ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Πτυχιακή Εργασία

«Ανάπτυξη μοντέλου μηχανικής/βαθιάς μάθησης για την εκτίμηση της πιθανότητας να συμβεί ισχυρός σεισμός με βάση τις καταγραφές δικτύου επίγειων σταθμών VLF/LF»

Φοιτητής: Δημήτριος Παναγιώτης Μπιτχαβάς
ΑΜ: 50106166

Επιβλέπων Καθηγητής

Στυλιανός Ποτηράκης
Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, Φεβρουάριος 2025

UNIVERSITY OF WEST ATTICA

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Degree Thesis

Development of a machine/deep learning model for the estimation of the probability for a strong earthquake to occur, based on the recordings of a VLF/LF ground-based stations network

Student: Dimitrios Panagiotis Bitchavas
Registration Number: 50106166

Supervisor

Stylianos Potirakis
Professor

ATHENS-EGALEO, February 2025

Η Πτυχιακή Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Σ. Ποτηράκης, Καθηγητής	Μ. Ραγκούση, Καθηγήτρια	Ε. Κυριάκης-Μπιτζάρος, Καθηγητής
Digitally signed by STYLIANOS POTIRAKIS Date: 2025.02.22 20:51:41 +02'00' (Υπογραφή)	MARIA RAGKOUSI Digitally signed by MARIA RAGKOUSI Date: 2025.02.20 14:46:29 +02'00' (Υπογραφή)	EFSTATHIOS KYRIAKIS-BITZAROS 2025.02.22 23:26:07 +02'00' (Υπογραφή)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Μπιτχαβάς Δημήτριος-Παναγιώτης του Κωνσταντίνου, με αριθμό μητρώου 50106166 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Ο/Η Δηλών/ούσα

Δημήτριος Παναγιώτης Μπιτχαβάς

Περίληψη

Η πρόβλεψη των σεισμών αποτελεί μία από τις σημαντικότερες, αλλά ταυτόχρονα εξαιρετικά δύσκολες προκλήσεις στον τομέα της γεωφυσικής. Η πρόβλεψη των σεισμών έχει τη δυνατότητα να μειώσει τους κινδύνους και να βελτιώσει τη διαχείριση καταστροφών, να βοηθήσει κατά τη διάρκεια μιας καταστροφής και να συμβάλει στον μετριασμό των επιπτώσεων. Στην παρούσα μελέτη, χρησιμοποιούνται τεχνικές μηχανικής μάθησης (ML) και βαθιάς μάθησης (DL) για την ανάλυση δεδομένων χρονοσειρών με σκοπό την πρόβλεψη σεισμικών φαινομένων. Η έρευνα αυτή μελετά την ανάπτυξη μεθόδων στην υπολογιστική επεξεργασία, προκειμένου να ανιχνευθούν δείκτες σεισμικών δεδομένων και να αυξηθεί η ακρίβεια της πρόβλεψης.

Η ανάλυση ξεκινά με την παρουσίαση της σημασίας και των στόχων των μεθόδων τεχνητής νοημοσύνης (AI) για την πρόβλεψη σεισμών, ακολουθούμενη από μια στατιστική και οπτική παρουσίαση των χρησιμοποιούμενων συνόλων δεδομένων. Αυτό γίνεται για να εξηγηθούν οι μετρικές και τα μέτρα που λαμβάνονται υπόψη, καθώς και οι πιο σημαντικές τάσεις ή ακραίες τιμές που υπάρχουν στα συλλεχθέντα σεισμικά δεδομένα. Η έρευνα στοχεύει στην παροχή μιας επεξήγησης των θεωριών και τεχνικών που χρησιμοποιούνται για την αναγνώριση, την ανάλυση και την ταυτοποίηση των βασικών προτύπων των εισερχόμενων σεισμικών δεδομένων, καθώς και όλων των στοιχείων που καθιστούν ένα πρότυπο πιθανό προάγγελο ενός επικείμενου σεισμού.

Η ροή εργασίας περιεγράφηκε λεπτομερώς όσον αφορά την προ επεξεργασία δεδομένων, την επιλογή μοντέλων και τις διαδικασίες που περιλαμβάνουν τη χρήση μη ισορροπημένων συνόλων δεδομένων. Δόθηκε ιδιαίτερη έμφαση στη μηχανική χαρακτηριστικών (feature engineering) και την επιλογή χαρακτηριστικών (feature selection) για την ανάπτυξη ενός πιο αποδοτικού μοντέλου. Χρησιμοποιήθηκαν οκτώ μοντέλα ML και DL, μεταξύ των οποίων δίκτυα Long Short-Term Memory (LSTM), Μονάδες Gated Recurrent (GRU) και Συνελικτικά Νευρωνικά Δίκτυα (CNN). Η βελτιστοποίηση των υπερπαραμέτρων πραγματοποιήθηκε μέσω grid search, με στόχο τη βελτίωση της ακρίβειας πρόβλεψης των μοντέλων.

Για την αξιολόγηση της αποτελεσματικότητας των μεθόδων, εξετάστηκαν τα αποτελέσματα πριν και μετά τη βελτιστοποίηση, λαμβάνοντας υπόψη την ακρίβεια και άλλες μετρικές, καθώς και τη σταθερότητα των αποτελεσμάτων. Το άρθρο ολοκληρώνεται με ένα συμπέρασμα, στο οποίο παρουσιάζονται τα πιο σημαντικά αποτελέσματα, τα οφέλη που προσφέρουν οι μέθοδοι τεχνητής νοημοσύνης για την πρόβλεψη σεισμών, καθώς και τα βέλτιστα μοντέλα και οι παράμετροι που προέκυψαν από τα πειράματα.

Abstract

The prediction of earthquakes stands as one of the most important, yet extremely difficult tasks to accomplish in the field of geophysics. The prediction of earthquakes has the ability to lessen risks and improve disaster management, assist during the time of catastrophe and mitigate risk. In this study machine learning (ML) and deep learning (DL) techniques are used to analyze time-series data in order to forecast seismic events. This research is proposed to develop methods in computing to detect indicators of seismic data in order to increase the prediction accuracy.

This analysis begins with the importance and goals of AI methods towards earthquake prediction, followed by a statistical and visual presentation of the data sets utilized to explain the metrics and measures we are taking into consideration along with the most important trends or outliers that are present in the gathered seismic data. The research tries to provide an important explanation of the theories and techniques to recognize, explain and identify the basic data patterns of the incoming seismic data and all the elements that make the pattern a precursor of a possible Earthquake.

The workflow was described in detail with regards to data preprocessing, model selection, and those processes which involve the usage of imbalanced datasets. More attention was given to feature engineering and feature selection for developing a better performing model. Eight ML and DL models were used which included Long Short-Term Memory networks (LSTM), Gated Recurrent Units (GRU), and Convolutional Neural Networks (CNN). Hyper parameter tuning was done through grid search with the goal of increasing the prediction accuracy in the models.

To understand the effectiveness of the methods, both the pre-optimization and post-optimization results were checked for accuracy and other metrics along with how robust the results are. The paper closes with a conclusion containing the most important results outlining the benefits that AI based methods offer to seismic predictions and the best models and parameters obtained from experiments.

Key Words

Word	**Definition**
Earthquake Prediction	The process of forecasting seismic events using scientific methods
Seismic Activity	The frequency, type, and size of earthquakes occurring in a specific region over time
Machine Learning	A subset of AI that enables systems to learn from data and make predictions without explicit programming
Deep Learning	A specialized field of ML using neural networks with multiple layers to model complex patterns in data
Time Series Analysis	The study of data points collected or recorded at successive time intervals to detect trends and patterns
LSTM	A type of recurrent neural network (RNN) designed to handle long-term dependencies in sequential data
GRU	A variant of RNN that uses gating mechanisms to efficiently capture dependencies in time series data
CNN	A deep learning architecture mainly used for image processing but also effective in analyzing spatial and sequential data
RNN	A neural network designed for processing sequential data by maintaining memory of previous inputs
Feature Selection	The process of selecting the most relevant variables in a dataset to improve model performance and efficiency
SHAP	An interpretability method that explains the impact of each feature on a model's predictions
RFE	A feature selection method that recursively removes less important features to improve model accuracy
Receiver Transmitter	A system that transmits and receives signals, often used in sensor networks for data collection
Imbalanced Data	A dataset where some classes are significantly underrepresented, leading to biased model predictions
Feature Engineering	The process of creating new input variables to improve ML model performance
Anomaly	The identification of unusual patterns or outliers in data that may indicate

Detection	significant events
Signal Processing	Techniques for analyzing, modifying, and interpreting signals from sensors or time series data
Time Series Forecasting	The use of statistical or ML methods to predict future values based on past time-dependent data
Disaster Preparedness	Strategies and measures taken to mitigate the impact of natural disasters before they occur
Risk Assessment	The evaluation of potential risks associated with natural disasters and their impact
Predictive Modeling	The process of creating models to predict future events based on historical data

List of tables:

Table 1: Grid Search parameters.. 43
Table 2: Modelling Pre Optimization results 44
Table 3: Modelling Post Optimization results... 45
Table 4: Best Models per metric 50

List of figures:

Figure 1: Map of the wider area around Japan showing the 8 subionospheric propagation paths of the EAL VLF network. [16] 18
Figure 2: Map of the wider area around Japan showing the 11 subionospheric propagation paths of the Hi-Sem VLF network [16] 18
Figure 3: Data sample of the NFM data utilized 19
Figure 4: Record count pre receiver for NFM 20
Figure 5: Timeseries visualization of the different metrics for NFM 20
Figure 6: Distribution for the different metrics for NFM 21
Figure 7: Boxplots for the different NFM Metrics per receiver, the dots outside the boxplots represent outliers per receiver 21
Figure 8: Seasonal variation Data Sample 22
Figure 9: Record count pre receiver for daylength 22
Figure 10: Distribution for the different metrics for Daylength 23
Figure 11: Boxplots for the different Daylength Metrics per receiver 23
Figure 12: Earthquake Data Sample. 24
Figure 13: Record count pre receiver for Eearthquake data 24
Figure 14: Commulative data overview 26
Figure 15: Chart of values daylight in relation to positive earthquake positive events. 27
Figure 16: Pre vs Post INOS Positive EQ events ratio. 29
Figure 17:Per vs Post INOS Optimization WEQ Positive events 30
Figure 18: Pre vs Post ESPO Optimization Positive EQ ratio 31
Figure 19: Pre vs Post ESPO Optimization Positive EQ events 32
Figure 20: Pre vs Post Optimization Positive EQ ratio of the combination of INOS and ESPO 32
Figure 21: F1 Score Model comparison 46
Figure 22: F1 Score Model Improvement 47
Figure 23: AUC Score Model Improvement 47
Figure 24: Post Optimization Model Comparison Heatmap 48
Figure 25: Pre vs Post Correlation Heatmap 49
Figure 26: Pre vs Post F1 Score Distribution range 49
Figure 27: Distribution range for different ML metrics across Models 50

List of Equations

(1) Running mean time series…………………………………………………………………..16
(2) Residual variation in amplitude………………………………………………………….16
(3) Trend…………………………………………………………………………………..17
(4) Dispersion……………………………………………………………………………..17
(5) Nighttime Fluctuation…………………………………………………………………17

Table of Contents

Contents

Περίληψη 5
Abstract 6
Key Words 7
List of tables: 9
List of figures: 10
Table of Contents 12
1. Introduction 14
2. Theory 15
   2.1. Introduction to Earthquakes 15
   2.2. Terminator Time Method (TTM) 15
   2.3. Nighttime Fluctuation Method (NFM) 16
3. Data 18
4. Strategic Approaches for Data Preparation and Model Optimization 25
   4.1 Data analysis - Data cleaning 26
   4.2. Balancing techniques 27
      4.2.1. INOS (Interpolation-Based Oversampling) 28
      4.2.2. ESPO (Edge-Based Synthetic Minority Oversampling) 30
      4.2.3. Combination of INOS and ESPO 31
   4.3. Feature selection 33
      4.3.1. SHAP (SHapley Additive exPlanations) values 33
      4.3.2. Recursive Feature Elimination (RFE) 34
5. Modeling 36
   5.1. Long Short-Term Memory (LSTM) 36
   5.2. Gated Recurrent Unit (GRU) 37
   5.3. Convolutional Neural Network (CNN) 38
   5.4. Recurrent Neural Network (RNN) 38
   5.5. Random Forest 39
   5.6. K-Nearest Neighbors 39
   5.7. XGBoost 40
   5.8. Gaussian Process 40
6. Model Optimization Methods 41
   6.1. Hyperparameter tuning 41

6.2. Grid Search 42
7. Results 44
7.1. Pre-Optimization Results 44
7.2. Post-Optimization Results 45
8. Summary 46
References 51

1. Introduction

Earthquakes are one of the most powerful natural forces on the planet, capable of taking countless lives whilst causing grave destruction to economies and the environment. There have been various researches in the technology and the causation of earthquakes that have improved the monitoring and management of seismic activities, however, there is still room for improvement in predicting earthquakes reliably and successfully.

Artificial intelligence as well as machine learning changed the model of prediction itself in numerous fields or areas, and the existence of big unorganized data has opened the opportunity to search for weak dependencies that are impossible for people to identify. To improve the way we predict earthquakes, deep learning, specifically time series analysis, became a suitable solution, as it was able to learn how to connect and cross reference correlations over time within the earthquake data.

This thesis investigates the development of a machine/deep learning models for the estimation of the probability for a strong earthquake to occur based on 6 years (2014-2020) of very low frequency (VLF) subionospheric propagation data from 19 VLF receivers in Japan. The data was transmitted via the JJI VLF transmitter, which has a frequency of 22.2 kHz. The analysis included earthquakes (ML $\geq$ 4.5, depth $\leq$ 50 km) that occurred during the same time frame within the area of Japan together with the VLF data.

