Mathematical Modelling and Self-Learning Systems

Masters Thesis

Volcanoes as dynamical systems: A neural network approach to eruption detection

Author: Patrick John Falvey

Supervisor: Dr. Andrew Keane

School of Mathematical Sciences
University College Cork,
Ireland
03/10/2022

Declaration of Authorship

This report is wholly the work of the author, except where explicitly stated otherwise. The source of any material which was not created by the author has been clearly cited.

Date: 03/10/2022

Signature: *Patrick Falvey*

Acknowledgements

I would like to express my sincere thanks to my supervisor, Dr. Andrew Keane, whose advice and guidance made this thesis possible.

Abstract

A volcano can be described as a dynamical system with two attractors. The steady state attractor reflects the volcano in a rest (dormant) interval and the active state attractor depicting it in a period of eruption. In this work a neural network is used to monitor actual time series seismic data to determine whether it can interpret the existence of the two different attractors within the system. The monitoring is done by a long-short term memory (LSTM) neural network which is trained to learn the dynamics of the 'normal activity' tremor.

Contents

List of Figures

Introduction

On the 9^{th} of Decemeber 2019, just after two o'clock in the afternoon local time, Whakaari (White Island), a volcano situated off the North Coastline of New Zealand's North Island experienced an phreatic eruption. The significance of this eruption was that 47 people were present on the island that day, of which twenty-one died, and twenty-six suffered injuries [1]. The event displayed the necessity for improved safety and efficient monitoring of volcanoes, in particular those that have a large number of visitors.

Due to the danger posed by volcanoes, it is crucial that there is an understanding of their dynamics in order to understand when volcanoes might erupt. Volcanologists and seismologists use various forms of data such as rising ground temperatures, an increase in sulfur dioxide gas from the volcano and increased seismic activity in the form of frequent minor earthquakes determine if a volcano is about to erupt. There has also been research into developing models that can interrupt this data so that a computer can understand when volcano might erupt.

Neural networks are powerful techniques which may be used to solve various complex problems using deep learning [2]. They have been applied to a number of different datasets and fields with great success, for example handwriting recognition [3] fraud detection [4], playing chess [5] and deep fakes [6]. The ability that Neural Networks posses to approximate arbitrary functions and process large multi-dimensional dataset are one of its key advantages. Neural Networks are a form of supervised learning. Supervised learning involves training a model using labelled (known) data to classify or predict outcomes accurately [7]. The growing availability data and the increase in computational power in recent years has led to neural networks becoming a popular choice for many types of analysis. Neural Networks are especially suited to non-parametric modeling given their ability to discover between inputs and outputs that other machine learning techniques are unable to achieve.

The power of artificial intelligence and neural networks can be used to better identify

and understand Volcanic Eruptions.

The aim of this thesis is to analyse the bifurcations of mathematical models for volcano dynamics and investigate evidence of critical transitions in observational data from monitored volcano sites through the application of Neural Networks. The performance of the model is tested on a volcanic tremor dataset representing the 2019 eruption at White Island (Whakaari), New Zealand.

The remainder of this paper is proportioned as follows:

- Chapter 2 is an examination of the background theory of the key concepts used throughout the report,
- Chapter 3 presents an overview of relevant literature. The focus is on volcanic eruptions as Dynamical Systems and the applications of ANN's to volcanic eruption detection and prediction,
- Chapter 4 describes the methodology: how the data was acquired and process and the modelling approach. The Results are also presented in this chapter,
- Chapter 5 presents a discussion of the result as well as briefly examining an outlook on future work.

Theoretical Background

The aim of this thesis is to apply neural a network-based approaches to the challenge of creating a model that is capable of volcanic eruption detection. As described in the previous chapter, Artificial Neural Networks are are a class of machine learning techniques that have found a variety of uses in modelling of dynamical system and provide a flexible yet robust approach to modelling unknown quantities or functions.

In this chapter, the relevant theoretical background and equations for the following topics will be examined:

- Dynamical Systems
- Artificial Neural Networks
- Volcanoes

2.1 Dynamical Systems

Infinitesimal calculus or differential and integral calculus was invented by Leibniz and Newton independently of each other in the late 17^{th} century [8]. This is an efficient manner to predict the future of a system provided that the system in governed by a differential equation. It was through it use that Poincaré started the qualitative theory of dynamical systems while working on celestial mechanics [9] and specifically the three-body problem and the equations of dynamics [10]. His work and the resulting methods developed are the basis for the analysis of nonlinear differential equations. This also included the use of first-return (Poincaré) maps, stability theory for fixed points and periodic orbits, stable and unstable manifolds, and the Poincaré recurrence theorem.

Dynamics is defined as the study of how things change with respect to time, as opposed to describing things simply in terms of their static properties [11]. In a dynamic system, this change is in accordance with a fix set of rules that determine how one state

of the system moves to another state. Examples of a dynamical system is any kind of season that evolves through time, including bacteria growth [12] and the predator prey model [13]. The equation for a simple model of a dynamical system is given in equation 2.1. It describes a system of coupled differential equations.

$$\frac{d}{dt}x = f(x, t, \mu) \tag{2.1}$$

Where:

x: is a vector that represents the state of system. It is a unique minimal description of the system in question.

f: is a set of functions that describe the dynamics of the state (the dynamics). f is a vector valued function that informs us given a current state x how does that state change in the next time instant.

t: time.

μ: are the parameters that can be manipulated in order to try to change the behavior of the system.

The equation 2.1, presents a one dimensional dynamical system given by $\frac{dx}{dt}$ which equates to some function of x and a parameter μ. If μ is a certain fixed value then this dynamical system will have a particular set of stable fixed points, unstable fixed points and half stable fix points. It will have some specific dynamical behavior. However, if μ is gradually changed to another value then it's possible that the dynamical behavior of the differential equation undergoes a drastic change. This transition point in the value of μ at which the behavior of the dynamical system undergoes a significant change is called a bifurcation [14]. An example of a bifurcation might be if for instance, the equation starts with μ_0 at which there is one stable fixed point but if the parameter value μ_0 is increased to μ_t the system might become one with two stable fixed points. Bifurcation is the transition and stability of the differential equation.

Almost all bifurcations can be classified into a selction of possible forms, which are called 'normal forms' or universal unfoldings of bifurcations. These included saddle node bifurcation, pitchfork bifurcation and hopf bifurcation to name a few. In this paper, saddle-node bifurcation is examined in further detail due to its relevance to the dynamical system of a volcano.

2.1.1 Saddle-node (fold) Bifurcation

Suppose an example of a dynamical system of the form:

$$\frac{dx}{dt} = x^2 + \mu \tag{2.2}$$

There exists three possible solutions for μ, positive, zero or negative

When $\mu > 0$:

$$\frac{dx}{dt} = 0 \rightarrow x^2 + \mu = 0 \rightarrow x^2 = -\mu \quad \text{No real soultion, no real fixed points}$$

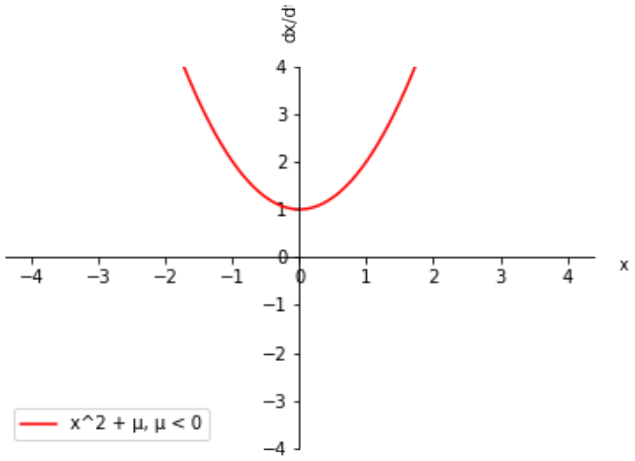

Figure 2.1: Phase Portrait for $\frac{dx}{dt} = x^2 + \mu = 0$, where $\mu > 0$

if μ is positive, there are no fixed points in the dynamical system, as shown in figure 2.1.

When $\mu = 0$:

$$\frac{dx}{dt} = 0 \rightarrow x^2 = 0 \rightarrow x^2 = 0 \quad x = 0$$

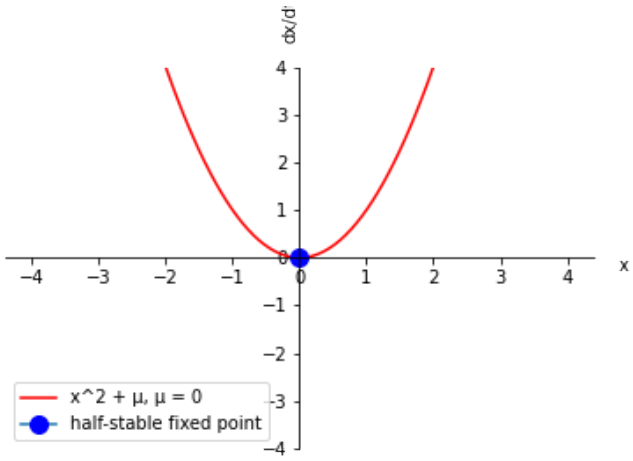

Figure 2.2: Phase Portrait for $\frac{dx}{dt} = x^2 + \mu = 0$, where $\mu = 0$

As shown in figure 2.2, if μ is zero, then solving for the fixed points returns one solution. Evaluating the stability of that solution, it is noted that the fix point is half stable meaning that if you start from a negative x the system will converge to zero but if you start at a positive x the system will diverge from zero.

When $\mu < 0$:

$$\frac{dx}{dt} = 0 \rightarrow x^2 + \mu = 0 \rightarrow x^2 = -\mu \quad x = \pm\sqrt{-\mu}$$

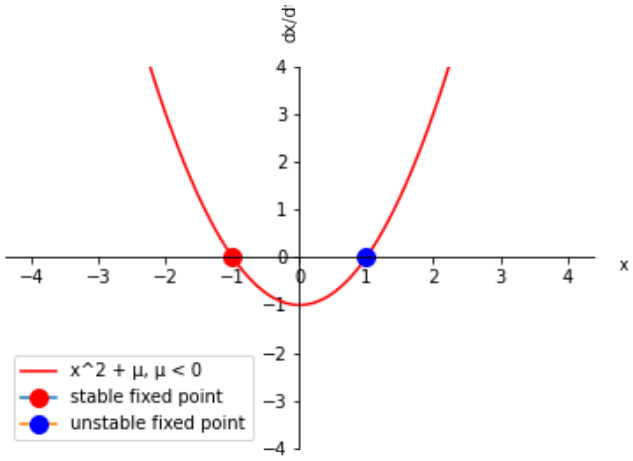

Figure 2.3: Phase Portrait for $\frac{dx}{dt} = x^2 + \mu = 0$, where $\mu < 0$

As shown in figure 2.2, if μ is negative, then solving for the fixed points returns two solutions. Evaluating the stability of those solutions, it is noted that one of those fix points is stable while the other is unstable. If starting from a value of x that is less than $\sqrt{-\mu}$ it will converge to $-\sqrt{-\mu}$ but if starting from a value greater than $\sqrt{-\mu}$ the

value will diverge from $\sqrt{-\mu}$.

The scenario described above as μ goes from a positive value to a negative values results in the creation of two fixed points from nothing. It's also possible to look at this from a reflected point of view, where μ goes from a negative value to a positive value which results in destruction of two fixed points to nothing. Depending on which of these approaches is taken this type of transition is called a saddle node bifurcation (also known as a fold bifurcation).

A bifurcation diagram is used a way to represent a bifurcation. A bifurcation diagram is a plot of the evolution of the fixed points. For our example above, this is a plot of the fixed point x versus μ, the varying parameter that is causing the bifurcation. It's evident that when $\mu < 0$, the fixed point x is a function of μ such that x equaled $\pm\sqrt{-\mu}$. There the bifurcation diagram would just be a plot of this.

The bifurcation diagram for $\frac{dx}{dt} = x^2 + \mu$ is created as follows:

- For $\mu > 0$ there are no fixed points, therefore there is nothing to plot on the part of the coordinate plane where $\mu > 0$, i.e. the right half of the plane.
- For $\mu = 0$ there is a half stable fixed point at $x = 0$, therefore it is plotted at (0,0) on the $\mu - x$ plane.
- For $\mu < 0$ there is a two fixed point at $x = \pm\sqrt{-\mu}$, therefore it is plotted at (-μ,$\sqrt{-\mu}$) and (-μ,$-\sqrt{-\mu}$) on the $\mu - x$ plane.

However, to complete the bifurcation diagram the stable fixed point and unstable fixed point needs to be distinguished on the bifurcation plot. The unstable fixed point was positive while the stable fixed point was negative on the phase portrait diagrams. This is represented on the phase diagram by keeping the bottom curve solid to represent the stable fixed points and make the top curve dashed to denote the fact that the represent unstable branch (made up of unstable fixed points).

