VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELGAVI-590018

A Project Report on

"Enhancing Seismic Data with Generative Adversarial Network for Affordable MEMs Sensors"

Submitted in fulfilment of the requirements for the award of the Degree of

Bachelor of Engineering
in
Computer Science and Engineering

submitted by

SPARSH BISEN [1DB21CS148]
SURYA M [1DB21CS155]
TEJASWINI P [1DB21CS158]
U JEEVAN HARI [1DB21CS161]

Under the Guidance of

Dr. Venugeetha Y
Professor
Dept. of CSE

Don Bosco Institute of Technology

Kumbalagodu, Mysore Road, Bengaluru-560 074

2024 - 2025

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

DON BOSCO INSTITUTE OF TECHNOLOGY

Kumbalagodu, Mysore Road, Bengaluru-560 074

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project entitled **"Enhancing Seismic Data with Generative Adversarial Network for Affordable MEMs Sensors"** has been successfully completed by SPARSH BISEN (1DB21CS148), SURYA M(1DB21CS155), TEJASWINI P(1DB21CS158) and U JEEVAN HARI (1DB21CS161) the bonafide students of the Department of Computer Science & Engineering, Don Bosco Institute of Technology of the Visvesvaraya Technological University, Belagavi – 590014, during the year 2024–2025. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated with the degree mentioned.

Signature of Guide
Dr. Venugeetha Y
Professor,
Dept. of CSE,
DBIT, Bengaluru

Signature of HOD
Dr. K B ShivaKumar
Professor and Head,
Dept. of CSE,
DBIT, Bengaluru

Signature of Principal
Dr. Nagabhushana B S
Principal,
DBIT, Bengaluru

External Viva:

Name of the Examiner **Signature with Date**

1) ____________________ ____________________

2) ____________________ ____________________

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned the efforts with success.

We would like to profoundly thank **Management** of **Don Bosco Institute of Technology** for providing such a healthy environment for the successful completion of project work.

We would like to express our thanks to **Dr. Nagabhushana B S,** Principal, Don Bosco Institute of Technology, Bengaluru for his encouragement that motivated us for the successful completion of project work.

It gives us immense pleasure to thank **Dr. K B ShivaKumar**, Professor and Head, Department of Computer Science and Engineering for his constant support and encouragement.

We would like to express our deepest sense of gratitude to our **Project Co- coordinator Dr. Asha K H,** Associate Professor, Department of Computer Science & Engineering for their constant support and guidance throughout the project work.

We would like to express our deepest sense of gratitude to our guide **Dr. Venugeetha Y,** Professor, Department of Computer Science & Engineering for her constant support and guidance throughout the project work.

Also, we would like to thank all teaching and non-teaching staff of Department of Computer Science & Engineering who have helped directly and indirectly throughout the project work.

Sparsh Bisen (1DB21CS148)
Surya M (1DB21CS155)
Tejaswini P (1DB21CS158)
U Jeevan Hari (1DB21CS161)

ABSTRACT

This project delves into the realm of seismic data processing, focusing particularly on data collected from low-cost Micro-Electro-Mechanical Systems (MEMS) sensors. These sensors have garnered significant attention in recent years due to their affordability and accessibility. However, the data they capture often suffer from various types of noise, posing a challenge for accurate analysis and interpretation. To address this challenge, the application explores the advanced machine learning techniques, with a primary focus on Generative Adversarial Networks (GANs), for augmenting seismic data. GANs have shown promise in generating synthetic data that closely resembles real-world samples, making them well-suited for addressing data scarcity issues in seismic research. The project conducts a comparative analysis of different unsupervised noise removal techniques, including Principal Component Analysis (PCA), Independent Component Analysis (ICA), and GAN models. By evaluating the effectiveness of each method in enhancing the quality of seismic data, the aim is to identify the most suitable approach for mitigating noise artifacts and improving data fidelity. Furthermore, supervised learning techniques such as Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks are employed to explore the feasibility of earthquake prediction using labeled data. By leveraging historical seismic data along with associated earthquake labels, these models are trained to recognize patterns and signals indicative of impending seismic events. The results of the project indicate that GANs offer promising outcomes in augmenting seismic data, effectively reducing noise contamination and enhancing the authenticity of generated samples. Moreover, the project highlights the potential of Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) models for earthquake prediction, demonstrating their ability to discern meaningful patterns and provide valuable insights into seismic activity. Overall, this project contributes to the advancement of data augmentation and earthquake prediction techniques in seismology. By exploring innovative machine learning approaches and evaluating their performance in real-world seismic datasets, this offers valuable insights for enhancing the effectiveness of seismic monitoring systems and improving our understanding of earthquake dynamics.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

Seismic monitoring is a critical element in the study and prediction of earthquakes, which are among the most destructive natural disasters, capable of causing extensive damage to infrastructure and significant loss of life. Understanding seismic activity is not only vital for scientific research but also essential for disaster preparedness and mitigation efforts. Traditional seismic sensors, typically employed in seismographs, provide high-quality and accurate data. However, these sensors are often prohibitively expensive, limiting their widespread deployment. This cost barrier is particularly problematic in regions that are most vulnerable to seismic activity, where comprehensive monitoring networks are crucial for early warning systems and effective response strategies.

In recent years, technological advancements have introduced low-cost MEMS (Micro-Electro-Mechanical Systems) sensors as a viable alternative for seismic data collection. These sensors are significantly more affordable than their traditional counterparts, enabling broader and more frequent monitoring. The cost-effectiveness of MEMS sensors opens the possibility for extensive seismic networks even in economically constrained regions. However, a significant drawback of MEMS sensors is their high susceptibility to noise, which can severely compromise the accuracy and reliability of the seismic data they collect. This noise issue poses a challenge for their effective use in seismic monitoring and necessitates advanced data processing techniques to ensure data quality.

This project seeks to address the noise inherent in seismic data collected from MEMS sensors by utilizing advanced data processing techniques. Specifically, it investigates the potential of Generative Adversarial Networks (GANs) to enhance the quality of seismic data. GANs, a class of machine learning frameworks, are particularly adept at generating and refining data through an adversarial process, making them suitable for noise reduction in an unsupervised manner. The project will compare the performance of GANs with conventional noise reduction methods such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA). These traditional methods have been widely used for signal processing and data reduction, and their comparison with GANs will provide insights into the most effective approach for improving MEMS sensor data.

In addition to noise reduction, the project explores supervised learning approaches for earthquake prediction using labeled seismic data. Earthquake prediction is a complex task that involves identifying patterns and anomalies in seismic data that precede seismic events. This project examines the efficacy of Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks in enhancing earthquake prediction accuracy. SVM is a robust classification technique known for its effectiveness in high-dimensional spaces, while LSTM networks, a type of recurrent neural network (RNN), are well-suited for time-series prediction due to their ability to capture long-term dependencies.

By leveraging both unsupervised and supervised techniques, the project aims to significantly improve the capabilities of earthquake prediction systems. The integration of noise reduction and predictive modeling addresses both the quality and utility of seismic data, providing a comprehensive approach to seismic monitoring.

The outcomes of this project hold substantial implications for the field of seismic monitoring and disaster preparedness. Enhanced data quality from MEMS sensors and improved earthquake prediction models contribute to more effective monitoring systems, which can lead to timely warnings and better-informed response strategies. By improving the efficiency and affordability of seismic data collection, this project supports the development of resilient infrastructure and communities, particularly in regions prone to seismic hazards.

1.1 Preamble

In the face of increasing natural disasters, particularly earthquakes, effective seismic monitoring and prediction are crucial for mitigating risks and ensuring the safety of communities. Earthquakes have the potential to cause widespread destruction, leading to loss of life, significant economic damage, and long-term social disruption. Accurate and timely seismic monitoring is therefore essential to provide early warnings, guide emergency responses, and inform infrastructure design and land-use planning. However, traditional seismic monitoring methods, which rely on high-precision equipment such as broadband seismometers and strong-motion accelerographs, often come with high costs. These expenses limit the extensive deployment of such equipment, particularly in regions with high seismic risk but limited financial resources. As a result, many vulnerable areas remain inadequately monitored, increasing the potential for catastrophic impacts when earthquakes occur.

The emergence of low-cost MEMS (Micro-Electro-Mechanical Systems) sensors offers a promising solution to the financial barriers associated with traditional seismic monitoring. MEMS sensors are compact, energy-efficient, and cost-effective, making them suitable for widespread deployment. They provide an opportunity to establish dense seismic networks that can enhance spatial coverage and improve the detection and analysis of seismic events. Despite their affordability and practical advantages, MEMS sensors face significant challenges, primarily related to their high susceptibility to noise. This noise can distort the seismic signals, making it difficult to accurately detect and analyze seismic events. Consequently, the reliability of the data collected by MEMS sensors is often questioned, posing a significant challenge to their effective utilization in seismic monitoring.

