

A Dynamic and Boosting Earthquake Prediction Through Neural Network : Extensible Framework

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

Jaggavarapu Tanuja (Reg. No. 37110281)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI,
CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **Jaggavarapu Tanuja**(Reg. No. 37110281) who carried out the project entitled "**A Dynamic and Boosting Earthquake Prediction Through Neural Network : Extensible Framework**" under my supervision from **August 2020** to **March 2021**.

Internal Guide

Mrs. S.L.Jany Shabu M.Tech.,(Ph.d)

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **Jaggavarapu Tanuja** hereby declare that the Project Report entitled **“A Dynamic and Boosting Earthquake Prediction Through Neural Network : Extensible Framework”** is done by me under the guidance of **Mrs.S.L.Jany Shabu,M.Tech.,(Ph.d)** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs.S.L.Jany Shabu,M.Tech.,(Ph.d) Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

An earthquake is shaking of the surface of the Earth, which caused as the result of movable plate boundary interactions. The sudden release of energy called seismic waves that destroys so many buildings and kills thousands of people. Earthquakes are measured using remarks from seismometers with Richter magnitude scale. Ground rupture, Landslides, Soil liquefaction and Tsunami are the main effects created by earthquakes. Today's earthquake warning system is used to provide regional notification of an earthquake in progress.

In this context, the core idea of this work is to predict whereas an event is classified as negative or positive major earthquake by applying different machine learning algorithms.

TABLE OF CONTENTS

ABSTRACT	v
LIST OF FIGURES	viii

CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	Overview	1
	Scope of the project	1
	Domain Overview	1
	<i>Deep Learning</i>	2
2.	AIM AND SCOPE	15
	2.1 a	15
	2.2 b	16
	2.2 c	17
	2.3 d	18
3.	SYSTEM DESIGN & METHODOLOGY	20
	3.1 Existing System	27
	3.2 Drawbacks of Existing System	28
	3.3 Proposed System	28
	3.4 Advantages of Proposed System	29
4.	REQUIREMENT SPECIFICATION	30
	4.1 INTRODUCTION	30
	4.2 HARDWARE & SOFTWARE SPECIFICATION	31
	<i>4.2.1 HARDWARE SPECIFICATION</i>	31
	<i>4.2.2 SOFTWARE SPECIFICATION</i>	31

	SOFTWARE DESCRIPTION	31
	<i>ABOUT PYTHON</i>	31
	<i>ABOUT IDE</i>	45
5.	SYSTEM DESIGN	51
	INTRODUCTION	51
	ARCHITECTURE DIAGRAM	51
6.	SYSTEM IMPLEMENTATION	53
	LIST OF MODULES	53
	<i>MODULE 1 : EXPLORATORY DATA</i>	53
	<i>EVALUATION</i>	
	<i>MODULE 2 : PRE-PROCESSING</i>	53
	<i>MODULE 3 : FEATURE ENGINEERING</i>	53
	<i>MODULE 4 : PREDICTION</i>	53
7.	SOFTWARE TESTING	55
	GENERAL	61
	TEST DRIVEN DEVELOPMENT	62
	UNIT TESTING	64
	BLACKBOX TESTING	65
	INTEGRATION TESTING	67
	SYSTEM TESTING	68
	SANITY TESTING	69
	REGRESSION TESTING	71
8.	CONCLUSION & FEATURE ENCHANCEMENT	75
	CONCLUSION	75
	FUTURE ENHANCEMENT	75
	APPENDIX-i SCREENSHOTS	76
	APPENDIX- ii SAMPLE CODING	77
	REFERENCES	86

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
5.2	ARCHITECTURE DIAGRAM	52
	SCREENSHORTS	76

Chapter 1

INTRODUCTION

Overview

We propose a pattern based feature extraction method to enhance the performance of earthquake prediction. Specially, the raw seismic data is firstly divided into fixed day time periods, and the magnitude of the largest earthquake in each fixed time period is labeled as main shock. The pattern is a part of seismic sequence before the main shock, on which the existing mathematical statistic features can be directly generated as seismic indicators. Based on these pattern based features, a simple yet effective classification algorithm is adopted to predict the label of main shock in a pre-defined future time period.

Scope of the Project

The main contributions of this project therefore are:

- Data Analysis

- Dataset Preprocessing

- Training the Model

- Testing of Dataset

Domain Overview

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing.

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology

behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before. In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.

Deep

Learning About

Pandas

Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. The DataFrame is one of these structures.

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Pandas is built on top of the NumPy package, meaning a lot of the structure of NumPy is used or replicated in Pandas. Data in pandas is often used to feed statistical analysis in SciPy, plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.

Jupyter Notebooks offer a good environment for using pandas to do data exploration and modeling, but pandas can also be used in text editors just as easily.

Jupyter Notebooks give us the ability to execute code in a particular cell as opposed to running the entire file. This saves a lot of time when working with large datasets and complex transformations. Notebooks also provide an easy way to visualize pandas' DataFrames and plots. As a matter of fact, this article was created entirely in a Jupyter Notebook.

There are two types of data structures in pandas: Series and DataFrames.

1. Series: a pandas Series is a one dimensional data structure (“a one dimensional ndarray”) that can store values — and for every value it holds a unique index, too.
2. DataFrame: a pandas DataFrame is a two (or more) dimensional data structure – basically a table with rows and columns. The columns have names and the rows have indexes.

Those who are familiar with R know the data frame as a way to store data in rectangular grids that can easily be overviewed. Each row of these grids corresponds to measurements or values of an instance, while each column is a vector containing data for a specific variable. This means that a data frame's rows do not need to contain, but can contain, the same type of values: they can be numeric, character, logical, etc.

Now, DataFrames in Python are very similar: they come with the Pandas library, and they are defined as two-dimensional labeled data structures with columns of potentially different types.

In general, you could say that the Pandas DataFrame consists of three main components: the data, the index, and the columns.

Firstly, the DataFrame can contain data that is:

- a Pandas DataFrame
- a Pandas Series: a one-dimensional labeled array capable of holding any data type with axis labels or index. An example of a Series object is one column from a DataFrame.
- a NumPy ndarray, which can be a record or structured
- a two-dimensional ndarray
- dictionaries of one-dimensional ndarray's, lists, dictionaries or Series.

Some of the key features of Python Pandas are as follows:

It provides DataFrame objects with default and customized indexing which is very fast and efficient.

- There are tools available for loading data of different file formats into in-memory data objects.
- It is easy to perform data alignment and integrated handling of missing data in Python Pandas.

- It is very simple to perform pivoting and reshaping of data sets in Pandas.
- It also provides indexing, label-based slicing, and sub-setting of large data sets.
- We can easily insert and delete columns from a data structure.
- Data aggregation and transformations can be done using group by.
- High-performance merging and joining of data can be done using Pandas.
- It also provides time series functionality.
- Inserting and deleting columns in data structures.
- Merging and joining data sets.
- Reshaping and pivoting data sets.
- Aligning data and dealing with missing data.
- Manipulating data using integrated indexing for DataFrame objects.
- Performing split-apply-combine on data sets using the group by engine.
- Manipulating high-dimensional data in a data structure with a lower dimension using hierarchical axis indexing.
- Subsetting, fancy indexing, and label-based slicing data sets that are large in size.
- Generating data range, converting frequency, date shifting, lagging, and other time-series functionality.
- Reading from files with CSV, XLSX, TXT, among other formats.

- Arranging data in an order ascending or descending.
- Filtering data around a condition.
- Analyzing time series.
- Iterating over a data set.

Numpy

Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. If you are already familiar with MATLAB, you might find this tutorial useful to get started with Numpy.

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

NumPy is, just like SciPy, Scikit-Learn, Pandas, etc. one of the packages that you just can't miss when you're learning data science, mainly because this library provides you with an array data structure that holds some benefits over Python lists, such as: being more compact, faster access in reading and writing items, being more convenient and more efficient.

NumPy is a Python library that is the core library for scientific computing in Python. It contains a collection of tools and techniques that can be used to solve on a computer mathematical models of problems in Science and Engineering. One of these tools is a high-performance multidimensional array object that is a powerful data structure for

efficient computation of arrays and matrices. To work with these arrays, there's a vast amount of high-level mathematical functions operate on these matrices and arrays.

an array is basically nothing but pointers. It's a combination of a memory address, a data type, a shape, and strides:

- The data pointer indicates the memory address of the first byte in the array,
- The data type or dtype pointer describes the kind of elements that are contained within the array,
- The shape indicates the shape of the array, and
- The strides are the number of bytes that should be skipped in memory to go to the next element. If your strides are (10,1), you need to proceed one byte to get to the next column and 10 bytes to locate the next row.

Or, in other words, an array contains information about the raw data, how to locate an element and how to interpret an element.

With NumPy, we work with multidimensional arrays. We'll dive into all of the possible types of multidimensional arrays later on, but for now, we'll focus on 2-dimensional arrays. A 2-dimensional array is also known as a matrix, and is something you should be familiar with. In fact, it's just a different way of thinking about a list of lists. A matrix has rows and columns. By specifying a row number and a column number, we're able to extract an element from a matrix. We can create a NumPy array using the `numpy.array` function. If we pass in a list of lists, it will automatically create a NumPy array with the same number of rows and columns. Because we want all of the elements in the array to

be float elements for easy computation, we'll leave off the header row, which contains strings. One of the limitations of NumPy is that all the elements in an array have to be of the same type, so if we include the header row, all the elements in the array will be read in as strings. Because we want to be able to do computations like find the average quality of the wines, we need the elements to all be floats.

NumPy has several advantages over using core Python mathematical functions, a few of which are outlined here:

1. NumPy is extremely fast when compared to core Python thanks to its heavy use of C extensions.
2. Many advanced Python libraries, such as Scikit-Learn, Scipy, and Keras, make extensive use of the NumPy library. Therefore, if you plan to pursue a career in data science or machine learning, NumPy is a very good tool to master.

NumPy comes with a variety of built-in functionalities, which in core Python would take a fair bit of custom code.

Matplotlib

Plotting of data can be extensively made possible in an interactive way by Matplotlib, which is a plotting library that can be demonstrated in Python scripts. Plotting of graphs is a part of data visualization, and this property can be achieved by making use of Matplotlib.

Matplotlib makes use of many general-purpose GUI toolkits, such as wxPython, Tkinter, QT, etc., in order to provide object-oriented APIs for embedding plots into applications.

John D. Hunter was the person who originally wrote Matplotlib, and its lead developer

was Michael Droettboom. One of the free and open-source Python library which is basically used for technical and scientific computing is Python SciPy. Matplotlib is widely used in SciPy as most scientific calculations require plotting of graphs and diagrams.

Matplotlib is a plotting library like GNUplot. The main advantage towards GNUplot is the fact that Matplotlib is a Python module. Due to the growing interest in python the popularity of matplotlib is continually rising as well.

Another reason for the attractiveness of Matplotlib lies in the fact that it is widely considered to be a perfect alternative to MATLAB, if it is used in combination with Numpy and Scipy. Whereas MATLAB is expensive and closed source, Matplotlib is free and open source code. It is also object-oriented and can be used in an object oriented way. Furthermore it can be used with general-purpose GUI toolkits like wxPython, Qt, and GTK+. There is also a procedural "pylab", which designed to closely resemble that of MATLAB. This can make it extremely easy for MATLAB users to migrate to matplotlib.

Matplotlib can be used to create publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Another characteristic of matplotlib is its steep learning curve, which means that users usually make rapid progress after having started. The official website has to say the following about this: "matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code."

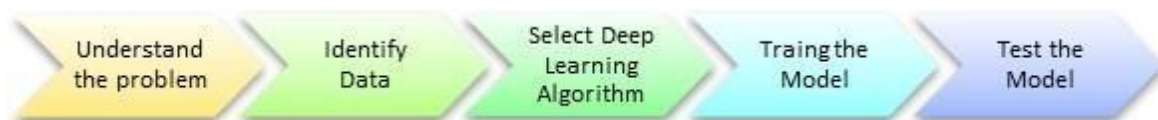
Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks.

Deep learning algorithms are constructed with connected layers.

- The first layer is called the Input Layer
- The last layer is called the Output Layer
- All layers in between are called Hidden Layers. The word deep means the network join neurons in more than two layers.

A deep neural network provides state-of-the-art accuracy in many tasks, from object detection to speech recognition. They can learn automatically, without predefined knowledge explicitly coded by the programmers.