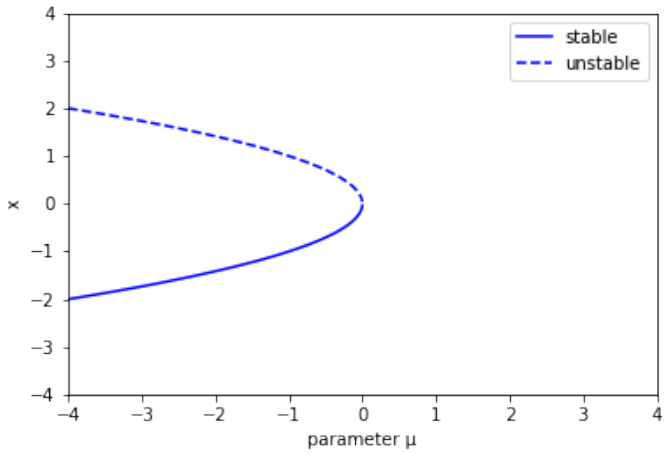

Figure 2.4: Bifurcation diagram for the system $\frac{dx}{dt} = x^2 + \mu = 0$, where μ is the changing parameter

This picture represents the bifurcation diagram which is an effective way of explaining the behavior of a dynamical system where bifurcations arise . For a positive μ, there are no fixed points but as we move to the left there is a half stable fixed point when $\mu = 0$ at $x = 0$. The half stable fixed point then separates into two branches; an unstable branch of fixed points and a stable branch of fixed points when $\mu < 0$.

2.2 Artificial Neural Networks

In real life, many of the relationship between inputs and outputs are non-linear and complex. A key advantage of Artificial Neural Networks is that they possess the ability to both learn and model non-linear and complex relationships.

In the following section, background theory of neural network will be presented in terms of model architecture, LSTMs and relevant algorithms.

2.2.1 Architecture

American psychologist, Frank Rosenblatt, is credited as the first person to develop the concept of a Neural Network. In 1958, he introduced the Perceptron [15]. The perception can be described as a single layer neural network that can classify a dataset as long as the data is linearly separable. For a dataset to be linearly separable it implies the existence of a hyperplane that distinguishes the data into the two classes. Graphically, this hyperplane can be represented as a line that demonstrates a clear distinction between the two classes, see figure 2.5 for an example.

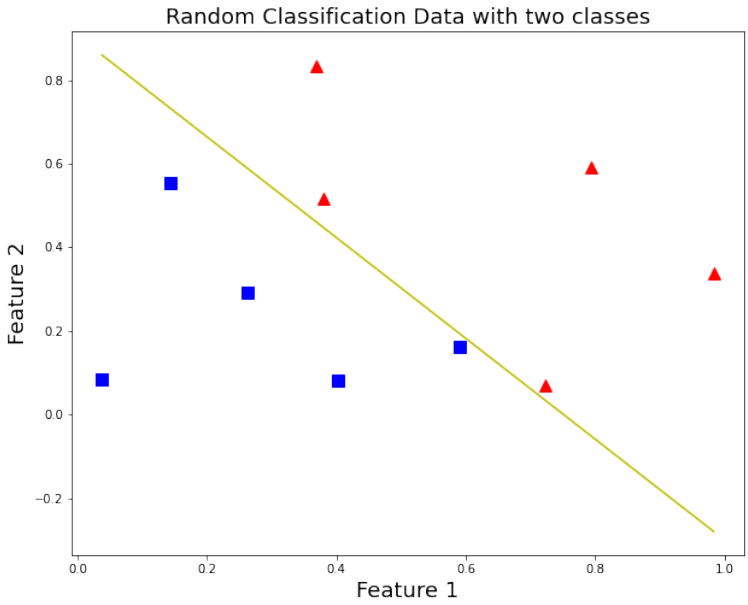

Figure 2.5: The Perceptron: The true separator of the two classes is linear hyperplane (line)

The concept of the Perception was a structure that loosely resembles the shape of neurons in the brain where there exists an

Figure 2.6, presents a typical neural network architecture of the Perceptron. It is explained as follows:

- There is a selected n number of inputs x_i
- A constant input b is selected, in this case = 1, which acts as an offset
- A weight w_i is applied to each of the inputs
- The sum of all the weighted inputs is generated
- A step function is applied to the weighted sum giving a single output y

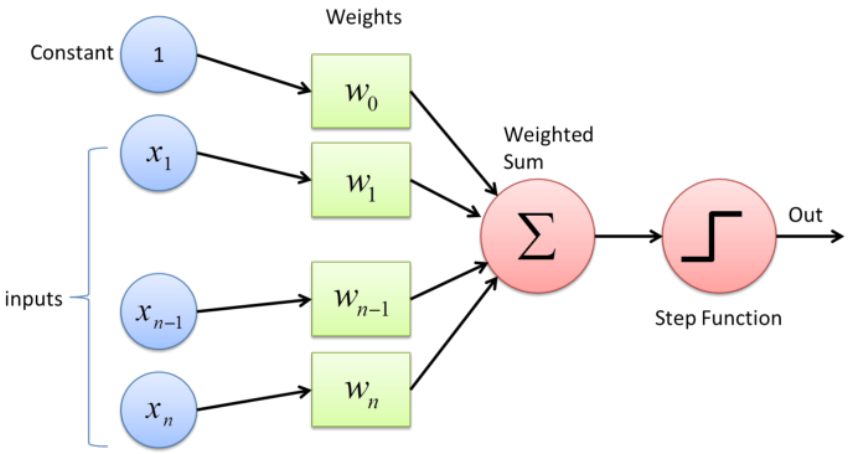

Figure 2.6: Perceptron structure, including inputs and weights [16]

This represents the basic structure of a neural network. The perceptron architecture allows for n number of inputs which are to be then mapped to a single output.

In order to improve on the performance of the perceptron, ANN's developed and, as a result, became more complex. Increasing the number of inputs, increasing the number of layers and applying a variety of activation functions are considered (in general) methods of improving the perceptron's ability to classify data. Another instrument that was developed was the concept of "training" the neural network structure. Training is when the structure learns from a given set of input and their corresponding outputs.

However research on Neural Network met a stagnation point beyond this concept. The reason for this can be attributed to a lack of adequate computing power at the time. In the last half century, computers developed and became more powerful. Their ability to both store and handle data increased almost exponentially. This afforded Neural Network research the instrumentation required to continue to progress. As computing power increased, neural networks have become increasingly more applicable to real-world analysis.

A important Theorem resulting from this progression is that given any continuous function at all, no matter how complicated it might be, it's always possible to find an ANN which can approximate that function as well as you would like. This Theorem is known as the Universal Approximation Theorem, which was proved in a 1989 paper by Cybenko [17]. In this paper, they proved this theorem for the specific case using a sigmoid activation function. This result proved that ANN have the ability to "learn" an unknown function, taking a series of known inputs, x, and mapping them to known outputs, y.

There are a number different types of ANNs including (but not limited to):

- Feedforward Neural Network (Artificial Neuron)
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- and more.

Each have their own attributes that make them preferable for dealing with certain dataset. However, all types of ANN's share a similar structure. That structure being made up of 3 fundamental elements:

- **An input layer** - brings the multidimensional data into the system
- n **number of hidden layers (where** $n \geq 1$**)** - location where all the computation ("learning") is done.
- **An output layer** - produces a single value or a series of values for given inputs through combining the outputs of the final hidden layer.

The layers of a NN are comprised of nodes (neurons). The number of nodes is a controllable parameter when assembling a NN. If the number of nodes are increased the performance of the network may (but not always) improve. The number of layers in a network regulates how the networks 'learns', an increased amount of layers assists the network identify higher dimensional patterns. Hidden layers are stacked in a linear order after the input layer and before the output layer.

An important element of applying neural networks is the activation function. An activation function is applied to each node within a layer. The purpose of an activation function of a node is to define the output of that node given an input or set of inputs. There is a number of different activation functions available to use depending on the problem to be solved. A sample of these activation fucntions are displayed below in Figure 2.7.

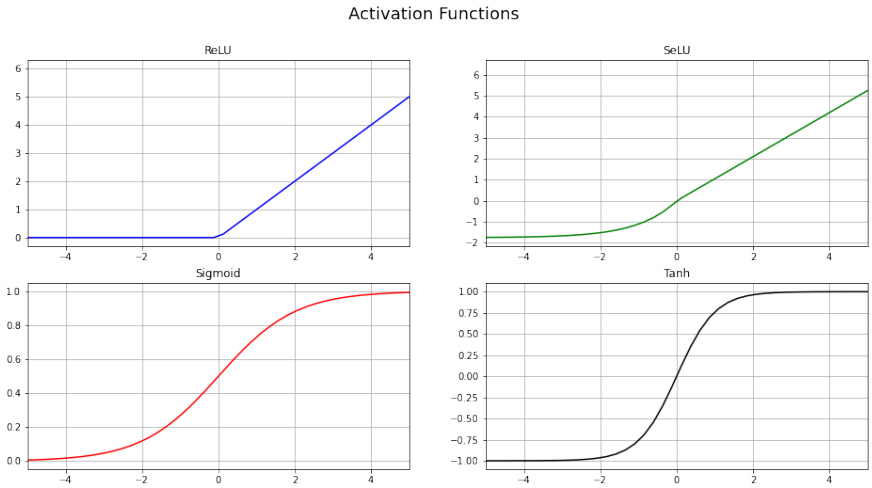

Figure 2.7: A sample of some Activation Functions: ReLU, SeLU, Sigmoid and Tanh

In a Deep Neural Network (DNN) often all nodes in a layer are connected to all nodes in the preceding and proceeding layers. This is often referred to as a dense Network. 'Deep' refers to the fact that there exists a number of layers in the model. In a Dense Network, each node i is scaled by a weight w_i and offset by a bias b_i. An example can be visualised in figure 2.8, the inputs take in the data of individuals height and weights and the output produced a prediction on individuals gender. A single hidden layer is employed in this arrangement.

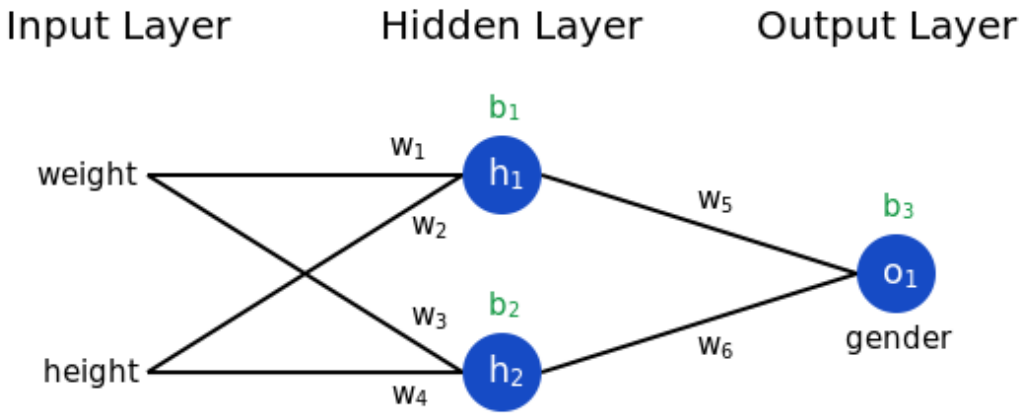

Figure 2.8: Visualisation of basic Artificial Neural Network, Section 2 of [18]

The value of each node in the neural network example above can be calculated as follows, where tanh is the selected activation function. A tanh activation is a suitable in this instance because it is a classification problem, a selected individual can either be the gender male or female. The input Height and Weights are defined by x_1 and x_2

respectively. The output of the "hidden" neurons is denoted as h_1 and h_2 whereas the output of the "output" neuron is denoted as o_1.

$$h_1 = tanh(x_1 \cdot w_1 + x_2 \cdot w_2 + b_1)$$

$$h_2 = tanh(x_1 \cdot w_3 + x_2 \cdot w_4 + b_1)$$

$$o_1 = tanh(h_1 \cdot w_5 + h_2 \cdot w_6 + b_3)$$

"Training" is a crucial part in using NN. In order for a Neural Network to predict an output for the input data that is fed to it, it must first "learn" what it is expected to predict. Through "Training", the neurons of the network adjusts their weights and biases, in order to recognise patterns and predict results. Supervised learning is a one such method used to train a network. In supervised learning, the NN is provided a set of inputs and their corresponding output. The NN then calculates its own output using the input and compares this generated output with the "known" output provided to it. The weights and biases are then adjusted through the use of derivative-based optimisation methods as to reduce the error between the actual data and the NN's output. This it an iterative process, repeated until the NN achieves a satisfactory level of accuracy. In general, to "train" a neural network a large amount of data is required, which may be considered a limitation of this type of machine learning.

2.2.2 Long-Short Term Memory (LSTM)

A LSTM is considered to be a more evolved version of a RNN. During Back propagation RNN's suffer from the vanishing gradient problem. The gradient value problem is where the gradient shrinks as it back propagates through time. If a gradient value becomes extremely small, it doesn't contribute to much learning, see example equation below.

$$NewWeight = Weight - LearningRate * Gradient$$

$$2.0999 = 2.1 - 0.001$$

So in RNN layers that get a small gradient update are unlikely to 'learn'. This occurs usually in the earlier layers.As these layers don't learn, an RNN can forget what it sees in longer sequences which is why an RNN is said to have a short term memory.

LSTMs were created to address this issue of short-term memory. They have internal mechanisms called gates that can regulate the flow of information. The gates can

'learn' which data in a sequence is important to retain or which can be forgotten. By doing this it learns to use relevant information to make predictions.