This project seeks to address these challenges by employing advanced data processing techniques to mitigate noise and improve the quality of seismic data collected from MEMS sensors. One of the primary techniques explored in this project is the use of Generative Adversarial Networks (GANs). GANs are a class of machine learning models that consist of two neural networks: a Generator and a Discriminator, which work together in a competitive process to enhance data quality. The generator creates synthetic data samples, while the discriminator evaluates them against real data, iteratively improving the generator's output. This adversarial process is particularly effective for noise reduction, as it can learn to distinguish and suppress noise patterns in the seismic data, resulting in cleaner and more reliable signals.

In addition to GANs, the project investigates traditional noise reduction methods such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA). PCA is a statistical technique that transforms data into a set of orthogonal components, emphasizing variance and reducing dimensionality, which can help isolate and remove noise. ICA, on the other hand, aims to separate a multivariate signal into additive, independent components, often used for blind source separation. Comparing the performance of GANs with these conventional methods provides a comprehensive understanding of the most effective approaches for noise mitigation in seismic data.

Beyond noise reduction, this project also explores supervised learning approaches for earthquake prediction using labelled seismic data. Accurate earthquake prediction involves identifying patterns and anomalies that precede seismic events, which can be challenging due to the complex and non-linear nature of seismic processes. This project examines the efficacy of Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks in predicting earthquakes. SVMs are powerful classifiers capable of handling high-dimensional data and identifying complex patterns, while LSTM networks, a type of recurrent neural network, excel at modelling sequential data and capturing long-term dependencies, making them suitable for time-series prediction tasks.

By integrating unsupervised and supervised methodologies, this project aims to create a holistic approach to seismic monitoring that enhances the efficiency, affordability, and accuracy of seismic systems. The outcomes of this research have significant implications for disaster preparedness and risk mitigation efforts. Improved data quality and predictive capabilities can lead to more reliable early warning systems, better - informed decision-making, and more effective emergency response strategies. Ultimately, this project contributes to the advancement of seismic monitoring technologies, supporting the development of resilient communities and infrastructure capable of withstanding the impacts of earthquakes.

1.2 Problem Statement

The growing reliance on low-cost MEMS sensors for seismic activity monitoring has gained significant attention due to the high cost of conventional sensors. However, the data collected by MEMS sensors is often polluted by various types of noise, posing challenges for accurate earthquake detection. To address the lack of high-quality seismic datasets, there is a critical need to enhance these datasets through intelligent data-augmentation techniques, enabling the generation of synthetic seismic data and improving the reliability of seismic activity monitoring systems.

1.3 Aim and Objectives of the Proposed Work

Aim:

The aim of this Project is to compare the effectiveness of Long Short-Term Memory (LSTM) and Support Vector Machine (SVM) models in earthquake prediction, and to evaluate the performance of Generative Adversarial Networks (GAN), Independent Component Analysis (ICA), and Principal Component Analysis (PCA) for noise reduction in seismic data. By augmenting the dataset to simulate real-time noise conditions, the project seeks to determine the most accurate and efficient methods for improving earthquake detection and data clarity.

Objectives:

1. Investigate the effectiveness of Generative Adversarial Networks (GANs) in augmenting seismic data collected from MEMS sensors to reduce noise levels.

2Explore supervised learning approaches, including Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks, for earthquake prediction using labeled seismic data.

1.4 Proposed System

Advantages of Proposed System:

• **Cost-effectiveness**: Utilizing low-cost MEMS sensors reduces the financial burden associated with seismic data collection, making monitoring more accessible, especially in resource-constrained regions.

• **Improved Data Quality**: Advanced data processing techniques such as GANs enhance data quality by effectively mitigating noise inherent in seismic signals, leading to more reliable and accurate data for analysis.

• **Enhanced Prediction Accuracy**: By leveraging supervised learning approaches, the system improves earthquake prediction accuracy, enabling timely alerts and proactive measures to mitigate potential risks.

• **Comprehensive Framework**: Integrating both unsupervised and supervised methodologies provides a comprehensive framework for seismic data processing and prediction, offering a holistic approach to seismic monitoring and risk management.

• **Increased Efficiency**: The system streamlines seismic monitoring processes, facilitating quicker data analysis and decision-making, ultimately enhancing disaster preparedness and response capabilities.

• **Scalability**: Due to the lower costs, the system can be scaled up more easily, covering wider geographical areas and providing more comprehensive seismic monitoring.

• **Accessibility**: The use of low-cost sensors and advanced data processing techniques democratizes access to seismic monitoring technology, making it available to a broader range of users and regions.

Models Implemented:

1. **PCA (Principal Component Analysis)**
    - **Description:** PCA is a statistical technique used for dimensionality reduction and noise filtering. It transforms the data into a set of orthogonal components, emphasizing variance and reducing complexity.
    - **Purpose in Project:** PCA is employed to analyze its effectiveness in filtering noise from seismic data.
2. **ICA (Independent Component Analysis)**
    - **Description:** ICA is a computational method for separating a multivariate signal into independent components. It is often used for blind source separation in signal processing.
    - **Purpose in Project:** ICA is used to assess its capability in isolating and removing noise from seismic data, providing a clean signal.
3. **Comparison**

    Description: The central box labeled "Comparison" signifies the core part of the project where different noise reduction and data enhancement techniques are evaluated against each other.
4. **SVM (Support Vector Machines)**
    - **Description:** SVM is a supervised learning model used for classification and regression analysis. It finds the optimal hyperplane that maximizes the margin between different classes.
    - **Purpose in Project:** SVMs are used for earthquake prediction based on labeled seismic data. Their performance in predicting seismic events is evaluated.
5. **LSTM (Long Short-Term Memory Networks)**
    - **Description:** LSTM networks are a type of recurrent neural network (RNN) capable of learning long-term dependencies and order dependence in sequence prediction problems.
    - **Purpose in Project:** LSTMs are implemented for earthquake prediction to capture temporal patterns and dependencies in seismic data.
6. **GAN (Generative Adversarial Networks)**
    - **Description:** GANs consist of two neural networks—the generator and the discriminator—that compete against each other. The generator creates synthetic data, while the discriminator evaluates them against real data.
    - augment seismic data and reduce noise, enhancing the overall quality of the data.

Comparison between Existing System and Proposed System:

- The proposed system for enhancing seismic data using Generative Adversarial Networks (GANs) with affordable MEMS sensors offers a cost-effective and scalable alternative to existing systems reliant on expensive high-quality seismic sensors like geophones.

- Unlike traditional methods, which provide high-quality data directly but at a significant cost, the proposed system utilizes low-cost MEMS sensors to collect raw seismic data. While these sensors inherently produce noisy and low-resolution data, the GAN-based model enhances this data to a quality comparable to that of high-end sensors.

- This is achieved through adversarial training, where the generator learns to transform noisy input into high-quality seismic data, and the discriminator validates its accuracy against real high-quality data. The system reduces deployment costs, making seismic monitoring accessible to smaller organizations and underserved regions.

- Furthermore, it eliminates the dependency on fixed, high-cost infrastructure and provides a scalable solution for applications such as earthquake monitoring, micro seismic event detection, and resource exploration.

Chapter 2

Literature Survey

Earthquake detection [1, 2, 3] and earthquake early warning (EEW) [4, 5] are the main tasks of seismological research. Many data processing techniques used in traditional seismological research originated from small datasets and limited computing power. Low- cost MEMS acceleration sensors have been extensively used in the monitoring system of Internet of Things (IoT) over the last few years, because of their low installation and operation costs, the examples include a wireless sensor network (WSN) [6,7], community seismic network (CSN) [8]. Although they have a great potential to replace the traditional expensive seismic networks whose coverage is hardly dense due to the high installation and operation costs, however, the large noises inherent in a low-cost MEMS acceleration sensor reduce the quality of data recorded [9], thus a novel approach is needed to adapt to the data with different Signal-to-Noise Ratios (SNR).In recent years, the Machine Learning (ML) has been widely applied to earthquake detection [10,11,12,13],including earthquake first arrival recognition [14,15,16] and source location [17,18].Compared with other time series (stock market price, WiFi signals), a high dimensional seismic sequence has many implict features (evolution process of different components and a single component, etc.) that are difficult to capture. Khan et al. [19] developed an Artificial Neural Network (ANN) model [20,21] to detect seismic events by artificially selecting labels.