To grasp the idea of deep learning, imagine a family, with an infant and parents. The toddler points objects with his little finger and always says the word 'cat.' As its parents are concerned about his education, they keep telling him 'Yes, that is a cat' or 'No, that is not a cat.' The infant persists in pointing objects but becomes more accurate with 'cats.' The little kid, deep down, does not know why he can say it is a cat or not. He has just learned how to hierarchies complex features coming up with a cat by looking at the pet overall and continue to focus on details such as the tails or the nose before to make up his mind.

A neural network works quite the same. Each layer represents a deeper level of knowledge, i.e., the hierarchy of knowledge. A neural network with four layers will learn more complex feature than with that with two layers.

The learning occurs in two phases:

1. The first phase consists of applying a nonlinear transformation of the input and create a statistical model as output.
2. The second phase aims at improving the model with a mathematical method known as derivative.

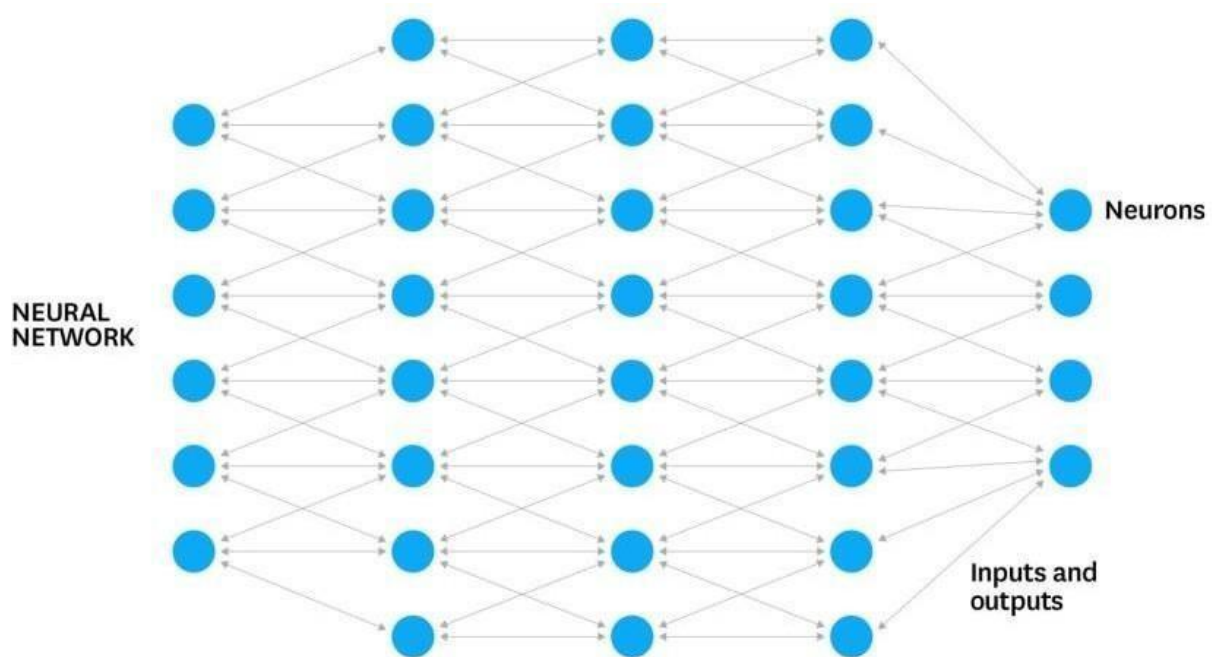
The neural network repeats these two phases hundreds to thousands of time until it has reached a tolerable level of accuracy. The repeat of this two-phase is called an iteration

Classification of Neural Networks

1. Shallow neural network: The Shallow neural network has only one hidden layer between the input and output.
2. Deep neural network: Deep neural networks have more than one layer. For instance, Google LeNet model for image recognition counts 22 layers.

The computational models in Deep Learning are loosely inspired by the human brain.

The multiple layers of training are called Artificial Neural Networks (ANN).



Neuron

Artificial Neural Networks contain layers of neurons. A neuron is a computational unit that calculates a piece of information based on weighted input parameters. Inputs accepted by the neuron are separately weighted.

Inputs are summed and passed through a non-linear function to produce output. Each layer of neurons detects some additional information, such as edges of things in a picture or tumors in a human body. Multiple layers of neurons can be used to detect additional information about input parameters.

Nodes Artificial Neural Network is an interconnected group of nodes akin to the vast network of layers of neurons in a brain. Each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another.

Inputs

Inputs are passed into the first layer. Individual neurons receive the inputs, with each of them receiving a specific value. After this, an output is produced based on these values. Outputs

The outputs from the first layer are then passed into the second layer to be processed. This continues until the final output is produced. The assumption is that the correct output is predefined.

Each time data is passed through the network, the end result is compared with the correct one, and tweaks are made to their values until the network creates the correct final output each time.

Some of the commonly used neural networks are as follows:

1. Artificial Neural Network (ANN)
2. Convolutional Neural Network (CNN)
3. Recurrent Neural Network (RNN)
4. Deep Neural Network (DNN)
5. Deep Belief Network (DBN)

Artificial neural networks are one of the main tools used in machine learning. As the “neural” part of their name suggests, they are brain-inspired systems which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize.

Generally, the working of a human brain by making the right connections is the idea behind ANNs. That was limited to use of silicon and wires as living neurons and dendrites. Here, neurons, part of human brain. That was composed of 86 billion nerve cells. Also, connected to other thousands of cells by Axons. Although, there are various inputs from sensory organs. That was accepted by dendrites. As a result, it creates electric impulses. That is used to travel through the Artificial neural network. Thus, to handle the different issues, neuron send a message to another neuron.

As a result, we can say that ANNs are composed of multiple nodes. That imitate biological neurons of the human brain. Although, we connect these neurons by links. Also, they interact with each other. Although, nodes are used to take input data. Further, perform simple operations on the data. As a result, these operations are passed to other neurons. Also, output at each node is called its activation or node value.

Chapter 2

Literature

Survey

a

Literature Survey 1	
Title	Earthquake Prediction based on Spatio-Temporal Data Mining: An LSTM Network Approach
Authors	Qianlong Wang, Yifan Guo, Lixing Yu, and Pan Li
Published Year	2017
Efficiency	• Can predict future seismic events accurately • Capture the nonlinear correlations among data • Facilitate the network training
Drawbacks	• Still cannot get good results • High Complexity • Lack of Training Data

Description	LSTMs are explicitly designed to avoid the long-term dependency problem. In particular, LSTMs have the same chain like structure, but they use a different way to implement function F in order to store information. It is to build a memory cell instead, which could be considered as a black box that, for example, at the first layer, takes the previous state h_{t-1} and current system input x_t as inputs and compute internally to decide what to keep in memory and output the hidden state h_t. All the cell state runs straight down the entire path with only some linear interactions, which makes it very easy for information to be propagated in time. To prevent our system from being overfitted, we apply a method called dropout to the output of the LSTM layer. System overfitting can lead to very high performance in training but very low in testing. This is because when overfitting occurs, the
-------------	---

	system focuses too much on historical data, which makes it too rigid to give satisfactory result on new input. Many works have proved that adding dropout in the system can efficiently prevent a neural network from Being overfitted [24]. In particular, by having dropout in the system, a certain number of randomly selected nodes are temporarily turned off in each training, along with all its conjoint connections
--	--

b

Literature Survey 2	
Title	Major earthquake event prediction using various machine learning algorithms
Authors	2019
Published Year	Roxane Mallouhy, Chady Abou Jaoude, Christophe Guyeux, Abdallah Makhoul
Efficiency	• Reliable Prediction of Major Events • Best for small data • Heavily simplified probabilistic model that calculates probabilities
Drawbacks	• Incapable of handling long-term time dependency in practice • Unsatisfactory in danger state prediction • Heavily dependent on the initial sample

Description	K-nearest neighbors (KNN) is an algorithm that can be used for regression and classification, but which is more frequently used for classification predictive problems. KNN is based on the principle that instances with similar properties exist generally in close proximity. The class value of the unclassified instance can be predicted by observing the class of its nearest neighbors. The KNN finds the only adjustable parameter in this algorithm, the K nearest neighbors to the query instance and classify it by determining the most frequent class label. The model can be made less or more flexible by modifying K. Multilayer perceptron is a neural network [12] that consists at least of one hidden layer of nodes other than the input and output layers. This means, the layers of MLP should be a minimum of 3 layers including hidden, input and output. Each node of the layer, excluded the input, is called neuron. The neuron, a processing component, which is the main element of the neural network, collects information from a certain number of inputs, applies a weight, add a bias term and send the result to an active function which generates an output
-------------	--

c

Literature Survey 3	
Title	Functional Link Neural Network Learning for Response Prediction of Tall Shear Buildings With Respect to Earthquake Data
Authors	Deepti Moyi Sahoo and Snehashish Chakraverty
Published Year	2017
Efficiency	• Better accuracy • Takes less computation time • Identifying dynamic characteristics of structures

Drawbacks	• Cannot estimate the structural response • Higher prediction complexity with higher dimensions • Very calculation intensive while training the model
Description	The ground acceleration data are taken as input, and structural responses of different floors of multistorey shear buildings are considered as output. It is worth mentioning that handling of large earthquake data has become a great challenge in the design of tall structures viz., that of shear buildings. FUNCTIONAL LINK NEURAL NETWORK FLNN is a single- layer neural network in which the hidden layer is removed and the input vectors are expanded to higher dimensional vectors by some orthogonal polynomials. The FLNN model consists of mainly two parts, one is functional expansion part and the other is learning part. Due to the absence of hidden layer, FLNN is computationally more efficient and faster than MNN. The structure of the ChNN comprises of one input node, a functional expansion block consisting of Chebyshev polynomials and one output node. Each input vector is expanded to several terms of Chebyshev orthogonal polynomials so that they may be viewed as a new input vector in the functional expansion part.

d

Literature Survey 4	
Title	A global classification and characterization of earthquake clusters
Authors	Ilya Zaliapin and Yehuda Ben-Zion
Published Year	2016

Efficiency	• Improve local seismic hazard assessments • Lowering the Complexity Threshold • Simplify Training Process
------------	--

Drawbacks	• Less accurate than those of the epicentres • Tedious data updating • dNarrowly specialized knowledge
Description	Analysis of statistical properties of the background and cluster modes is one of the main tools of this study. The bimodality of the earthquake distance distribution allows one to use a suitably chosen nearest-neighbour threshold η_0 to formally attribute each event to either the background. In each family, the earthquake with the largest magnitude is called main shock. If there are several earthquakes with same largest magnitude within a family, the first one is considered to be the main shock, so each family has a single main shock. All events in a family that occurred after the main shock are called aftershocks. All events that occurred prior to the main shock are called foreshocks. Each single is also considered to be a main shock (that has no foreshocks and aftershocks). Connecting each earthquake in the catalogue to its nearest neighbour (parent) according to the nearest-neighbour distance produces a single cluster (spanning network) that contains all examined events. From a graph-theoretical perspective, this cluster is a tree graph, which means that it does not have loops.

