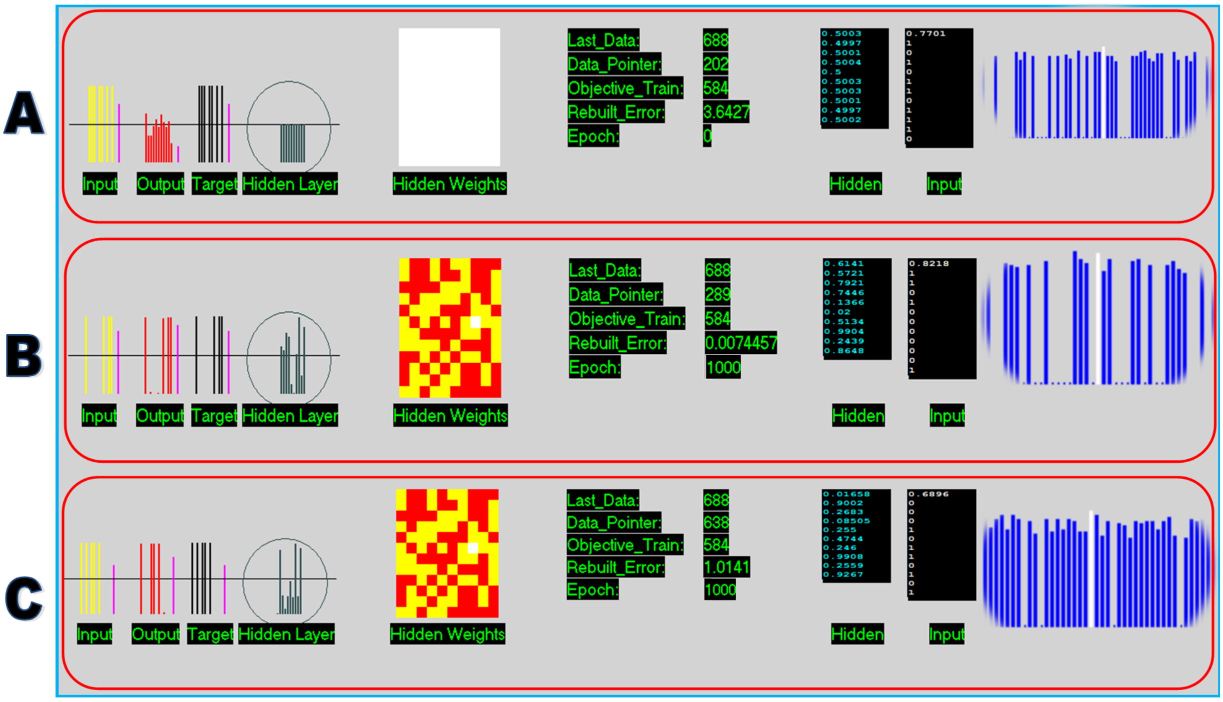


Figure 5.6: This Graph represents the conditions of how the Autoencoder works.
Form A: is the beginning of the work in the Autoencoder implementation. Here, Hidden weights have the value O. For this reason, the diagram is white. The output and the Hidden Layer have randomized values, and they do not have the required representation.
Form B: is the finish of the training phase. Here, Hidden weights take positives and negatives values, red and yellow respectively, that make a diagram of the main characteristics of input data. The output representation is the same as the target that is expected.
From C: is the testing phase. Here, the Hidden weights diagram is the same that the training phase. output representation shows a similar shape to the target which is the prediction expected.