Also, researchers have developed different seismic detection models based on different Convolutional Neural Network (CNN) methods [3,22,23]. Nevertheless, no matter which method is adopted, the abovementioned models are supervised. A real seismic waveform needs to be identified by subjectively selecting feature labels, which will affect the detection performance of the model. Therefore, seismic detection methods based on the ML gradually used to eliminate the influence of subjective factors. Considering that the performance of the DL algorithm depends on the size and quality of the training dataset, in our work, however, we utilize low-cost MEMS sensors to record ground motion signals instead of smartphones, which are polluted by different noise levels (human activities or the sensor themselves), resulting in the lack of high-quality seismic data (High SNR). Too few training datasets easily leads to overfitting [5].

Therefore, to solve this problem, the current research is mainly based on the following three solutions:

• To Use Transformation: To overcome the problems of incomplete real seismic waveform and low SNR, Dokht et al. [24] designed a general deep convolution network model for the seismic event and phase detection based on time-frequency representation and convolution neural network. Saad et al. [25] used automated unsupervised approaches to extract waveform signals from continuous micro seismic data according to the time-frequency representation of micro seismic trajectory, which was also applicable in an environment with a low SNR, corning that the waveform-based inverse time migration method could be used in the model to improve the resolution of micro seismic imaging.

• To Train Model Using a Generalized Deep Learning Model [31,32] Based on a Small Dataset: Using a generalized deep learning architecture to extract the most representative features from limited/small training datasets, Saad et al. [26] successfully proposed the SCALODEEP model to detect ground demand signals. Similarly, Zhu et al. [27] proposed a CNN-based Phase Recognition Classifier (CPIC) for phase detection and picked up from small and medium-sized training datasets. While Saad et al. [28] used a Capsule neural Network [33] (CapsNet) to identify and detect earthquakes automatically and confirmed that it could learn from small datasets with a good generalization performance.

• To Develop a Data Augmentation-Approach: Data augmentation is also an effective method to increase data samples. The conditional GAN [29] was used to generate the seismic dataset effectively. Wang et al. [30] developed the Earth quake Gen to generate a short seismic waveform and verified its rationality. Although seismic data can be generated [34,35] by inferring the implicit and explicit characteristics of the seismic waveform, it is not easy to ensure the diversity or efficiency of the data generated In-Depth Exploration of Methodologies:

• Ensemble Learning for Improved Robustness: Researchers are exploring ensemble methods that combine multiple deep learning models to enhance prediction accuracy and robustness in seismic event detection. Techniques like bagging, boosting, and stacking are being adapted to this domain to amalgamate diverse model predictions effectively.

• Graph Neural Networks (GNNs) for Seismic Graph Data: Graph-based approaches particularly Graph Neural Networks (GNNs), are gaining prominence in analyzing seismic data represented as graphs. GNNs facilitate the modeling of spatial and temporal dependencies in seismic networks or fault systems, enabling better understanding and prediction of seismic events.

• Adversarial Training and Robustness Enhancement: Adversarial training methods are being explored to improve the robustness of deep learning models against adversarial attacks and variations in seismic data. Adversarial examples crafted to perturb seismic signals are used to train models to become more resilient to such perturbations.

• Transition from Traditional Historical Methods: Traditional seismological research heavily relied on small datasets and limited computing power, shaping data processing techniques. However, these methods were constrained by the quality of data and computing capabilities of that time.

• Advent of MEMS Acceleration Sensors: The emergence of low-cost MEMS acceleration sensors in IoT-based monitoring systems, such as Wireless Sensor Networks (WSN) and Community Seismic Networks (CSN), brought promise due to their affordability but raised challenges due to inherent noise impacting data quality.

• Machine Learning in Earthquake Detection Applications in Seismology: Machine Learning techniques have gained significant traction in earthquake detection, including earthquake first arrival recognition and source location. ML models offer potential advantages in handling high-dimensional seismic sequences and implicit features within them.

• Supervised Approaches: Initial ML-based seismic detection models were supervised, requiring the subjective selection of feature labels from real seismic waveforms, potentially impacting detection performance.

• Eliminating Subjective Factors: Recent trends focus on reducing the influence of subjectivity in seismic detection by moving away from subjective feature labeling. This shift aims to enhance the detection performance of models by leveraging ML methods that don’t heavily rely on manual feature selection.

Chapter 3

System Requirements and Specification

3.1 Hardware and Software requirements

Hardware Requirements

- Processor: Intel Core 3 and above
- RAM: 8GB
- Operating System: Windows 10

Software Requirements

- Programming language: Python 3.9
- IDE: Google Colab

3.2 Libraries Used in the Project

The libraries used in the Project is as follows:

1. **ObsPy:**

   Description: ObsPy is an open-source project dedicated to providing a Python framework for processing seismological data. It includes a variety of tools for reading, writing, and manipulating seismic waveform data.

   Installation: pip install obspy

   from obspy import Trace, Stream, UTCDateTime

2. **Pandas:**

   Description: Pandas is a powerful data manipulation and analysis library for Python. It provides data structures such as Data Frames, which are essential for handling structured data and performing operations such as merging, reshaping, and aggregating datasets.

   Installation: pip install pandas

   import pandas as pd

3. **NumPy:**

Description: NumPy is the fundamental package for scientific computing in Python. It provides support for arrays, matrices, and many mathematical functions to operate on these data structures efficiently.

Installation: pip install numpy

import numpy as np

4. **Matplotlib:**

Description: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is widely used for plotting graphs, histograms, bar charts, and other types of plots.

Installation: pip install matplotlib

import matplotlib.pyplot as plt

5. **Scikit-learn:**

Description: Scikit-learn is a machine learning library for Python that provides simple and efficient tools for data mining, data analysis, and machine learning. It includes a wide range of algorithms for classification, regression, clustering, and dimensionality reduction.

Installation: `pip install scikit-learn`

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import StandardScaler, MinMaxScaler

from sklearn.decomposition import PCA, FastICA

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score

6. **TensorFlow:**

Description: TensorFlow is an open-source platform for machine learning developed by Google. It provides a comprehensive ecosystem for building and deploying machine learning models, including neural networks.

Installation: pip install tensorflow

```
import tensorflow as tf
from tensorflow.keras import layers, models
```

7. **SciPy:**

Description: SciPy is a Python library used for scientific and technical computing. It builds on NumPy and provides additional functionality for optimization, integration, interpolation, eigenvalue problems, and other advanced mathematical computations.

Installation: pip install scipy

```
from scipy.signal import resample_poly
```

8. **Keras:**

Description: Keras is an open-source neural network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Keras is user-friendly, modular, and easy to extend.

Installation: `pip install keras`

```
from keras.models import Sequential
from keras.layers import LSTM, Dense
```

Chapter 4

System Architecture and Implementation

4.1 Methodology

- PCA (Principal Component Analysis):
  - Missing values in the seismic data are imputed using the mean strategy.
  - Features are standardized using Standard Scaler.
  - PCA is applied to reduce the dimensionality of the data to 2 principal components.
  - Explained variance ratio is calculated and plotted to determine the optimal number of principal components.
- ICA (Independent Component Analysis):
  - Similar preprocessing steps are applied, including imputation and scaling.
  - ICA is used to decompose the data into independent components.
  - The original and augmented data are plotted along with the ICA-transformed augmented data.
- SVM (Support Vector Machine):
  - Data augmentation techniques like resampling are used.
  - The SVM classifier is trained on the preprocessed and augmented data.
  - Model evaluation is performed using accuracy score on the test set.
  - Predicted labels are plotted against the real and augmented data
- GAN (Generative Adversarial Network):
  - Missing values are imputed and features are standardized as in previous steps.
  - GAN architecture is defined, comprising a generator and discriminator.
  - The generator is trained to generate synthetic seismic data that resembles real data.
  - The progress of discriminator and generator losses is printed during training.

- ➢ Original and predicted data are plotted to compare.

- LSTM (Long Short-Term Memory):
  - ➢ Data preprocessing involves imputation and scaling as before.
  - ➢ LSTM model is defined and trained on the preprocessed data.
  - ➢ Model evaluation is performed using loss and accuracy metrics.
  - ➢ Original and predicted labels are plotted for comparison.