Chapter 3

System Analysis

One of the primary choices we face for our project implementations is “Which development methodology should we use?” There are various software development methodologies available in the ICT industry. They are

- Agile Software Development
- Crystal Methods
- Dynamic Systems Development Model (DSDM)
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Joint Application Development (JAD)
- Lean Development (LD)
- Rapid Application Development (RAD)
- Rational Unified Process (RUP)
- Scrum
- Spiral
- Systems Development Life Cycle (SDLC)
- Waterfall (a.k.a. Traditional)

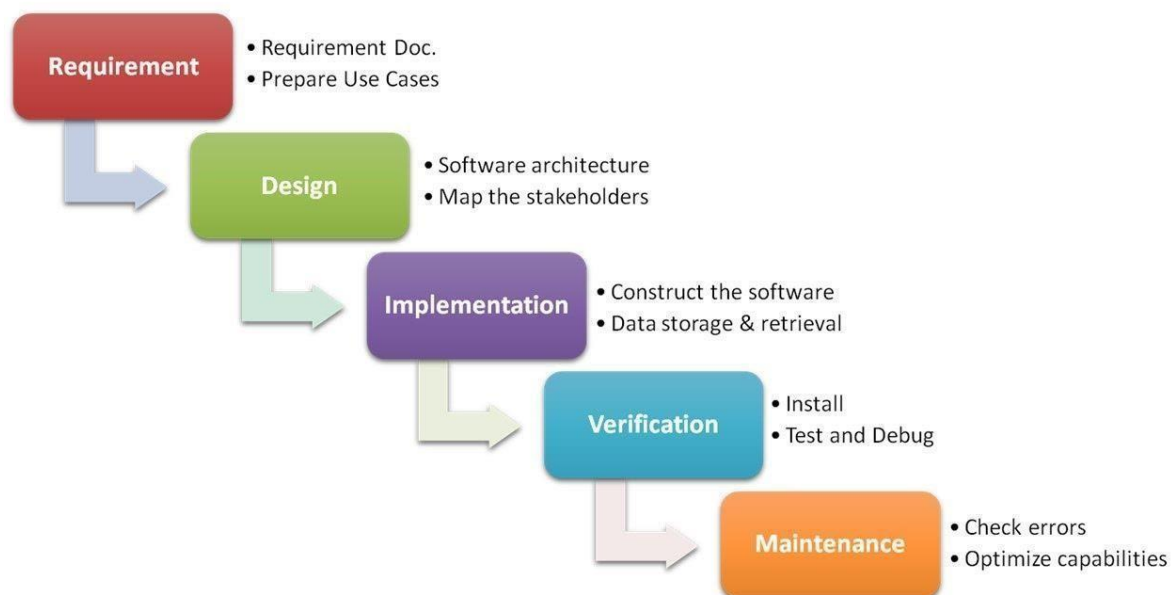
Among all these methodologies, we have decided to compare the two most popular and widely used methodologies:

- Waterfall Model
- RAD Model

Waterfall Model

It is called as traditional approach. Waterfall is a linear method (sequential model) for any software application development. Here application development is segregated into a sequence of pre-defined phases.

Waterfall Model (Taken from Google.com)



(Reference:<https://www.lucidchart.com/blog/waterfall-project-management-methodology>)

Requirement gathering and documentation

In this stage, you should gather comprehensive data approximately what this challenge requires. You may gather this data in a diffusion of ways, from interviews to questionnaires to interactive brainstorming. By means of the stop of this section, the task requirements have to be clean, and also you have to have a necessities file that has been allotted on your team.

2. System design

Using the well-known requirements, your team designs the solutions. During this phase, no development will be happening. But the project team starts specification such as programming language or hardware requirements.

3. Implementation

During this phase software development coding will be happening. Web application programmers take data from the previous stage and create a functional product. Web application programmers write source code in small pieces, which are integrated at the end of this phase or the beginning of the next.

4. Testing

Once all coding is done, testing of the product can begin. Testers methodically find and report any problems. If serious issues arise, your project may need to return to phase one for reevaluation.

5. Delivery/deployment

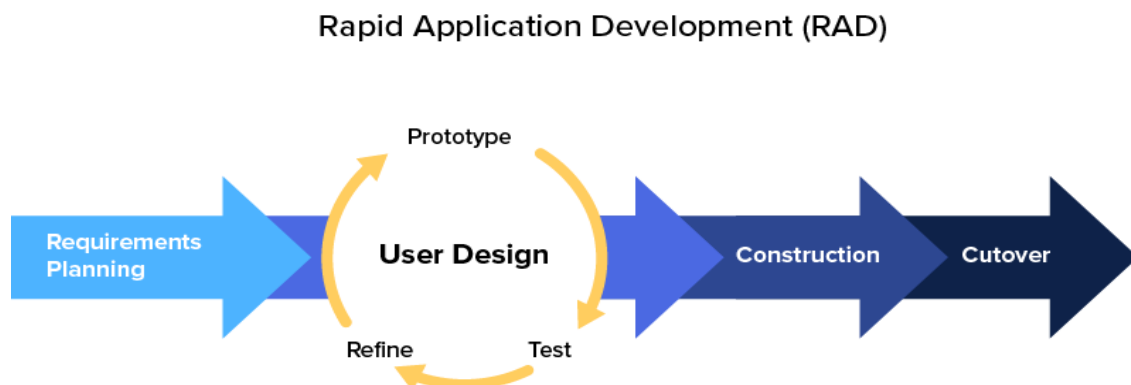
In this phase, the solution is complete, and your project team submits the deliverables to be deployed or released.

6. Maintenance

The final solution has been implemented to the client and is being used. As troubles arises, the project team might also want to create patches and updates might also to deal with them. Again, huge troubles may also necessitate a return to segment one.

RAD Model

Agile is an iterative, team-based approach to development. This method highlights the speedy delivery of an software application in comprehensive functional components



RAD Model (Taken from Google.com)

References: (<https://www.lucidchart.com/blog/rapid-application-development-methodology>)

Phase 1: Requirements planning

This phase is equivalent to a project scoping meeting. Although the planning phase is condensed compared to other project management methodologies, this is a critical step for the ultimate success of the project.

During this stage, web application programmers, clients (software users), and team members communicate to determine the goals and expectations for the project as well as current and potential issues that would need to be addressed during the build.

A basic breakdown of this stage involves:

Researching the current problem

Defining the requirements for the project

Finalizing the requirements with each stakeholder's approval

It is important that everyone has the opportunity to evaluate the goals and expectations for the project and weigh in. By getting approval from each key stakeholder and web application programmer, teams can avoid miscommunications and costly change orders down the road.

Phase 2: User design

Once the project is scoped out, it's time to jump right into development, building out the user design through various prototype iterations.

This is the meat and potatoes of the RAD methodology—and what sets it apart from other

project management strategies. During this phase, clients work hand in hand with web application programmers to ensure their needs are being met at every step in the design process. It's almost like customizable software development where the users can test each prototype of the product, at each stage, to ensure it meets their expectations.

All the bugs and kinks are worked out in an iterative process. The web application programmer designs a prototype, the client (user) tests it, and then they come together to communicate on what worked and what didn't.

This method gives web application programmers the opportunity to tweak the model as they go until they reach a satisfactory design.

Both the software web application programmers and the clients learn from the experience to make sure there is no potential for something to slip through the cracks.

Phase 3: Rapid construction

Phase 3 takes the prototypes and beta systems from the design phase and converts them into the working model. Because the majority of the problems and changes were addressed during the thorough iterative design phase, web application programmers can construct the final working model more quickly than they could by following a traditional project management approach.

The phase breaks down into several smaller

steps: Preparation for rapid construction

Program and application development

Coding

Unit, integration, and system testing

The software development team of programmers, coders, testers, and web application programmers work together during this stage to make sure everything is working smoothly and that the end result satisfies the client's expectations and objectives.

This third phase is important because the client still gets to give input throughout the process. They can suggest alterations, changes, or even new ideas that can solve problems as they arise.

Phase 4: Cutover

This is the implementation phase where the finished product goes to launch. It includes data conversion, testing, and changeover to the new system, as well as user training.

All final changes are made while the coders and clients continue to look for bugs in the system.

Benefits of RAD methodology

RAD is one of the most successful software development programs available today, with numerous benefits for both software development teams as well as their clients.

Here are just a few advantages:

- RAD lets you break the project down into smaller, more manageable tasks.
- The task-oriented structure allows project managers to optimize their team's efficiency by assigning tasks according to members' specialties and experience.
- Clients get a working product delivered in a shorter time frame.

- Regular communication and constant feedback between team members and stakeholders increases the efficiency of the design and build process.

Existing System

In the existing techniques based on Artificial Intelligence (AI) are well known for their capability to find hidden patterns in data. In the case of earthquake prediction, these models also produce a promising outcome. This work systematically explores the contributions made to date in earthquake prediction using AI-based techniques. A total of 84 scientist research papers, which reported the use of AI-based techniques in earthquake prediction, have been selected from different academic databases. These studies include a range of AI techniques including rule-based methods, shallow machine learning and deep learning algorithms. Covering all existing AI-based techniques in earthquake prediction, this article provides an account of the available methodologies and a comparative analysis of their performances. The performance comparison has been reported from the perspective of used datasets and evaluation metrics. Furthermore, using comparative analysis of performances the paper aims to facilitate the selection of appropriate techniques for earthquake prediction.

The first layer generates grade membership functions like Gaussian functions, triangular functions, and trapezoid functions, which are used to generate firing strength. The second layer uses the membership function's grade to calculate the firing strength. The output of each model is compared, and the product or minimum of them is selected. In the third layer, normalization is done by dividing the firing strength of a rule by the combined firing strength. Defuzzification is the next layer where the output is calculated using the weighted parameters. These activation functions take the inputs and biases of the previous layers. calculate the weighted sum, and scale it within some range. This way, the data move forward, and the output layer produces predicted output. After this forward pass, the error is Calculated by the sum square function which is the residue of actual output and predicted output. This error function needs to be minimized. The BP algorithm is adopted to minimize the error and adjust the biases and weights accordingly. The first derivative of the error is calculated with respect to

the weights. Then learning rate is multiplied with these values and is deducted from the weights to adjust them. This method can be slow to converge to the optimal value.

Drawbacks of Existing System

- Not a reliable prediction of major events
- Patterns are not discovered correctly
- Features are not correctly capture according characteristics of earthquake
- Magnitude of earthquakes was not considered
- High Computational cost

Proposed System

The proposed work heavily simplified probabilistic model that calculates probabilities by counting combinations of values and frequency in a data set.classification assumes that the data belongs to a particular class, then the probability for the hypothesis to be true is calculated. It operates on a strong independence assumption, in other terms, the conditional probability of one attributes will not affect the probability of the other.

However, the hypothesis can be updated each time new evidence is added.

The core idea of this work is to predict whereas an event is classified as negative or positive major earthquake by applying different machine learning algorithms. It is well known that there is no best algorithm or one solution that fits all the problems and datasets for machine learning since the performance of algorithms depends on many factors. Some algorithms are best for small data, while others perform better for a tons of data sample. Some algorithms require categorical inputs, while others need quantitative. Another important criterion while choosing the algorithm is the complexity of the dataset and how many features the model needs to learn and predict. A multilayer perceptron model consists mainly of a linear activation function of all the neurons and backpropagation process for the training. The activation function maps the weighted supervised learning process that occurs with a continuous adjustment after each processing (based on the error in output) of the weights of the connections.

Advantages of Proposed System

- Infer the effective time range of earthquake prediction results
- Can estimate the magnitude of the earthquake.
- Tolerates Variations
- Maximum Utilization of Unstructured Data
- Ability To Deliver High Quality Results

Chapter 4

Requirement Specification

Introduction

Clearly defined requirements are essential signs on the road that leads to a successful project. They establish a formal agreement between a client and a provider that they are both working to reach the same goal. High-quality, detailed requirements also help mitigate financial risks and keep the project on a schedule. According to the Business Analysis Body of Knowledge definition, requirements are a usable representation of a need.

Creating requirements is a complex task as it includes a set of processes such as elicitation, analysis, specification, validation, and management. In this article, we'll discuss the main types of requirements for software products and provide a number of recommendations for their use.