The use of ionospheric anomalies for the prediction of earthquakes through the nighttime fluctuation method (NFM) and terminator time method (TTM) is still a controversial subject among the scientific community. So there is a strong need to evaluate the effectiveness and limitations of these approaches in order to contribute to the ongoing dialogue of creating a way to accurately predict high magnitude earthquake occurrences. [16]

2. Theory

2.1. Introduction to Earthquakes

Earthquakes are abrupt violent shakes of the ground that are usually caused by movements along faults or volcanic activities. They happen when tectonic plates move under great stress, releasing energy in the Earth's crust which creates seismic waves. During the event, the energy that is released is measured by the Richter or moment magnitude scales and quantifies the energy released. Events of high magnitude can destroy human life, impact infrastructure, and heavily disrupt ecosystems.

With the advances of modern technology, predicting earthquakes is still a huge challenge that many scientists face. One of the main reasons is the variety of factors that play a role during an earthquake, along with the lacking precursors. Because of this, earthquakes are still regarded as inaccurate natural processes that are caused by tectonic movements.

In our efforts to accurately predict this natural process we are employing different methods and tools to create a system that can identify the precursors of an earthquake and provide warning to the areas affected.

2.2. Terminator Time Method (TTM)

The Terminator Time Method (TTM) is a Method that aims to examine the alterations experienced by the ionosphere during the inferior and superior transitions, commonly referred to as 'terminator times.' These periods are characterized by alterations of the ionospheric activities, due to the effects of the sun on the Earth's atmosphere, during these periods experiencing rapid changes. It is during these transitions that TTM are utilized to observe how earthquakes and other extreme phenomena do alter these periods of amplitude dips. Moderate to significant seismic activities may also induce minor shifts in the time of transitions/terminator times. Studying these shifts using TTM, we may have an opportunity to detect various upcoming seismic activities and hence contribute to research of how to accurately predict earthquakes.

In simple terms: TTM uses the times of sunrises and sunsets to try and locate, and research, the unusual patterns in the Ionization layer region above the earth that might be linked to earthquakes.
The Terminator Time Method (TTM) is designed for monitoring the timing of sunrise and sunset as it pertains to the minima found in the amplitude or phase of a signal.

The sunrise and sunset signals exhibit terminator times (TTs) and include the following:

- The amplitude SRT is also known as the sunrise terminator.
- The amplitude SST is referred to as the sunset terminator.

The variances around SRT and SST are caused T by the interference of different propagating waves, such as the ground wave and the sky wave. Notably, significant shifts in the TTs from the adjacent or neighboring days' TTs are considered as possible precursors to seismic activity. The TTM also exhibits flexibility in its approach. A sliding window of ±2 days (five days altogether) is used for calculating running means of time series t_m and t_e for the morning and evening terminators respectively.

Finally, the running mean time series are subtracted from the respective TT time series to form the residual TT time series $dt_m = t_m - \langle t_m \rangle$ and $dt_e = t_e - \langle t_e \rangle$

The VLF-daylength is calculated as $D_{VLF} = t_e - t_m$, as the time difference between evening and morning terminator times, and the running mean time series D_{VLF} as

$$dD_{VLF} = D_{VLF} - \langle D_{VLF} \rangle. \quad (1)$$ [18]

2.3. Nighttime Fluctuation Method (NFM)

Nighttime Fluctuation Method (NFM) is an approach used to study geophysical and environmental parameters during the night, focusing on fluctuations in electric fields in the ionosphere, Ionospheric activity, and data from ground-based sensors. The above data is collected and analyzed in order to detect geophysical and seismic events.

The basic assumption is that nighttime measurements of the ionospheric conditions are more stable compared to the day due to noise and human activity, thus making it easier to identify subtle anomalies that could indicate the possibility of an upcoming earthquake. This can be achieved by detecting alterations such as increase and decrease of ionospheric activity. The procedure involves processing raw nighttime amplitude data obtained from daily variation in the amplitude signal. It is necessary to select a specific night period to ensure enough data is collected while excluding daytime periods that are more susceptible to anthropogenic noise.

For better analysis, terminator time represented by minimum amplitudes is excluded from the night interval. Ionosphere-influencing extreme events may change terminator times and these shifts are analyzed separately. Once the appropriate nighttime interval has been identified, mean value of amplitude data over ±15 days sliding window (with center at the day of interest) including the day of interest itself is calculated. This windowing technique minimizes long term variations, thus allowing focus on short-term fluctuations.

The equation for residual variation in amplitude is as follows:

$$dA(t) = A(t) - \langle A(t) \rangle, \quad (2)$$

where $A(t)$ represents the amplitude at time t and $\langle A(t) \rangle$ denotes mean amplitude over the sliding window.

The daily values of three parameters are calculated as below:

$$TR = \frac{\sum_{N_s}^{N_e} dA(t)}{N_e - N_s}, \tag{3}$$

where TR is the mean value of $dA(t)$, and N_e and N_s are the selected nighttime start and end times respectively. [16]

$$DP = \sqrt{\frac{1}{N_e - N_s} \sum_{N_s}^{N_e} (dA(t) - TR)^2}, \tag{4}$$

where DP is actually the standard deviation of $dA(t)$.

$$NF = \sum_{N_s}^{N_e} (dA(t))^2 \tag{5}$$

3. Data

Receivers are positioned across various regions of Japan, with the map generated from precise coordinates in our dataset.

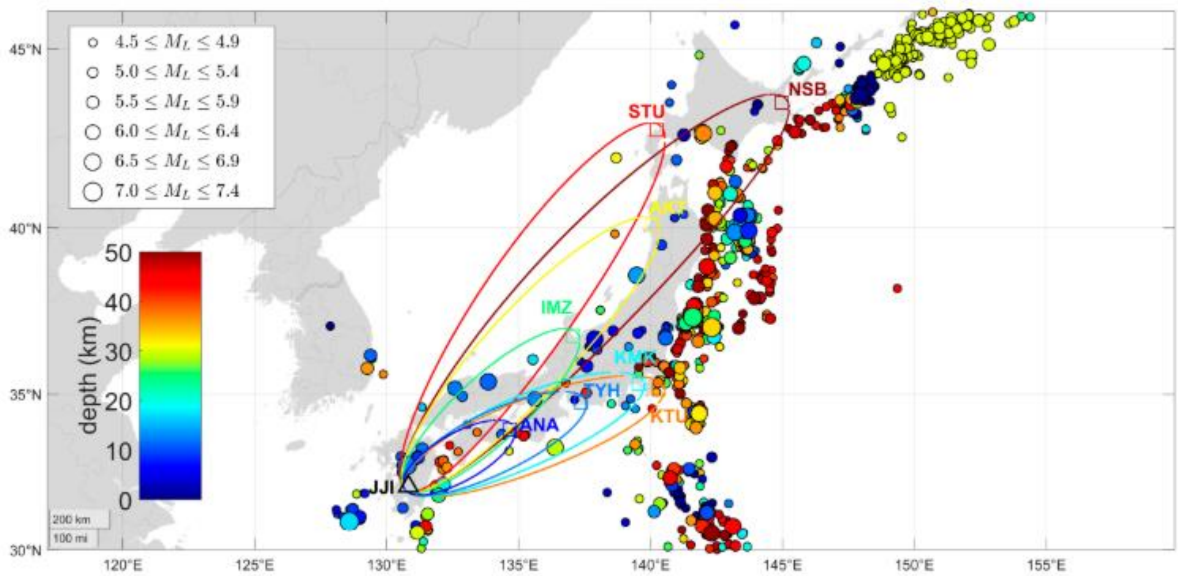

Figure 1: Map of the wider area around Japan showing the 8 subionospheric propagation paths of the EAL VLF network. [16]

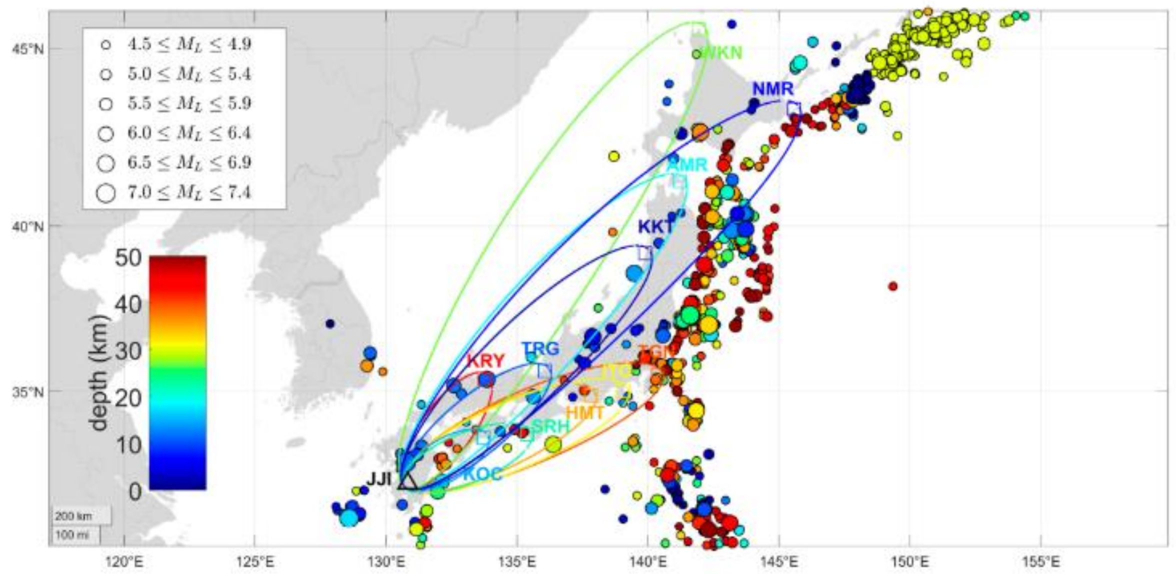

Figure 2: Map of the wider area around Japan showing the 11 subionospheric propagation paths of the Hi-Sem VLF network [16]

We used data in CSVs from these 19 receivers that include the transmitter's name along with each receiver's name and the range of the y-axis in dB. For example, we have "JJI_NSB_85_50." Furthermore, the selected nighttime interval is 20:30–02:30 LT, which is common for all sub-ionospheric paths and throughout the year. Additionally, parts of the nighttime data were excluded where the transmitter was "off" (i.e., the recording was noise) or there was high noise. Other parts were excluded due to temporal overlap with geomagnetic storms (Dst < -50 nT and Kp > 5) and solar flares (C, M, and X classes).

For the data processed with NFM (see figure 2), we have the three statistical normalized parameters: Trend, Dispersion, and NF The analysis was conducted based on the nighttime signal amplitude from the images. Nineteen csv files were used as input which were later combined to gather all the information. The NFM csv file structure is the following:

- The 1st column is the date.
- The 2nd column is the normalized Trend.
- The 3rd column is the normalized Dispersion.
- The 4th column is the normalized Nighttime fluctuation (NF).

Date	Trend	Dispersion	NF	Filename
2014-01-31	-0.785869	-0.013816	-0.429489	AKT
2014-02-01	-0.839924	-0.687260	-0.606656	AKT
2014-02-02	-1.185717	2.465816	1.259553	AKT
2014-02-03	-0.797102	-0.774903	-0.531919	AKT
2014-02-04	-1.205561	2.448145	1.433150	AKT

Figure 3: Data sample of the NFM data utilized

Filename	record_count
KRY	1300
HMT	1190
KMK	1100
TYH	1100
STU	1100
NSB	1100
KTU	1100
AKT	1100
IMZ	1100
ANA	1100
ITO	1080
TGN	820
NMR	780
KKT	730
SRH	730
TRG	730
WKN	730
KOC	460
AMR	450

Figure 4: Record count pre receiver for NFM

The time series plot shows significant fluctuations in Trend, Dispersion, and NF over time. No clear periodicity or seasonal patterns are evident by looking at the diagrams.

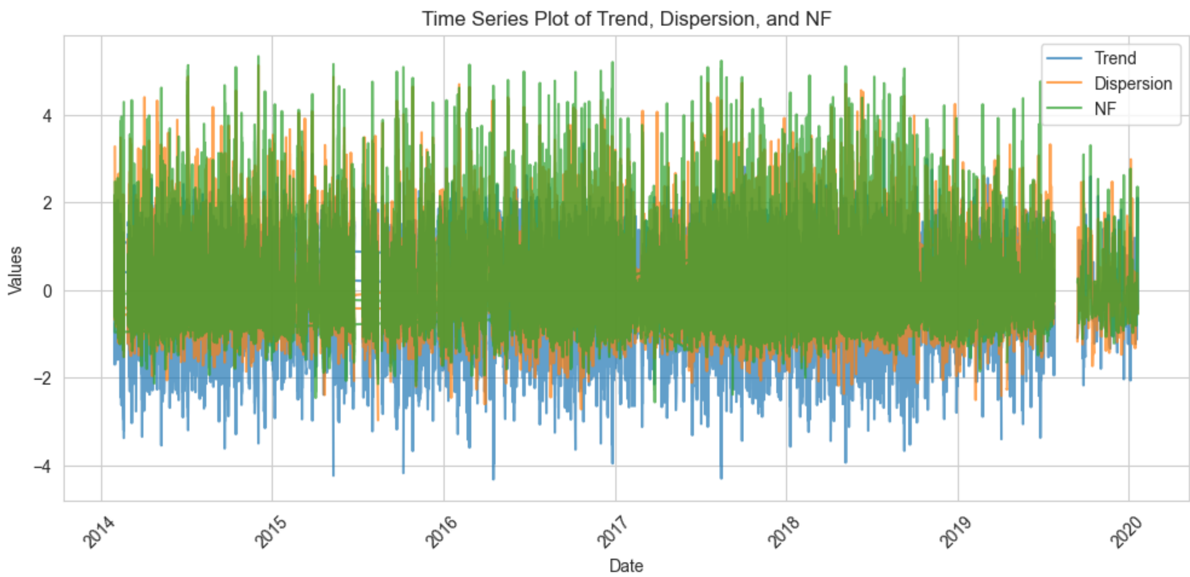

Figure 5: Timeseries visualization of the different metrics for NFM

The diagrams below suggest that Trend follows a roughly normal distribution, centered around zero. Dispersion and NF are skewed slightly to the left, indicating that most NF values are small, but occasionally high values exist in both dispersion and NF. The distributions appear dense around their mean values, indicating that the majority of the data lies within a standard range.