4.2 System Design

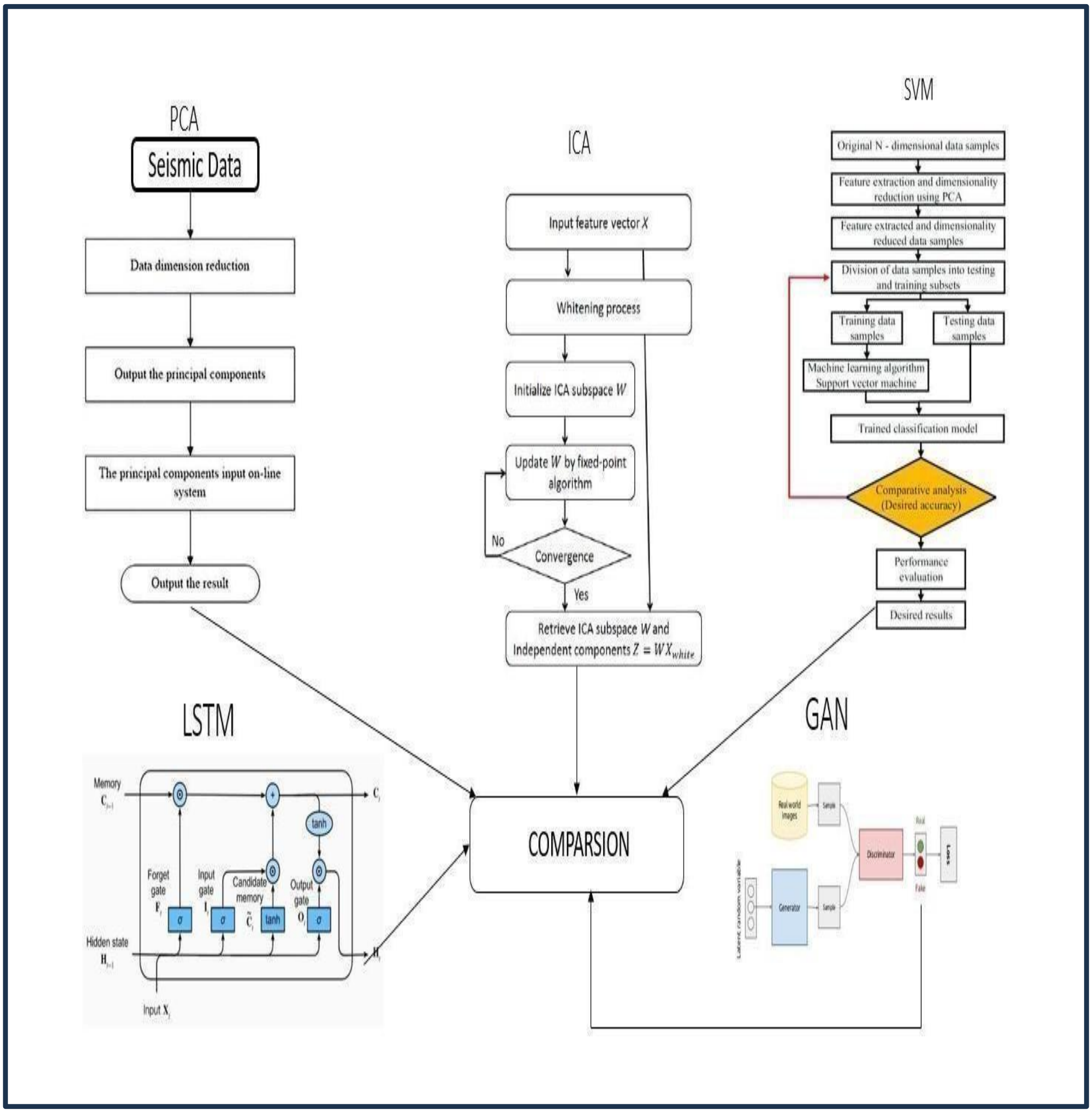


Figure 4.2 System Design

4.3 Pseudo code

Seismic Data Processing and Visualization Pipeline

```
# Import necessary libraries
from obspy import Trace, Stream, UTCDateTime
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# Read CSV files containing seismic data
df_fas = pd.read_csv('ESM_flatfile_FAS.csv', delimiter=';', nrows=50)
df_sa = pd.read_csv('ESM_flatfile_SA.csv', delimiter=';', nrows=50)
df_sd = pd.read_csv('ESM_flatfile_SD.csv', delimiter=';', nrows=50)
# Merge datasets based on event_id
df_merged_fas_sa = pd.merge(df_fas, df_sa, on='event_id', how='inner')
df_combined = pd.merge(df_merged_fas_sa, df_sd, on='event_id',
how='inner')
# Create an empty ObsPy Stream object
stream = Stream()
# Iterate over rows in the combined DataFrame
for index, row in df_combined.iterrows():
    # Create a new ObsPy Trace object
    trace = Trace()
   # Set metadata attributes for the trace
    trace.stats.network = row['network_code']
    trace.stats.station = str(row['station_code'])
    trace.stats.location = str(row['location_code'])
    trace.stats.channel = f"{row['V_channel_code']}_{row.get('frequency_hz'
    , row.get('Frequency'))}"
    trace.stats.starttime = UTCDateTime(row['event_time_x'])
```

```
waveform_data = []
for col_name in df_combined.columns:
    if col_name.startswith('rotD00_T'):
        waveform_data.append(row[col_name])
trace.data = np.array(waveform_data, dtype=float)
# Add station metadata to trace SAC header
trace.stats.sac = {
    'epi_dist': row['epi_dist'],
    'epi_az': row['epi_az'],
    'JB_dist': row['JB_dist'],
    'rup_dist': row['rup_dist'],
    'Rx_dist': row['Rx_dist'],
    'Ry0_dist': row['Ry0_dist'],
    'instrument_type_code': row['instrument_type_code'],
    'late_triggered_flag_01': row['late_triggered_flag_01'],
    'V_channel_code': row['V_channel_code'],
    'V_azimuth_deg': row['V_azimuth_deg'],
    'V_hp': row['V_hp'],
    'V_lp': row['V_lp'],
    'U_pga': row['U_pga'],
    'U_pgv': row['U_pgv'],
    'U_pgd': row['U_pgd'],
    'U_T90': row['U_T90'],
    'U_housner': row['U_housner'],
    'U_ia': row['U_ia'],
    'U_CAV': row['U_CAV'],
    'U_T0_010': UTCDateTime(row['event_time_x']),
}
```

```
    # Append the trace to the ObsPy Stream
    stream.append(trace)
    # Select a specific trace from the stream
    graph = stream[7]
    # Plot seismic data for the selected trace
    graph.plot(type='relative', color='black', size=(800, 600))
    plt.show()
```

Principal Component Analysis:

```
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
# Assuming X contains your seismic data with NaN values
# Assuming real_data contains your real seismic data without NaN values
# Impute missing values
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(stream)
# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)
# Apply PCA
pca = PCA(n_components=2)
X_reduced = pca.fit_transform(X_scaled)
# Plot the reduced data
plt.figure(figsize=(10, 5))
```

```
plt.subplot(121)
plt.scatter(X_reduced[:, 0], X_reduced[:, 1], label='Reduced Data')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Seismic Data')
# Plot real data
plt.subplot(122)
real_data = np.array([trace.data for trace in stream])
# Assuming real_data is a 2D array containing real seismic data
plt.scatter(real_data[:, 0], real_data[:, 1], label='Real Data', color='orange')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Real Seismic Data')
plt.tight_layout()
plt.show()
```

Independent Component Analysis:

1. Import necessary libraries:
   - numpy
   - matplotlib.pyplot
   - sklearn.decomposition.FastICA
   - sklearn.impute.SimpleImputer
   - scipy.signal.resample_poly
2. Define a function to resample data:

```
resample_data(data, target_length):
    resampled_data = resample_poly(data, target_length, len(data))
    return resampled_data
```

3. Set the target length for resampling.

4. Generate or load augmented data and real data:

- Create synthetic augmented data by resampling synthetic data to the target length.

- Create real data by adding random noise to the augmented data.5. Preprocess the data to handle NaN values:

- Create a SimpleImputer object with the strategy set to 'mean'.

- Fit the imputer on the augmented data and transform it.

- Transform the real data using the same imputer.

6. Apply Independent Component Analysis (ICA) for dimensionality reduction:

- Create a FastICA object with the desired number of components.

- Fit and transform the augmented data using ICA.

- Transform the real data using the fitted ICA model.

7. Plot the results:

- Create three subplots using matplotlib for visualization.

- Scatter plot 1: Real data

- Scatter plot 2: ICA of real data

- Scatter plot 3: ICA of augmented data

- Each plot visualizes the independent components against each other.