LSTM has the same control flow as a recurrent neural network. It processes data sequentially, passing on information as it propagates forward. The difference between both LSTMs and RNNs being the operations within the LSTM cells. These operations are used to allow LSTM to forget or remember information.

The core concepts of LSTM are the cell states and its various gates [19]. The cell state acts as a transport highway that transfers relevant information all the way down to the sequence chain. It can be described as the memory of the network. This is because the cells they can carry information throughout the processing sequence . In theory, even information from earlier time steps could be carried all the way to the last time step thus reducing the effects of short-term memory. As it goes on its journey, information gets added or removed to the cell state via gates. These gates are neural networks that decide which information is allowed to pass to the cell state. The gates learn what information is relevant to keep and what information is t be forgotten during training.

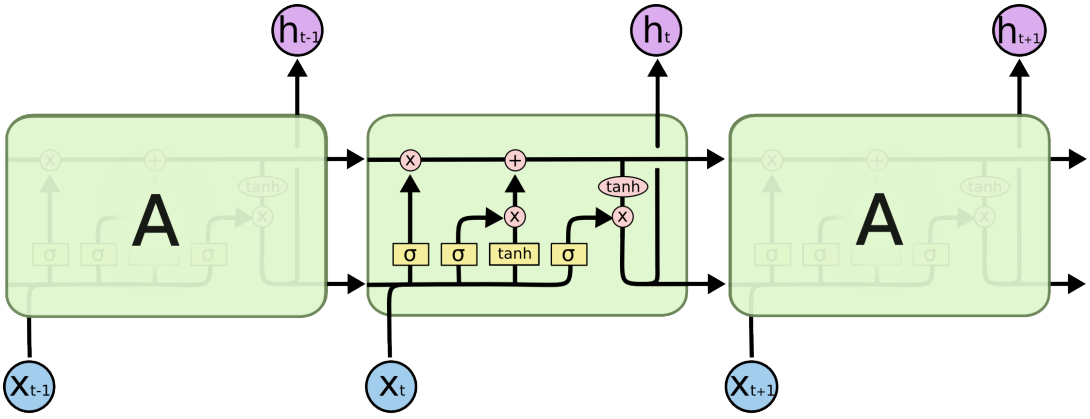

Figure 2.9: The repeating module in an LSTM contains four interacting layers [19]

Gates contain sigmoid activation functions. Sigmoid activation is similar to the tanh activation function. Instead of shrinking values between -1 and 1, the sigmoid function shrinks values between zero and one, see figure 2.7. This is helpful to update or forget data. A number multiplied by 0 is 0, which causes the value to disappear or be forgotten. A number multiplied by 1 is the same value therefore that value stays the same (or is retained). This is how the network can learn what data should be forgotten and what data should be remembered.

There is three different gates that regulate information flow in an LSTM cell:

- **A forget Gate** - This gate decides what information should be discarded or retained. Information from the previous hidden state and the current input is

passed through a sigmoid function. Values come out between zero and one. The closer a value is to zero tells the network to forget it and a value closer to one means to keep. The output is known as the forget vector.

- **An Input Gate** - This gate updates the cell state. The previous hidden state and the current input are passed through a sigmoid function. This decides which values will be updated. The values are transformed to values between zero (meaning not important) and one (meaning important). As well as being passed through a sigmoid function, the previous hidden state and the current input are passed through and tanh function. The tanh function scales all the values down between negative one and one. The purpose of this is to regulate the network and avoid exploding gradients. The outputs of both the tanh and sigmoid functions are then multiplied. The sigmoid output decides what information is important to keep from the tanh output.

  The information required to calculate the cell state has now been complied. The previous cell state is multiplied by the forget vector. The purpose of this is to investigate whether there are values in the previous cell state that can be dropped on account of the information from the current input (values near zero). Then, the output from the input gate and updated cell state are added together through pointwise addition. This returns the New Cell state with updated values.

- **An Output Gate** - This decides what the next hidden state should be. The hidden state contains information from the previous inputs. The hidden state is used for prediction.

  The previous hidden state and the current input are passed through a sigmoid function. The newly updated cell state is passed through a tanh function. the output of both are multiplied deciding what information will be carried over in the new hidden state.

Once generated the new cell state and the hidden state is carried over to the next timestep.

2.2.3 Computational Framework

While training an ANN, a large amount of calculations are complete. This requires a large amount of computational processing power. Machine learning can be carried out on the central processing unit (CPU) on a computer however depending on the size of the input information and the complexity of the model this may not be feasible. A Graphics Processing unit (GPU) is considered a more suited hardware to deal with computations required for machine learning. GPUs have traditionally been used for

rendering visuals in video games and by nature are capable of handling large volumes of repetitive computations.

In recent years, the focus has become to make GPUs more optimised for machine learning applications. NVIDIA along with a number of similar companies have developed GPU hardware that is purpose-built for machine learning and artificial intelligence (AI) applications. An example of this is the Tesla GPU series by NVIDIA, which boasts a significantly reduced training time for Neural Networks [20] in comparison to a standard CPU, see figure 2.10.

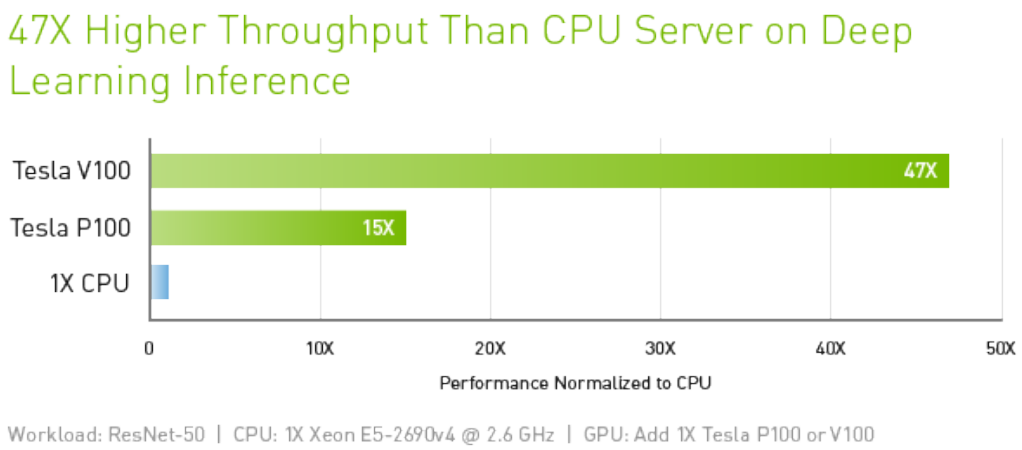

Figure 2.10: Comparison of GPU and CPU-based Deep Learning, AI Inference in [21]

Neural Networks tend to be trained and analysed through the use of bespoke software packages on account of repetitive training process and complexity of the computation involved. These software packages include the likes of Pytorch in Python [22] and NeuralNet in R [23]. The Tensorfow python package, with a keras frontend [24] was used for this paper. Tensorflow is a free and open-source software platform by Google, specifically for machine learning and artificial intelligence. It is a library that supports a range of optimizers, layer types as well as metrics that measure model performance.

2.3 Volcanoes

A Volcanic eruption is caused by magma escaping through vents in the Earth's Crust. During an Volcanic eruption the hot magma in the magma chamber is less dense than the surrounding rock and as a result it rises. As it rises, gas bubbles form increasing pressure. The magma flows upwards through the crust (taking the path of least resistance). When the magma hits the surface, the gases cause it to erupt explosively.

White Island (Whakarri) erupted just at approximately 14:11 New Zealand Daylight

Time (NZDT) on the 9th of December 2019. It was a phreatic eruption (a release of volcanic gases and steam) that resulted in a explosion which launched rock and ash into the air. There was 47 people present on the island during the eruption. Twenty two and twenty five suffered injuries, with the majority requiring intensive care for severe burns.

However, this is not an isolated case and volcanic eruptions happen regularly. There exists approximately 1,350 volcanoes worldwide, although there are countless others found across the sea floor [25].

The 2019 event at White Island will be the focus of this study for this paper going forward. Before proceeding to look at the data surrounding this incident, a look back this volcanoes history will provide us with some insight into the volcanoes dynamics.

2.3.1 History of Volcanic eruptions at Whakaari/White Island

Whakaari (White Island) is the exposed summit of an active stratovolcano located approximately 50 km offshore Whakatane, New Zealand [26]. It has been and remains to be a highly reactive hydro-thermal system for the past 1000 years.

- **9th December 2019:** At approximately 14.11 NZDT, a phreatic eruption occurred. A phreatic eruption is the release of volcanic gases and steam. The release resulted in a explosion which launched rock and ash into the air. The eruption was short lived only lasting between 1 and 2 minutes.
- **27 April 2016:** A phreatic eruption occurred at 09:36 Coordinated Universal Time (UTC). The erruption triggered and eruption sequence which contained a number of eruptive pulses. The eruptive pulses occured over the first 30 minutes. A continuing tremor signal lasted 2 hours after the 30 minute erupive pulsing sequence.
- **5 August 2012:** An eruption occured following a rapid water level rise within the crator. Minor ash emission continued until the 17th of August. No magma surfaced during the duration of the eruption period.
- **1975 to July 2000:** The volcano was in a continuous state of eruption for a 25 year period, the last significant eruption occurring in July 2000. This is noted as the longest and largest historic eruption episode in the volcano's history. The volcano was subject strombolian activity in the late 70's trough to the mid 80's. A strombolian eruption is the release of basaltic magma and volcanic gases through moderately explosive eruptions.
- Older recorded eruptions, resulting in the formation of craters, include

- July 1971
- 1968
- 13 Novemeber 1966
- 1962
- 1947
- April 1933

Literature Review

To date, a large amount of research has focused on the topic of critical transition in the natural world. Such studies include critical transitions in climate (so called "climate tipping events"). An example of such studies would be the transition between an 'on' and 'off' state of the Gulf Stream [27]. The fundamental bifurcation that underlies this transition is the same as that of a volcano that transitions from a non-erupting to an erupting state.

This section of the report will exhibit a number of relevant papers, that highlight the developments and approaches applied to the study of volcanic eruptions as dynamical systems. It will also give an overview of various application of Artificial Neural Networks (ANN) to identifying volcanic eruptions.

3.1 A volcanic Eruption as a Dynamical System

The basic concept of a dynamical system stems from Newton's 3 laws of motion [28]. As opposed to describing things simply in terms of their static properties, a dynamical system refers to the pattern observed in how the state of things change over time. The relationship of a dynamical system can be described by a differential equation, difference equation.

Yu.B. Slezin et al [29] described an *'eruption as an event, separated from other similar events by a rest (or dormant) interval'*. Slezin argues that a high magma ascent velocity inhibits the release of volatile gases upward either through the melt or through porous rock. This in turn creates the conditions conducive to an explosive eruption. He noted the strong relationship between the velocity of the rising magma and the length of the conduit (or magma chamber) itself. He identified the magma chamber depth as the main splitting parameter. This means that conduit length controls the existence of unstable magma flow with sharp regime changes. Looking at figure 3.1, it is clear that if the magma chamber depth is less that the critical magma chamber depth, $H < H_c r$,

a catastrophic rise of eruption intensity is possible. This picture gives an insight into why some volcanoes are more explosive than others.

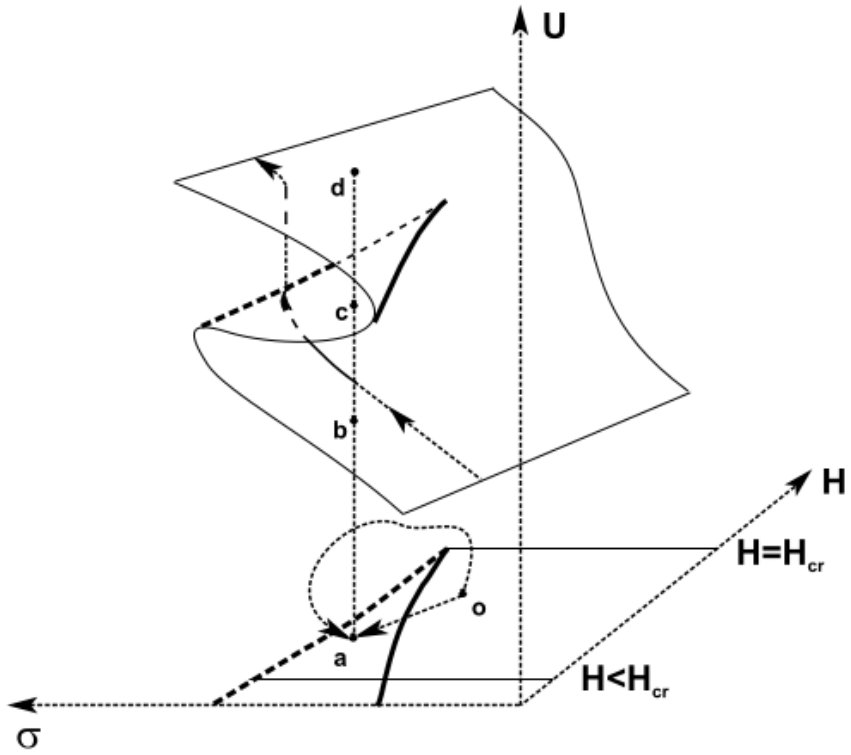

Figure 3.1: Phase Diagram: Magma accent velocity ,U, as a function of the governing parameters, conduit length, H, and conduit conductivity, σ [29]

where:

- $\boldsymbol{U}$ - Magma accent velocity
- $\boldsymbol{H}$ - Conduit Length/ Magma chamber depth
- σ - Conduit Conductivity
- H_{cr}- Critical Conduit Length

He identified conduit conductivity as a normal parameter, meaning that a change in conduit conductivity changes the state of the system resulting in an increase or decrease of the magma discharge rate. Conduit conductivity is defined by the formula:

$$\sigma = b^2/n \tag{3.1}$$

- b is the characteristic cross-dimension of the conduit,
- n is is magma viscosity

Reviewing the picture 3.1, when the state of the system has a conduit length less than

the critical conduit length the system experiences saddle bifurcation. The equilibrium surface has three branches (limbs). Any vertical line that passes through these three branches will have three points, that reflect three different regimes, that corresponded to one value of conduit conductivity. As a result, we can say that this systems has three possible states at a given set of parameters. Between these parameters the system has bistability, meaning that the system has two stable equilbrium state. The upper and lower branches describe stable regimes, whereas the middle branch describes an unstable regime.