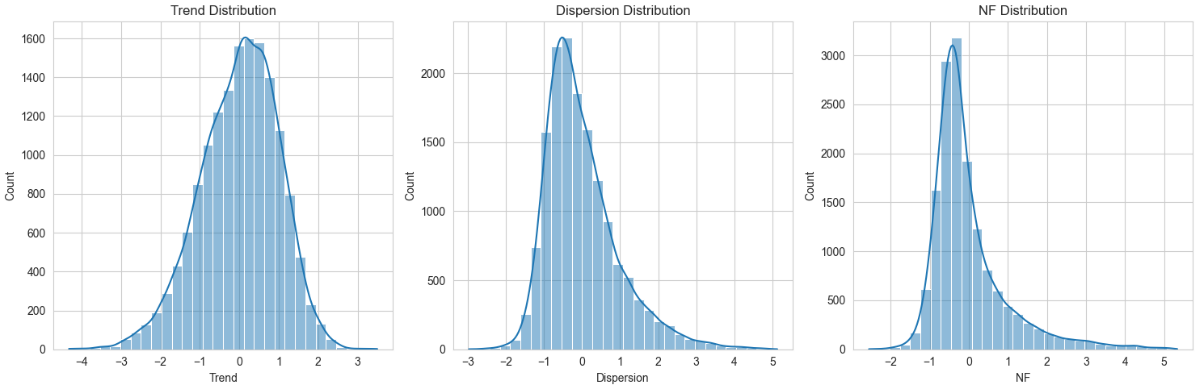

Figure 6: Distribution for the different metrics for NFM

The Boxplot diagrams in Figure 7 indicate that several outliers are present, especially in Dispersion and NF, suggesting certain filenames may have unusual or extreme records. The median values across filenames are relatively stable, indicating no drastic shifts in overall central tendency.

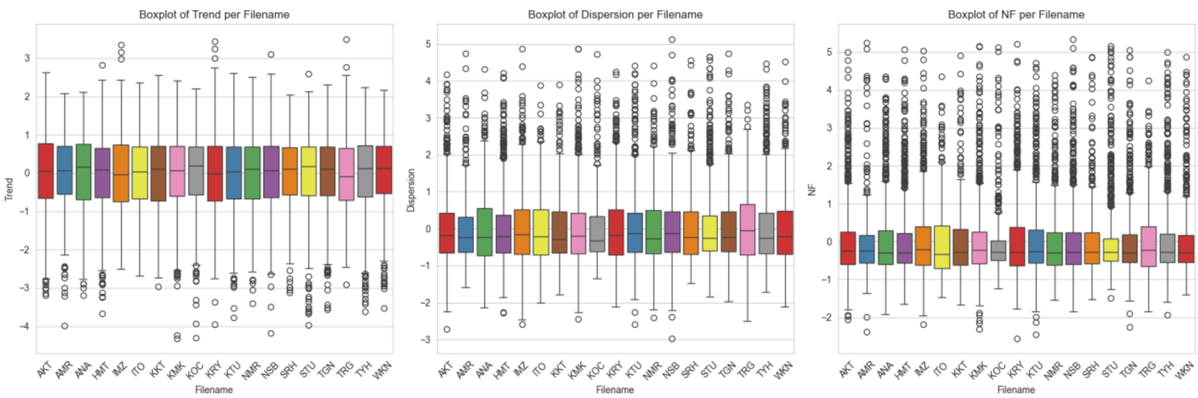

Figure 7: Boxplots for the different NFM Metrics per receiver, the dots outside the boxplots represent outliers per receiver

Additionally we used data on the seasonal variation of terminator times and daylength. The data was provided in 3 csv files per receiver, one for the morning, one for the evening and one for the daylength values. A total of 54 csv items were combined to gather all the information using as keys the Date and the Filename

.

The combined data contained the following information:

- The 1st column is the Date in Local Time.
- The 2nd column is "morning TT" (Morning Terminator Time).
- The 3rd column is "evening TT" (Evening Terminator Time).
- The 4th column is "daylength".

Date	Value_dayl	Filename	Value_ev	Value_mo
2014-04-01	11.89884	AKT	NaN	NaN
2014-04-01	NaN	AKT	17.600859	NaN
2014-04-01	NaN	AKT	NaN	5.702019
2014-04-01	NaN	ANA	NaN	NaN
2014-04-01	NaN	ANA	17.557179	NaN

Figure 8: Seasonal variation Data Sample

Filename	record_count
KRY	3943
HMT	3592
TGN	3472
TYH	3462
STU	3462
IMZ	3460
KMK	3460
AKT	3410
ANA	3380
KTU	3298
NSB	3265
NMR	2359
TRG	1936
WKN	1674
KOC	1486
SRH	1458
KKT	1367
AMR	1295

Figure 9: Record count pre receiver for daylength

The diagrams below suggest that the distribution of Value_day is right-skewed (positively skewed), meaning that lower values are less frequent, while higher values occur more often.

The distribution of Value_ev is slightly right-skewed, but it is more symmetric compared to Value_day. Value_mo is also right-skewed with a longer tail on the right, meaning some values are much higher but occur infrequently.

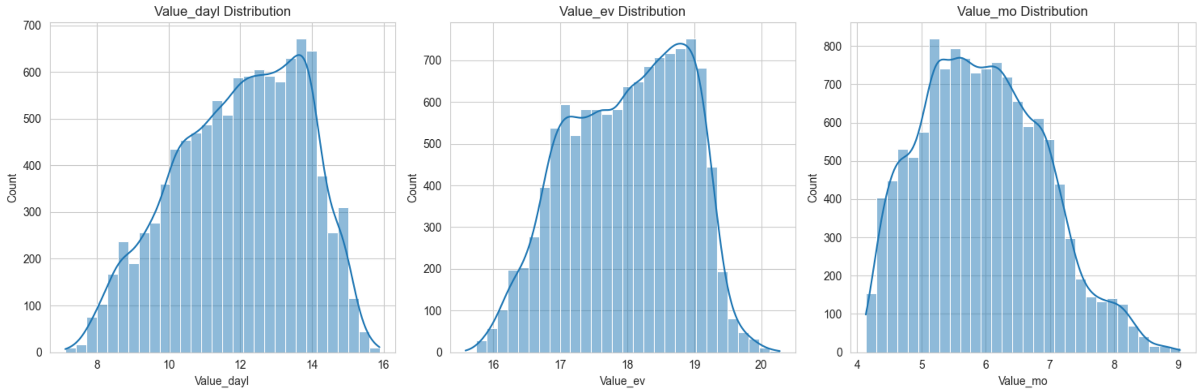

Figure 10: Distribution for the different metrics for Daylength

The Boxplot diagrams in Figure 11 indicate Variability across receivers. There are outliers present, especially in Value_day, which may indicate anomalies or fluctuations in the dataset. The Value_ev presents a relatively more stable distribution which suggests it is more consistent compared to the other two variables.

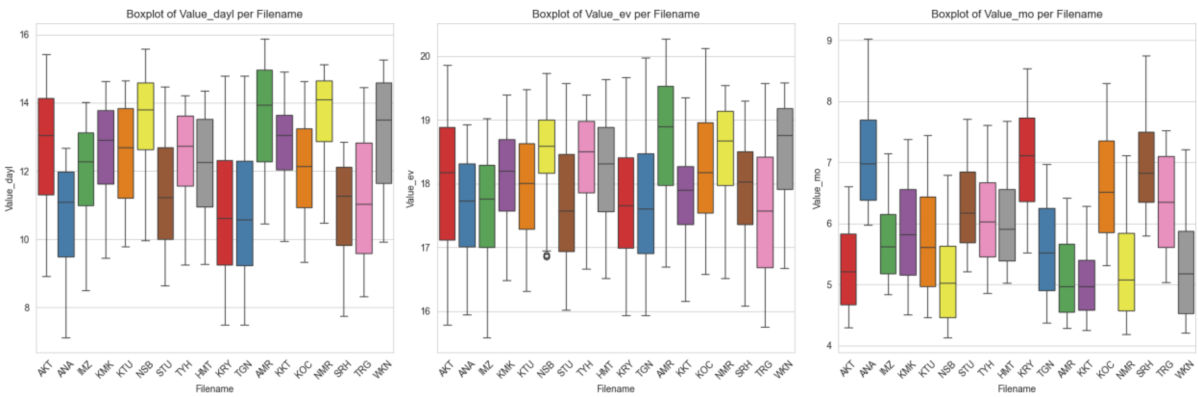

Figure 11: Boxplots for the different Daylength Metrics per receiver

Segments have been discarded from the data as there was abnormal winter dispersion, outliers, and temporal overlap with geomagnetic storms and solar flares. Furthermore, another category labeled 'corrupted data' includes data in which no terminator times (TTs) could be defined because of the absence of signal, noise, or the signal's form.[16]

For the case of earthquakes, data have been obtained from JMA national catalog filtering for earthquakes with Magnitude metric greater than 4.5 and focal depth less or equal to 50 kilometers. Following that, filtering of the sub-ionospheric paths was done separately for each path considering only the earthquakes located within the 5th zone of Fresnel diffraction. Moreover, the critical radius of the Fresnel zone intersection with the seismic zone also needed to be taken into account. The 5th Fresnel zone is a region of wave propagation that contains the impact of additional diffraction on the signal caused by seismic activity. It is one of the higher order Fresnel zones and for this reason, it is expected interference of waves within the zone can produce measurable disturbance of the ionosphere.[19]

The Earthquake data was provided in nineteen csv files which were combined, containing the following information:

- The 1st column is the date and time in local time.
- The 2nd column is the Earthquake Magnitude.
- The 3rd column is the Longitude.
- The 4th column is Latitude.
- The 5th column is the focal Depth of the earthquake.

Date	Time	Magnitude	Latitude	Longitude	Depth	Filename
2014-08-29	04:14:35	6	32.14	132.145	18	AKT
2014-08-29	04:32:03	4.5	32.1116666666667	132.143333333333	19	AKT
2014-11-22	22:08:17	6.7	36.6916666666667	137.89	5	AKT
2014-11-22	22:37:49	4.5	36.7783333333333	137.91	3	AKT
2015-02-06	10:25:12	5.1	33.7333333333333	134.37	11	AKT

Figure 12: Earthquake Data Sample.

Filename	record_count
NMR	240
NSB	206
TGN	110
AMR	106
KKT	105
AKT	102
WKN	92
KMK	90
KTU	90
IMZ	89
STU	89
ITO	86
HMT	85
TRG	84
TYH	82
KRY	80
ANA	78
KOC	72
SRH	69

Figure 13: Record count pre receiver for Eearthquake data

4. Strategic Approaches for Data Preparation and Model Optimization

The most important goal of our project was to identify the best combination of techniques which allow us to create a system that can predict the occurrence of seismic events with a good deal of accuracy. Given the importance of the goal, we tried to calibrate before and during data modeling and even employed less sensitive techniques, in order to increase the chances of identifying any relationships that would suggest an earthquake is likely to happen.

In order to accomplish our objective and build competent models, we utilized certain strategies:

- **Data Analysis & Data Cleansing:** Validates data accuracy and integrity by eliminating noise, outliers, and missing values, thus making it appropriate for analysis.
- **Data Balancing:** Prevention of biased predictions through the use of Balancing techniques to ensure that all classes in the dataset (earthquakes and non-earthquakes) are adequately represented.
- **Feature Selection:** This step details the relevant features (or variables) needed in relation to an Earthquake prediction, which raises the accuracy of the model while improving interpretability because of noise reduction and dimensionality constriction.
- **Modeling & Hyperparameter Tuning:** It also includes the selection of appropriate machine learning models like LSTM, GRU, CNN, LSTM, etc. and the tuning of hyperparameters to achieve maximum accuracy and optimize performance of the model in regard to the data provided.

4.1 Data analysis - Data cleaning

```
Percentage of empty values per column:
Date                         0.000000
Filename                     0.000000
Value_dayl                  46.163312
Value_ev                    41.162573
Value_mo                    35.541665
Time                        90.377516
Magnitude                   90.377516
Latitude                    90.377516
Longitude                   90.377516
Depth                       90.377516
Trend                       23.039819
Dispersion                  23.039819
NF                          23.039819
Angle_from_transmitter       0.000000
Distance_from_transmitter    0.000000
receiver_long                0.000000
receiver_lat                 0.000000
flag_EQ                      0.000000
Receiver_AKT                 0.000000
Receiver_AMR                 0.000000
Receiver_ANA                 0.000000
Receiver_HMT                 0.000000
Receiver_IMZ                 0.000000
```

Figure 14: Commulative data overview

A combination of the existing datasets with the aim of having one dataset that contains all the information of the other datasets, allows for improving the understanding of the information offered. This enables one to properly analyze the data, find missing values, or outlier data – For these reasons and to achieve a better understanding of the information captured by each receiver, we first created a single consolidated dataset. For this purpose, all the datasets were combined using the dates and the receivers for each specific dataset as the keys.