8. Display the plots using plt.show().

Support Vector Machine:

1. Import necessary libraries:

- numpy

- pandas

- train_test_split from sklearn.model_selection

- Trace, Stream from obspy

- resample_poly from scipy.signal

- MinMaxScaler from sklearn.preprocessing

- SVC from sklearn.svm

- accuracy_score from sklearn.metrics

- SimpleImputer from sklearn.impute

2. Define a function to resample data:

   - resample_data(data, target_length):

      - resampled_data = resample_poly(data, target_length, len(data))

      - return resampled_data

3. Split data into training and testing sets:

   - Split the data into training and testing sets using train_test_split.

   - Adjust test_size and random_state as needed.

4. Convert the DataFrames back to Stream for training:

   - Convert the DataFrames obtained from the split into Stream objects.

5. Resample or pad the data to ensure a length of 1000:

   - Use the resample_data function to resample each trace in the streams to a target length of 1000.

6. Convert the Stream to a list of NumPy arrays for DataFrame creation:

   - Convert the resampled streams to lists of NumPy arrays.

7. Reshape the data to have a shape of (batch_size, 1000, 1):

   - Expand the dimensions of the arrays to include a channel dimension.

8. Reshape train and test arrays to 2D:

   - Reshape the arrays to be 2D.

9. Rescale or Normalize Data:

   - Use MinMaxScaler to scale the data between 0 and 1.

10. Impute NaN values:

   - Use SimpleImputer to handle any NaN values in the data.

11. Prepare target labels for training:

- Create target labels, assuming the first half of the training data is actual earthquake data and the second half is non-earthquake data.

12. Train SVM classifier:

- Initialize an SVC model with desired parameters (kernel='rbf', gamma='scale', random_state=42).

- Fit the model on the training data and target labels.

13. Predict on test set:

- Use the trained model to predict on the test data.

14. Evaluate model:

- Assuming the first half of test_data is actual earthquake data and the second half is augmented data, create true labels.

- Ensure predicted labels have the same length as true labels.

- Evaluate the accuracy of the model using accuracy_score and print the result.

Generative Adversarial Network:

1. Import necessary libraries:

- numpy

- tensorflow as tf

- layers, models from tensorflow.keras

2. Define the Generator model:

- Input: Latent dimension

- Sequential model with Dense layers:

  - Dense layer with 128 units, ReLU activation

  - Dense layer with 256 units, ReLU activation

  - Dense layer with 512 units, ReLU activation

  - Dense layer with X_scaled.shape[1] units, tanh activation (Output layer)

- Return the model

3. Define the Discriminator model:

- Input: Shape of the data
- Sequential model with Dense layers:
  - Dense layer with 512 units, ReLU activation
  - Dense layer with 256 units, ReLU activation
  - Dense layer with 128 units, ReLU activation
  - Dense layer with 1 unit, sigmoid activation (Output layer)
- Return the model

4. Combine Generator and Discriminator into a GAN model:
   - Set discriminator's trainable attribute to False
   - Define GAN model input as noise
   - Generate fake data using the generator
   - Pass the generated data through the discriminator
   - Return the GAN model
5. Define hyperparameters:
   - Latent dimension
   - Number of epochs
   - Batch size
6. Build and compile the Discriminator:
   - Use binary cross entropy loss and Adam optimizer with specified learning rate and beta_1
7. Build the Generator:
   - Use the defined latent dimension to build the generator model
8. Build and compile the GAN model:
   - Use binary cross entropy loss and Adam optimizer with specified learning rate and beta_1
9. Training loop:
   - Iterate over epochs:
     - Generate random noise as input to the Generator
     - Generate fake data with Generator

- Select a random batch of real data from the dataset

- Concatenate real and generated data to create the training dataset for the Discriminator

- Train the Discriminator using real and generated data separately

- Generate new random noise as input to the GAN

- Train the Generator via the GAN model by providing noise and labels indicating the data is real (tricking the discriminator)

- Print progress periodically

Long Short-Term Memory:

1. Import necessary libraries:

   - numpy as np

   - train_test_split from sklearn.model_selection

   - Sequential from keras.models

   - LSTM, Dense from keras.layers

2. Split the data into training and testing sets:

   - Use train_test_split to split the data into X_train, X_test, y_train, and y_test.

   - Adjust the test_size and random_state parameters as needed.

3. Reshape the input data for LSTM:

   - Reshape the training input data (X_train) to have the shape (samples, time steps, features) for LSTM.

   - Assuming each trace is a time series and you want to use all features.

4. Define the LSTM model:

   - Initialize a Sequential model.

   - Add an LSTM layer with specified units and input shape.

   - Add a Dense layer with 1 unit and 'sigmoid' activation for binary classification.

5. Compile the model:

   - Compile the model with 'adam' optimizer and 'binary_crossentropy' loss function.

- Optionally, define metrics such as accuracy.

6. Train the model:

- Fit the model to the training data (X_train_reshaped, y_train) with specified epochs, batch_size, and validation split.

7. Transpose the input data for prediction:

- Transpose the reshaped testing input data (X_test_reshaped) to match the expected input shape of the model.

8. Predict using the transposed input data:

- Use the trained model to predict the labels for the transposed input data (X_test_reshaped_transposed).

9. Calculate loss and accuracy:

- Evaluate the model's performance on the testing data to calculate loss and accuracy.

- Print the test loss and accuracy.

10. Manually calculate accuracy:

- Calculate accuracy manually by comparing the predicted labels with the true labels (y_test).

Chapter 5

Results and Discussion

5.1 Snapshots

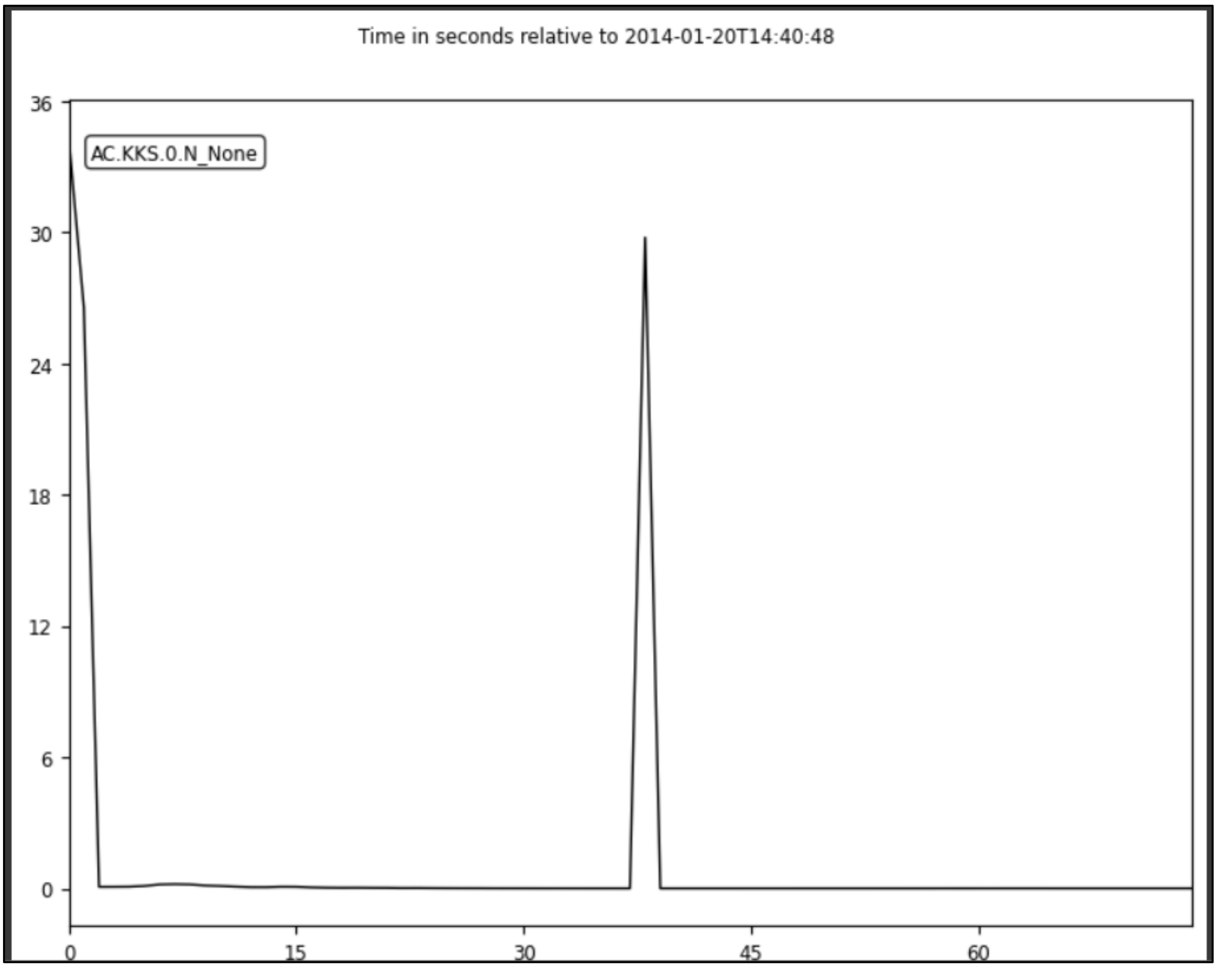
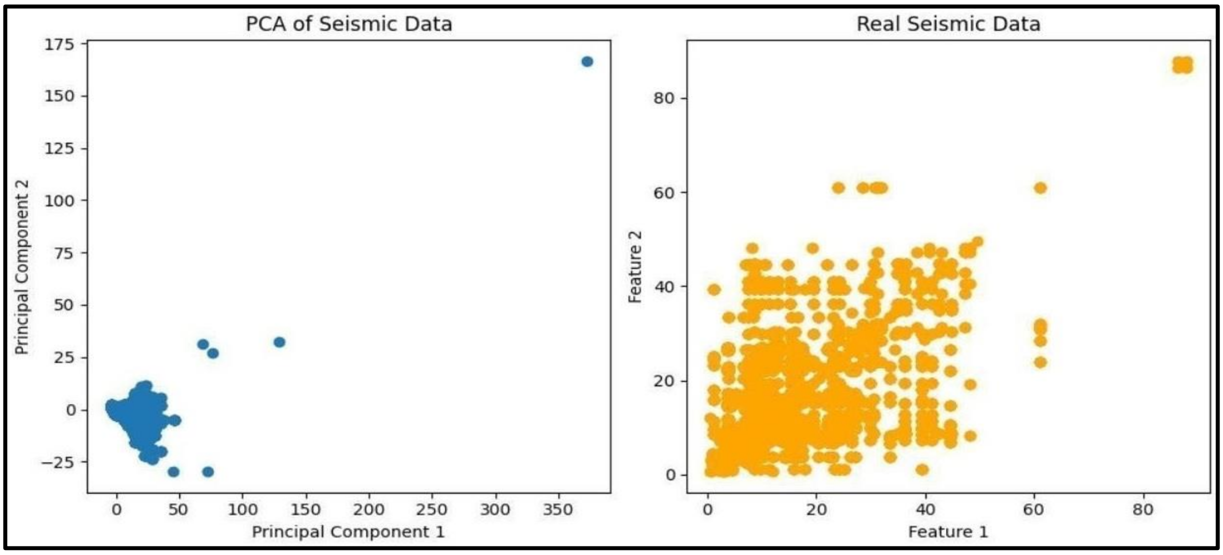