When $H < H_{cr}$, the trajectory due to the conductivity increase is shown by the line with the arrows on the equilibrium surface. If the point describing the state of the system is on the lower branch of the graph, and the conductivity of the conduit gradually increases, at the saddle node bifurcation point (reversal of the curve, see section 2.1) the conditions must jump to the upper branch. This results in an large magnitude sharp increase in the magma velocity, and therefore the mass discharge rate. This change is irreversible, as if the thermal conductivity reduces the conditions do not return at the lower branch at the same point. Instead the state of the system continues along the higher branch until it reaches the half saddle node bifurcation point where the conditions return back to the lower branch.

He applied this theory to well documented eruptions including Plosky Tolbachik, Russia and Mount St.Helens, Washington State, USA.

3.2 Artificial Neural Networks

Artificial Neural Networks have been applied to a variety of problems including a number of well-known examples, such as handwritten digit recognition [30], which utilises Deep Neural Networks (DNN), and weather prediction[31], which uses Long Short-Term Memory (LSTM) network. Korstnje wrote in his book *'Advanced Forecasting with Python'* that LSTM Networks are one of the most advanced models existing to forecast time series [32]. This is because they possess an ability remember past information, which is an aid to pattern detection during training [33]. It has only be in recent years that LSTM have found use in number of areas of study, including volcanic and earthquake detection and analysis, in particular in terms of parameter estimation and prediction of significant seismic events. In this section, a number of papers will be presented that demonstrate the ways in which ANN can be applied to aid in the understanding of volcanic eruptions and tremor analysis.

Carniel et al [34], is one of the earliest application of the concept of neural networks to volcanic tremor analysis. His study is extremely interesting given that is was pub-

lished in 1996. His study incorporated the concept that a volcano is a dynamical system that moves from a state of 'rest' to a state of 'eruption'. For this study, volcanic tremors recorded at Stromboli Volcano, off the north coast of Silicy, Italy, were used as the input data. He describes a back-propagated network that interprets that there is two different attractors. The neural network is trained on the dynamics of the persistent tremor, a volcano in a rest (or dormant state), it then tries to detect structural changes in its behaviour. These structural changes are represented by larger ground velocities and are indicative of significant ground tremors that are associated with a volcanic eruption.

Carniel et al [35] followed this with the 2012 review article *'Time Series Analysis: Dynamical Evolution of a Spectral, Deterministic and Stochastic parameter for the characterisation of Volcanic Activity*. In this , a study of the existing literature was complete. Carniel also examined the prospects the future of understanding the physical changes that occur in a volcanic system as well as their possible consequences.

The Paper by Boue et al [36] uses the Failure Forecast Method (FFM) with a Bayesian Approach to attempt predict eruptions in real-time. The paper builds on studies that predict on time series that have been completed previous with the intention of carrying out real-time forecasting with a time series that is partially know. The output of the network is the probability of the forecast time at each observation time before the eruption. The model is tested on precursory accelerations of long-period seismicity prior to vulcanian explosions at Volcán de Colima (Mexico).

Curilem et al [37] present an alternative, but interesting, approach where by instead of dealing with raw seismic signal they examine spectrograms. Using Fourier Transforms, the spectograms are calculated for every collected signal. The input data becomes a collection of spectograms which are 2d images. The images are passed into a deep neural network which classifies the seismic activity through analyzing the spectrogram shapes.

Raw Data Retrieval and Reprocessing

All of the data was accessed and processed and using the packages Obspy in Python. The LSTM models were written in Python using tensorflow 2.9.0 . All code used in this thesis can be found at https://github.com/Falvey94.

In the following sections, the methodology for the both the data acquisition and modelling approaches in question will be outlined and relevant results presented and discussed.

4.1 Data

Seismic data is readily available in the form of time seires data through a number of seismic datasets including GeoNET, IRIS and FDSN. These datasets provide a live geological hazard monitoring systems. The systems are comprised of geophysical instruments (e.g. seismometers) which detect ground movement, in particular significant ground movements that reflect volcanic activity, earthquakes, large landslides and tsunami/hurricanes. They provide free access to an open format precision measurement service that provides timely and high quality data to the public and for research, emergency management and industry use.

The seismic datasets that best captures the activity at White Island are provided by GeoNet and the Institute of Geological and Nuclear Sciences Limited (GNS). On White Island there is two permanent GeoNet stations (WIZ and WSRZ) and four GNS Science seismic staions (WI01, WI02, WI04, and WI13), see figure 4.1. The seismic and acoustic instruments at all of the recording sites mentioned are sampled at 100 Hz [38]. For this study, The data station WIZ was chosen.

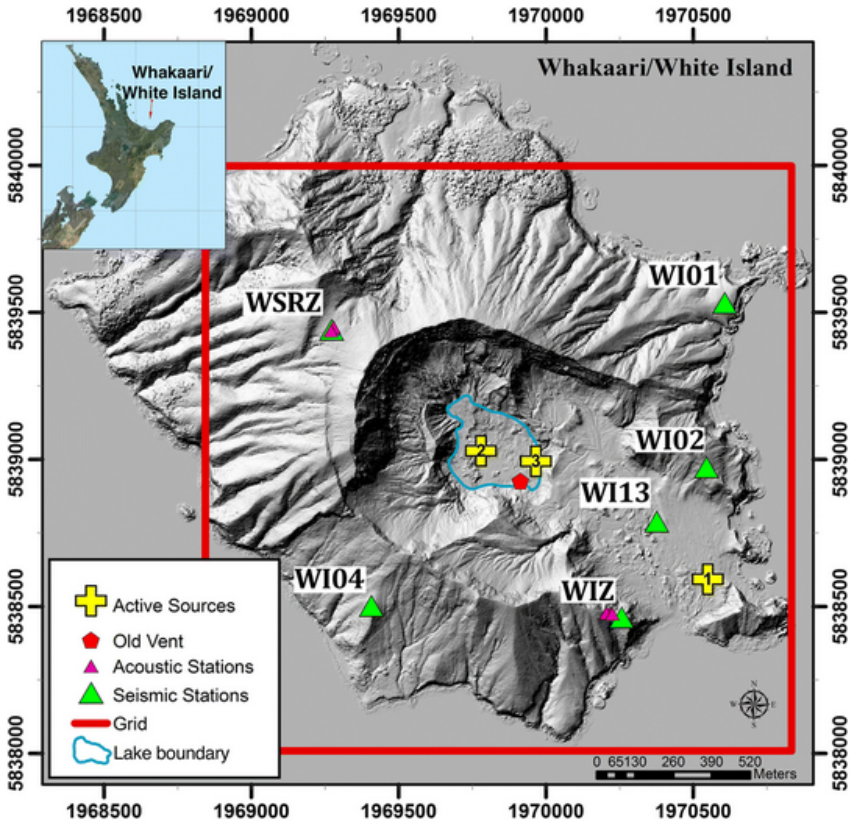

Figure 4.1: A map of Whakaari Island, New Zealand, depicting the locations of seismometers (green triangles), acoustic sensors (magenta triangles), and active source drops (yellow crosses) [38]

Seismometers, CPS networks and many other types of devices of measurement is how data is collected by GeoNet and GNS. The data and images are freely available in an open format through the GeoNet Project. This is so that a range of industries and organisations can benefit from a precision measurement services and crucially, to facilitate research into hazards and assessment of risk.

In Figure 4.2, the raw seismic data has 3 data channels of seismic data. The channel codes follow the SEED conventions, unless they proceed the convention and have not been updated [39]:

- The first letter of the code represents a combination of sampling rate and sensor bandwidth, where H represents High Broad band sampled at or above 80Hz, generally 100 or 200 Hz.
- The second letter represents the sensor type, where H represents Weak motion sensor, e.g. measuring velocity.
- The third letter either represents the sensor orientation or a processing stage.

The three channels are HH1, HH2 and HHZ which correspond to the north-south,

east-west, and vertical components of the seismogram. The data reflects the amount of ground movement by the instrument on each of the these axes.

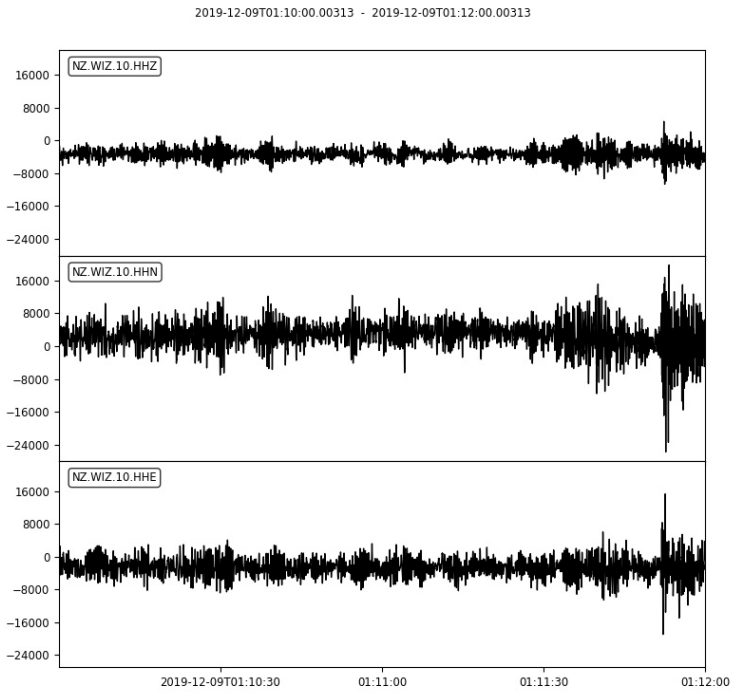

Figure 4.2: The 3 data channels of Raw seismic data: HH1, HH2 and HHZ

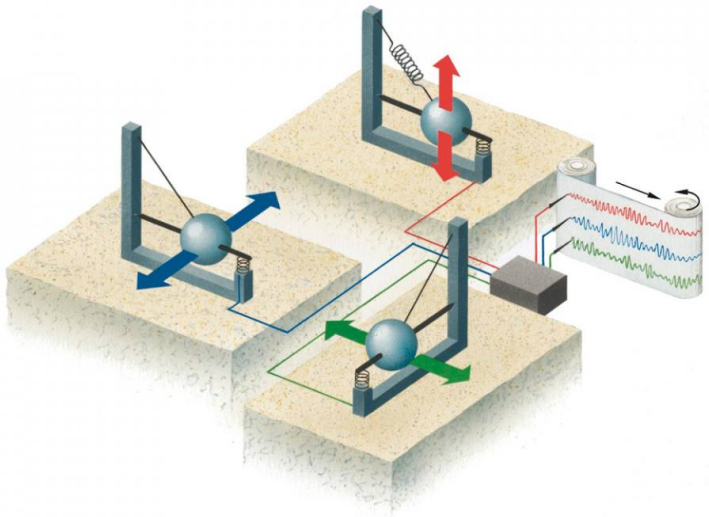

Figure 4.3: A three-component seismometer. Z (red) measures up/down motion; E (green) measures east/west motion; N (blue) measures north/south motion [40].

It was recommended by a seismology contact, to observe and analyses the data according to the HHZ channel for this experiment. This is because the channels HH1 and HH2 are affected by additional external forces, for example the movement of water which has the potential to additional noise to the data.

A interval of data was selected to represent the White Island (whakaari) volcano eruption. As discussed, the event took place on the 9th of December 2019 at 14:11 NZDT (01:11 UTC) and lasted an estimated 2-3 minutes. In order to attain a suitable amount of required 'normal activity' (rest state) data to train the proposed model, an interval of fourteen hours (twelve hours before until two hours after the eruption) was used.

4.1.1 Removing Instrument Response

The reason why it is necessary to remove instrument response is due to the manner of how a seismometer actually records data.

In seismology, there is a wide bandwidth of frequencies and signals that are of interest and are to be recorded. In order to record this entire range of frequencies, their instruments have to be designed to be sensitive to them. The instrument is known as a seismograph/seismometer.