When available, latitude and longitude values were adjusted to coordinates. Additionally, recognizing the potential importance of distinguishing that each data point comes from a certain receiver, each receiver was given flag columns. All the information was then captured in a single final data frame where the following was the output.

The columns Time, Magnitude, Latitude, Longitude, and Depth are only populated during a reported/recorded earthquake. Because of this, these columns were omitted from the dataset and were changed to an earthquake flag as their model features selection would introduce a bias into the model.

4.2. Balancing techniques

From the graphs below it is evident that our data is imbalanced as the number of negative events is much larger than the number of positive events, in specific only 9.6% of our data are positive earthquake events

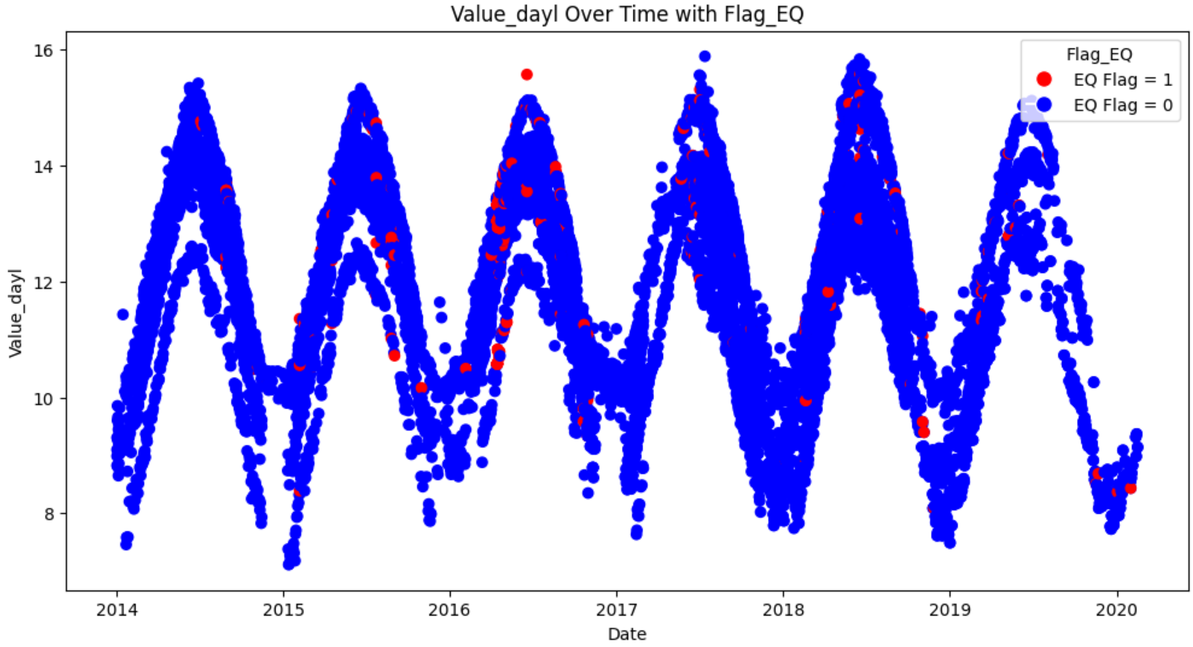

Figure 15: Chart of values daylight in relation to positive earthquake positive events.

In the context of machine learning, an imbalanced dataset is when the instances of a class exceed that of another class by a large margin. As an illustration, consider seismic events as the primary class and non-seismic events as the secondary class. In earthquake prediction, the former class occurs far less often, thus resulting in a skewed model training which will produce biased predictions and poor performance metrics for the minority class.

An Imbalanced Dataset causes Challenges in modeling, as mentioned above. In our case we are trying to predict the minority class (positive earthquake occurrences) and to avoid having biased predictions and poor performance metrics we had to manipulate the data to produce a more consistent and robust training dataset.

The challenges we faced due to the Imbalanced Dataset are

- **Model Bias: Majority Class is Preferred:** With the majority of classes in a dataset, machine learning models seek to minimize overall error distribution often resulting in prioritized class predictions more frequently. The result is often a lower recall and precision measure which is the opposite of what the analysis aims to achieve.
- **Misleading Metrics:** Performance accuracy and other standard metrics can be extremely misleading. For example, in our dataset with a ratio of 91:9, a model who predicts the only majority class achieved 98% accuracy but could not fulfill the needs of the minority class.

- **Overfitting to Majority Class:** One of the other implications of such imbalanced datasets is that models may overfit the majority class resulting in a failure to generalize on class instances of the minority.
- **Real World Impact:** In high stake situations such as the prediction of earthquakes, erroneously classifying a minority class event and failing to detect it could lead to catastrophic outcomes.

There are multiple strategies that have been created to alleviate the problems caused by imbalanced datasets. These mentioned strategies include:

- **Resampling Techniques:**
  - Over-Sampling: Creating new synthetic examples for the less represented class.
  - Under-Sampling: Selecting a smaller number of observations from the overrepresented class.
- **Combination Methods**: Blending both over-sampling and under-sampling methods.
- **Algorithmic Adjustments:** Applying cost-sensitive learning by increasing penalties for the mistakes made in the training set for the minority class.
- **Hybrid Approaches:** Integrating data and algorithm methods to achieve better performance with biased datasets.

For earthquake prediction, a dataset representing the occurrence of seismic events is usually heavily imbalanced which makes it one of the most difficult tasks due to the scarcity of the disasters.

This imbalance can be dealt with very effectively by deploying machine learning:

- Better recall and precision for earthquakes determined in the datasets.
- Less false negatives when a particular marker is missed that indicates a pinnacle to an earthquake.
- Better trustworthiness of models when applied to the ground reality builds this in a large-scale manner.

In our case we decided to test two different methods of handling the imbalanced dataset and compare results to identify the methods that produce a better outcome for our data. In particular we tested INOS (Interpolation-Based Oversampling) and ESPO (Edge-Based Synthetic Minority Oversampling).[1]

4.2.1. INOS (Interpolation-Based Oversampling)

Informative Over-Sampling, or InOS, focuses on the improvement of the predictive performance of a model through the augmentation of the minority class in class imbalanced datasets. InOS differs from conventional methods in that it seeks to produce domain driving informative time-series samples, ensuring an allocation of the original data distribution. This method is quite useful in the prediction of extreme but rare events like earthquakes.

The steps we implemented to perform INOS are the following:

- **Identification of the Minority Class:** Analysis of the dataset to determine the minority event classes such as seismic events.
- **Evaluation and determination of the level of unbalance** within the dataset using the imbalance ratio.
- **Feature Space Analysis:** Analysis of the features in the majority class and features of the class with fewer samples,
- **Clustering:** Identification of feature space regions with a high density of class samples.
- **Generation of Synthetic Samples:** Generation of New samples by using a mixture of features from the few samples in that cluster to increase the sample size of that class.

InOS creates synthetic examples that are realistic and informative as they are created by sample points in the neighborhoods without disturbing the original data structure. Using informatic or statistical techniques to guide the creation of the synthetic examples to be representative of the patterns seen in the underrepresented class, we were able to prevent the creation of unnecessary or irrelevant samples which could potentially lower the performance of the model's predictions such as precision and recall. After the addition of the synthetic samples into the original data set, the class distribution is much more even.

In our case, after implementing INOS, our dataset gained the following characteristics:

- It created realistic simulated samples of what one could expect before the shock occurs, keeping intact the statistical characteristics of the minority group.
- It controlled the exposure of the models to samples that create noise and very simplistic samples reducing the risk of over-fitting.
- It increased the model's sensitivity to important events that are infrequent, improving the recall and diminishing the false negatives.

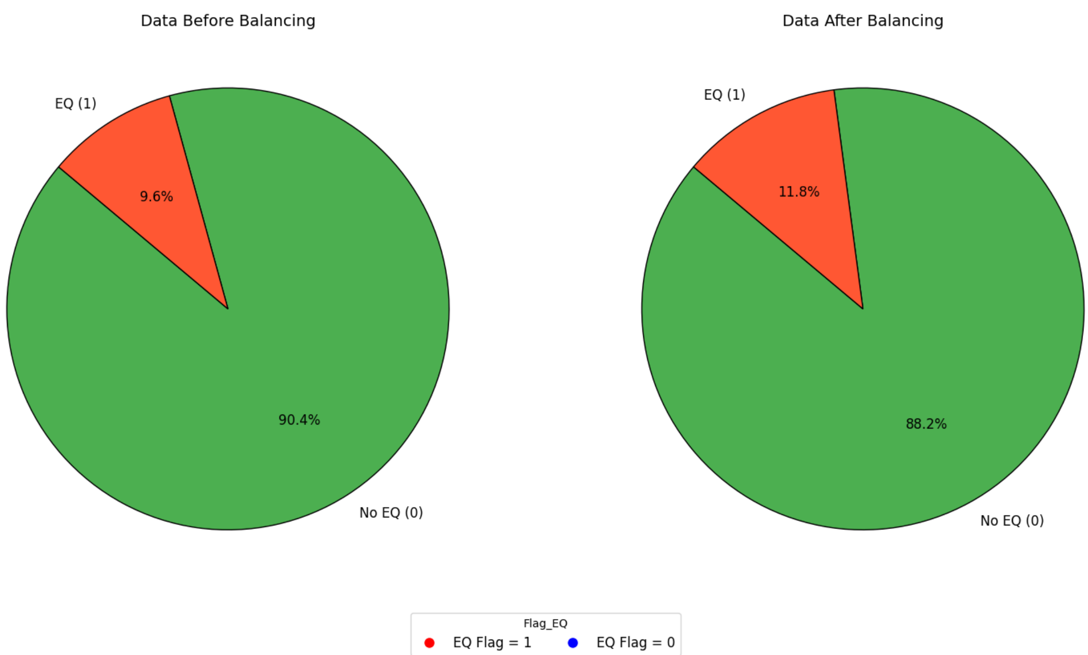

Figure 16: Pre vs Post INOS Positive EQ events ratio.

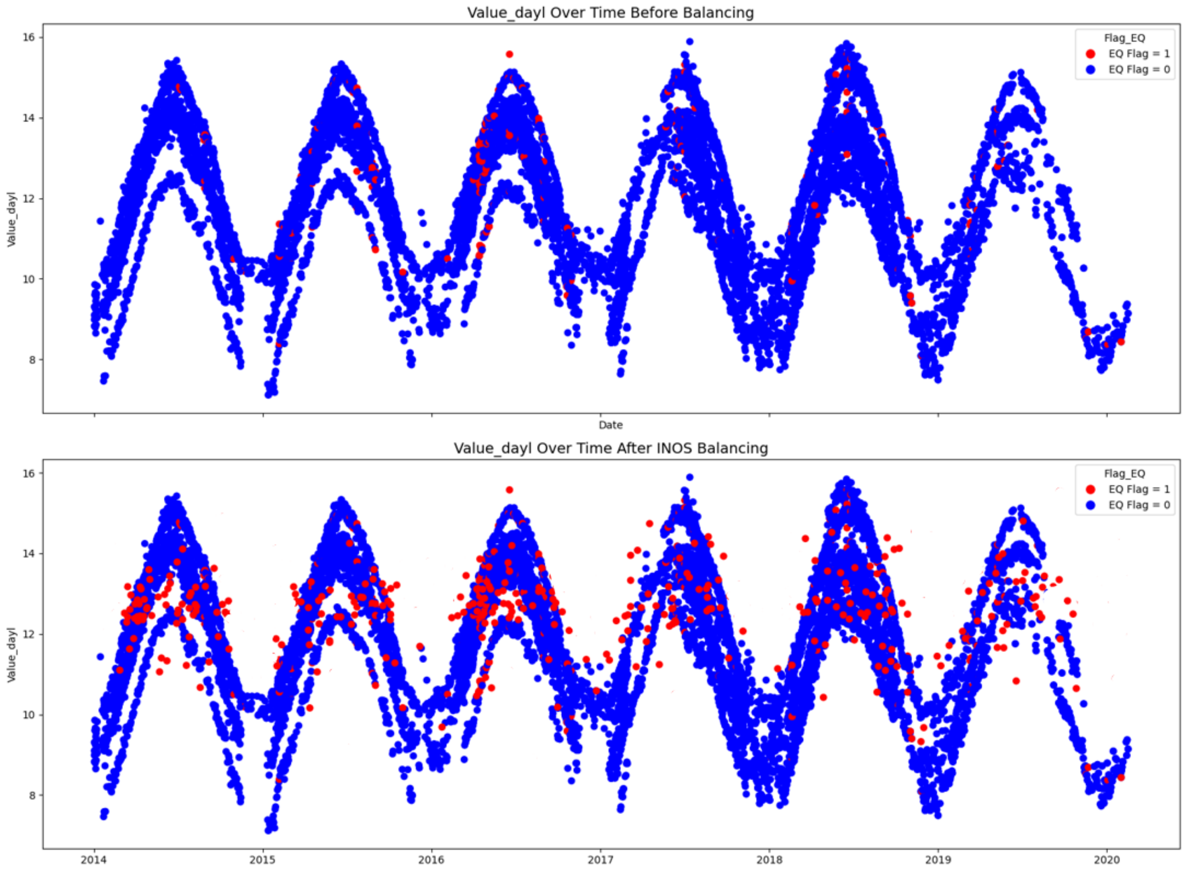

Figure 17:Per vs Post INOS Optimization WEQ Positive events

4.2.2. ESPO (Edge-Based Synthetic Minority Oversampling)

ESPO refers to Edge-Based Synthetic Minority Oversampling, which is a technique of data augmentation used to address the problems of class imbalance by creating synthetic samples along the boundaries of the minority class distribution. ESPO's initial step is to increase the class boundary of the decision-making boundary or border. This is unlike traditional methods such as SMOTE, which "dissolve" all instances of minority class to create new samples. This addresses the underrepresentation issue. Because ESPO aims to improve the classification accuracy, it works best in datasets in which the boundaries of the minority class are sparse and achieving precision in classification is crucial.