Figure 5.1 Seismic Waveform

The Figure 5.1 displays a seismic waveform plot for a specific trace. The x-axis represents time, while the y-axis represents the relative amplitude of the seismic waves. The waveform shows variations in ground motion intensity over time, likely corresponding to seismic activity recorded by the sensor. The plot is presented in black color against a white background, with clear waveform patterns visible throughout the duration of the plot.



(a) (b)

Figure 5.2: a) PCA of Seismic Data, b) Real Seismic Data

The above Figure 5.2 consists of two scatter plots side by side. (a) depicts the reduced seismic data after PCA, showcasing the distribution of data points in the principal component space. (b) displays the original seismic data, providing a comparison to the reduced data obtained through PCA.

`

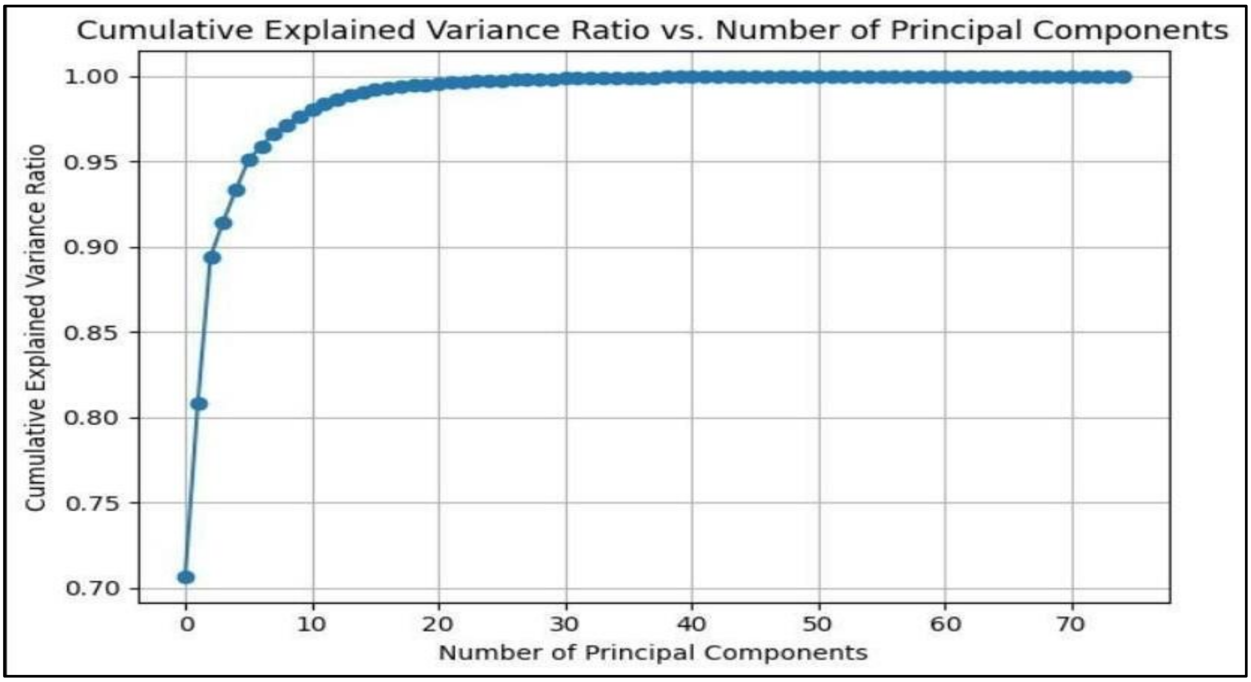
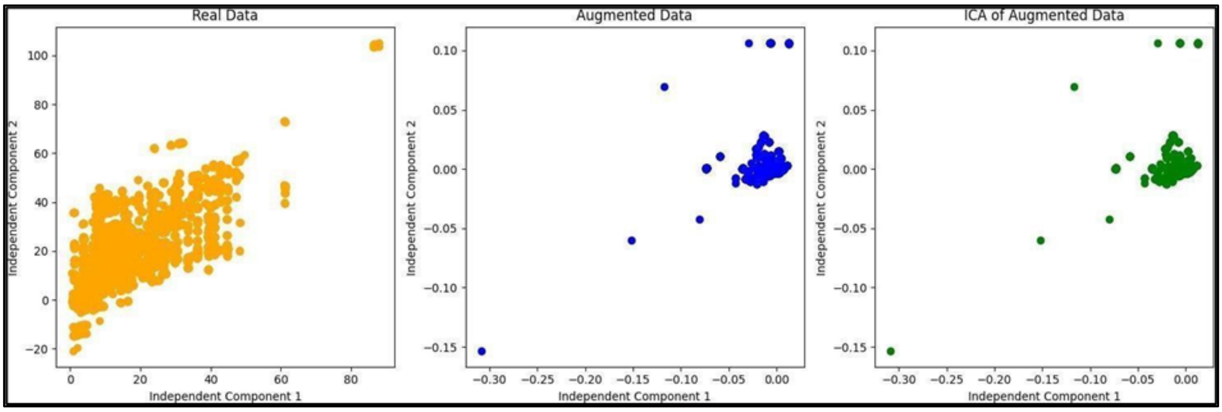


Figure 5.3 Cumulative Explained Variance Ration vs Number of Principal Components