A seismometer operates on the principle of a pendulum. The typical picture of a basic seismometer presents as an instrument that records the movement (motion) of the seismometers frame/housing relative to a heavy inert mass. The mass possess a certain resistance to movement (i.e. inertia) due to its weight is suspended from the frame by a spring that allows movement. However, this physical measure has a limited dynamic range in which frequencies and signals can be recorded. In order to overcome this limitation, electronic feedback loops in the instruments are used to increase the range over which measurements can be recorded. This means that rather than recording the motion of the frame, it is recording the amount of voltage required to keep the mass inside the instrument stable, while the ground is moving. The seismometer is connected to digitiser which converts this analog signal into a digital record of counts. This record of counts can be successfully passed and read by a computer.When the data has been downloaded from a FDSN (Federation of Digital Seismograph Networks) web service (e.g. IRIS, GeoNET), what is actually acquired is a time series of counts from the digitiser that is attached to the selected instrument.

In order to convert to displacement, acceleration or, in the case of this procedure, ground velocity, it is important to account for the digitiser and the response of the seismometer itself to ground motion which is frequency dependant. Theoretically, this is quite involved and considered quite complicated. The mathematics and derivations of how to account for the response are discussed further in *'Instrumentation in Earthquake*

Seismology' [41]. Whereas in practice, obspy makes correcting for instrument response quite straight forward. The main point to highlight being that when downloading the data the user must make sure to also download the response.

When the user downloads the data, the instrument response must be attached to it. Similar to how the data is downloaded from the selected "client" (GeoNet), the obspy command `st = client.get_waveform()` is used. In order to attach the instrument response, an additional response is added inside the parentheses `st = client.get_waveform(attach_response = True)`. This is to inform the command to attach the response to the data that is being downloaded. `attach_response = True` is a boolean operation, meaning that if 'true', the instrument response is to be attached to the data downloaded.

When the selected data with the instrument response attached is obtained, response removal can proceed using `st.remove_response()`. The user can select the units that they want the ground velocity to be represented in using the output response option `output = 'VEL'` inside the parentheses. The user can select from an number of outputs including velocity (*'VEL'*), displacement (*'DISP'*) and acceleration (*'ACC'*). When the user runs this they an see that the units on the y access have now changed from a count to measure of ground velocity (m/s), see figure below.

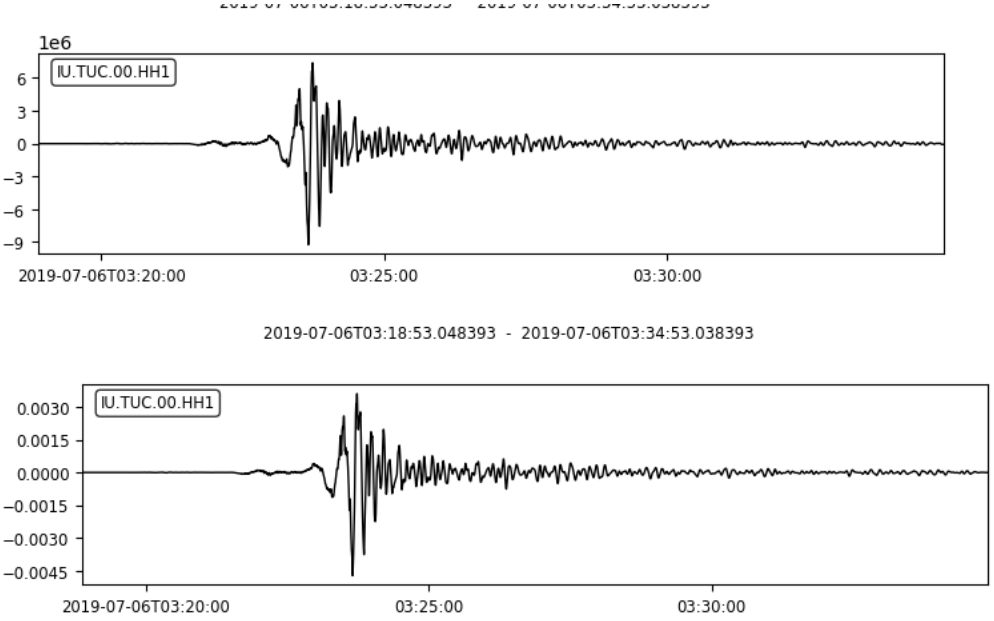

Figure 4.4: Response Removal: The first image shows a selected stream of data with no response removal carried out. The second image shows selected stream of data with response removal. The waveform looks exactly the same in this case but notice that the units on the y axis have changed. Instead of a digital counts, the amplitude is measuring the velocity of the ground motion (meters per second).

However it is important to note that this method is consider an oversimplification of the response removal process. This is because the data is inherently noisy. If the data is divided at low and high frequencies by the response, there exists a risk of amplifying the noise considerably. This is the reason why bandpass filtering is commonly included as a part of the response removal process. You can included bandpass filtering by adding additional options inside the parentheses when removing the response. This is particularly important if an instrument that doesn't have a flat response at a range of frequencies is used. Using filters, is considered to be a specialised skill and it can be quite difficult to identify and select what is the correct filter that is appropriate for the frequencies that the user is hoping to look at. For the sake of simplicity, it was decided that this stage was omitted as to keep the data treatment process as straightforward as possible.

The response removal process can be visualised using the plot response option `plot = True` inside the parentheses. This is a boolean operation, meaning that if true the plot of the removal of instrument response should be plotted.

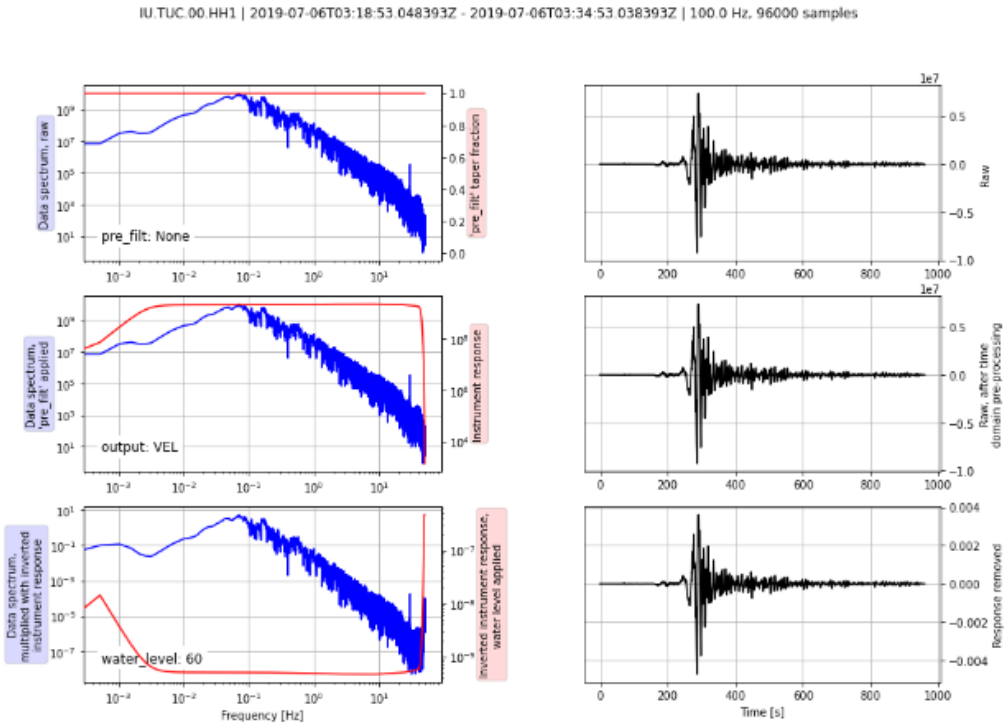

Figure 4.5: Response Removal: The plots on the left are showing the instrument response (shown in red) and the data (shown in blue). The plots on the right of the figure are displaying what happens after each step in the process of removing the response

The plots on the left are showing the instrument response (shown in red) and the data (shown in blue). If the user had decided to do a band pass filter the data would have

been cut at certain frequencies here. The plots on the right of the figure are displaying what happens after each step in the process of removing the response. In this example, as no filtering was carried out, they all look the same as other than the units on the y axis changing from counts into ground velocity (m/s).

4.1.2 Detrending the Data

Detrending the Data involves the removal of any underlying trend from the data. This is done so that there is no distortion existing within the data so that subtrends in the data, that may be seasonal or cyclical, can be more easily identified. The Figure 4.6, displays and an example of a random dataset before and after undergoing `scipy.signal.detrend()`.

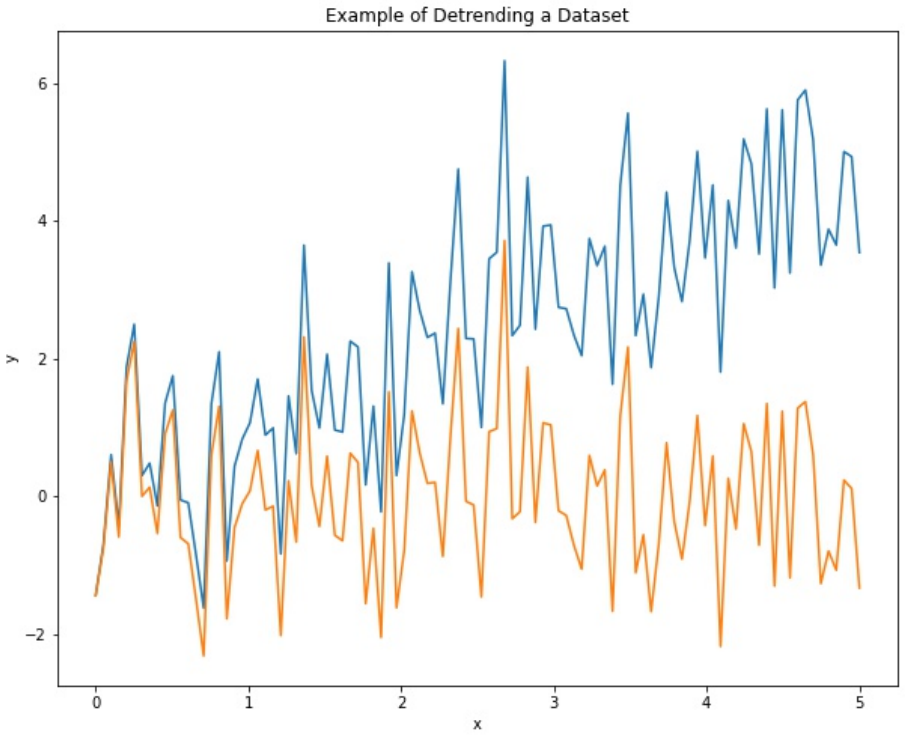

Figure 4.6: Example of Detrending a Dataset

There are two main methods to detrend time series data:

- Detrending by differencing: Creating a new dataset where each data observation is the difference between itself and the previous observation.
- Detrending by model fitting: Fitting a regression model to the dataset and calculating the difference between the observed values and the predicted values from

the model.

For this dataset, the `.detrend` command in the obspy python package is used.

4.1.3 Applying a Filter

The purpose of applying a seismic filter is so that the user can observe a specific frequency range of data. This is be helpful in identifying large earth movements that would be indicative of earthquakes or volcanic eruptions. There are many different filter methods available including:

- bandpass
- bandstop
- lowpass
- highpass

In the case of this study we will only use a highpass filter. A highpass filter is a filter that passes signals from a selected sample with a frequency higher than a selected frequency cut-off. The signals with a frequency lower than the selected cut-off will be attenuated. To filter the signals, the command `st.filter("highpass", freq=3.0)` is used. Additional responses are added inside the parentheses in order to specify what type of filter method is to be applied and also to specify the condition on the filter method. It was decided to choose a highpass filter for a frequency of $3Hz$. This means that any signals with a frequency of less than $3Hz$ is to be ignored in this study. This is to eliminate time periods of little to no activity.

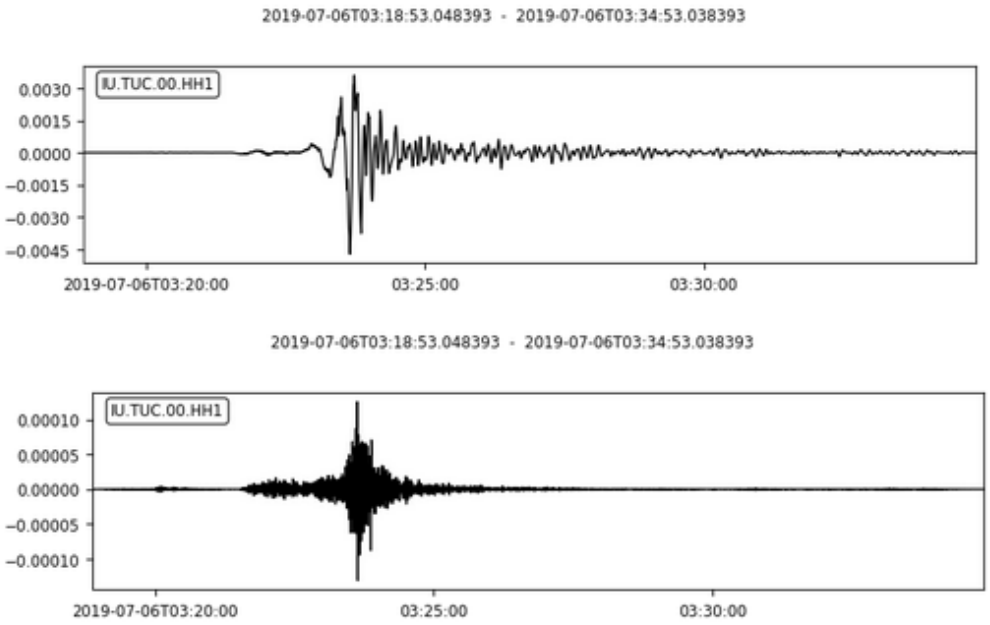

Figure 4.7: Filtering out Lower Frequency Signals

It's visible from the example above that the original data (Top image) has lower frequency information present. When the filter is applied, these low frequency signals are removed so that only frequencies higher than 3 Hz are visible.