Hardware & Software Specification

Hardware Specification

Hardware Requirement

Hardware Minimum Requirement

Disk Space 32 GB or more, 10 GB or more for Foundation

Edition Processor 1.4 GHz 64 bit

Memory 512 MB

Display (800 × 600) Capable video adapter and monitor

Software Specification

Backend Technologies

- Python
- Numpy
- Sci-learn
- Eclipse IDE

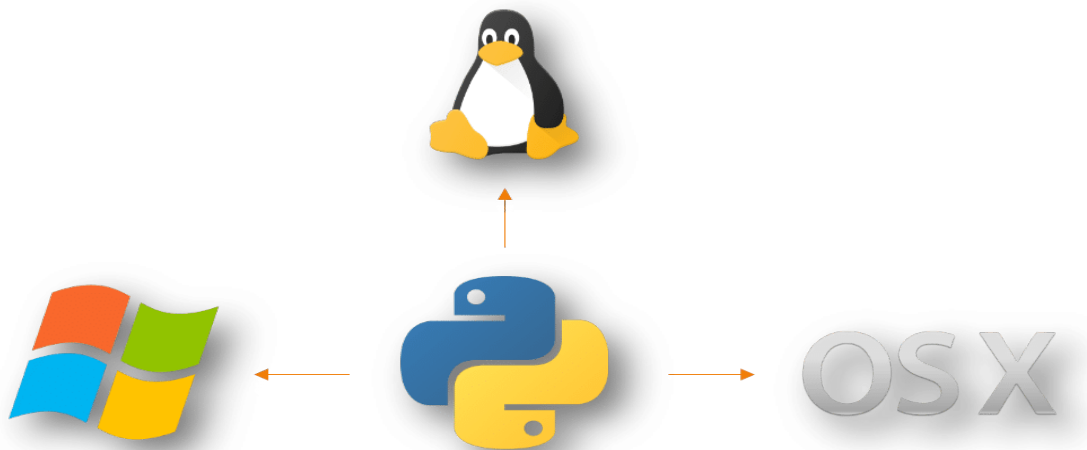
Frontend Technologies

- Web Technologies
- Bootstrap

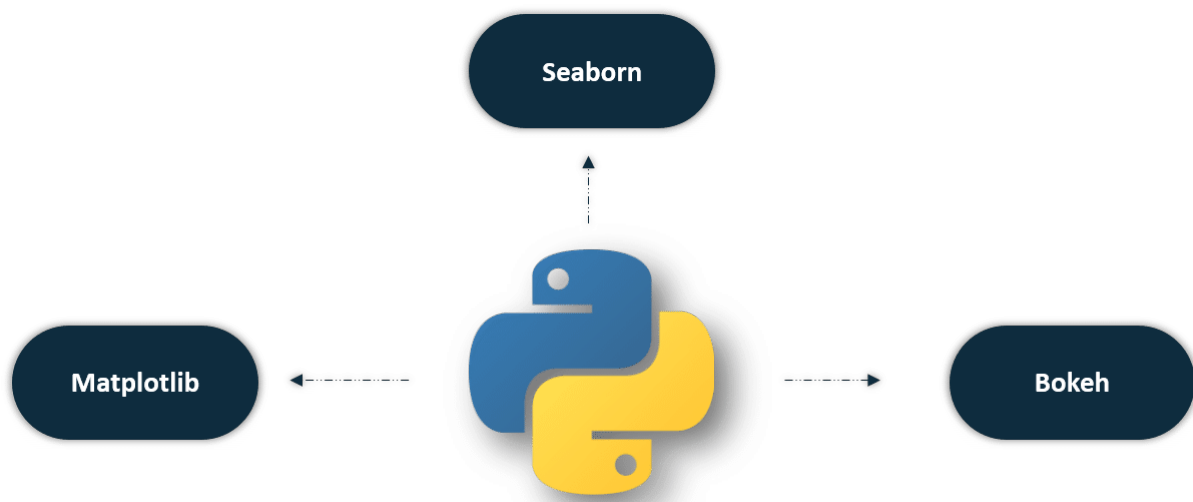
Software Description

About Python

Python is a free, open-source programming language. Therefore, all you have to do is install Python once, and you can start working with it. Not to mention that you can contribute your own code to the community. Python is also a cross-platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. Whether you have a Windows, Mac or Linux, you can rest assure that Python will work on all these operating systems.



Python is also a great visualization tool. It provides libraries such as Matplotlib, seaborn and bokeh to create stunning visualizations.



In addition, Python is the most popular language for machine learning and deep learning. As a matter of fact, today, all top organizations are investing in Python to implement machine learning in the back-end.

Python Line Structure

Python coding style comprises physical lines as well as logical lines or statements. A physical line in a Python program is a sequence of characters, and the end of the line terminates the line sequence as opposed to some other languages, such as C and C++ where a semi-colon is used to mark the end of the statement. A logical line, on the other hand, is composed of one or more physical lines. The use of a semi-colon is not prohibited in Python, although it's not mandatory. The NEWLINE token denotes the end

of the logical line. A logical line that only contains spaces, comments, or tabs are called blank lines and they are ignored by the interpreter.

As we saw that in Python, a new line simply means that a new statement has started. Although, Python does provide a way to split a statement into a multiline statement or to join multiple statements into one logical line. This can be helpful to increase the readability of the statement. Following are the two ways to split a line into two or more lines:

Explicit Line Joining

In explicit line joining, we use a backward slash to split a statement into a multiline statement.

Implicit Line Joining

Statements that reside inside [], {}, or () parentheses can be broken down into two or more physical lines without using a back slash.

Multiple Statements on a Single Line

In Python, it is possible to club multiple statements in the same line using a semi-colon; however, most programmers do not consider this to be a good practice as it reduces the readability of the code.

Whitespaces and Indentation

Unlike most of the programming languages, Python uses indentation to mark a block of

code. According to Python coding style guideline or PEP8, we should keep an indent size of four.

Most of the programming languages provide indentation for better code formatting and don't enforce to have it. But in Python it is mandatory. This is why indentation is so crucial in Python.

Comments in any programming language are used to increase the readability of the code. Similarly, in Python, when the program starts getting complicated, one of the best ways to maintain the readability of the code is to use Python comments. It is considered a good practice to include documentations and notes in the python syntax since it makes the code way more readable and understandable to other programmers as well, which comes in handy when multiple programmers are simultaneously working on the same project. The code can only explain how it does something and not why it does that, but Python comments can do that. With Python comments, we can make documentations for various explanations in our code itself. Comments are nothing but tagged lines of codes which increase the readability of a code and make it self-explanatory. There are different ways of creating comments depending on the type of comment we want to include in our code. Following are different kinds of comments that can be included in our Python program:

1. Single Line Comments
2. Docstring Comments
3. Multiline Comments

Single line Python comments are marked with # character. These comments end at the end of the physical line, which means that all characters starting after the # character (and lasts till the end of the line) are part of the comment.

Python has the documentation strings (or docstrings) feature which is usually the first statement included in functions and modules. Rather than being ignored by the Python Interpreter like regular comments, docstrings can actually be accessed at the run time using the dot operator.

It gives programmers an easy way of adding quick notes with every Python module, function, class, and method. To use this feature, we use triple quotes in the beginning of the documentation string or comment and the closing triple quotes at the end of the documentation comment. Docstrings can be one-liners as well as multi-liners.

Unlike some programming languages that support multiline comments, such as C, Java, and more, there is no specific feature for multiline comments in Python. But that does not mean that it is totally impossible to make multiline comments in Python. There are two ways we can include comments that can span across multiple lines in our Python code. Python Block Comments: We can use several single line comments for a whole block. This type of comment is usually created to explain the block of code that follows the Block comment. Python Block comment is the only way of writing a real comment that can span across multiple lines. It is supported and preferred by Python's PEP8 style guide since Block comments are ignored by Python interpreter or parser. However, nothing is stopping programmers from using the second 'non-real' way of writing multiline comments in Python which is explained below.

Using Docstrings: Docstrings are largely used as multiline comments in Python by many programmers since it is the closest thing to having a multiline comment feature in Python. While it is not wrong to use docstrings when we need to make multiline comments, it is important to keep in mind that there is a significant difference between docstrings and comments. Comments in Python are totally ignored by the Python Interpreter, while docstrings, when used inside the Python function, can be accessed at the run time.

Python Data Types

One of the most crucial part of learning any programming language is to understand how data is stored and manipulated in that language. Users are often inclined toward Python because of its ease of use and the number of versatile features it provides. One of those features is dynamic typing.

In Python, unlike statically typed languages like C or Java, there is no need to specifically declare the data type of the variable. In dynamically typed languages such as Python, the interpreter itself predicts the data type of the Python Variable based on the type of value assigned to that variable.

Advantages of Python

- Universal Language Construct
- Support both High Level and Low Level Programming
- Language Interoperability
- Fastest Development life cycle therefore more productive coding environment
- Less memory used because a single container hold multiple data types

and each type doesn't require its own function

- Learning Ease and open source development

- Speed and user-friendly data structure
- Extensive and extensible libraries.
- Simple & support IoT
- And many more

Based on the 3 languages that are described above, we decided to use Python as programming language for developing this e-Voting web based application. The important motives are

- Easy to learn, even non experienced programmers can use it. Ex: spacing and tabbing instead of extra syntax
- Interactive mode
- Large and comprehensive standard libraries
- Python programs resemble to that of pseudo-code. This makes it a basis and a must have for beginner programmers due to its extreme ease a difficulty when compared to C++ Java, Perl, and so forth

About Pandas

Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. The DataFrame is one of these structures.

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames

allow you to store and manipulate tabular data in rows of observations and columns of variables.

Pandas is built on top of the NumPy package, meaning a lot of the structure of NumPy is used or replicated in Pandas. Data in pandas is often used to feed statistical analysis in SciPy, plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.

Jupyter Notebooks offer a good environment for using pandas to do data exploration and modeling, but pandas can also be used in text editors just as easily.

Jupyter Notebooks give us the ability to execute code in a particular cell as opposed to running the entire file. This saves a lot of time when working with large datasets and complex transformations. Notebooks also provide an easy way to visualize pandas' DataFrames and plots. As a matter of fact, this article was created entirely in a Jupyter Notebook.

There are two types of data structures in pandas: Series and DataFrames.

1. Series: a pandas Series is a one dimensional data structure (“a one dimensional ndarray”) that can store values — and for every value it holds a unique index, too.
2. DataFrame: a pandas DataFrame is a two (or more) dimensional data structure — basically a table with rows and columns. The columns have names and the rows have indexes.

Those who are familiar with R know the data frame as a way to store data in rectangular grids that can easily be overviewed. Each row of these grids corresponds to

measurements or values of an instance, while each column is a vector containing data for a specific variable. This means that a data frame's rows do not need to contain, but can contain, the same type of values: they can be numeric, character, logical, etc. Now, DataFrames in Python are very similar: they come with the Pandas library, and they are defined as two-dimensional labeled data structures with columns of potentially different types. In general, you could say that the Pandas DataFrame consists of three main components: the data, the index, and the columns.

Firstly, the DataFrame can contain data that is:

- a Pandas DataFrame
- a Pandas Series: a one-dimensional labeled array capable of holding any data type with axis labels or index. An example of a Series object is one column from a DataFrame.
- a NumPy ndarray, which can be a record or structured
- a two-dimensional ndarray
- dictionaries of one-dimensional ndarray's, lists, dictionaries or Series.

Some of the key features of Python Pandas are as follows:

It provides DataFrame objects with default and customized indexing which is very fast and efficient.

- There are tools available for loading data of different file formats into in-memory data objects.
- It is easy to perform data alignment and integrated handling of missing data in Python Pandas.
- It is very simple to perform pivoting and reshaping of data sets in Pandas.
- It also provides indexing, label-based slicing, and sub-setting of large data sets.
- We can easily insert and delete columns from a data structure.
- Data aggregation and transformations can be done using group by.
- High-performance merging and joining of data can be done using Pandas.
- It also provides time series functionality.
- Inserting and deleting columns in data structures.
- Merging and joining data sets.
- Reshaping and pivoting data sets.
- Aligning data and dealing with missing data.
- Manipulating data using integrated indexing for DataFrame objects.
- Performing split-apply-combine on data sets using the group by engine.

- Manipulating high-dimensional data in a data structure with a lower dimension using hierarchical axis indexing.
- Subsetting, fancy indexing, and label-based slicing data sets that are large in size.
- Generating data range, converting frequency, date shifting, lagging, and other time- series functionality.
- Reading from files with CSV, XLSX, TXT, among other formats.
- Arranging data in an order ascending or descending.
- Filtering data around a condition.
- Analyzing time series.
- Iterating over a data set.

Numpy

Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. If you are already familiar with MATLAB, you might find this tutorial useful to get started with Numpy.

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

NumPy is, just like SciPy, Scikit-Learn, Pandas, etc. one of the packages that you just can't miss when you're learning data science, mainly because this library provides you with an array data structure that holds some benefits over Python lists, such as: being more compact, faster access in reading and writing items, being more convenient and more efficient.

NumPy is a Python library that is the core library for scientific computing in Python. It contains a collection of tools and techniques that can be used to solve on a computer mathematical models of problems in Science and Engineering. One of these tools is a high-performance multidimensional array object that is a powerful data structure for efficient computation of arrays and matrices. To work with these arrays, there's a vast amount of high-level mathematical functions operate on these matrices and arrays.

an array is basically nothing but pointers. It's a combination of a memory address, a data type, a shape, and strides:

- The data pointer indicates the memory address of the first byte in the array,
- The data type or dtype pointer describes the kind of elements that are contained within the array,
- The shape indicates the shape of the array, and
- The strides are the number of bytes that should be skipped in memory to go to the next element. If your strides are (10,1), you need to proceed one byte to get to the next column and 10 bytes to locate the next row.

Or, in other words, an array contains information about the raw data, how to locate an element and how to interpret an element.