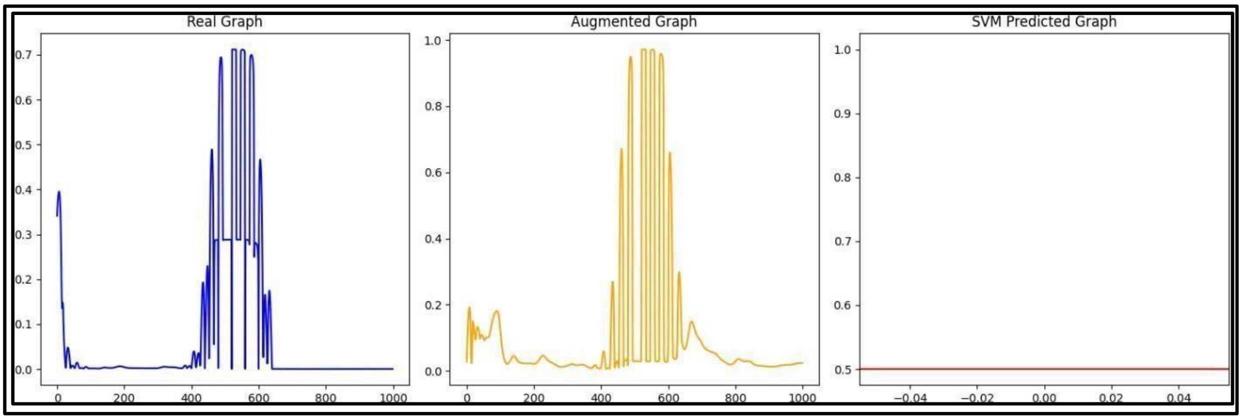
The figure 5.3 depicts the cumulative explained variance ratio against the number of principal components retained after PCA. It visually demonstrates the trade-off between retaining more components and the cumulative variance explained, aiding in determining the optimal dimensionality reduction while preserving significant information from the original data.



(a) **(b)** **(c)**

Figure 5.4: a) Real Data, b) Augmented Data, c) ICA Augmented Data

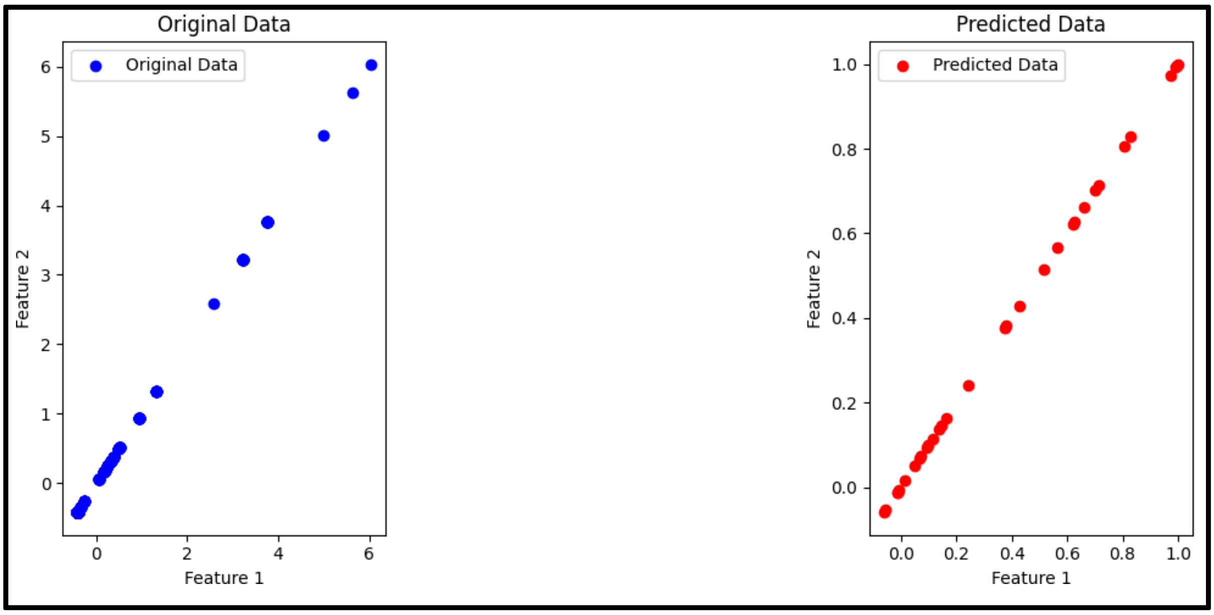
The Figure 5.4 shows scatter plots visualize the independent components for both the real and augmented data, aiding in understanding their underlying structures and similarities. (a) represents the original dataset's graph. (b) is the scatter plot for Augmented data. (c) represents the ICA of the augmented data.



(a) **(b)** **(c)**

Figure 5.5: (a)Real Graph, b) Augmented Graph, c) SVM Predicted Graph

In the figure 5.5, the results showcase real and augmented data samples, alongside the SVM-predicted graph for a specific data point. The real and augmented graphs depict raw and augmented seismic data, while the SVM-predicted graph illustrates model predictions, aiding in visualizing model performance.



(a) (b)

Figure 5.6: (a)Original Data, b) Predicted Data

In the figure 5.6, the plotted graphs compare original and predicted data points in a two-dimensional feature space. (a) shows the original data points, while the (b) illustrates the synthetic data generated by a model. These visualizations aid in assessing the similarity between the original and generated datasets.

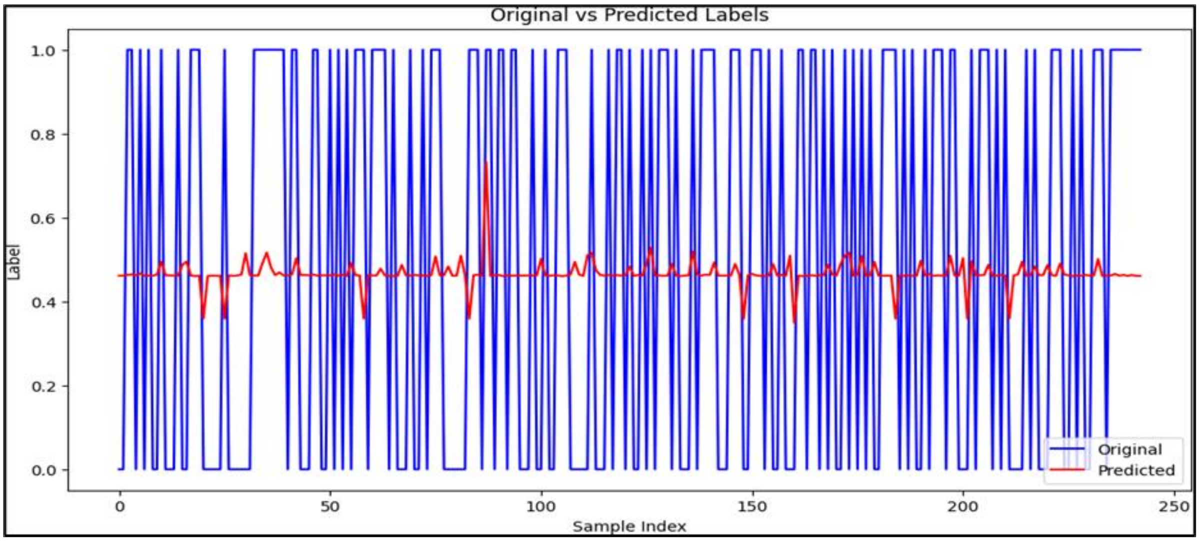


Figure 5.7 Originated vs Predicted Labels

The figure 5.7 compares original (blue) and predicted labels (red), helping evaluate the model's performance by visually assessing how closely the predictions match the actual labels.

5.2 Discussion

The results show that both the LSTM and SVM models achieved around 50% accuracy in earthquake detection, where labels were randomly assigned as 0 or 1 for unsupervised data and then converted into a supervised learning format. Additionally, GAN, ICA, and PCA were applied for noise reduction on the unsupervised data, with augmented data from the dataset used as input.

Here's a discussion on these results:

1. **Accuracy and Performance**: The accuracy achieved for LSTM and SVM models are around 45% and 50%. The SVM algorithm is a simpler prediction method, while LSTM networks are far more complicated. Overall, LSTM performs better than SVM in all the scenarios. This is because of its ability to remember or forget the data in an efficient manner than SVM.

2. **Model Comparison**: Comparing LSTM and SVM models, we find out that LSTM model works better compared to SVM for earthquake. When comparing LSTM and SVM models for earthquake prediction, LSTMs generally perform better due to their ability to handle temporal dependencies and learn features automatically from time-series data. LSTMs excel at capturing non-linear patterns and long-term dependencies inherent in seismic data, leading to more accurate predictions. In contrast, SVMs require extensive feature engineering and struggle with large datasets and temporal complexities, making them less effective for this task. Overall, LSTMs' specialized architecture makes them a superior choice for earthquake forecasting.