4.1.3.1 Resampling

To resample a dataset of time series observations means to change the frequency. There are two main types of resampling. These are:

- Downsampling where the frequency of the data is decreased. An example would be changing your readings from one reading per hour to one reading per day.
- Upsampling where the frequency of the data is decreased. An example would be changing your readings from one reading per hour to one reading per minute.

Whether a time series observation is downsampled or upsampled there is a requirement to invent data from the original data. In the case of upsampling, data points must be created inbetween the existing data point through interpolation. If a time series is downsampled, it is important that the aggregated values accurately reflect the existing data. Therefore care is needed in selecting the summary statistic.

The main reason for resampling time series observations for this paper is to replicate the experiment by Carniel et al, see section 3.2. The seismic tremor data that they used in their experiment was sampled at 80Hz. As the incoming data from GeoNet sampled at a rate of 100 Hz, the data is to be resampled.

To resample the data, the command `.resample(80, no_filter=True)` is used. Additional responses are added inside the parentheses in order deactivate the automatic filtering.

4.1.4 Writing the data to a file

Once the data has been tailored through response removal, detrending, filtering and resmapling it now has to be saved in the appropriate file format so that a model can be trained, validated and tested. In this study, it was decided to save the data into its own comma-separated value file (.csv). The reason as to why it was decided to save the data in this format was due to a .csv file having the attribute of a file that is smaller in size but also given that it can be read by the open source python package pandas. Pandas is an useful package when it comes to data science/data analysis and machine learning tasks. Another reason why the data is saved to a file, is so that easy access to the file is possible and the csv file can be loaded into different scripts.

The first step that is required is separating both the data and the corresponding times. For any type of graph or plot, an independent variable (times)and a dependant variable (ground motion in terms of velocity) is required. It is quite straightforward to parse the data from the tracing stream objects. Using the command `st[0].data`, we are extracting the data from the first stream of traces. These values are then stored in a choosen variable. As the variable contains the ground motion records in velocity (the data values) it makes sense to name the variable 'data'. Similarly, in order to extract the corresponding times from the first streams of traces we use the command `st[0].times`. This data is be assigned to the varaible 'times'.

This leaves us with all the required data in 2 separate variables labelled 'data' and 'times'. These variables are both in the form of an array. In order to successfully create a dataframe containing these 2 variables, we must convert the variables from arrays to lists. THese array can be converted to a list through using the `.tolist()` pandas function.

The data and the corresponding times variables are in the correct form so that the can be successfully combined in a dataframe. Using the pandas fuction `pd.DataFrame())` to create a dataframe, the 'Data' and 'Times' lists can be zipped together by using the zip command inside the parentheses, `list(zip(data_list,times_list))`. The dataframe can be customized as it is being created. For example, labels for each column in the data frame can be provided which in this case represents the 'data' and 'times' columns. This id done by adding a supplementary response option inside the parentheses, `columns= ['Data', 'Times'])`.

The data has now been successfully complied in a dataframe. The dataframe must

now be converted into a .csv file so that it can be accessed from other scripts. To convert a data frame to to a .csv file the following line of code is to be used `pandas_dataframe.to_csv('choosen_name.csv', index=False)/`. A name and location of storage of the output .csv file is to be provided to the function. The index response is set to 'False' so that an index column is not created. It is intended to use the 'Times' Column as an index column. Once the `to_csv` function has been run, the name file is added to the location provided. The file can then be accessed from other scripts using the pandas command `pd.read_csv('choosen_name.csv')`. This is how the data was accessed throughout the project period.

4.2 Volcanic Eruption Detection

The objective of this study is to to train a recurrent neural network using seismic signals resulting from White Island 2019 volcanic eruption.

4.2.1 Accessing the Data and further data processing

Firstly the data that was previously prepared must be accessed. The file is in the comma-separated values (.csv) format. This file can be accessed using the command `pd.read_csv` from the pandas package. It is possible to view what the data looks like at this point using `print(df)` which will output the dataframe to the screen.

	Data	Times
0	5.776560e-04	0.00
1	2.933398e-04	0.01
2	8.107005e-05	0.02
3	-7.032297e-05	0.03
4	-1.710582e-04	0.04
...	...	...
4319996	5.186805e-10	43199.96
4319997	-7.274885e-10	43199.97
4319998	4.526382e-10	43199.98
4319999	-5.501588e-10	43199.99
4320000	4.356862e-10	43200.00

4320001 rows × 2 columns

Figure 4.8: The dataframe as read in from the pandas command $pd.read_csv$|

It is clear from the jupyter notebook extract presented above, that there are 3 columns. There is a column that holds ground motion records in velocity, the data values, their corresponding time values and there is also an index column present. The index column labels each row of data and time values starting from an index of 0. Depending on the data, it might be more convenient to index the samples by one of its attributes. In this scenario, the data will be indexed by the 'times'.

In order to set the 'times' column as the index for the dataframe going forward, the index is to be specifyed and the original 'times' column is to be dropped from the dataframe. This is to avoid having two columns of repeated information. To set the 'Times' column to become the index column, the index of the column must be specified as Times column of the dataframe. Once 'times' has been set as the index of the dataframe, the original 'Times' column is to be removed. This can be done by popping a column from a dataframe using the command `df.pop('Times')`.

	Data
Times	
0.0	5.776560e-04
10.0	1.770613e-08
20.0	-9.938918e-09
30.0	4.024192e-08
40.0	-1.305654e-10
...	...
43160.0	2.014696e-08
43170.0	-1.156946e-08
43180.0	-1.441832e-08
43190.0	-6.000574e-09
43200.0	4.356862e-10

4321 rows × 1 columns

Figure 4.9: The dataframe where the 'Times' column has now been set as the Index column and the original 'Times' column has been dropped

Now that the data is in the correct form, it is ready to be observed by the model.

4.3 Modelling Approach

Initially, the aim of this project was to apply the approach as discussed by Roberto Carniel et al in their 1995 paper 'Neural networks and dynamical system techniques for volcanic tremor analysis' [34]. However, in the initial stages of this study it became clear that this approach was not feasible within the timeframe of this project. Due to this, it was decided to investigate whether a model could be assembled that recognises the dynamical system of a volcano, recognising that it has two attactors, these being the a rest steady state and and active state.

4.3.1 Splitting the Input

The input is required to be divided into sets so that the performance of the neural network model can be evaluated. For this study, the input is divided into 3 individual sets: the train, validation and test sets. They are defined by Brian Ripley [42] as

follows:

- **Training set:** A set of examples used for learning, that is to fit the parameters of the classifier.
- **Validation set:** A set of examples used to tune the parameters of a classifier, for example to choose the number of hidden units in a neural network.
- **Test set:** A set of examples used only to assess the performance of a fully-specified classifier.

As shown in figure 4.10, the input will be split so that 70% of the data will be utilised for the training, 10% for the validation and 20% for the test. It's also important to note that the data is not randomly shuffled. The reason for this being that it ensures that chopping the data into windows of consecutive samples is still possible. It also ensures that the validation/test results are more realistic, being evaluated on the data collected after the model was trained.

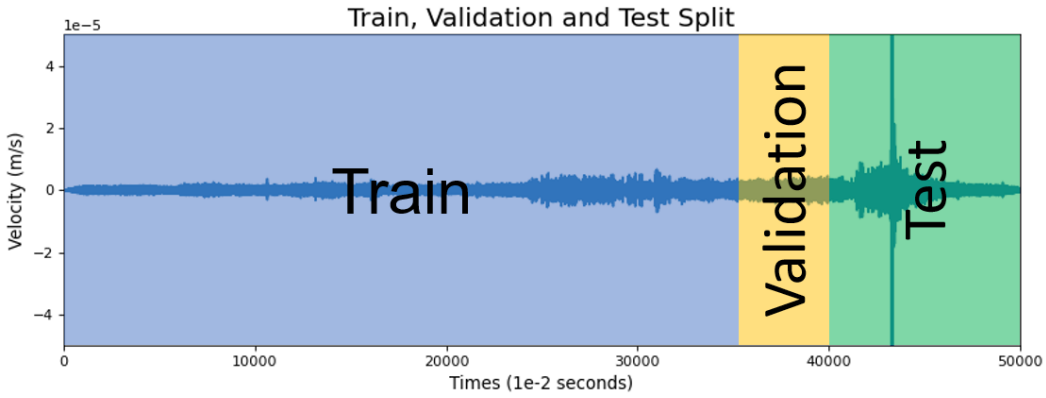

Figure 4.10: The train, validation and test split for the seismic waveform data

4.3.2 Normalising the Input

Normalization is the scaling of features. It is considered an important pre-process before training a neural network. To normalise each feature, one must subtract the mean and divide by the standard deviation.

$$n_i = \frac{x - \bar{x}}{\sigma} \tag{4.1}$$

Where:

- n_i = Normalised data
- x = Original data

- $\bar{x}$ = Mean of the original data
- σ = Standard deviation of the original data

To ensure that the models do not have access to values in either the validation and the test sets, the mean and standard deviation are to be calculated using the training data only. It should be noted that a simple average is considered a simplification. It's arguable that the model shouldn't have access to future values within the training data when the model is training, and that normalisation should be instead carried out using moving averages. However, in the interest of simplicity, it was decided to proceed using a simple average.

This is a typical z-score normalisation. The data looks exactly the same as before. This is because normalization does not change the shape of the data. The purpose of z-score normalisation is to make sure that the mean is now centered around zero and the standard deviation of the data is 1.

4.4 Data Windowing

For forecasting, the model needs to observe a select amount of the data that has come before. The proposed models will be expected to make a set of predictions based on a preceding window of consecutive samples from the data. For example, the consecutive values from the value at t to t+5 is used to predict the value at time t+6, where t is represents a single time step. The design of the sliding window design will affect the architecture of the neural network.

A class was created in order to perform a data windowing operation on the input data. The class was named `df_to_X_y`, figure 4.11 and the main features included are:

- The number of time steps (width) of the input window, default size of 5 time steps.
- The selected time series data - in this case the seismic ground movement data

```
#Converting a forecasting learning problem to a supervised learning problem

def df_to_X_y(df, window_size=5): #Where the X is going to be your input values and y is your output value
  df_as_np = df.to_numpy()        #convert the df to numpy
  X = []                          #create list X
  y = []                          #create list y
  for i in range(len(df_as_np)-window_size):
    row = [[a] for a in df_as_np[i:i+window_size]] #Take the values from i to i+window(not including itself)
    X.append(row)
    label = df_as_np[i+window_size] #true actually value for that input row
    y.append(label)
  return np.array(X), np.array(y) #convert lists into arrays using np.array
```

Figure 4.11: The data windowing class

The class split windows of features into (features, labels) pairs. Inside the class, the

input time series data is converted from a dataframe to an array. Two empty lists are assigned to the variables X and y. A `for` loop is used to cycle trough the all values in the array of data. This is excluding the last number of values that equate to the length of the window itself because the index of the loop would become out-of-bounds. Inside the loop a new variable called `row` is used to grab our first row of input values (`i` to `i+Window_Length`) from the data array and is then appended to the list `X`. Similarly, for the labels (the next value in the array `i+Window_Length`) is grabbed by the new variable `label` and is then appended to the list `y`.

For this study, a window size of 5 was chosen in order to make a single-output prediction on the 6^{th} time-step. The `df_to_X_y` class was applied to each of the Training, Validation and Test sets individually.

The output of this class is two arrays X and y where

- X - stores the the input window of a select width
- y - stores the corresponding next value in the sequence, ie. the corresponding label.

4.5 Assembling the Model

As the data is now presented in the correct format and split into the training, validation and test sets, the model can now be compiled. The model was created using Tensorflow with Keras, as per section 2.2.3. The model was assembled in the form of a sequential model.

The following neural network architecture was used to train on the waveform data. This Recurring Neural Network (RNN) consisted of:

- An Input Layer - The input is specified. It takes in the form of a (5,1) vector, which is the previous 5 five values preceding the value we are aiming to make a prediction on.
- First Hidden Layer (LSTM) - 64 units that represent the dimensionality of outer space.
- Second Hidden Layers - Convert to 8 tanh activation. Each LSTM layer applies the tanh function. The tanh function regulates the values flowing through the network. The tanh function squishes values to always be between negative 1 and 1. Thus regulating the neural networks output.
- An Output Layer with linear activation - The output value is to be linear as the model is trying to predict some ground velocity which is going to be positive or

negative value.

We can view a summary of the model through `model.summary()`. This displays the number of trainable parameters in each layer.