With NumPy, we work with multidimensional arrays. We'll dive into all of the possible types of multidimensional arrays later on, but for now, we'll focus on 2-dimensional arrays. A 2-dimensional array is also known as a matrix, and is something you should be familiar with. In fact, it's just a different way of thinking about a list of lists. A matrix has rows and columns. By specifying a row number and a column number, we're able to extract an element from a matrix. We can create a NumPy array using the `numpy.array` function. If we pass in a list of lists, it will automatically create a NumPy array with the same number of rows and columns. Because we want all of the elements in the array to be float elements for easy computation, we'll leave off the header row, which contains strings. One of the limitations of NumPy is that all the elements in an array have to be of the same type, so if we include the header row, all the elements in the array will be read in as strings. Because we want to be able to do computations like find the average quality of the wines, we need the elements to all be floats.

NumPy has several advantages over using core Python mathematical functions, a few of which are outlined here:

1. NumPy is extremely fast when compared to core Python thanks to its heavy use of C extensions.

2. Many advanced Python libraries, such as Scikit-Learn, Scipy, and Keras, make extensive use of the NumPy library. Therefore, if you plan to pursue a career in data science or machine learning, NumPy is a very good tool to master.

NumPy comes with a variety of built-in functionalities, which in core Python would take a fair bit of custom code.

Matplotlib

Plotting of data can be extensively made possible in an interactive way by Matplotlib, which is a plotting library that can be demonstrated in Python scripts. Plotting of graphs is a part of data visualization, and this property can be achieved by making use of Matplotlib.

Matplotlib makes use of many general-purpose GUI toolkits, such as wxPython, Tkinter, QT, etc., in order to provide object-oriented APIs for embedding plots into applications. John D. Hunter was the person who originally wrote Matplotlib, and its lead developer was Michael Droettboom. One of the free and open-source Python library which is basically used for technical and scientific computing is Python SciPy. Matplotlib is widely used in SciPy as most scientific calculations require plotting of graphs and diagrams.

Matplotlib is a plotting library like GNUplot. The main advantage towards GNUplot is the fact that Matplotlib is a Python module. Due to the growing interest in python the popularity of matplotlib is continually rising as well.

Another reason for the attractiveness of Matplotlib lies in the fact that it is widely considered to be a perfect alternative to MATLAB, if it is used in combination with Numpy and Scipy. Whereas MATLAB is expensive and closed source, Matplotlib is free and open source code. It is also object-oriented and can be used in an object oriented way. Furthermore it can be used with general-purpose GUI toolkits like wxPython, Qt, and GTK+. There is also a procedural "pylab", which designed to closely resemble that of MATLAB. This can make it extremely easy for MATLAB users to migrate to matplotlib.

Matplotlib can be used to create publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Another characteristic of matplotlib is its steep learning curve, which means that users usually make rapid progress after having started. The official website has to say the following about this: "matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code."

About IDE

IDE – Integrated Development Environment. It is software suite. It contains different instruments for web application software engineers to compose and test applications. Programming Web application software engineers utilize wide scope of devices habitually all through application source code creation, building and application testing. Programming Development apparatuses frequently incorporate word processors for source code creation, code libraries for reuse, compilers and test stages. An Integrated

Development Environment (IDE) is a product application that gives a programming.

domain to streamline creating and troubleshooting programming. Instead of playing out every one of the means required to make an executable program as disconnected individual assignments, it brings every one of the devices required into one application and workspace. Every one of the devices has a familiarity with the earth, and they cooperate to display a consistent improvement set for the designer.

An IDE typically has an editor and a compiler window. There, we can write our code, run it, and compile it. Hence, it becomes much easier to develop applications using an IDE. The major reasons for using an IDE are given below:

- We can easily write our code in its text editor window.
- To store the resource files of a project, we have a project editor window.
- It makes debugging of the code easier.
- An IDE's autosuggest feature helps us in writing efficient code.
- We can import other projects for integrating with the existing one.

Without an IDE, a product web application software engineer must choose, send, incorporate and deal with these instruments independently. An IDE brings huge numbers of those improvement related instruments all together system, application or administration. The incorporated toolset is intended to streamline programming advancement and can distinguish and limit coding missteps and grammatical errors.

- Komodo
- NetBeans
- PyCharm

- Python for Visual Studio.NET
- PyDev
- PyStudio
- LiClipse (Eclipse)

LiClipse is at the core for all Python based application development activities. It's been available for many years. LiClipse is a set of plugins to enhance Eclipse. It improves the overall Eclipse experience. It is best IDE tool to develop Python Applications.

Advantages of LiClipse

- Perfect Debugger
- Error checking features
- An enhanced ability to analyze the code for quality and security concerns
- Auto complete feature

Based on the various IDE that are described above, we decided to use LiClipse as IDE (Integrated Development Environment) for developing this e-Voting web based application.

Important Features That Make LiClipse our IDE of Choice

- Rich Reusable Libraries are available
- Model Driven Development

- Enterprise Python Tooling
- Powerful Debugging options

IDE is a program that is used to write applications (other programs) in a variety of programming languages. It allows you to complete many distinct program development steps in one place (program window). A program which allows you to do this is known as an integrated development environment, or IDE. The program is free and easy to import onto your home computer.

LiClipse is used for writing programs in Python. There are several steps that must be completed to set up LiClipse and further steps to start up a new Python project (set of Python components which work together to make up an application).

PyCharm is an Integrated Development Environment (IDE) that provides tools for writing efficient code. It supports various frameworks such as Django, Flask, and Pyramid. PyCharm is majorly used by developers for application development with Python language.

Let's suppose, we are making a project using Python programming language. We know that Python is extensively popular among the developer's community, as well as in industries, due to its following features:

- Simple to code: Python is a very simple language compared to other programming languages. Also, it is a developer-friendly language.
- Object-oriented language: Python is an object-oriented language that uses concepts of classes and objects.
- GUI programming support: We can create graphical user interfaces using Python Tkinter library. Besides, we can use modules such as PyQt5, PyQt4, and wxPython to create a GUI application.

- High-level language: Python is a high-level programming language. We do not need to take care of managing memory or the system architecture while writing Python codes.
- Portability: Python is a portable programming language, i.e., we can run Python codes on any platform. For example, we can run a code written for Windows in Linux, UNIX, and Mac, without changing it.
- Integrated language: Python is an integrated programming language. We can easily integrate C, C++, etc. with Python.
- Interpreted language: Python is an interpreted programming language as well. Python's source code is changed to bytecode. Due to this, it is directly executed without compiling.
- Large set of standard libraries: Python consists of a rich set of modules, packages, and libraries. It helps in various stages of application development.

"notebook" or "notebook documents" denote documents that contain both code and rich text elements, such as figures, links, equations, ... Because of the mix of code and text elements, these documents are the ideal place to bring together an analysis description, and its results, as well as, they can be executed perform the data analysis in real time. The Jupyter Notebook App produces these documents.

As a server-client application, the Jupyter Notebook App allows you to edit and run your notebooks via a web browser. The application can be executed on a PC without Internet access, or it can be installed on a remote server, where you can access it through the Internet.

Its two main components are the kernels and a dashboard.

A kernel is a program that runs and introspects the user's code. The Jupyter Notebook

App has a kernel for Python code, but there are also kernels available for other programming languages.

The dashboard of the application not only shows you the notebook documents that you have made and can reopen but can also be used to manage the kernels: you can which ones are running and shut them down if necessary.

Chapter 5

System Design

Introduction

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

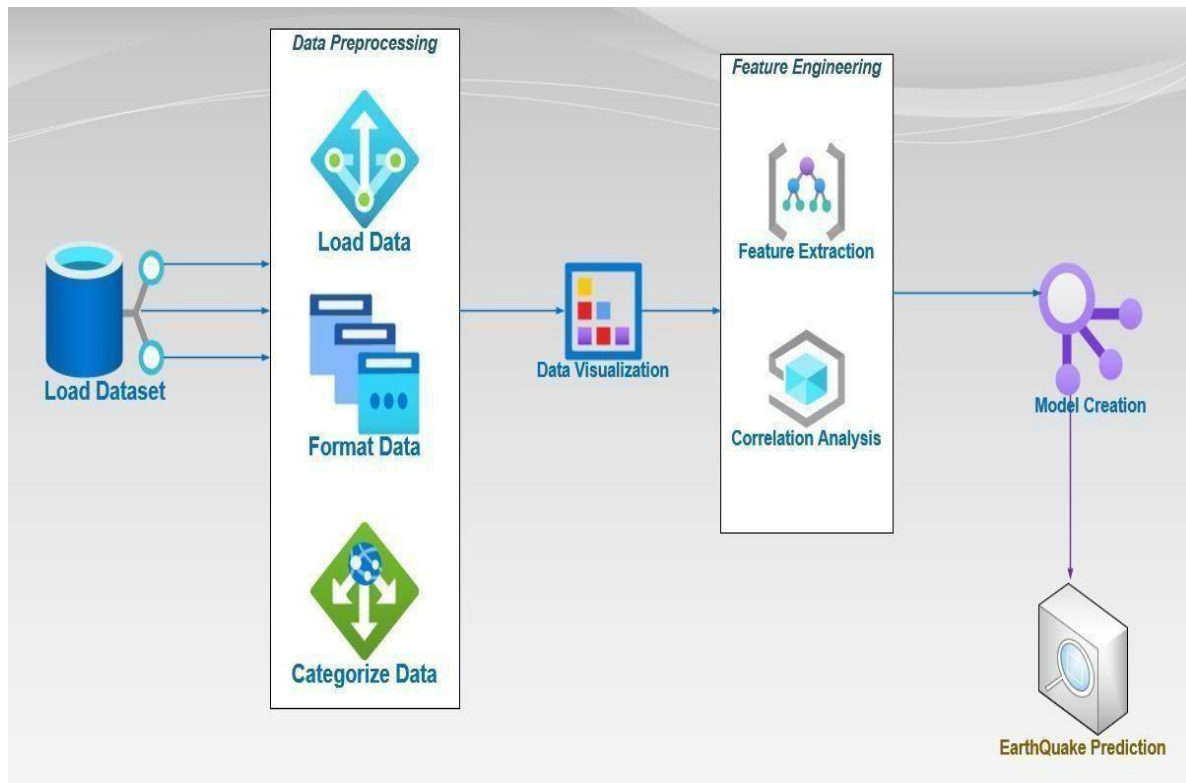
Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficiency.

Architecture Diagram

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. What is a diagram? What are the types of diagrams for architecture? The Dragon1 open EA Method makes it very clear: if a diagram does not show a concept, principle or part of a principle, it is NOT an architecture diagram, because it does not show (a part of) the architecture. There are many kinds of architecture diagrams, like a software architecture diagram, system architecture diagram, application architecture diagram, security architecture diagram, etc.

For system developers, they need system architecture diagrams to understand, clarify, and communicate ideas about the system structure and the user requirements that the

system must support. It's a basic framework can be used at the system planning phase helping partners understand the architecture, discuss changes, and communicate intentions clearly.



5.2 ARCHITECTURE DIAGRAM

Chapter 6

System Implementation

List of Modules

Module 1 : Data Analysis

Module 2 : Feature Engineering

Module 3 : Model Creation and Evaluation

Module 1 : Data Analysis

These techniques are used to explain a value from a series of discrete or continuous variables. These are fairly efficient, non-parametric and non-linear methods of partitioning individuals, producing groups of individuals as homogeneous as possible from the point of view of the variable to be predicted, taking into account a hierarchy of the predictive capacity of the variables considered. This hierarchy makes it possible to visualize the results in a tree, and to constitute explicit explanatory rules.

Module 2 : Feature Engineering

The method is useful for detecting potential earthquake patterns, and the feature extraction method proposed by can estimate the time range of the prediction results. In order to predict both the magnitude range of future earthquakes as well as obtain the effective time range of prediction results, we propose a pattern based features extraction method by combining the advantages of these two methods. To be specific, the sequence of historical seismic records is firstly divided into N-day time period. The magnitude range of the largest earthquake in the corresponding time period is the prediction target (i.e. main shock), labeled . Different from the work [6] that a set of fixed length of events before main shock is defined as the pattern, in this paper, the pattern is the pattern sequence before the main shock that consists two part,i.e. the internal pattern sequence P_{li} and the external pattern sequence.

Module 3 : Model Creation and Evaluation

These techniques is based on the principle that instances with similar properties exist

generally in close proximity. The class value of the unclassified instance can be predicted by observing the class. several trees are used. And to avoid having equal trees, randomness is added: each tree has a fragmented vision of the problem, randomly drawn from the input observations, and from the explanatory variables. More precisely, the assembly of decision trees built on the basis of a random draw among the observations is the tree bagging algorithm.