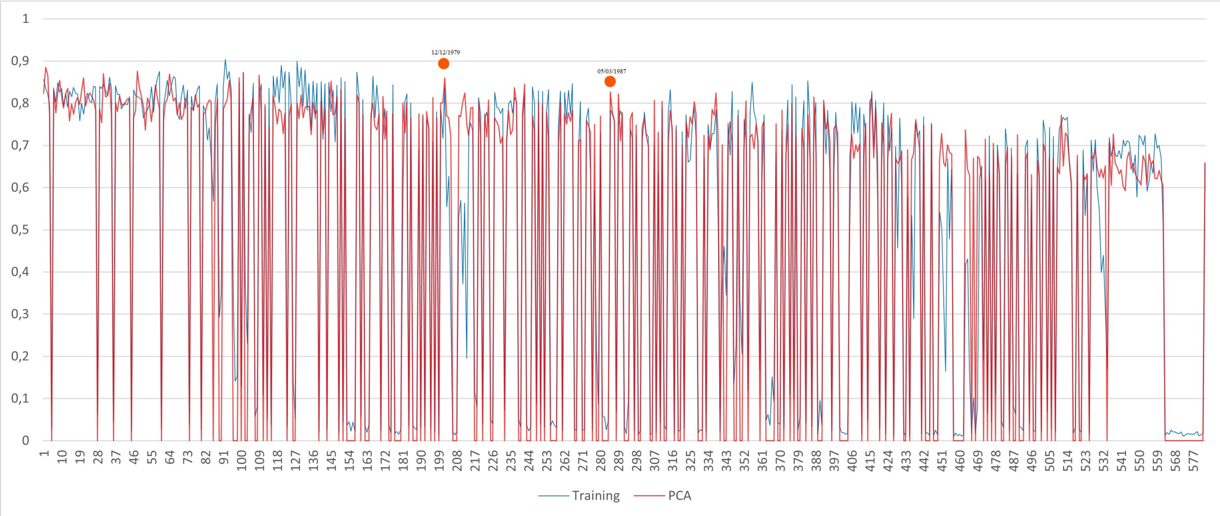


Figure 5.7: PCA vs Output values in training phase.

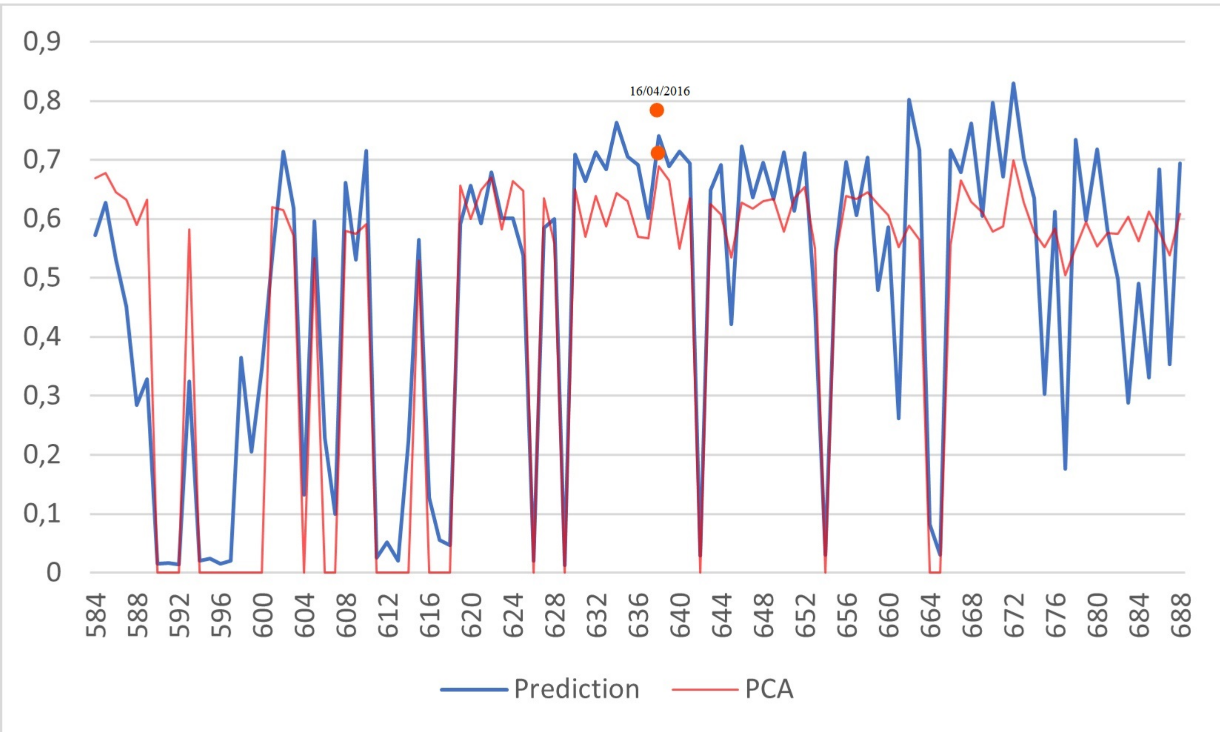


Figure 5.8: PCA vs Prediction values in testing Phase.