In our case we performed the following steps to handle the dataset Imbalance using ESPO:

- **Identification of Minority Class and Edges**: Examined the data and focused on the edges of the minority class while also figuring out the class imbalance ratio.
- **Utilization of k-nearest neighbors (k-NN)** to identify the instances that are located closer to the boundary of the minority class.
- **Selected Pairs for Interpolation**: Identified and paired together instances that are considered edge instances for interpolation.
- **Choose pairs** that are located closer to the decision boundary and regions with lower concentration of minority class instances.

- **Created Artificial Samples**: Interpolation of new instances between the given edge instances to ensure that the produced instances are close to the edge while still being in the realistic distribution boundaries of the minority class.
- **Refined Artificial Samples**: The synthetic samples needed to be verified using some domain knowledge or constraints, such as their relevance towards the actual problem within context (seismic precursors to earthquake forecasting).
- **Augmented the Dataset:** Included the artificial samples to the minority class. This resulted in a more balanced dataset.
- **Model Training and Evaluation**: Trained the model with the augmented dataset focusing on recall, F1-score and AUC-ROC metrics for the minority class improvements.

Why we use ESPO on Earthquake Prediction: When ESPO is applied to the areas of earthquake prediction, prediction accuracy is much higher for the following reasons:

- **Focuses on edge instances**, which best helps delineate the border between seismic and non-seismic phenomena, which is critical in event characterization.
- **Produces synthetic samples** which enhance model sensitivity in these regions, therefore increasing recall and reducing false negatives.
- Helps achieve the **preservation of the minority class distribution**, ensuring that the produced synthetic data is credible and useful.

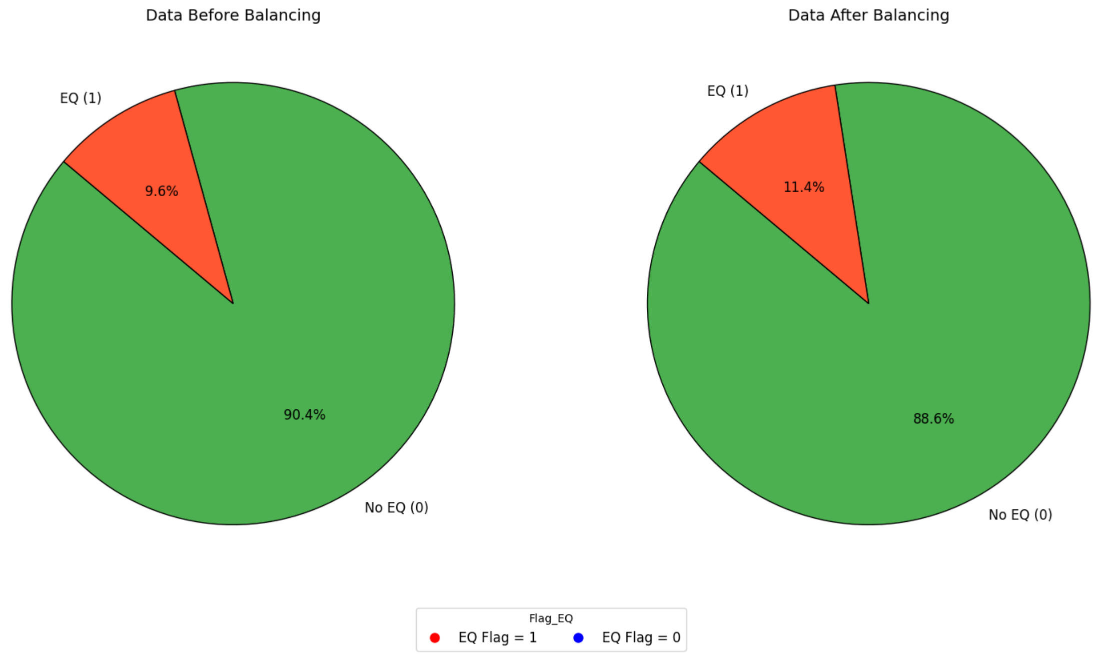

Figure 18: Pre vs Post ESPO Optimization Positive EQ ratio

4.2.3. Combination of INOS and ESPO

In concert with the best features of INOS and ESPO, we adopted a hybrid method that joins their strengths together. INOS produces highly believable and statistical driven synthetic samples by interpolating dense regions of the minority class feature space, whereas ESPO performs better on boundaries to increase accuracy and reduce false negatives. These methods together gave us a dataset that has domain-representative samples balanced with edge focused enhancements. This combination improved recall, F1 score, and model bias in a more positive direction leading to a robust prediction framework for earthquakes.

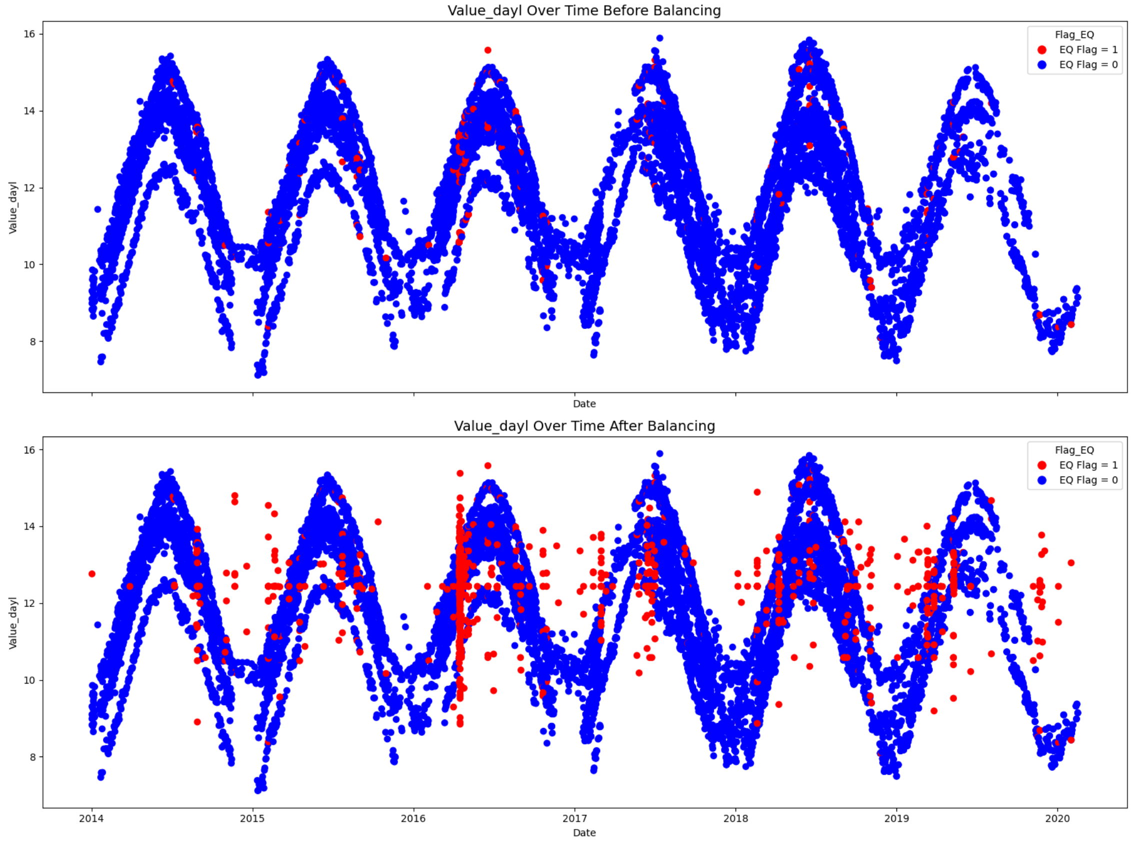

Figure 19: Pre vs Post ESPO Optimization Positive EQ events

The combination of the two methods further increased the Minority Sample percentage thus creating a more robust dataset.

Figure 20: Pre vs Post Optimization Positive EQ ratio of the combination of INOS and ESPO

4.3. Feature selection

Before training our models into the earthquake data, we implemented different methods to identify the features that were most relevant in order to reduce complexity, minimize overfitting, and enhance interpretability. Owing to the dataset complexity and scale, a systematic feature selection approach was necessary, allowing isolating the most relevant variables for seismic events prediction. In this chapter, we present the methods of feature selection applied, SHAP and Recursive Feature Elimination (RFE) in particular. With these, we have been able to assess separately the contribution of every feature and arrange the models in accordance with the most significant predictors. These tests proved beneficial not only in reducing the model training time but also in exposing what was truly driving the predictions made by the models and contributed to building models that were reliable and free from bias.

4.3.1. SHAP (SHapley Additive exPlanations) values

SHAP (SHapley Additive exPlanations) is a certain output from a machine learning mode. It provides an inclusive approach to the use of cooperative game theory, which involves the use of Shapley values, to complement a specific prediction by assigning an importance to each feature involved. These weights serve as a measure of a feature's impact and thus serve the purpose of making complex models easier to understand

SHAP values are calculated based on all possible combinations of features in the model and measuring the additional value that outstanding features bring to the prediction of the model:

- **Model Agnostic**: SHAP can be used in any kind of machine learning model, be it a linear regression to a nonlinear XGBoost and even random forest which is considered to be complex.
- **Additivity**: For each feature in the instance, it is measured that the total SHAP values add up to the output of the model minus the average prediction (baseline).
- **Fairness**: Feature contributions are made in an equitable manner using basic alliances mechanisms in game theory.

The key concepts of the SHAP are shown below:

- **Shapley Values**: These were originally derived from cooperative game theory where all the features that helped make a prediction having a go before the "pay" is made to every single one of them equally.
- **Baseline Prediction**: This estimates what the model will predict, when there are no input features available, which is an average of the output to be trained.
- **Feature Contribution**: Shapley Heuristic additively approximates the contributions of entire features to or from the baseline prediction.

In our case we applied SHAP in Feature Selection with the below steps:

- **Feature Importance Ranking**: Ranked by their SHAP values, features were ordered according to their impact in the models' predictions. To reduce the complexity in the model, features with low average SHAP values were ignored.
- **Visualizing Feature Impact**: Features were evaluated according to their degree of importance and interaction by showing the distribution of shap values across the dataset for every feature. So, we were able to pinpoint features that are significantly relevant to the prediction by tracking how the shap values vary in relation to various features being combined.

Pros of SHAP in earthquake prediction:

- **Works for any Model**: Is useful for any kind of prediction model.
- **Global and Local Interpretability**: Gives insight into an individual prediction(local) and also gives an insight into how the model performs overall(global).
- **Fairness**: Explains mathematically a fair mechanism to approve the feature contributions.

Cons of SHAP in earthquake prediction:

- **Costly**: Deriving exact SHAP values tends to be time-consuming and an expensive exercise mostly with complex models and large data sets.
- **Underpinning Approximations**: There are simpler approaches such as TreeSHAP or KernelSHAP that perform the calculation, but only approximate the Shapley values with Shapley values, which entails some loss of accuracy.[15]

4.3.2. Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a feature selection method that aims to determine the best predictors for a particular machine learning model. RFE performs this by taking a model and removing the features that are the least useful, and then re-training the model on the remaining features. This process is repeated until a certain number of features have been chosen, or other performance-related criteria have been met.

RFE adopts a particularly structured approach to feature ranking and elimination of those features that do the least towards enhancing the performance of the model in the following steps:

1. **Train model**: The machine learning model (linear regression, Random Forest, SVM, etc.) is trained on the dataset with all features.
2. **Ranking Features**: During this training session, the model sets scores to the features relative to their importance (linear models contain coefficients; tree-based models incorporate feature importance scores).
3. **Removing Features**: The model is retrained after excluding the least important feature(s) of the model.
4. **Repeat**: Perform Step 2 and Step 3 until optimal features are reached or the model starts to decrease in performance.

The application of RFE in our dataset produced the following results:

- **RFE shrunk the predictor space** by targeting only the most important features that needed to be used. This reduced overfitting and could save the costs involved in computations, in particular on high-dimensional data.
- **Feature Ranking**: RFE develops a ranked list of features that were used pertaining to the model's performance. The prediction tasks that have the highest rank were given the topmost gears.
- **Model Optimization**: RFE took care of the elimination of irrelevant or redundant features which helped in the interpretation of the model and also improved the accuracy considerably.

Advantages of RFE:

- **Model-Specific Feature Importance**: RFE analyzes the model at hand and uses suitable features to that model so that it does not make any mistakes on that model.
- **Scalability**: RFE is an approach that can be used in different arbitrary types of machine learning models.
- **Improved Performance**: RFE is useful in removing the noise or the undesirable features making the generalization of the model relatively better as well as reducing overfitting.