Chapter 7

Software Testing

Testing documentation is the documentation of artifacts that are created during or before the testing of a software application. Documentation reflects the importance of processes for the customer, individual and organization. Projects which contain all documents have a high level of maturity. Careful documentation can save the time, efforts and wealth of the organization.

If the testing or development team gets software that is not working correctly and developed by someone else, so to find the error, the team will first need a document. Now, if the documents are available then the team will quickly find out the cause of the error by examining documentation. But, if the documents are not available then the tester need to do black box and white box testing again, which will waste the time and money of the organization. More than that, Lack of documentation becomes a problem for acceptance.

Benefits of using Documentation

- Documentation clarifies the quality of methods and objectives.
- It ensures internal coordination when a customer uses software application.
- It ensures clarity about the stability of tasks and performance.
- It provides feedback on preventive tasks.
- It provides feedback for your planning cycle.

- It creates objective evidence for the performance of the quality management system.

The test scenario is a detailed document of test cases that cover end to end functionality of a software application in liner statements. The liner statement is considered as a scenario. The test scenario is a high-level classification of testable requirements. These requirements are grouped on the basis of the functionality of a module and obtained from the use cases.

In the test scenario, there is a detailed testing process due to many associated test cases. Before performing the test scenario, the tester has to consider the test cases for each scenario.

In the test scenario, testers need to put themselves in the place of the user because they test the software application under the user's point of view. Preparation of scenarios is the most critical part, and it is necessary to seek advice or help from customers, stakeholders or developers to prepare the scenario.

As per the IEEE Documentation describing plans for, or results of, the testing of a system or component, Types include test case specification, test incident report, test log, test plan, test procedure, test report. Hence the testing of all the above mentioned documents is known as documentation testing. This is one of the most cost effective approaches to testing. If the documentation is not right: there will be major and costly problems. The documentation can be tested in a number of different ways to many different degrees of complexity. These range from running the documents through a spelling and grammar checking device, to manually reviewing the documentation to remove any ambiguity or inconsistency.

Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed. The most popular testing documentation files are test reports, plans, and checklists. These documents are used to outline the team's workload and keep track of the process. Let's take a look at the key requirements for these files and see how they contribute to the process.

Test strategy. An outline of the full approach to product testing. As the project moves along, developers, designers, product owners can come back to the document and see if the actual performance corresponds to the planned activities.

Test data. The data that testers enter into the software to verify certain features and their outputs. Examples of such data can be fake user profiles, statistics, media content, similar to files that would be uploaded by an end-user in a ready solution.

Test plans. A file that describes the strategy, resources, environment, limitations, and schedule of the testing process. It's the fullest testing document, essential for informed planning. Such a document is distributed between team members and shared with all stakeholders.

Test scenarios. In scenarios, testers break down the product's functionality and interface by modules and provide real-time status updates at all testing stages. A module can be described by a single statement, or require hundreds of statuses, depending on its size and scope.

Test cases. If the test scenario describes the object of testing (what), a scenario describes a procedure (how). These files cover step-by-step guidance, detailed conditions, and current inputs of a testing task. Test cases have their own kinds that depend on the type of testing — functional, UI, physical, logical cases, etc. Test cases compare available resources and current conditions with desired outcomes and determine if the functionality can be released or

not.

Traceability Matrix. This software testing documentation maps test cases and their requirements. All entries have their custom IDs — team members and stakeholders can track the progress of any tasks by simply entering its ID to the search.

External documentation collects information from inner documentation but also emphasize on providing a visual data representation — graphs, diagrams, etc. External reports — these documents collect information on test results and can describe an entire project or a particular piece of functionality.

Test summary report — the file with final test results and findings, presented to stakeholders.

Bug reports — such files keep track of newly encountered bugs and their fixes. We prefer to keep our bug documentation numbered, so it's easier to mention them in further documentation. Reports are concise and focus on offering tangible solutions. Sometimes, bug reports can only include issue description, if the team hasn't yet found the best approach to fixing the problem.

The combination of internal and external documentation is the key to a deep understanding of all testing processes. Although stakeholders typically have access to the majority of documentation, they mostly work with external files, since they are more concise and tackle tangible issues and results. Internal files, on the other hand, are used by team members to optimize the testing process.

Black Box testers don't care about Unit Testing. Their main goal is to validate the application against the requirements without going into the implementation details.

But as a curiosity or Out of the box thinking, have you ever wondered how developers test their own code? What method do they use to test before releasing code for

testing?

How is dev-testing important in an agile process? The answer to all this is Unit Testing. I want to educate you on the importance of Unit Testing so that development and testing teams can work more collaboratively to design, test and release an excellent application.

Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development).

Most of us might know the classic definition of Unit Testing

“Unit Testing is the method of verifying the smallest piece of testable code against its purpose.” If the purpose or requirement failed then the unit test has failed.

In simple words, Unit Testing means – writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.



Unit Testing is used to design robust software components that help maintain code and

eliminate the issues in code units. We all know the importance of finding and fixing defects in the early stage of the software development cycle. Unit Testing serves the same purpose.

Unit Testing is an integral part of the agile software development process. When a nightly build run unit test suite should run and report should be generated. If any of the unit tests have failed then the QA team should not accept that build for verification. If we set this as a standard process, many defects would be caught in the early development cycle, saving much testing time.

I know many developers hate to write unit tests. They either ignore or write bad unit test cases due to tight scheduled or lack of seriousness (yea they write empty unit tests, so 100% of them pass successfully ;-)). It's important to write good unit tests or don't write them at all. It's even more important to provide sufficient time and a supportive environment for real benefits.

- Testing can be done in the early phases of the software development lifecycle when other modules may not be available for integration
- Fixing an issue in Unit Testing can fix many other issues occurring in later development and testing stages
- Cost of fixing a defect found in Unit Testing is very less than the one found in the system or acceptance testing
- Code coverage can be measured
- Fewer bugs in the System and Acceptance testing

- Code completeness can be demonstrated using unit tests. This is more useful in the agile process. Testers don't get the functional builds to test until integration is completed. Code completion cannot be justified by showing that you have written and checked in the code. But running Unit tests can demonstrate code completeness.
- Expect robust design and development as developers write test cases by understanding the specifications first.
- Easily identify who broke the build
- Saves development time: Code completion may take more time but due to decreased defect count overall development time can be saved.

General

Unit Testing frameworks are mostly used to help write unit tests quickly and easily. Most of the programming languages do not support unit testing with the inbuilt compiler. Third- party open source and commercial tools can be used to make unit testing even more fun.

List of popular Unit Testing tools for different programming languages:

- Java framework – JUnit
- PHP framework – PHPUnit
- C++ frameworks – UnitTest++ and Google C++
- NET framework – NUnit
- Python framework – py.test

to write. The idea is simple, but like most simple things, the execution is hard. TDD requires a completely different mind set from what most people are used to and the tenacity to deal with a learning curve that may slow you down at first.

Functional Testing types include:

1. Unit Testing
2. Integration Testing
3. System Testing
4. Sanity Testing
5. Smoke Testing
6. Interface Testing
7. Regression Testing
8. Beta/Acceptance Testing

9. Non-functional Testing types include:

Performance Testing

- Load Testing
- Stress Testing
- Volume Testing
- Security Testing
- Compatibility Testing

- Install Testing
- Recovery Testing
- Reliability Testing
- Usability Testing
- Compliance Testing
- Localization Testing

Unit Testing

UNIT TESTING is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

A unit can be almost anything you want it to be -- a line of code, a method, or a class. Generally though, smaller is better. Smaller tests give you a much more granular view of how your code is performing. There is also the practical aspect that when you test very small units, your tests can be run fast; like a thousand tests in a second fast.

Black Box testers don't care about Unit Testing. Their main goal is to validate the

application against the requirements without going into the implementation details.

Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test driven development TDD or test-first development).

Most of us might know the classic definition of Unit Testing –

“Unit Testing is the method of verifying the smallest piece of testable code against its purpose.” If the purpose or requirement failed then the unit test has failed.

In simple words, Unit Testing means – writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.

Blackbox Testing

During functional testing, testers verify the app features against the user specifications. This is completely different from testing done by developers which is unit testing. It checks whether the code works as expected. Because unit testing focuses on the internal structure of the code, it is called the white box testing. On the other hand, functional testing checks app's functionalities without looking at the internal structure of the code, hence it is called black box testing. Despite how flawless the various individual code components may be, it is essential to check that the app is functioning as expected, when all components are combined. Here you can find a detailed comparison between functional testing vs unit testing.

Integration Testing

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults

in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

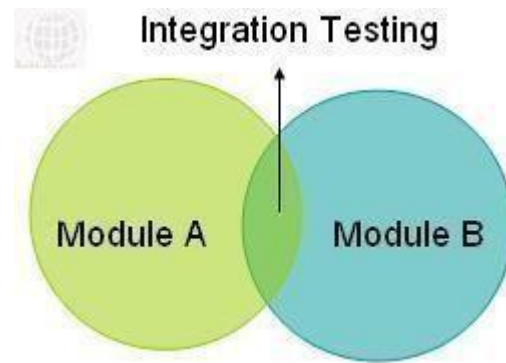
integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.

component integration testing: Testing performed to expose defects in the interfaces and interaction between integrated components. System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Integration tests determine if independently developed units of software work correctly when they are connected to each other. The term has become blurred even by the diffuse standards of the software industry, so I've been wary of using it in my writing. In particular, many people assume integration tests are necessarily broad in scope, while they can be more effectively done with a narrower scope.

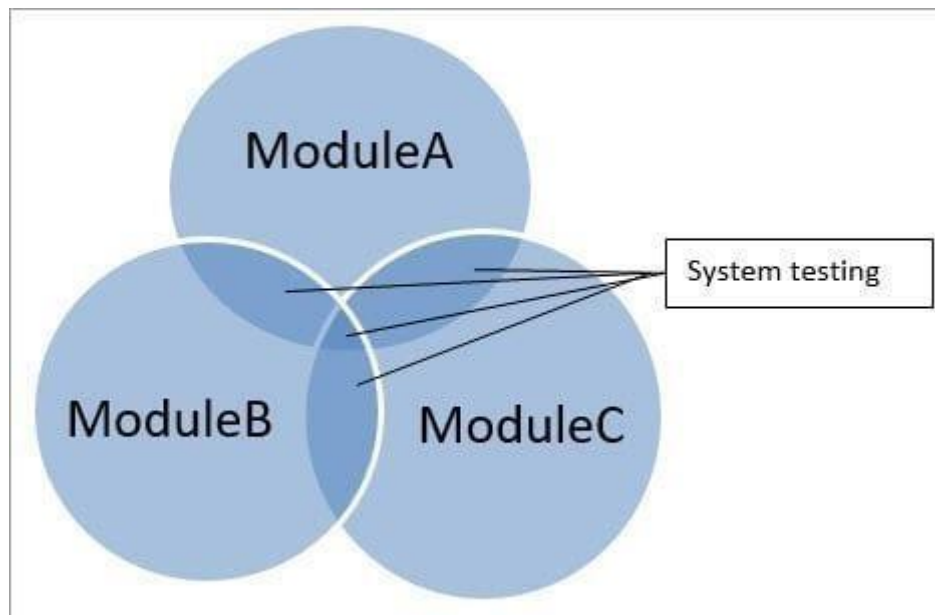
As often with these things, it's best to start with a bit of history. When I first learned about integration testing, it was in the 1980's and the waterfall was the dominant influence of software development thinking. In a larger project, we would have a design phase that would specify the interface and behavior of the various modules in the system. Modules would then be assigned to developers to program. It was not unusual for one programmer to be responsible for a single module, but this would be big enough that it could take months to build it. All this work was done in isolation, and when the programmer believed it was finished they would hand it over to QA for testing.

Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems. Integration testing is a key aspect of software testing.



System Testing

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not. System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product.



System testing is a method of monitoring and assessing the behaviour of the complete and fully-integrated software product or system, on the basis of pre-decided specifications and functional requirements. It is a solution to the question "whether the complete system functions in accordance to its pre-defined requirements?"

It's comes under black box testing i.e. only external working features of the software are evaluated during this testing. It does not requires any internal knowledge of the coding, programming, design, etc., and is completely based on users-perspective.

A black box testing type, system testing is the first testing technique that carries out the task of testing a software product as a whole. This System testing tests the integrated system and validates whether it meets the specified requirements of the client.