Chapter 6

Conclusions and Future Works

Since human beings built their homes on the earth's crust, earthquakes have caused the enormous economic and human loss through the ages. For example, while writing the present undergraduate work; Haiti was shaken by a seismic of 7.2 on the Richter scale, causing enormous human and physical losses [35] [24]. On the other hand, artificial neural networks, and machine learning are tools that scientists use to predict this phenomenon's occurrences. Thus, this work through the techniques of Support Vector Machine, Principal Component Analysis and Autoencoder presents a promising tool to efficiently predict earthquakes in Ecuador.

The data provided by IGEPN was selected in a good manner to do a homogenized table with one meta attribute and eight features. This selection ensured that SVM and PCA inside to Orange Application work efficiently to calculate the outliers and compute one component of PCA. In addition, normalize PCA vector helps to found standardized values, which becomes the inputs to the autoencoder.

The autoencoder was implemented in Borland C++ and received as input unlabeled data the PCA component and the time at which the attached earthquake event happens. In particular, the time variable was converted to a binary coded decimal (BCD) arrangement, where the related input neurons assume the values 0 or 1, and the time always moves forward. The logic behind this procedure is that in its quest to reconstruct pure binary information the autoencoder also reconstructs earthquake prediction in efficient way. When writing the code, a difficult task was to find appropriate hyperparameter values like the number of neurons in the hidden layer, percent used to training and testing phase, the number of the epoch, and the reduction and standardization of the irregular available data. The combination of PCA and a binary reconstructing autoencoder proves to be a powerful combination that produces auspicious results, with a better prediction performance as compared with other studies carried out in the same field.

The results obtained in this work could help the Secretaría de Gestión de Riesgos, which is the Ecuadorian government organization that aims to protect the life and property of the affected population, caring for and protecting their rights [36], as well as other public and private entities to plan, manage and minimize the serious effects of earthquakes. In addition, with time as a predictor variable, they would have a reasonable time to develop effective measures that contribute to avoiding life and economic losses at the time of the earthquake.

6.1 Future Works

A future task that remains to be carried out is the prediction of the place where an earthquake might occur. This aim is possible because the data provided by IGEPN has parameters values of latitude and longitude. Therefore, with machine learning techniques and artificial intelligence, it is possible to map these attributes and make clusters to locate possible places where an earthquake will occur.

Finally, earthquake prediction is a very difficult task to perform, but at the same time, good results have been bringing about using different ANN techniques. Although, if this objective is achieved, the governmental institutions of the respective countries would have the opportunity to save several human lives and build contingency plans in the possible affected areas.

Bibliography

[1] D. Guha-Sapir and F. Vos, “Earthquakes, an epidemiological perspective on patterns and trends,” in *Human Casualties in Earthquakes.* Springer Netherlands, Dec. 2010, pp. 13–24. [Online]. Available: https://doi.org/10.1007/978-90-481-9455-1_2

[2] A. Woodruff. (2019) What is a neuron? [Online]. Available: https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron

[3] H. J. Shin, D.-H. Eom, and S.-S. Kim, “One-class support vector machines—an application in machine fault detection and classification,” *Computers & Industrial Engineering*, vol. 48, no. 2, pp. 395–408, Mar. 2005. [Online]. Available: https://doi.org/10.1016/j.cie.2005.01.009

[4] O. IQ. (2021) Understand autoencoders by implementing in tensorflow. [Online]. Available: https://iq.opengenus.org/implementing-autoencoder-tensorflow/