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 lstm_4 (LSTM)                (None, 64)                16896

 dense_8 (Dense)              (None, 8)                 520

 dense_9 (Dense)              (None, 1)                 9

=================================================================
Total params: 17,425
Trainable params: 17,425
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.12: Model Summary

In our first dense layer, eight nodes are specified and therefore the output shape is `(None, 8)` the number of parameters are the weights connecting the 64 outputs of the previous layers with the 8 inputs of the current layer. Each of 8 nodes has one bias parameter. The final number of parameters in the layer can be calculated as follows:

$$(64 + 1) * 8 = 520$$

The final layer is calculated in a similar fashion, one node is specified and therefore its output is `(None,1)`. The number of parameters are the weights connecting the 9 outputs of the previous layer with the 1 input of the current layer. The single node in this current layer has one bias parameter. The final number of parameters can be calculated as follows:

$$(9 + 1) * 1 = 10$$

The `Total params` figure is the addition of all parameters in each layer:

$$16,896 + 520 + 8 = 17,425$$

Once the model is assembled it must be configured. The compiling stage is used to configure the model. To compile the model a number of necessary arguments need to be defined including:

- The Loss function - a method of measuring how well the chosen algorithm models the dataset. It is a measure of the deviation between the model's prediction and the actual observation. Depending on the type of learning task, loss functions can be classifieds by two major categories - regression losses and classification losses. The seismic waveform data is a continuous dataset, therefore a regression loss is required for this model.

  Mean Square Error (MSE) is the loss function used for this model. MSE s measured as the average of squared difference between predictions and actual observations [43]. The benefit of using MSE is that is has nice mathematical properties which makes it easier to calculate gradients.

$$MSE = \frac{1}{N}\sum_{I=1}^{N}(x_i - \hat{x}_i)^2 \tag{4.2}$$

  Where:

  x_i: observed value

  $\hat{x}_i$: predicted value

  N: number of data points

- The optimizer - is an algorithm that modifies the attributes (weights and learning rate) of the neural network so that the losses are reduced.

  The Adam (Adaptive Moment Estimation) optimiser was used in this model as the method converges rapidly. An initial learning rate of 0.0001 was specified at the compile stage so as to prevent the loss from decreasing at a fast rate. The reason for this is to prevent the model from trying to decrease the loss quickly, meaning that the model will not find its local minimum.

- The metrics - are used to monitor and measure the performance of the model. As the model is a regression model, a metric that calculates the distance between the observed and predicted values is required.

  Root Mean Squared Error (RMSE) was used as the metric for this model. The RMSE corresponds to the square root of the average of the squared difference between the target value and the value predicted by the regression model [44].

$$RMSE = \sqrt{\frac{1}{N}\sum_{I=1}^{N}(x_i - \hat{x}_i)^2} \tag{4.3}$$

Where:

x_i: observed value

$\hat{x}_i$: predicted value

N: number of data points

Once complied, the model is fitted to the data. This is point at which the model is trained and validated. The model is fed the input and the corresponding labels. The model is passed both the training and validation datasets. One epoch is one training iteration and the number of epochs indicates the number of times the model runs through the given data (training and validation). The number of epochs chosen to fit this model is 10 so as to allow the learning algorithm to run until the error from the model has been sufficiently minimized [45]. The partially-trained model is saved at the end of each epoch using callbacks. Callbacks will save the best performing model in training.

4.6 Results

In this section, analysis is carried out the on the best performing model i.e. the model with the lowest validation loss from training. This model was saved during training and can be accessed through the command `load_model`.

4.6.1 Training and Validation Results

A visualisation of a brief subsection of the training results given are shown in figure 4.13. The red solid line shows a graph of the actual data. The green dashed line shows a graph of the predictions after the training process. The blue line shows the error signal. It depicts the difference between the actual data and the prediction.

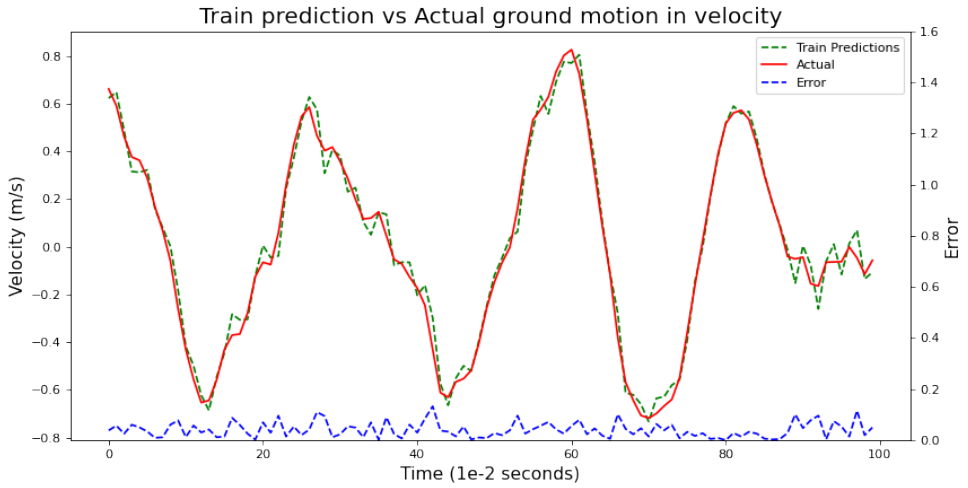

Figure 4.13: Train prediction (green, dashed) vs observed (red, solid) ground motion in velocity. The blue line represents the error signal, the divergence between the observed and prediction

Visually, it shows a high degree of agreement between both the actual seismic data and the simulated data. It's also clear that the error signal is quite low confirming that the distance between the observed and trained is quite small. While there is some minor deviation between the two curves (particularly in the peaks), the LSTM training performance is very good.

Figure 4.14 includes a graph of the RMSE accuracy metric, for both the training and validation datasets. The training RMSE appears to be consistently decreasing. It undergoes a large initial decrease from 0.1334% to 0.1074% RMSE from the first to the second epoch. Between the second epoch and the tenth epoch the RMSE decrease to 0.1051%. This indicates that the majority of the models learning on the training set occurs in the first epoch. The validation RMSE appears to be decreasing overall however it does experience a slight increase after the second epoch. The overall decrease in RMSE is quite small similarly indicating that majority of the models learning on the validation set occurs in the first epoch. Overall, the neural network is able to achieve very low validation errors 0.1446% and a validation loss error of 0.0209 (MSE). This highlights that this approach is able to consistently and accurately predict on seismic tremors in the absence of interesting seismic activity.

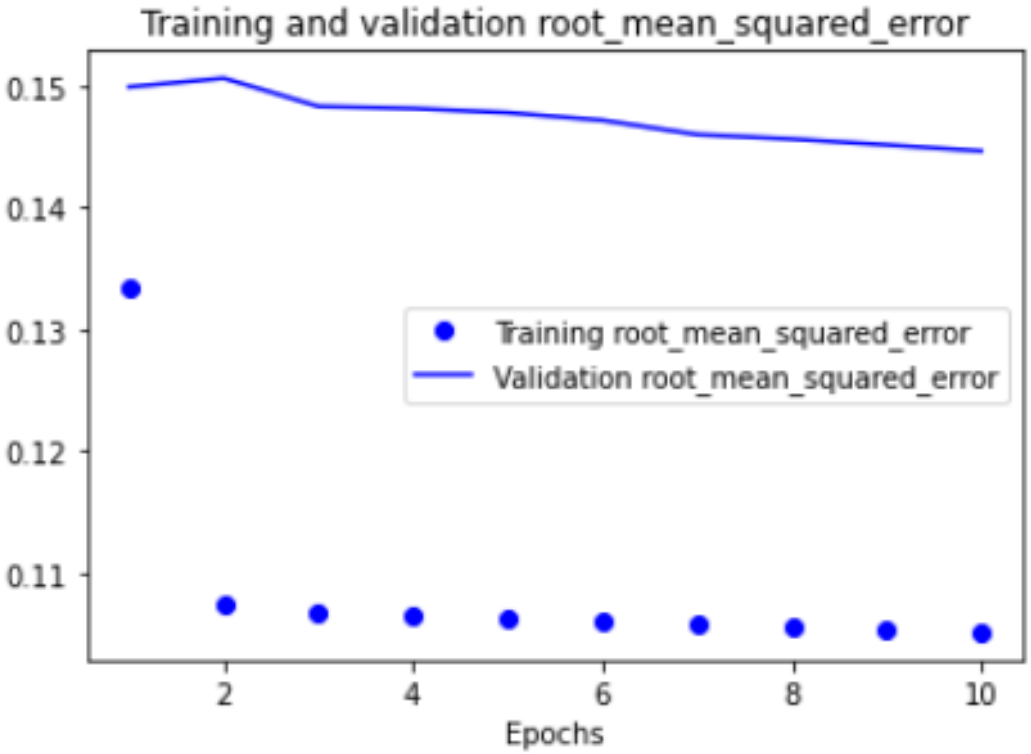

Figure 4.14: LSTM Train/Validation RMSE

4.6.2 Test Results

A visualisation of a subsection of the test results is provided in figure 4.15. Similar to figure 4.13, the actual data is represented by the red solid line, the test predictions are represented by the green dashed and the error signal is represented by the blue line.

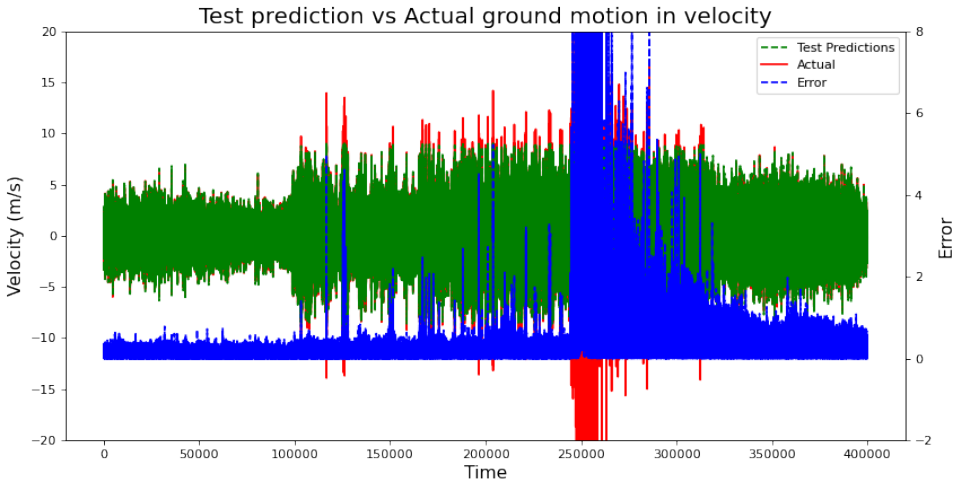

Figure 4.15: Test prediction (green, dashed) vs observed (red, solid) ground motion in velocity. The blue line represents the error signal, the divergence between the observed and prediction

Again visually, it shows a high degree of agreement between both the actual seismic data and the simulated data at both the beginning an end of this sample. However the the model performance is poor at the middle section and the error signal becomes very large indicating that the predicted values and the actual observations have diverged significantly. The poor performance is associated with the period of the volcanic eruption. The LSTM prediction performance is very good in the absence of interesting seismic data however the model fails to predict on significant seismic ground velocity resulting from volcanic eruption.

4.6.3 Data Analysis

Data analysis was performed on the processed seismic signal. An example of a single seismic waveform is shown below in figure 4.16.

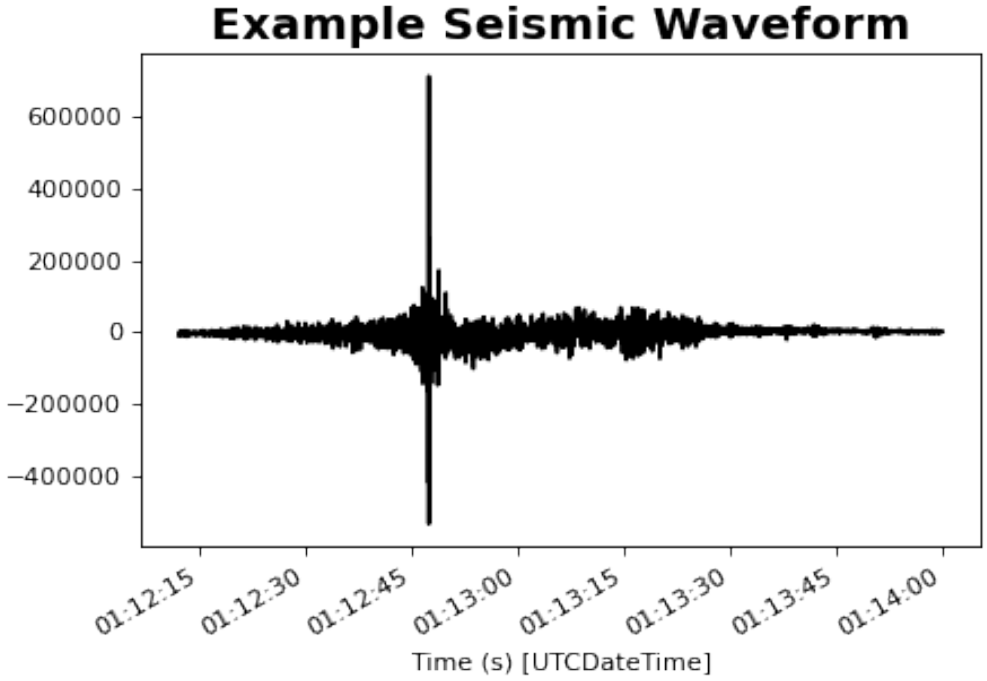

Figure 4.16: n example of a single seismic waveform

Figure 4.17 provides a visualisation of the the two attactors present in the data. For the purposes of visualisation, a short window of the data was selected to represent the 'normal activity' and the activity during an eruption.