Limitations of RFE:

- **Computational Cost**: RFE has the disadvantage of retraining the model many times, thus this takes a long time for big datasets and heavily parameterized models.
- **Dependency on Initial Model**: The RFE scales to how good the model specified in the rank ordering of the feature was. For example, tree-based models may rank the features differently than some linear models.
- **Risk of Over-Elimination**: Over-Removing features can risk cutting out relevant features that have a minimum contribution to the prediction.[11]

5. Modeling

This chapter describes the machine learning models which were trained and tested with a view to predicting earthquakes. In carrying out this work, both deep learning structures as well as traditional machine learning structures were developed and compared. The chosen models are sufficiently different from one another and are each tailored to give optimal results with time series data, inter-feature relations, and predicting times.

Deep learning models include Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs). These architectures were selected because of their successful dependability to model temporal relationships and derive intricate structures from ordered data which is essential in earthquake prediction.

Also, the deep learning models, traditional machine learning algorithms such as Random Forest, K-Nearest Neighbors (KNN), XGBoost, and Gaussian Process were trained and tested. These models offer different strategies that do not demand High Performance Computing and large amounts of data but are still reasonably good.

Every model is discussed in the next sections, including their architectural design and significant features and the rationale behind their selection. This comparison and assessment clearly bring out the strengths and weaknesses of all the models providing great insight into their use in earthquake prediction.

5.1. Long Short-Term Memory (LSTM)

How LSTMs Work: Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to handle sequential information by capturing long-term dependencies. Traditional RNNs often suffer from the vanishing gradient problem making it difficult for them to learn relationships over extended sequences. LSTMs solve this by incorporating memory cells and gating mechanisms that enable selective storing or forgetting.[13]

Crucial parts of an LSTM model include:

- **Forget Gate**: Decides which portions of prior information should be thrown away so as to ensure the model concentrates on relevant facts.
- **Input Gate**: Determines what new data should be saved in the memory thus allowing the model to effectively assimilate recent observations.
- **Output Gate**: Controls how much of the memory is used in generating output or passed to the next step thereby providing contextually informed predictions made by the model.[9]

Thus, these features make LSTMs capable of capturing both short and long term dependencies hence they are suitable for time series data such as seismic or atmospheric signals.

This study uses LSTMs to forecast earthquake incidences through the analysis of sequential data. The input data is a series of historical time steps which assist the model in identifying relationships with seismic events.[10]

Input Data: Sequences like ionospheric anomalies are sent into an LSTM model.
Training Process: LSTM learns temporal dependencies by iterating through multiple time steps, capturing patterns that may indicate precursors to Earthquakes.[9]
Prediction: Given new data, the trained LSTM looks back on its previous observations and can tell whether there is a possibility of earthquake or not.[13]

5.2. Gated Recurrent Unit (GRU)

Gated Recurrent Units (GRUs) are a variant of recurrent neural networks (RNNs), which were simplified in order to fix the issues with previous RNN models, such as vanishing gradient problem, while also providing more straightforward design compared to Long Short-Term Memory (LSTM) networks. By reducing the number of gates and parameters, GRUs have shown that they can be efficient without compromising on their ability to capture long-term dependencies.

There are two parts in the GRU architecture:

- **Update Gate**: This determines how much of past information should be preserved in the present hidden state. It balances the relevance of past and actual inputs.
- **Reset Gate**: When calculating the new candidate hidden state, it decides how much of the previous information to erase out. This upholds flexibility for recent changes in input sequences.

Unlike LSTMs, GRUs do not employ an additional memory cell thus simplifying them whilst still allowing them to model both short and long term dependencies.[5]

Within this study, the GRUs are applied to analyze and predict earthquakes by modeling their ability to simulate temporal patterns in this kind of data. Since they are efficient and simple, GRUs can handle huge datasets or applications that require faster training.

Data Input: The GRU model takes sequential data such as seismic signals or ionospheric anomalies for analysis.

Training: The GRU identifies patterns among input sequences associated with seismic activities, hence improving its ability to make predictions.[4]

5.3. Convolutional Neural Network (CNN)

CNNs, Convolutional Neural Networks, are a type of deep learning model constructed to handle structured data such as grid-like data or images. Its abilities in the recognition of spatial and temporal patterns make it highly effective for feature extraction when dealing with tasks like image classification and time series.[17]

The major components constituting CNNs are as follows:

- **Convolutional Layers**: These are layers with filters (kernels) that slide over input data and identify local patterns. The key features of the input like edges in images or temporal patterns in signals can be highlighted by these filters creating feature maps.
- **Pooling Layers**: Retaining important information while reducing computational complexity is what pooling layers do to spatial dimensions of feature maps. In this step, small variations of the input should not affect results significantly (Bengio et al., 2013).
- **Fully Connected Layers**: Fully connected layers map extracted features during feature extraction to output predictions whether classification or regression.

In this study, we apply CNNs for earthquake prediction on spatial or temporal grids structured data. Hierarchical structure of CNNs helps them to interpret meaningful patterns within the input data e.g. changes in atmospheric signals. We have data which is structured as grids or sequences to represent spatiotemporal nature of observed phenomena.

Feature Extraction: The convolutional layers detect patterns such as discontinuities or relationships that may indicate seismic activity.

Prediction: The fully connected layers will use the extracted features to predict the chance, place, and size of earthquakes.

5.4. Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) are a type of neural network designed for sequential data processing. Unlike ordinary feed-forward networks, they contain feedback loops that remember past inputs and thus can represent temporal dependencies and patterns.

At each time step, present input is taken along with the prior hidden state by the RNN. As far as remembrance of what went before is concerned, this concealed status serves as the memory of the network. The net’s output at every time step will be influenced by both current information and all previously accumulated memories. Although recurrent neural networks learn well during short term relations, they usually fail to capture long-term dependencies due to problems such as vanishing gradients.[2]

This research will employ RNNs in earthquake prediction given its sequential nature. They are useful in analyzing seismic signals or atmospheric anomalies because they have a capacity for capturing temporal links.

- **Input Data**: This is when data flows continuously into an RNN like ionospheric variations or seismic signal measurements over time.
- **Training Process**: The RNN learns to connect definite templates in the sequence with seismic activity, using feedback loops to include interconnections across many time steps.
- **Prediction**: After training, the RNN ascertains chances of an earthquake happening from new input sequences by taking advantage of its remembrance of previous data.[8]

5.5. Random Forest

Random Forest is a machine learning technique that combines many decision trees to improve predictive precision and robustness. It is done by constructing a bunch (forest) of decision trees during training and collecting their predictions (through majority voting for classification or averaging for regression) into one final result. Random Forest relies on two elements: bootstrapping and random feature selection:

Bootstrapping (Bagging): Every single decision tree is constructed based on a randomly chosen fraction of the training data with replacements. This ensures different trees resulting in reduced overfitting.

Random Feature Selection: While making each split in a decision tree only some features are considered instead of considering all features. Thus, reducing correlation among trees thereby boosting generalization.

Random forest **averages predictions** from numerous trees so as to minimize errors arising from overfitting to specific samples or variables.[3]

5.6. K-Nearest Neighbors

K-Nearest Neighbors (k-NN) is a simple, but effective algorithm in classification and regression. Hence, it uses the principle of instance-based learning, which makes predictions for new data points with respect to their likeness with already existing data points in the training set. The algorithm does not need any form of training; instead, it just relies on memory.[6]

Essential Steps in k-NN:

- **Distance Calculation**: This includes finding the distance between a new data point and all the points present in the training dataset.
- **Finding neighbors**: By use of k closest data points that are associated with a novel instance are taken as its neighbours.

- **Prediction**: Among the k nearest neighbours, prediction is that whose class is majority one.

5.7. XGBoost

XGBoost (Extreme Gradient Boosting) is a very sophisticated ensemble machine learning model applicable for both classification and regression tasks. It is based on the paradigm of gradient boosting where models are built in an auto-regressive manner whereby each new model is built with the intention of correcting the mistakes of the last one. XGBoost makes advancements like regularization, parallel processing, and tree building optimization which enables its users to build much quicker and accurate solutions as compared to just gradient boosting.

In XGBoost the main steps include:

- **Model Initialization**: Algorithm computes base learner such as predictions for regression could be the mean or for classification the most common category.
- **Gradient Calculation**: In every iteration, the model computes the gradient of the loss function for each training example as a measure of how the model is currently predicting compared to the actual value.
- **Tree Construction**: A decision tree is constructed in order to limit the value of the loss function where the calculated gradients act as weights for the splits.
- **Model Update**: The predictions from the new tree are appended to the previous ensemble and hence the quality of the overall prediction is enhanced.
- **Regularization**: XGBoost implements L1 and L2 regularization during the model training stage to limit the model complexity and hence eliminate overfitting.

5.8. Gaussian Process

Gaussian Processes constitute a class of probabilistic non-parametric models which can be used for regression and classification tasks. Instead of a direct output prediction, GP's model a distribution of functions that can explain the data, making it a useful and uncertainty-informed approach (Rasmussen & Williams, 2006).

Key Concepts of Gaussian Processes:

- **Prior Distribution**: There is a belief that GP models function values which are multivariate Gaussian variables. Said belief is formed after observing the data and this function is an estimated mean function with zero and a kernel type function.
- **Kernel (Covariance) Function**: Data points that are in closer proximity are weighted more heavily than outlying data points, which is determined by the kernel function. Well known kernels are the radial basis function (RBF) and Matérn kernel.
- **Posterior Distribution**: A posterior distribution is therefore defined as the distribution which is obtained through multiplication of the prior distributions with a given

likelihood of the data function and a model. This posterior serves to refine the earlier guesses which were made regarding the underlying function that generated the data.

6. Model Optimization Methods

In order to achieve the optimal modeling results we tried to improve model accuracy by utilizing optimization methods and techniques. This chapter covers different methods of model optimization with an emphasis on hyperparameter adjustment and search strategies. Hyperparameter adjustment is one of the most important parts in adjusting a model's accuracy level, preventing it from overfitting or underfitting, and saving on computation power. Among the numerous methods, grid search is presented as the most organized technique for determining the best combination of hyperparameters by systematically examining all preset configurations in order to enhance accuracy and reliability. In undertaking these approaches, we hope to optimize the models for broader application and use in tremor forecasting.

6.1. Hyperparameter tuning

Hyperparameter tuning refers to the procedure of selecting the best values of a model's hyperparameters for a task. In contrast to model parameters such as weights in a neural network learned through training, hyperparameters that define the learning are set. These include the learning rates, layers or units in neural networks, the depth of decision trees, or the number of estimators in the case of ensemble methods.[2]

Hyperparameters significantly affect the performance of the model and its ability to generalize. It goes without saying that hyperparameters must be well optimized for performance to improve. [7]

Good hyperparameters can for example.

- **Improve Accuracy**: So long as a suitable hyperparameter is constantly utilized, it results in better performance in most predictions anyways.
- **Prevent Overfitting or Underfitting**: If, entirely for example, the regularization parameter is exceedingly high, it could easily lead to under fitting, conversely if it is lower the former becomes overfitting.
- **Optimize Computational Resources**: Hyperparameters like batch size, or the number of estimators when properly suited can be said to help decrease time and memory consumption during training.
- **Ensure Robustness**: To be applicable in real life models that have been fitted needs to be able to provide performance when data that it has never been exposed to is provided.

6.2. Grid Search

Grid search is a type of methodical search used for hyperparameter optimization to such a depth of which each parameter within a specific set would be scanned. The thorough optimization would initiate for a desired model till all combinations have been checked.[14]

Each combination is tested with performance measures:

- **Systematic Exploration**: The Grid search guarantees the usage of every combination within the set parameters during the model building phase thus ensuring no combinations are missed or ignored.
- **Optimizing Model Performance**: Since multiple combinations will be checked through the Grid Search process the best combination of the hyper parameters to increase the performance would be sought.
- **Reproducibility**: Considering the systematic and extensive nature of the grid search tuning process, replication of the process for the purposes of research and operational usage is effortless.
- **Hidden Size**: The number of neurons in the hidden layers of a neural network, determining its capacity to learn complex patterns.
- **Batch Size**: The number of training samples processed together in a single forward and backward pass.
- **Epochs**: The number of complete passes through the entire training dataset during training.
- **Learning Rate**: The step size that controls how much the model's weights are updated during training.[13]

In our case, we tested every hyperparameter combination by training the models multiple times in order to find the best performing combination for each model.

Table 1: Grid Search parameters

Model	**Tuned Hyperparameters**	**Best performing Configuration**
LSTM	Hidden Size: [64, 128, 256] Learning Rate: [0.01, 0.001, 0.0001] Batch Size: [16, 32, 64] Dropout Rate: [0.1, 0.2, 0.5] Layers: [1, 2, 3]	Hidden Size: 128 Learning Rate: 0.001 Batch Size: 32 Dropout Rate: 0.2 Layers: 2
GRU	Hidden Size: [64, 128, 256] Learning Rate: [0.01, 0.001, 0.0001] Batch Size: [16, 32, 64] Dropout Rate: [0.1, 0.2, 0.5] Layers: [1, 2, 3]	Hidden Size: 128 Learning Rate: 0.001 Batch Size: 32 Dropout Rate: 0.2 Layers: 2
CNN	Filter Size: [32, 64, 128] Kernel Size: [3x3, 5x5] Learning Rate: [0.01, 0.001] Pooling Type: [max, average]	Filter Size: 64 Kernel Size: 3x3 Learning Rate: 0.001 Pooling Type: max
RNN	Hidden Size: [64, 128, 256] Learning Rate: [0.01, 0.001, 0.0001] Batch Size: [16, 32, 64] Dropout Rate: [0.1, 0.2, 0.5] Layers: [1, 2, 3]	Hidden Size: 128 Learning Rate: 0.001 Batch Size: 32 Dropout Rate: 0.2 Layers: 2
Random Forest	Number of Trees: [50, 100, 200] Max Depth: [5, 10, 20] Min Samples Split: [2, 5, 10]	Number of Trees: 100 Max Depth: 10 Min Samples Split: 5
K-Nearest Neighbors	Number of Neighbors (k): [3, 5, 7, 9] Distance Metric: [Euclidean, Manhattan] Weights: [Uniform, Distance-based]	Number of Neighbors: 5 Distance Metric: Euclidean Weights: Distance-based
XGBoost	Learning Rate: [0.1, 0.01, 0.001] Max Depth: [3, 6, 9] Subsample Ratio: [0.8, 1.0] Number of Estimators: [50, 100, 150]	Learning Rate: 0.01 Max Depth: 6 Subsample Ratio: 0.8 Number of Estimators: 100
Gaussian Process	Kernel Type: [RBF, Matern, Rational Quadratic] Noise Level: [1e-3, 1e-2, 1e-1] Length Scale: [0.1, 1.0, 10.0]	Kernel Type: RBF Noise Level: 1e-2 Length Scale: 1.0

7. Results

This chapter describes the performance of our machine learning models in predicting seismic activities, including the results obtained from optimization. Several elements of model performance, including accuracy, recall, precision, false positive rate (FPR), G-means, F1 score, Matthews correlation coefficient (MCC), and area under the curve (AUC) were assessed for the performance of each model.