System testing is a process of testing the entire system that is fully functional, in order to ensure the system is bound to all the requirements provided by the client in the form of the functional specification or system specification documentation. In most cases, it is

done next to the Integration testing, as this testing should be covering the end-to-end system's actual routine. This type of testing requires a dedicated Test Plan and other test documentation derived from the system specification document that should cover both software and hardware requirements. By this test, we uncover the errors. It ensures that all the system works as expected. We check System performance and functionality to get a quality product. System testing is nothing but testing the system as a whole. This testing checks complete end-to-end scenario as per the customer's point of view. Functional and Non-Functional tests also done by System testing. All things are done to maintain trust within the development that the system is defect-free and bug-free. System testing is also intended to test hardware/software requirements specifications. System testing is more of a limited type of testing; it seeks to detect both defects within the "inter- assemblages".

Sanity Testing

Sanity Testing is done when as a QA we do not have sufficient time to run all the test cases, be it Functional Testing, UI, OS or Browser Testing. Sanity testing is a subset of regression testing. After receiving the software build, sanity testing is performed to ensure that the code changes introduced are working as expected. This testing is a checkpoint to determine if testing for the build can proceed or not.

The main purpose of this testing is to determine that the changes or the proposed functionality are working as expected. If the sanity test fails, the build is rejected by the testing team to save time and money. It is performed only after the build has cleared the smoke test and been accepted by the Quality Assurance team for further testing. The focus of the team during this testing process is to validate the functionality of the application and not detailed testing. Let's assume that in an e-commerce project, there

are mainly five modules- like login page, home page, user profile page, new user

creation, etc. There is a defect in the login page when the password field accepts less than four alpha numeric characters and the requirement mentions that this password field should not be below eight characters. So, the bug is reported by the testing team to the development team to resolve it. When the development team fixes the defect and passes it to the testing team. Then the testing team checks if the changes done are working fine. It also checks if it does not have an impact on other related functionalities. Now there is a functionality to update the password in the user profile page. As part of the sanity check, we would need to validate the login page as well as the profile page to ensure that the checks are working fine at both the places. Sanity tests are generally performed on build where the production deployment is required immediately - like a critical bug fix.

Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is also called as “Day 0” check. It is done at the “build level”.

It helps not to waste the testing time to simply testing the whole application when the key features don't work or the key bugs have not been fixed yet. Here our focus will be on primary and core application work flow.

To conduct smoke testing, we don't write test cases. We just pick the necessary test cases from already written test cases.

Do we really write test cases for all testing types? Here in this article, we have given clear idea on choosing testing types to write test cases.

As mentioned earlier, here in Smoke Testing, our main focus will be on core application work flow. So we pick the test cases from our test suite which cover major functionality of the application. In general, we pick minimal number of test cases that won't take more than half an hour to execute.

After receiving a Software build with the minor issues fixes in code or functionality, Sanity testing is carry out to check whether the bugs reported in previous build are fixed & there is regression introduced due to these fixes i.e. not breaking any previously working functionality. The main aim of Sanity testing to check the planned functionality is working as expected. Instead of doing whole regression testing the Sanity testing is perform. Sanity tests helps to avoid wasting time and cost involved in testing if the build is failed. Tester should reject the build upon build failure.

After completion of regression testing the Sanity testing is started to check the defect fixes & changes done in the software application is not breaking the core functionality of the software. Typically this is done nearing end of SDLC i.e. while releasing the software. You can say that sanity testing is a subset of acceptance testing. We can also say Tester Acceptance Testing for Sanity testing.

Regression Testing

Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to make sure the product works fine with new functionality, bug fixes or any change in the existing feature. Previously executed test cases are re-executed in order to verify the impact of change.

Regression Testing is a Software Testing type in which test cases are re-executed in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs.

This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix.

For regression testing to be effective, it needs to be seen as one part of a comprehensive testing methodology that is cost-effective and efficient while still incorporating enough variety—such as well-designed frontend UI automated tests alongside targeted unit testing, based on smart risk prioritization—to prevent any aspects of your software applications from going unchecked. These days, many Agile work environments employing workflow practices such as XP (Extreme Programming), RUP (Rational Unified Process), or Scrum appreciate regression testing as an essential aspect of a dynamic, iterative development and deployment schedule. But no matter what software development and quality-assurance process your organization uses, if you take the time to put in enough careful planning up front, crafting a clear and diverse testing strategy with automated regression testing at its core, you can help prevent projects from going over budget, keep your team on track, and, most importantly, prevent unexpected bugs from damaging your products and your company's bottom line.

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size. The process incorporates “performance” indicators such as:

- Browser, page, and network response times
- Server request processing times
- Acceptable concurrent user volumes

- Processor memory consumption; number and type of errors that might be encountered with app

Performance testing gathers all the tests that verify an application's speed, robustness, reliability, and correct sizing. It examines several indicators such as a browser, page and network response times, server query processing time, number of acceptable concurrent users architected, CPU memory consumption, and number/type of errors which may be encountered when using an application.

Performance testing is the testing that is performed to ascertain how the components of a system are performing under a certain given situation. Resource usage, scalability, and reliability of the product are also validated under this testing. This testing is the subset of performance engineering, which is focused on addressing performance issues in the design and architecture of a software product.



Software Performance testing is type of testing perform to determine the performance

of system to major the measure, validate or verify quality attributes of the system like

responsiveness, Speed, Scalability, Stability under variety of load conditions. The system is tested under a mixture of load conditions and check the time required responding by the system under varying workloads. Software performance testing involves the testing of application under test to ensure that application is working as expected under variety of load conditions. The goal of performance testing is not only find the bugs in the system but also eliminate the performance bottlenecks from the system.

Load Testing is type of performance testing to check system with constantly increasing the load on the system until the time load is reaches to its threshold value. Here Increasing load means increasing number of concurrent users, transactions & check the behavior of application under test. It is normally carried out underneath controlled environment in order to distinguish between two different systems. It is also called as “Endurance testing” and “Volume testing”. The main purpose of load testing is to monitor the response time and staying power of application when system is performing well under heavy load. Load testing comes under the Non Functional Testing & it is designed to test the non-functional requirements of a software application.

Load testing is perform to make sure that what amount of load can be withstand the application under test. The successfully executed load testing is only if the specified test cases are executed without any error in allocated time.

Simple examples of load testing:

- Testing printer by sending large job.
- Editing a very large document for testing of word processor
- Continuously reading and writing data into hard disk.
- Running multiple applications simultaneously on server.
- Testing of mail server by accessing thousands of mailboxes

- In case of zero-volume testing & system fed with zero load.

Chapter 8

Conclusion & Feature Enhancement

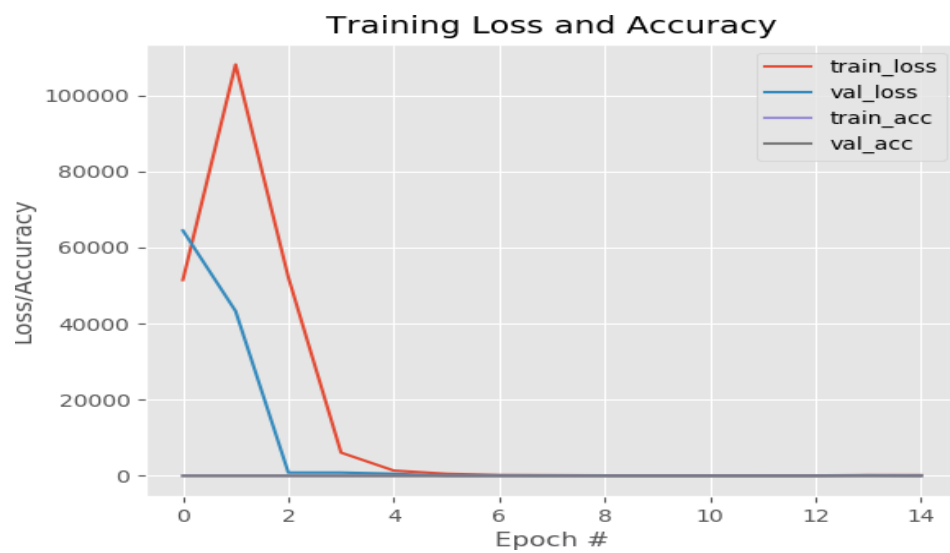
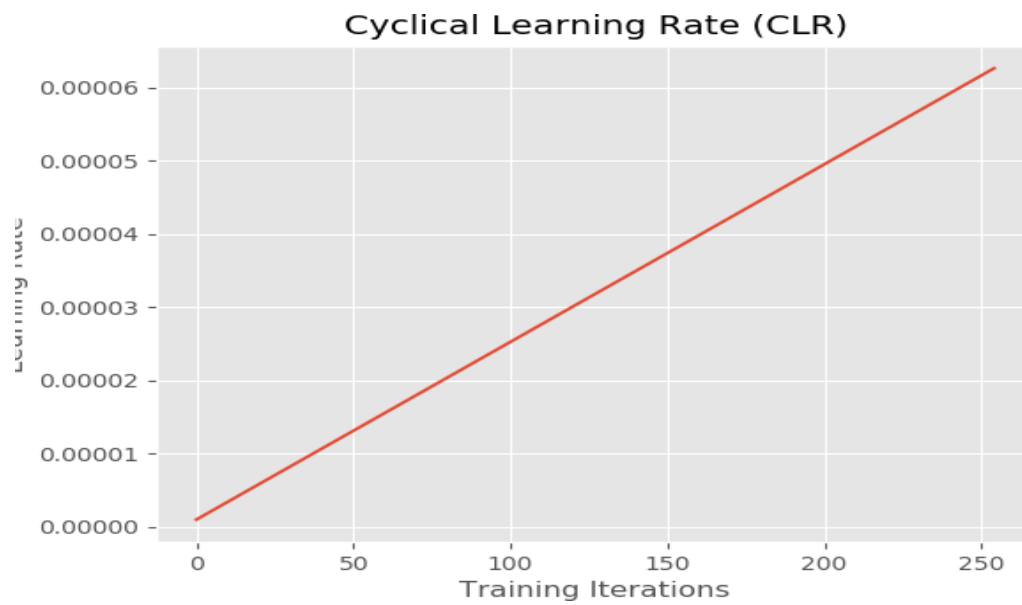
Conclusion

In this paper, we proposed a precursory pattern based feature extraction method with the selected approach for accurate earthquake prediction. From the experimental results, the performance of all comparison algorithms has indeed improved by using algorithm for feature selection. However, with the proposed precursory pattern based feature extraction method still get the best performance since that it can do feature selection automatically according to the information gain of each feature in the process of tree building. In summary, the proposed precursory pattern based feature extraction method with the selected model is a promising method for solving the task of earthquake prediction.