[5] P. Baldi, “Autoencoders, unsupervised learning, and deep architectures,” in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, ser. Proceedings of Machine Learning Research, I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver, Eds., vol. 27. Bellevue, Washington, USA: PMLR, 02 Jul 2012, pp. 37–49. [Online]. Available: http://proceedings.mlr.press/v27/baldi12a.html

[6] M. Agius, S. Hicks, W. Bohon, P. Denton, C. Eakin, A. Gil, L. Gualtieri, C. Mora-Stock, K. Van Noten, A. Schaeffer, S. Tharimena, S. Parolai, and R. Sarkar, “An international day for seismology: Rationale, co-ordination, and planning,” 04 2018.

[7] Q. Kong, D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, and P. Gerstoft, “Machine learning in seismology: Turning data into insights,” *Seismological Research Letters*, vol. 90, no. 1, pp. 3–14, Nov. 2018. [Online]. Available: https://doi.org/10.1785/0220180259

[8] Wikipedia, “Terremoto de colombia de 1979 — wikipedia, la enciclopedia libre,” 2021, [Internet; descargado 12-septiembre-2021]. [Online]. Available: https://es.wikipedia.org/w/index.php?title=Terremoto_de_Colombia_de_1979&oldid=139221020

[9] G. Berz, “List of major natural disasters, 1960-1987,” vol. 1, no. 1, pp. 97–99, 1988. [Online]. Available: https://doi.org/10.1007/bf00168223

[10] R. M. F. Gerstner, F. Lara-Lara, E. Vasconez, G. Viscor, J. D. Jarrin, and E. Ortiz-Prado, “Correction to: Earthquake-related stressors associated with

suicidality, depression, anxiety and post-traumatic stress in adolescents from muisne after the earthquake 2016 in ecuador," vol. 21, no. 1, May 2021. [Online]. Available: https://doi.org/10.1186/s12888-021-03238-7

[11] P. comin Charamonti and C. de Barros Gomes, *ALKALINE MAGMATISM IN CENTRAL - EASTERN PARAGUAY*, 1st ed. Av. Prof. Luciano Gualberto, Travessa j , 374: Universidad de Sao Paulo, 1995, ch. SEISMICITY IN PARAGUAY AND NEIGHBOURING REGIONS, pp. 63–64.

[12] J. Daniell, F. Wenzel, and B. Khazai, "The cost of historic earthquakes today – economicanalysis since 1900 through the use of catdat," 01 2010.

[13] C. Beauval, H. Yepes, P. Palacios, M. Segovia, A. Alvarado, Y. Font, J. Aguilar, L. Troncoso, and S. Vaca, "An earthquake catalog for seismic hazard assessment in ecuador," *Bulletin of the Seismological Society of America*, vol. 103, no. 2A, pp. 773–786, Mar. 2013. [Online]. Available: https://doi.org/10.1785/0120120270

[14] I. Bodnar, "Aristotle's Natural Philosophy," in *The Stanford Encyclopedia of Philosophy*, Spring 2018 ed., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2018.

[15] M. Nielsen, *Neural Networks and Deep Learning*, 1st ed. Determination Press, 2015.

[16] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. [Online]. Available: https://doi.org/10.1037/h0042519

[17] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," vol. 4, no. 11, p. e00938, Nov. 2018. [Online]. Available: https://doi.org/10.1016/j.heliyon.2018.e00938

[18] C. Cortes and V. Vapnik, *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: https://doi.org/10.1023/a:1022627411411

[19] S. Abe, "Feature selection and extraction," in *Support Vector Machines for Pattern Classification*. Springer London, 2010, pp. 331–341. [Online]. Available: https://doi.org/10.1007/978-1-84996-098-4_7

[20] M. Buhmann, "Radial basis function," *Scholarpedia*, vol. 5, no. 5, p. 9837, 2010, revision #137035.