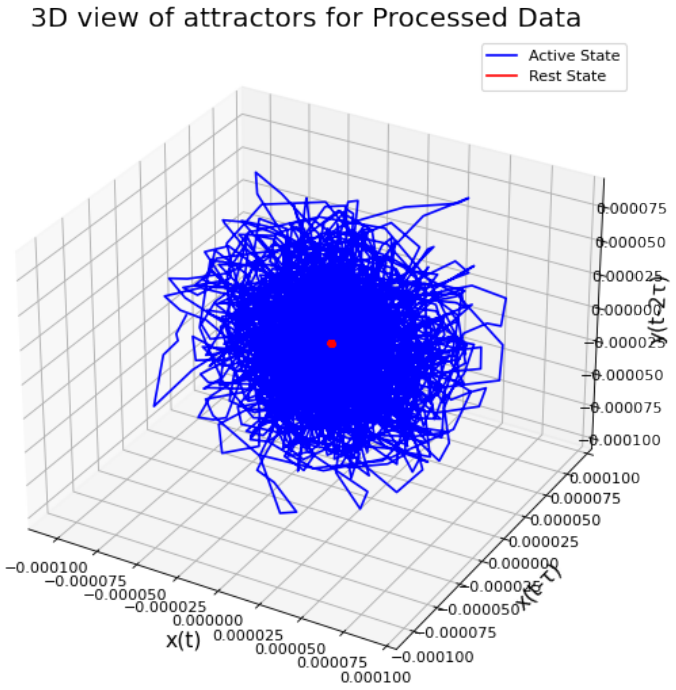

Figure 4.17: 3D view of two attractors present in the dynamical system representing volcanic eruptions

The figure presents a structure that is topologically equivalent to the attractors in the system. It is generated through the means of a three-dimension delayed embedding [46]. This is achieved by plotting the values for some system x at some time t against the values for the same system at time $t - \tau$ against the values for the same system at time $t - 2\tau$ where τ is a small lag time. The value for τ in this scenario was chosen to be 5 time steps, which is 0.0625s $(5 \times 0.0125s)$.

Visually, this figure shows the both attactors, the steady state (or rest/dormant state) and the active state (the eruptive state). The picture seems to depict the resting attractor trajectory (red) surrounded by an erupting chaotic attractor trajectory (blue). The noisy depiction of both at tractors is due to oscillation of the seismic waveform about zero ground motion. Although, this is just an estimate it does gives an insight into what the volcano dynamical system might look like.

In summary, overall performance is very promising and it can be seen that the model interprets that two different attractors exists in this system. The final chapter will provide some concluding remarks, as well as outline potential extensions to the approach and additional analysis that could be carried out in a future work.

Discussion

In this work, a neural network-based approach to observing the existence of two attractors in the dynamics of a volcano was investigated.

As discussed in section 4, the LSTM training performance is very good. From the obtained results, it has been observed that the adam optimizer with an learning rate of 0.0001 has helped in obtaining the minimum validation loss of 0.0209 overall but importantly, a validation loss of 0.0225 after the first run through of the data (the first epoch). This is evident from figure 4.13, which shows the plot of the predicted and observed data. The error signal between the predicted and observed values is low but also the predicted data appears to resemble the trend in the observed data in training. This makes sense as the model is aware of what the label value is and should be able to learn to match this during training.

The performance of the LSTM is poor when trying to predict on interesting seismic activity, such as the volcanic eruption at White Island. Figure 4.15 shows the plot of the test predictions and observed data. Like in training, the model predicts on the observed data that reflects the volcanic system in a period of rest well. However, the model appears to have some difficulty in predicting the observed data in the eruption interval. The error signal between the predicted and observed values is very large.

The models inability to correctly predict and replicate the seismic waveform record of the eruption interval is a positive result. The model is trained and validated on the normal activity seismic data. Given that it doesn't go on to predict the eruption correctly indicates the model recognises the steady state attractor of the dynamical system of the volcano only. From this result, its clear that the the model interprets that there exists two different attractors.

With regards to data acquisition, a number of different possible sources were investigated at the earlier stages of the project period. In Carniel et al original paper [34] he elected to use seismic data from before and after one of the 1993 paroxysmal phases at the Stromboli Volcano, on of the Acolian Islands situated off the North Coast of Sicily,

Italy. The initial intention was to replicate his experiment by using the same data set however this data was not easily accessible. As a compromise, the decision was made to utilise the data available on White Island through the seismic dataset GeoNet, given that it is considered the most active volcano in New Zealand.

With regards to the model performance, a number of issues were encountered before reaching the presented results. One issue being that we were getting a similar prediction trend to the observed result however there was an offset of the predicted and observed result. This was rectified by introducing normalisation of the data before passing it to the model for fitting. The purpose of z-score normalisation is to make sure that the mean is centered around zero. The data was normalised on the training data so as to prevent the access to values in either the validation and/or the test sets.

5.1 Future Work

Although not explored due to time constraints, there were a number of further aspects that provide an interesting research direction for future work on the topic.

- When normalising the data, the mean and standard deviation were calculated using the training data only. It's arguable that the model shouldn't have access to future values within the training data when the model is training, and that normalisation should be instead carried out using moving averages.
- There is potential to use a much larger dataset. Increase the number of epochs in order to increase the accuracy of the model.
- More Hidden dense layers can be use to improve the accuracy.
- Attempt to carry the experiment using gated recurrent units (GRU's) to determine whether it performs better than LSTM. GRU's tend to train faster than LSTM's on account of the fact that a GRU has less tensor operations.
- Update the data window class so that the model can be used for multi-output predictions as well as Single-time-step and multi-time-step predictions.

Bibliography

[1] Mark Harvey. "Sentinel-1 InSAR captures 2019 catastrophic White Island eruption". In: *Journal of Volcanology and Geothermal Research* 411 (2021), p. 107124.

[2] Chandrahas Mishra and DL Gupta. "Deep machine learning and neural networks: An overview". In: *IAES International Journal of Artificial Intelligence* 6.2 (2017), p. 66.

[3] Anil Chandra Naidu Matcha. *How to Easily do Handwriting Recognition Using Deep Learning*. URL: https://nanonets.com/blog/handwritten-character-recognition/ (visited on 09/14/2022).

[4] Zhaohui Zhang et al. "A model based on convolutional neural network for online transaction fraud detection". In: *Security and Communication Networks* 2018 (2018).

[5] URL: https://lczero.org/ (visited on 09/14/2022).

[6] Sally Adee. *What Are Deepfakes and How Are They Created?* URL: https://spectrum.ieee.org/what-is-deepfake (visited on 09/14/2022).

[7] IBM Cloud Education. *Supervised Learning*. URL: https://www.ibm.com/cloud/learn/supervised-learning (visited on 09/14/2022).

[8] Jason Socrates Bardi. *The calculus wars: Newton, Leibniz, and the greatest mathematical clash of all time*. Hachette UK, 2009.

[9] Henri Poincaré. "Des fondements de la géométrie: à propos d'un livre de M. Russell". In: *Revue de métaphysique et de morale* 7.3 (1899), pp. 251–279.

[10] Henri Poincaré. "Sur les équations aux dérivées partielles de la physique mathématique". In: *American Journal of Mathematics* (1890), pp. 211–294.

[11] Ian C Percival and Derek Richards. *Introduction to dynamics*. Cambridge University Press, 1982.

[12] Olga A Nev et al. "Predicting microbial growth dynamics in response to nutrient availability". In: *PLoS computational biology* 17.3 (2021), e1008817.

[13] Martin Bohner, Meng Fan, and Jimin Zhang. "Existence of periodic solutions in predator–prey and competition dynamic systems". In: *Nonlinear Analysis: Real World Applications* 7.5 (2006), pp. 1193–1204.

[14] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

[15] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[16] Sagar Sharma. *What the hell is Perceptron?* Oct. 2019. URL: https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53 (visited on 09/13/2022).

[17] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[18] Victor Zhou. *Machine learning for beginners: An introduction to neural networks*. URL: https://victorzhou.com/blog/intro-to-neural-networks/ (visited on 09/14/2022).

[19] URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 09/14/2022).

[20] URL: https://developer.nvidia.com/deep-learning (visited on 09/14/2022).

[21] URL: https://www.nvidia.com/en-gb/data-center/tesla-v100/ (visited on 09/14/2022).

[22] URL: https://pytorch.org/ (visited on 09/14/2022).

[23] URL: https://www.rdocumentation.org/packages/neuralnet/versions/1.44.2/topics/neuralnet (visited on 09/14/2022).

[24] Martın Abadi et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems". In: *arXiv preprint arXiv:1603.04467* (2016).

[25] URL: https://www.usgs.gov/faqs/how-many-active-volcanoes-are-there-earth (visited on 09/14/2022).

[26] Raymond Russell Dibble, Ian Alistair Nairn, and Vincent Ernest Neall. "Volcanic hazards of North Island, New Zealand-Overview". In: *Journal of geodynamics* 3.3-4 (1985), pp. 369–396.

[27] Johannes Lohmann and Peter D Ditlevsen. "Risk of tipping the overturning circulation due to increasing rates of ice melt". In: *Proceedings of the National Academy of Sciences of the United States of America* 118.9 (Mar. 2021), e2017989118. ISSN: 0027-8424. DOI: 10.1073/pnas.2017989118. URL: https://europepmc.org/articles/PMC7936283.

[28] I. Newton, I.B. Cohen, and A. Whitman. *The Principia: Mathematical Principles of Natural Philosophy*. The Principia: Mathematical Principles of Natural Philosophy. University of California Press, 1999. ISBN: 9780520088160. URL: https://books.google.ie/books?id=k%5C_NgQgAACAAJ (visited on 12/09/2022).

[29] Yu.B Slezin. "The mechanism of volcanic eruptions (a steady state approach)". In: *Journal of Volcanology and Geothermal Research* 122.1 (2003), pp. 7–50. ISSN: 0377-0273. DOI: https://doi.org/10.1016/S0377-0273(02)00464-X. URL: https://www.sciencedirect.com/science/article/pii/S037702730200464X.

[30] Yann Le Cun et al. "Handwritten digit recognition: Applications of neural network chips and automatic learning". In: *IEEE Communications Magazine* 27.11 (1989), pp. 41–46.

[31] Afan Galih Salman et al. "Single layer & multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting". In: *Procedia Computer Science* 135 (2018), pp. 89–98.

[32] Joos Korstanje. *Advanced Forecasting with Python*. Springer, 2021.

[33] *Understanding LSTM networks*. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 09/13/2022).

[34] R. Carniel. "Neural networks and dynamical system techniques for volcanic tremor analysis". In: 39.2 (1996). ISSN: 0377-0273. DOI: https://doi.org/10.

4401/ag-3967. URL: https://www.annalsofgeophysics.eu/index.php/annals/article/view/3967/4032.

[35] Roberto Carniel. *Time Series Analysis: Dynamical Evolution of Spectral, Deterministic and Stochastic Parameters for the Characterization of Volcanic Activity*. 2012.

[36] Anaıs Boué et al. "Real-time eruption forecasting using the material Failure Forecast Method with a Bayesian approach". In: *Journal of Geophysical Research: Solid Earth* 120.4 (2015), pp. 2143–2161.

[37] Millaray Curilem et al. "Using CNN to classify spectrograms of seismic events from Llaima Volcano (Chile)". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–8.

[38] Braden Walsh et al. "Geophysical examination of the 27 April 2016 Whakaari/White Island, New Zealand, eruption and its implications for vent physiognomies and eruptive dynamics". In: *Earth, Planets and Space* 71.1 (2019), pp. 1–18.

[39] *Geonet stream naming conventions*. URL: https://www.geonet.org.nz/data/supplementary/channels (visited on 09/14/2022).

[40] *How are earthquakes detected, located and measured?* URL: https://www.bgs.ac.uk/discovering-geology/earth-hazards/earthquakes/how-are-earthquakes-detected/.

[41] Jens Havskov and Gerardo Alguacil. *Instrumentation in Earthquake Seismology*. Jan. 2004. ISBN: 978-1-4020-2968-4. DOI: 10.1007/978-1-4020-2969-1.

[42] Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007.

[43] *Common Loss functions in machine learning*. URL: https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23 (visited on 09/14/2022).

[44] *Performance Metrics in Machine Learning [Complete Guide]*. URL: https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide (visited on 09/14/2022).

[45] *Difference Between a Batch and an Epoch in a Neural Network*. URL: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/ (visited on 09/14/2022).

[46] Ethan R Deyle and George Sugihara. "Generalized theorems for nonlinear state space reconstruction". In: *Plos one* 6.3 (2011), e18295.