Future Enhancement

The future research work includes earthquake prediction algorithm and model design and earthquake prediction practice based on the feature set, algorithm, and model

Appendix – I Screenshots



Appendix – II Sample Coding

```
import os
import pickle

import sys

import cv2

import argparse

import numpy as np

import matplotlib

matplotlib.use("Agg"
)

import matplotlib.pyplot as plt

from imutils import paths

import tempfile

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.applications import VGG16

from tensorflow.keras.layers import Dropout

from tensorflow.keras.layers import Flatten

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import Input

from tensorflow.keras.models import Model

from tensorflow.keras.optimizers import SGD

from sklearn.preprocessing import LabelBinarizer

from tensorflow.keras.callbacks import LambdaCallback

from tensorflow.keras import backend as K

from tensorflow.keras.callbacks import *

from tensorflow.keras import backend as K

from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report

DATASET_PATH = "Dataset"
```

```
CLASSES = ["Earthquake",  
"Flood"] TRAIN_SPLIT = 0.75  
VAL_SPLIT = 0.1
```

```

TEST_SPLIT =
0.25 MIN_LR =
1e-6 MAX_LR =
1e-4 BATCH_SIZE
= 32
STEP_SIZE = 8
CLR_METHOD = "triangular"
NUM_EPOCHS = 48
class LearningRateFinder:
    def init(self, model, stopFactor=4, beta=0.98):
        self.model = model
        self.stopFactor =
        stopFactor self.beta = beta
        self.lrs = []
        self.losses =
        [] self.lrMult =
        1
        self.avgLoss = 0
        self.bestLoss = 1e9
        self.batchNum = 0
        self.weightsFile =
        None
    def reset(self):
        self.lrs = []
        self.losses =
        [] self.lrMult =
        1
        self.avgLoss = 0
        self.bestLoss = 1e9
        self.batchNum = 0
        self.weightsFile =
        None
    def is_data_iter(self, data):

```

```
iterClasses = ["NumpyArrayIterator",  
               "DirectoryIterator", "DataFramelIterator", "Iterator",  
               "Sequence"]  
  
return data.__class__.__name__in  
iterClasses  
  
def on_batch_end(self, batch, logs):  
    lr = K.get_value(self.model.optimizer.lr)  
    self.lrs.append(lr)  
    l = logs["loss"]
```

```

self.batchNum += 1
self.avgLoss = (self.beta * self.avgLoss) + ((1 - self.beta) *
l) smooth = self.avgLoss / (1 - (self.beta ** self.batchNum))
self.losses.append(smooth)
stopLoss = self.stopFactor * self.bestLoss
if self.batchNum > 1 and smooth > stopLoss:
    self.model.stop_training = True
    return
if self.batchNum == 1 or smooth <
    self.bestLoss: self.bestLoss = smooth
lr *= self.lrMult
K.set_value(self.model.optimizer.lr, lr)
def find(self, trainData, startLR, endLR, epochs=None,
stepsPerEpoch=None, batchSize=32,
sampleSize=2048, verbose=1):
self.reset()
useGen = self.is_data_iter(trainData)
if useGen and stepsPerEpoch is None:
    msg = "Using generator without supplying
stepsPerEpoch" raise Exception(msg)
elif not useGen:
    numSamples = len(trainData[0])
    stepsPerEpoch = np.ceil(numSamples / float(batchSize))
if epochs is None:
    epochs = int(np.ceil(sampleSize /
float(stepsPerEpoch))) numBatchUpdates = epochs *
stepsPerEpoch
self.lrMult = (endLR / startLR) ** (1.0 /
numBatchUpdates) self.weightsFile =
tempfile.mkstemp()[1]
self.model.save_weights(self.weightsFile)
origLR = K.get_value(self.model.optimizer.lr)
K.set_value(self.model.optimizer.lr, startLR)

```

```
callback = LambdaCallback(on_batch_end=lambda batch,
    logs: self.on_batch_end(batch, logs))
if useGen:
    self.model.fit
    (
```

```

        x=trainData,
        steps_per_epoch=stepsPerEpoch
        , epochs=epochs,
        verbose=verbose,
        callbacks=[callback])
else:
    self.model.fit(
        x=trainData[0], y=trainData[1],
        batch_size=batchSize,
        epochs=epochs,
        callbacks=[callback],
        verbose=verbose)
self.model.load_weights(self.weightsFile)
K.set_value(self.model.optimizer.lr, origLR)
def plot_loss(self, skipBegin=10, skipEnd=1,
    title=""): lrs = self.lrs[skipBegin:-skipEnd]
    losses =
    self.losses[skipBegin:-skipEnd]
    plt.plot(lrs, losses)
    plt.xscale("log")
    plt.xlabel("Learning Rate (Log
    Scale)") plt.ylabel("Loss")
    if title != "":
        plt.title(title)
class CyclicLR(Callback):
    def __init__(self, base_lr=0.001, max_lr=0.006, step_size=2000., mode='triangular',
        gamma=1., scale_fn=None, scale_mode='cycle'):
        super(CyclicLR, self).__init__()

    self.base_lr = base_lr
    self.max_lr = max_lr
    self.step_size =
    step_size self.mode =

```

```
mode self.gamma =  
gamma  
if scale_fn == None:  
    if self.mode == 'triangular':
```

```

        self.scale_fn = lambda x: 1.

        self.scale_mode = 'cycle'
    elif self.mode == 'triangular2':
        self.scale_fn = lambda x: 1/(2.**x-1))

        self.scale_mode = 'cycle'
    elif self.mode == 'exp_range':
        self.scale_fn = lambda x:
            gamma**(x) self.scale_mode =
                'iterations'
    else:
        self.scale_fn = scale_fn

        self.scale_mode =
            scale_mode
    self.clr_iterations = 0.
    self.trn_iterations =
    0. self.history = {}

    self._reset()

def _reset(self, new_base_lr=None, new_max_lr=None,
            new_step_size=None):
    if new_base_lr != None:
        self.base_lr =
            new_base_lr
    if new_max_lr != None:
        self.max_lr =
            new_max_lr
    if new_step_size != None:
        self.step_size = new_step_size
    self.clr_iterations = 0.

def clr(self):
    cycle = np.floor(1+self.clr_iterations/(2*self.step_size))

```

```
x = np.abs(self.clr_iterations/self.step_size - 2*cycle + 1)
if self.scale_mode == 'cycle':
    return self.base_lr + (self.max_lr-self.base_lr)*np.maximum(0, (1-x))*self.scale_fn(cycle)
else:
    return self.base_lr + (self.max_lr-self.base_lr)*np.maximum(0, (1-x))*self.scale_fn(self.clr_iterations)
```

```

def on_train_begin(self, logs={}):
    logs = logs or {}

    if self.clr_iterations == 0:
        K.set_value(self.model.optimizer.lr, self.base_lr)
    else:
        K.set_value(self.model.optimizer.lr, self.clr())

def on_batch_end(self, epoch, logs=None):

    logs = logs or {}
    self.trn_iterations +=
    1
    self.clr_iterations += 1

    self.history.setdefault('lr', []).append(K.get_value(self.model.optimizer.lr))
    self.history.setdefault('iterations', []).append(self.trn_iterations)

    for k, v in logs.items():
        self.history.setdefault(k, []).append(v)

    K.set_value(self.model.optimizer.lr, self.clr())
    imagePaths = list(paths.list_images(DATASET_PATH))
    data = []
    labels = []
    for imagePath in imagePaths:
        label = imagePath.split(os.path.sep)[-2]
        image = cv2.imread(imagePath)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        image = cv2.resize(image, (224, 224))
        data.append(image)
        labels.append(label)
    print("[INFO] processing data...")

```

```
data = np.array(data,  
dtype="float32") labels =  
np.array(labels)  
lb = LabelBinarizer()
```

```

labels = lb.fit_transform(labels)
(trainX, testX, trainY, testY) = train_test_split(data, labels,
    test_size=TEST_SPLIT, random_state=42)
(trainX, valX, trainY, valY) = train_test_split(trainX, trainY,
    test_size=VAL_SPLIT, random_state=84)
aug = ImageDataGenerator(
    rotation_range=30,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest")
baseModel = VGG16(weights="imagenet",
    include_top=False, input_tensor=Input(shape=(224, 224,
    3)))
headModel = baseModel.output
headModel = Flatten(name="flatten")(headModel)
headModel = Dropout(0.5)(headModel) headModel = Dense(512,
activation="relu")(headModel) headModel = Dropout(0.5)(headModel)
headModel = Dense(len(CLASSES), activation="softmax")(headModel)
model = Model(inputs=baseModel.input, outputs=headModel)
print("[INFO] compiling model...")
opt = SGD(lr=MIN_LR, momentum=0.9)
model.compile(loss="sparse_categorical_crossentropy", optimizer=opt,
    metrics=["accuracy"])
print("[INFO] finding learning
rate...") lrf =
LearningRateFinder(model) lrf.find(
    aug.flow(trainX, trainY, batch_size=BATCH_SIZE),
    1e-10, 1e+1,
    stepsPerEpoch=20,
    epochs=10,

```

```
    batchSize=BATCH_SIZE)  
print("[INFO] learning rate finder complete")  
print("[INFO] examine plot and adjust learning rates before training")
```

```

stepSize = STEP_SIZE * (trainX.shape[0] // BATCH_SIZE)
clr = CyclicLR(
mode=CLR_METHOD
    , base_lr=MIN_LR,
    max_lr=MAX_LR,
    step_size=stepSize)
print("[INFO] training
network...") H =
model.fit_generator(
    aug.flow(trainX, trainY, batch_size=BATCH_SIZE),
    validation_data=(valX, valY),
    steps_per_epoch=trainX.shape[0]//3 // BATCH_SIZE,

    epochs=15,
    callbacks=[clr],
    verbose=1)
print("[INFO] evaluating network...")
predictions = model.predict(testX, batch_size=BATCH_SIZE)
print(classification_report(testY.argmax(axis=1),
    predictions.argmax(axis=1), target_names=CLASSES))
MODEL_PATH = os.path.sep.join(["earthquake_disaster.model"])
print("[INFO] serializing network to '{}...'".format(MODEL_PATH))
model.save(MODEL_PATH)
TRAINING_PLOT_PATH = os.path.sep.join(["training_plot.png"])

N = np.arange(0, 15)
plt.style.use("ggplot")
plt.figure()
plt.plot(N, H.history["loss"], label="train_loss")
plt.plot(N, H.history["val_loss"], label="val_loss")
plt.plot(N, H.history["acc"], label="train_acc")
plt.plot(N, H.history["val_acc"], label="val_acc")
plt.title("Training Loss and Accuracy")

```

```
plt.xlabel("Epoch #")  
plt.ylabel("Loss/Accuracy")  
plt.legend(loc="upper  
right")
```

```
plt.savefig(TRAINING_PLOT_PATH)
```

```
CLR_PLOT_PATH = os.path.sep.join(["clr_plot.png"])
```

```
N = np.arange(0, len(clr.history["lr"]))
```

```
plt.figure()
```

```
plt.plot(N, clr.history["lr"])
```

```
plt.title("Cyclical Learning Rate
```

```
(CLR)") plt.xlabel("Training Iterations")
```

```
plt.ylabel("Learning Rate")
```

```
plt.savefig(CLR_PLOT_PATH)
```

References

- S.M. Potirakis, A. Schekotov, T. Asano and M. Hayakawa. "Natural time analysis on the ultra-low frequency magnetic field variations prior to the 2016 Kumamoto (Japan) earthquakes", *Journal of Asian Earth Sciences*, 154(4): 419- 427, 2018
- S.M. Potirakis, M. Hayakawa and A. Schekotov. "Fractal analysis of the ground- recorded ULF magnetic fields prior to the 11 March 2011 Tohoku earthquake ($M_W = 9$): discriminating possible earthquake precursors from space-sourced disturbances", *Natural Hazards*, 85(1): 59-86, 2017
- X. Wang, S.Yong, B. Xu and J. Huang. "Research and implementation of Multi- component seismic monitoring system AETA", *ACTA Scientiarum Naturelium Universitatis Pekinensis*, 54(3): 487-494, 2018
- S.Yong, X.Wang, R.Pang and J.Zeng. "Development of inductive magnetic sensor for Multi-component seismic monitoring system AETA", *ACTA Scientiarum Naturelium Universitatis Pekinensis*, 54(3): 495-501, 2018
- Y. Lu, E. Sawada, and G. Chakraborty, "Earthquake aftershock prediction based solely on seismograph data," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2019, pp. 1-6.
- E. Florido, F. Martínez-Álvarez, A. Morales-Esteban, J. Reyes, and J. L. Aznarte- Mellado, "Detecting precursory patterns to enhance earthquake prediction in Chile," *Comput. Geosci.*, vol. 76, pp. 112-120, Mar. 2015.
- E. Florido, G. Asencio Cortés, J. L. Aznarte, C. Rubio-Escudero, and F. Martínez Álvarez, "A novel tree-based algorithm to discover seismic patterns in earthquake catalogs," *Comput. Geosci.*, vol. 115, pp. 96-104, Jun. 2018.
- M. N. Shodiq, A. R. Barakbah, and T. Harsono, "Spatial analysis of earthquake distribution with automatic clustering for prediction of earthquake seismicity in Indonesia," in *Proc. KCIC*, 2015, pp. 24-26.
- M. N. Shodiq, D. H. Kusuma, M. G. Rifqi, A. R. Barakbah, and T. Harsono, "Neural network for earthquake prediction based on automatic clustering in Indonesia," *JOIV, Int. J. Informat. Vis.*, vol. 2, no. 1, p. 37, Feb. 2018.

S. Tombaz, K. W. Sung, and J. Zander, "On metrics and models for energyefficiency in wireless access networks," IEEE Wireless Commun. Lett., vol. 3, no. 6, pp. 649– 652, Dec. 2014. [7] S. Althunibat, M. Di Renzo, and F. G

J. Zheng, Y. Cai, X. Shen, Z. Zheng, and W. Yang, "Green energy optimization in energy harvesting wireless sensor networks," IEEE Commun. Mag., vol. 53, no. 11, pp. 150–157, Nov. 2015.