[21] Wikipedia contributors, "Radial basis function — Wikipedia, the free encyclopedia," 2021, [Online; accessed 16-August-2021]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Radial_basis_function&oldid=1032256175

[22] J. Shlens, "A tutorial on principal component analysis," *CoRR*, vol. abs/1404.1100, 2014. [Online]. Available: http://arxiv.org/abs/1404.1100

[23] O. Chang, P. Constante, A. Gordon, and M. Singaña, "A novel deep neural network that uses space-time features for tracking and recognizing a moving object," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 7, no. 2, pp. 125–136, Feb. 2017. [Online]. Available: https://doi.org/10.1515/jaiscr-2017-0009

[24] "Eeuu advierte de "probable" alto número de vÍctimas por el sismo en haití," Aug 2021. [Online]. Available: https://www.latimes.com/espanol/eeuu/articulo/2021-08-14/eeuu-advierte-de-probable-alto-numero-de-victimas-por-el-sismo-en-haiti

[25] K. M. Asim, A. Idris, T. Iqbal, and F. Martínez-Álvarez, "Earthquake prediction model using support vector regressor and hybrid neural networks," *PLOS ONE*, vol. 13, no. 7, p. e0199004, Jul. 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0199004

[26] G. Asencio-Cortés, F. Martínez-Álvarez, A. Troncoso, and A. Morales-Esteban, "Medium–large earthquake magnitude prediction in tokyo with artificial neural networks," *Neural Computing and Applications*, vol. 28, no. 5, pp. 1043–1055, Nov. 2015. [Online]. Available: https://doi.org/10.1007/s00521-015-2121-7

[27] M. N. Shodiq, D. H. Kusuma, M. G. Rifqi, A. R. Barakbah, and T. Harsono, "Spatial analisys of magnitude distribution for earthquake prediction using neural network based on automatic clustering in indonesia," in *2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*. IEEE, Sep. 2017. [Online]. Available: https://doi.org/10.1109/kcic.2017.8228594

[28] D. Ghosh and A. Vogt, *Section on Survey Research Methods JSM 2012*, p. 3455–3460. [Online]. Available: http://www.asasrms.org/Proceedings/y2012/Files/304068_72402.pdf

[29] J. Demsar, T. Curk, A. Erjavec, B. Zupan, M. Zitnik, J. Zbontar, L. Zagar, L. Umek, M. Stajdohar, A. Stari, and et al., "Orange: Data mining toolbox in python," *Journal of Machine Learning Research 14 (2013)*, p. 2349–2353, Aug 2013. [Online]. Available: https://www.jmlr.org/papers/volume14/demsar13a/demsar13a.pdf

[30] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37-52, Aug. 1987. [Online]. Available: https://doi.org/10.1016/0169-7439(87)80084-9

[31] A. Gogna and A. Majumdar, "Semi supervised autoencoder," in *Neural Information Processing*. Springer International Publishing, 2016, pp. 82–89. [Online]. Available: https://doi.org/10.1007/978-3-319-46672-9_10

[32] S. Loussaief and A. Abdelkrim, "Convolutional neural network hyper-parameters optimization based on genetic algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, 2018. [Online]. Available: https://doi.org/10.14569/ijacsa.2018.091031

[33] D. Dooley, R. Catalano, S. Mishra, and S. Serxner, "Earthquake preparedness: Predictors in a community survey1," *Journal of Applied Social Psychology*, vol. 22, no. 6, pp. 451–470, Mar. 1992. [Online]. Available: https://doi.org/10.1111/j.1559-1816.1992.tb00984.x

[34] A. Unwin, "Multivariate outliers and the o3 plot," *Journal of Computational and Graphical Statistics*, vol. 28, no. 3, pp. 635–643, Apr. 2019. [Online]. Available: https://doi.org/10.1080/10618600.2019.1575226

[35] "El terremoto de haití deja ya más de 2.100 muertos y casi 10.000 heridos." [Online]. Available: https://www.bbc.com/mundo/noticias-internacional-58214474

[36] [Online]. Available: https://www.gestionderiesgos.gob.ec/wp-content/uploads/downloads/2018/08/Plan-Nacional-de-Respuesta-SGR-RespondeEC.pdf

Appendices

.1 Appendix 1.

[Your appendix]