We start off with the pre-optimization results which serve as the primary evaluation of model performance, against which the results achieved after applying the refinement techniques are compared. Lastly, we turn to the post optimization results to validate the effects of hyperparameter tuning and grid search strategies on overall model accuracy and reliability.

7.1. Pre-Optimization Results

Upon applying the Machine Learning Models covered in the previous chapters we collected various performance metrics to provide a comparative overview. In this section, the results obtained before an optimization process is applied are described with a focus on the numerous performance measures achieved which include accuracy, recall, precision, false positive rate (FPR), G-means, F1 score, Matthews correlation coefficient (MCC), and the area under the curve (AUC). Resulting values shed light on every model's ability to make predictions and set a comparison against the metrics collected after the data cleaning, feature extraction, and model's parameters optimization.

Table 2: Modelling Pre Optimization results

Model	Accuracy	Recall	Precision	False Positive Rate (FPR)	G-means	F1 Score	MCC	AUC
LSTM	0.96	0.67	0.86	0.01	0.81	0.75	0.74	0.91
GRU	0.95	0.53	0.89	0.01	0.73	0.67	0.67	0.9
CNN	0.96	0.61	0.9	0.01	0.78	0.73	0.72	0.9
RNN	0.95	0.67	0.8	0.02	0.81	0.73	0.71	0.9
Random Forest	0.96	0.68	0.92	0.01	0.82	0.78	0.77	0.93
K-Nearest Neighbors	0.92	0.77	0.57	0.06	0.85	0.65	0.62	0.89
XGBoost	0.97	0.7	0.92	0.01	0.83	0.79	0.78	0.91
Gaussian Process	0.93	0.73	0.63	0.05	0.83	0.67	0.64	0.83

7.2. Post-Optimization Results

Through the first assessment of machine learning models, optimization approaches were put to the test to improve accuracy. Hyperparameter tuning and Grid search was applied to all the Models being evaluated to optimize their results. Assuming that performing different refinements on different parameters of the models would improve several performance parameters like accuracy, recall, precision etc., various modeling configurations were tested, and we have selected the best optimization output per Model.

Table 3: Modelling Post Optimization results

Model	Accuracy	Recall	Precision	False Positive Rate (FPR)	G-means	F1 Score	MCC	AUC
LSTM	0.96	0.7	0.88	0.01	0.83	0.77	0.75	0.92
GRU	0.96	0.56	0.9	0.01	0.74	0.7	0.69	0.9
CNN	0.96	0.63	0.92	0.01	0.79	0.76	0.73	0.91
RNN	0.96	0.69	0.82	0.02	0.82	0.75	0.72	0.91
Random Forest	0.97	0.71	0.94	0.0	0.84	0.81	0.79	0.94
K-Nearest Neighbors	0.93	0.79	0.6	0.05	0.86	0.69	0.63	0.9
XGBoost	0.97	0.72	0.94	0.0	0.85	0.82	0.79	0.94
Gaussian Process	0.94	0.76	0.66	0.04	0.85	0.71	0.69	0.89

8. Summary

The following evaluation metrics were used to compare the modeling results:

- **F1 Score:** A measure of a test's accuracy which is calculated by taking both false positives and false negatives into consideration. Especially applicable in situations where there is a disproportionate ratio of sample elements.
- **AUC (Area Under the Curve):** Evaluates the ability of the model to differentiate one class from another. Positive AUC values imply a higher degree of separation between the positive and negative classes.
- **Accuracy:** The ratio of correctly predicted outcomes to all the predicted outcomes. For skewed or imbalanced datasets accuracy may not be viably reliable.
- **Recall (Sensitivity):** Measures the percentage of relevant instances that are retrieved by the system. High recall ensures that we do miss events that actually happened.
- **MCC (Matthews Correlation Coefficient):** Informative metric that combines true positive, true negative, false positive and false negative results in classification tasks.

Figure 21 illustrates the F1 Scores across different models with and without hyperparameter tuning. As shown in Figure 21, the F1 Score for all models improved after tuning was applied. The greatest advances were noted during XGBoost, K-Closest Neighbors, Gaussian Process, while also having meaningful improvement in LSTM, GRU, CNN, RNN. F1 is a harmonic mean of precision and recall, which balances false positives and false negatives, which in our case serves as a good metric since our data is indeed imbalanced.

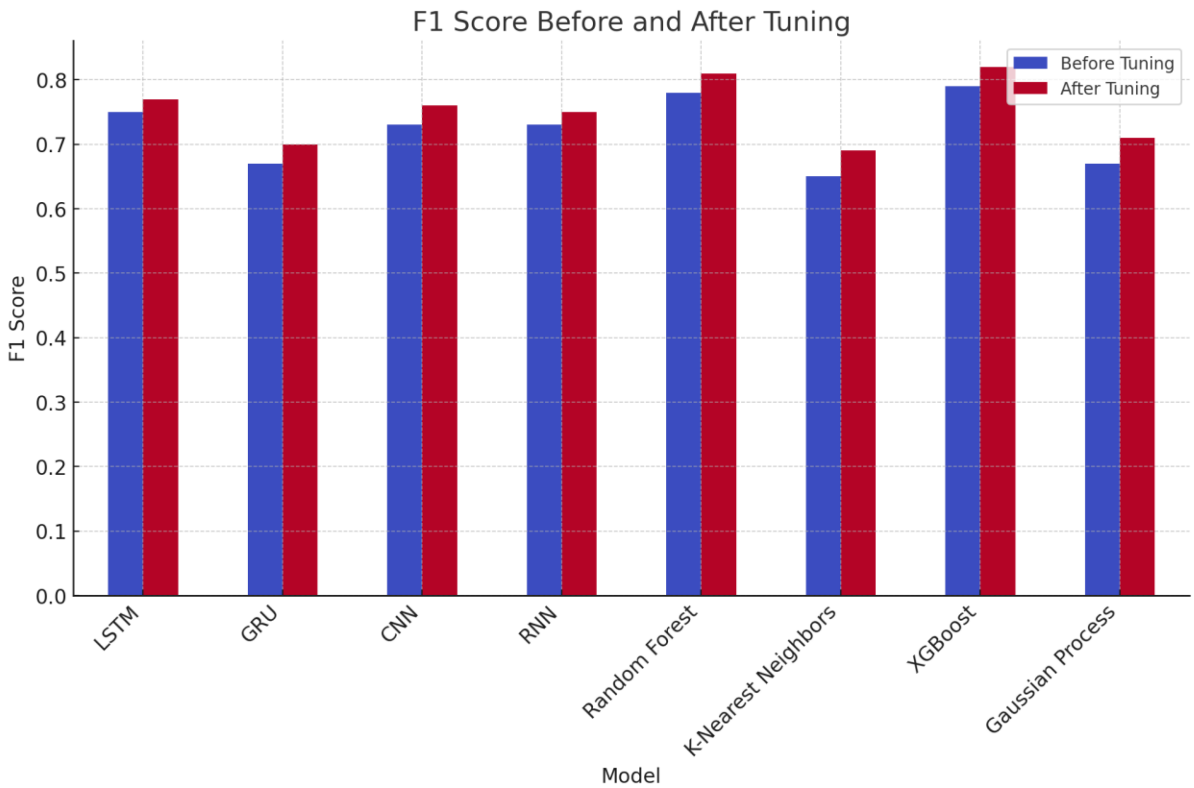

Figure 21: F1 Score Model comparison

A detailed assessment of the improvements made on the F1 Score are depicted in Figure 22. The results indicate that XGBoost attained the greatest enhancement followed in turn by K-Nearest Neighbors and Gaussian Process. Of the deep learning models, LSTM and GRU have shown slight improvements which signify that being optimized has made a positive impact on their classification capability.

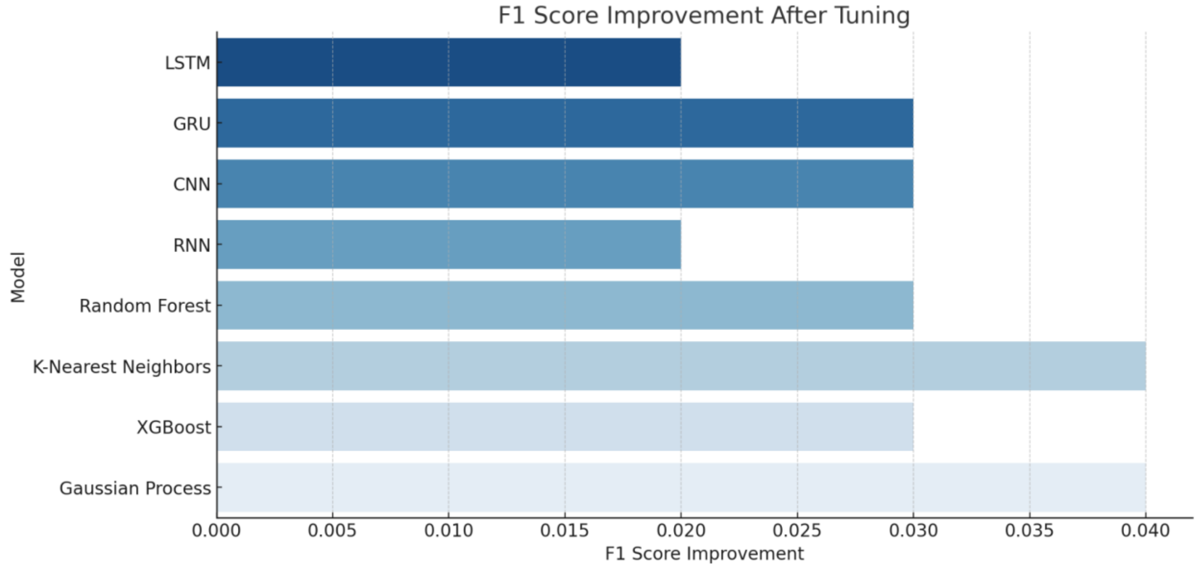

Figure 22: F1 Score Model Improvement

AUC (Area Under the Curve): Indicates the effectiveness in distinguishing different classes. AUC of 1 means perfect prediction and of 0 means random guessing. As observed in Figure 23, the AUC Score improvement analysis was highest in the Gaussian Process and XGBoost models and further justified their utility in the earthquake prediction challenges. Whereas LSTM, GRU and CNN also gained improvement.

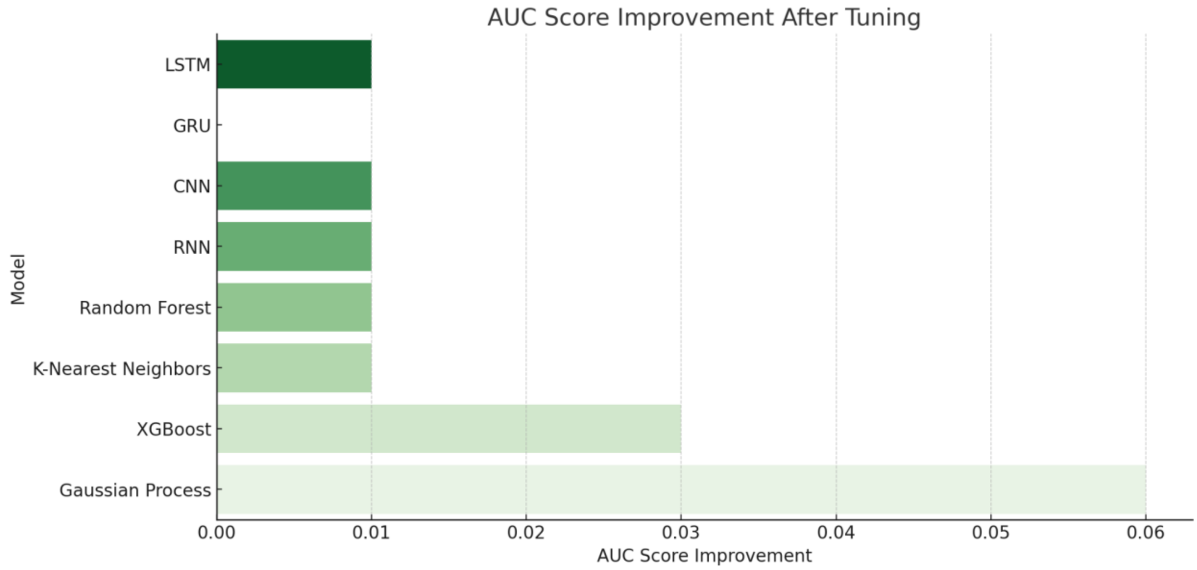

Figure 23: AUC Score Model Improvement

A cumulative analysis of the changes in Improvement for Accuracy, Recall, F1 Score, AUC, and MCC are consolidated in Figure 24. Gaussian Process displayed the most improvement in

AUC and MCC, while K- Nearest Neighbors had marked improvement in F1 Score. Deep learning-based models like LSTM, GRU, and CNN had consistent enhancements for the Recall and F1 Score Measures. It is apparent from these results that although all the models 'performed' better after tuning, there was a relative gain for all models.

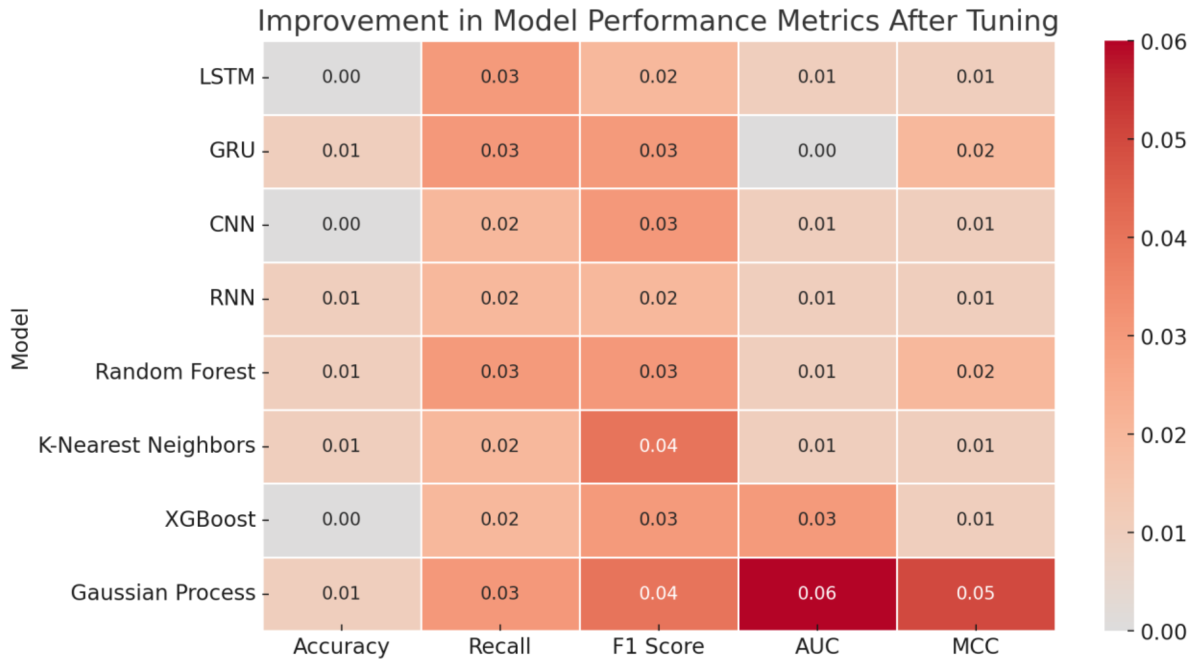

Figure 24: Post Optimization Model Comparison Heatmap

The heat maps depicting the correlations before and after the tuning processes are portrayed in Figure 25. The most important points are as follows:

- After tuning, the correlation between Accuracy and F1 Score improved which showed a positive trend as the classification model predictions became more consistent.
- The trade-off of some models that may have tried to achieve higher Recall but in return lowered their precision is suggested through the negative trend of Recall correlation with Accuracy.
- The improvement in the Gaussian Process and XGBoost models was further corroborated by significantly higher correlations between AUC and MCC after tuning which proved the efficacy of these models.

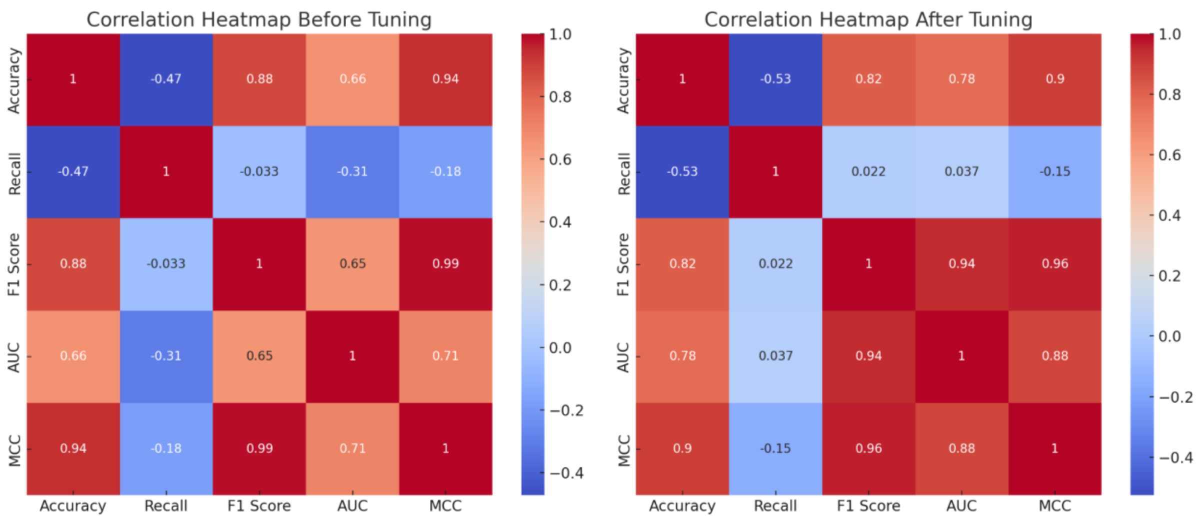

Figure 25: Pre vs Post Correlation Heatmap

Figure 26 presents the difference in distributions of F1 Scores before and after tunings. It can be observed that the median performance of all of the models improved and so did the variance, though to a smaller extent. Hence, this implies that there was an increase in reliability.

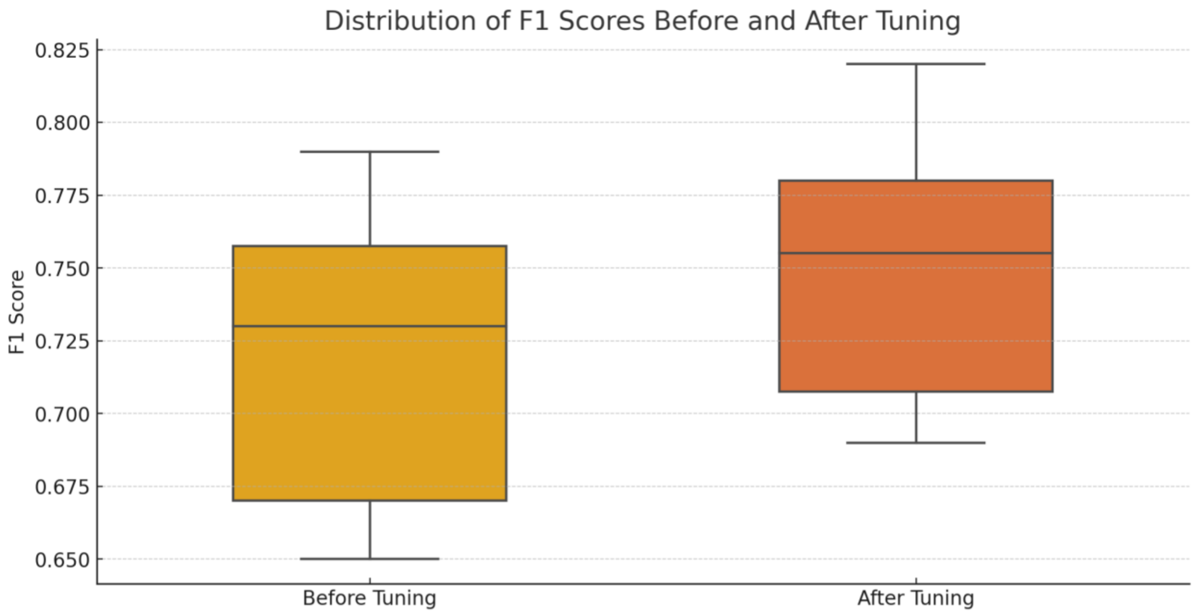

Figure 26: Pre vs Post F1 Score Distribution range

Furthermore, in Figure 27, displays the distribution of the improvements for different metrics which indicates that the maximum improvements were achieved in F1 Score and Recall, while MCC and AUC had relatively smaller but still significant improvements.

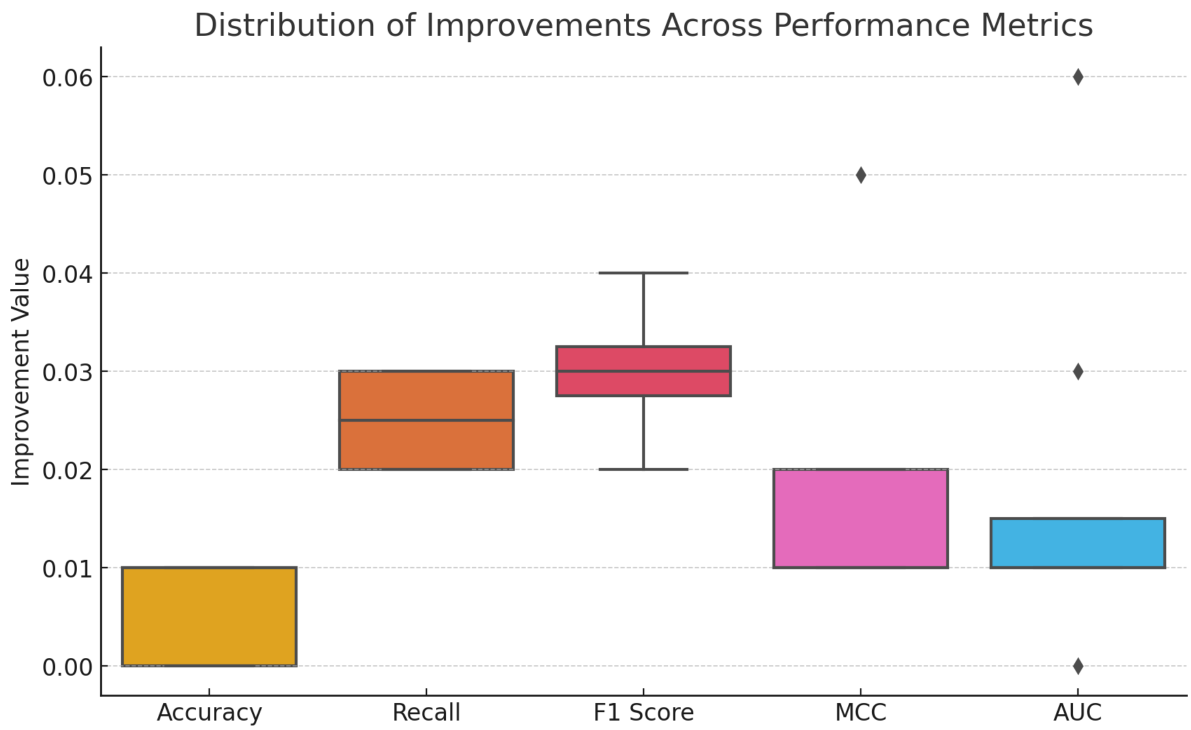

Figure 27: Distribution range for different ML metrics across Models

In this study we used advanced machine learning and deep learning techniques, balancing methods, feature selection, and model optimization strategies. By integrating data preparation, strategic resampling, feature engineering, and model tuning, we aimed to enhance the predictive accuracy and reliability of various models applied to seismic data.

Table 4: Best Models per metric

Model	Best in Metric	Precision	Recall	G-means	F1 Score	MCC
Random Forest	Precision F1 Score MCC	0.94	0.71	0.84	0.81	0.79
K-Nearest Neighbors	Recall G-means	0.6	0.79	0.86	0.69	0.63

References

[1] Abokadr, S., Azman, A., Hamdan, H., & Amelina, N. (2023). Handling imbalanced data for improved classification performance: Methods and challenges. *2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*. IEEE. https://ieeexplore.ieee.org/document/10293442

[2] Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *arXiv preprint arXiv:1206.GHF3*.

[3] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

[4] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[5] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[6] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21–27.

[7] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

[8] Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks*. Springer Science & Business Media.

[9] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems, 28*(10), 2222–2232.

[10] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation, 12*(10), 2451–2471.

[11] Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*(1–3), 389–422.

[12] Hayakawa et al. (1996a, 1996b); Molchanov et al. (1998).

[13] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

[14] Keskar, N. S., et al. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

[15] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30*, 4765–4774.

[16] Politis, D. Z., Sasmal, S., Hayakawa, M., Haralambous, H., Datta, A., & Potirakis, S. M. (2024). A six-year (2014–2020) statistical correlation study of VLF terminator time shift with earthquakes in Japan. *Remote Sensing, 16*(22), 4162. https://doi.org/10.3390/rs16224162

[17] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

[18] Molchanov, O. A., & Hayakawa, M. (1998). Subionospheric VLF signal perturbations possibly related to earthquakes. *Journal of Geophysical Research: Space Physics, 103*(A8), 17,489–17,504. https://doi.org/10.1029/98JA00999

[19] Cerveny, V., & Soares, J. E. P. (1992). Seismic wave propagation and Fresnel zones in heterogeneous media. Geophysical Journal International, 108(1), 42–54