3. **Unsupervised Data Processing**: When employing Generative Adversarial Networks (GAN), Independent Component Analysis (ICA), and Principal Component Analysis (PCA) for noise reduction in unsupervised data, GANs tend to give better results. GANs are highly effective in learning the underlying distribution of the data and generating realistic, denoised outputs by leveraging their adversarial training mechanism. In contrast, ICA and PCA, while useful for dimensionality reduction and separating independent components or principal components, may not capture complex data distributions as effectively as GANs. Thus, GANs outperform ICA and PCA in noise reduction tasks by producing cleaner and more accurate representations of the original data.

4. **Augmentation for Noise Reduction**: To evaluate noise reduction techniques for seismic data, the dataset was augmented to simulate the noise present in real-time seismic data. This augmented dataset was then used as input for Generative Adversarial Networks (GAN), Independent Component Analysis (ICA), and Principal Component Analysis (PCA). Among these methods, GANs yielded the best results. GANs excel at learning the underlying data distribution and generating clean, denoised outputs, whereas ICA and PCA, while useful for dimensionality reduction and component isolation, were less effective at handling the complex noise patterns typical of seismic data. Thus, GANs provided superior noise reduction, producing more accurate and realistic seismic data.

In conclusion, while both LSTM and SVM models achieved around 45-50% accuracy in earthquake prediction with randomly assigned labels, LSTM models demonstrated superior performance due to their ability to handle temporal dependencies and complex patterns in seismic data. Additionally, for noise reduction in unsupervised seismic data, GANs outperformed ICA and PCA. GANs effectively learned the underlying data distribution and produced cleaner, more accurate denoised outputs. This highlights the advantage of using LSTM for prediction and GANs for noise reduction in seismic data analysis, providing more reliable and realistic results in earthquake forecasting.

5.3 Advantages, Disadvantages and Applications

Advantages

- **Cost-Effective Data Enhancement**: GANs allow for the generation of synthetic seismic data, reducing the reliance on expensive, high-precision sensors. This significantly lowers the cost of data collection and analysis.
- **Increased Data Diversity**: Augmenting real seismic data with synthetic data increases the diversity of the dataset, helping to capture a broader range of seismic patterns and conditions.
- **Improved Model Generalization**: Augmented data can enhance the performance of machine learning models, making them more robust and accurate, especially when dealing with limited or noisy data from low-cost sensors.
- **Mitigation of Data Imbalances**: In some cases, seismic events might be rare or unevenly distributed. GAN-generated data can balance the dataset, making it more

suitable for training models.

- **Enhanced Training Efficiency**: GANs can speed up the training process by providing additional data, potentially reducing the need for extensive real data collection.
- **Flexibility and Customization**: GANs can be fine-tuned to generate synthetic data that mimics specific characteristics of the real data, allowing for customization according to the application's requirements.
- **Data Augmentation at Scale**: GANs can generate a large volume of augmented data efficiently, making it possible to train models at scale.
- **Environmental Monitoring**: Low-cost MEMS sensors can be deployed in a wider range of locations, including remote or hazardous areas, to gather more comprehensive environmental monitoring data.
- **Scientific Research and Geophysical Studies**: Augmented data can be used to simulate various seismic scenarios, enabling researchers to conduct geophysical studies and simulations that would be impractical or costly with only real sensor data.
- **Early Warning Systems**: Improved data quality can enhance the performance of early warning systems, allowing for more accurate and timely alerts in the event of seismic activity.
- **Exploration and Education**: Augmented data can be valuable for educational purposes, helping students and researchers better understand seismic phenomena.

Disadvantages

- The methods adopted previously mentioned model that are supervised a real seismic waved form needs to be identified by subjectively selecting feature label which effects the detection performance of the model

- Many data processing techniques used in traditional seismological research originated from small data sets and limited computing power

- The traditional seismic network are expensive and the operation costs are high

Applications

1. **Earthquake Early Warning Systems**

   Timely Alerts: The enhanced data quality and prediction accuracy provided by the proposed system enable the development of more reliable earthquake early warning systems. These systems can deliver timely alerts to communities, allowing them to take preventive measures to protect lives and property.

Emergency Preparedness: Governments and disaster management agencies can utilize these improved warning systems to better prepare for imminent earthquakes, including mobilizing resources and coordinating emergency response efforts.

2. **Urban Planning and Infrastructure Development**

   Seismic Risk Assessment: Urban planners and engineers can use the accurate seismic data and predictive insights from the system to assess the seismic risks of different regions. This information is crucial for designing earthquake-resistant buildings and infrastructure.

   Retrofitting Strategies: The data can guide decisions on retrofitting existing structures to withstand potential seismic events, thereby enhancing the safety and resilience of urban environments.

3. **Insurance and Risk Management**

   Insurance Models: Insurance companies can integrate the accurate seismic data and prediction models into their risk assessment frameworks. This integration can lead to more precise pricing of insurance premiums based on the actual seismic risk of specific areas.

   Claim Assessments: Post-earthquake, insurers can use the data to more accurately assess the extent of damage and expedite the claims process, ensuring faster relief to affected policyholders.

4. **Scientific Research and Seismology**

   Earthquake Mechanisms: Researchers can utilize the high-quality, noise-reduced data to gain deeper insights into the mechanisms of earthquakes. This understanding can contribute to the development of more sophisticated models for predicting seismic activity.

   Seismic Patterns: The system can help identify patterns and anomalies in seismic activity over time, contributing to the broader field of seismology and our understanding of earth's geophysical processes.

5. **Public Safety and Community Awareness**

   Educational Programs: The data and predictive insights can be used to develop educational programs aimed at increasing public awareness about earthquake preparedness and safety measures.

Community Drills: Communities can organize regular earthquake drills based on accurate prediction data, ensuring that residents are well-prepared to respond effectively during an actual event.

6. **Infrastructure Monitoring and Maintenance**

Structural Health Monitoring: The system can be integrated into the monitoring of critical infrastructure such as bridges, dams, and nuclear power plants. Continuous seismic data can help in the early detection of structural weaknesses and guide maintenance efforts.

Real-Time Monitoring: Real-time seismic monitoring of infrastructure can provide immediate data on the impact of seismic events, helping authorities to quickly assess and address any damage.

7. **Environmental Impact Studies**

Impact Assessments: The system can aid in conducting environmental impact assessments for regions prone to seismic activity. This information is essential for planning safe and sustainable development projects.

Conservation Efforts: Accurate seismic data can also inform conservation efforts, particularly in protecting natural habitats and wildlife from the adverse effects of earthquakes.

8. **Technological Innovation**

Sensor Development: The insights gained from using low-cost MEMS sensors and advanced data processing techniques can spur innovation in sensor technology, leading to the development of even more cost-effective and reliable seismic monitoring devices.

Software Solutions: The project's success can lead to the creation of new software solutions for seismic data analysis and prediction, benefiting various industries and sectors that rely on accurate seismic information.

The proposed seismic monitoring system has a wide range of applications that extend beyond mere earthquake prediction. It holds the potential to revolutionize the approach of earthquake preparedness, urban planning, insurance, scientific research, public safety, infrastructure maintenance, environmental impact assessments, and technological innovation. By enhancing the accuracy, reliability, and accessibility of seismic data, the system can significantly contribute to mitigating the risks associated with earthquakes and improving disaster preparedness on multiple fronts.

Chapter 6

Conclusion and Future Scope

In this project, the formidable challenges surrounding seismic data collection and earthquake prediction, particularly focusing on leveraging the affordability and accessibility of low-cost MEMS sensors. These sensors help in seismic data collection but are often plagued by high levels of noise, which can severely compromise the accuracy and reliability of the data. To overcome data processing techniques used are Generative Adversarial Networks (GANs), to effectively mitigate the noise inherent in seismic data. By harnessing the power of GANs, and enhancing the quality of seismic data, thus laying a robust foundation for analysis. The extended supervised learning approaches, such as Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks are used to refine earthquake prediction accuracy. The approach demonstrates the potential of these supervised techniques in improving the reliability of earthquake prediction, thereby to anticipate the seismic events.

This project introduces EQGAN, a deep generative model that leverages Generative Adversarial Networks (GANs), long short-term memory (LSTM), attention mechanisms, and neural networks (NN) to address challenges in seismic data analysis. EQGAN effectively captures the multi-dimensional temporal evolution of seismic sequences and generates high-quality P-wave and S-wave data, demonstrating superior performance in terms of efficiency (81%) and stability compared to standard GAN, LSTM, and NN models. The model's ability to align with seismic data patterns makes it a promising tool for data augmentation and earthquake prediction, overcoming limitations such as noise pollution in MEMS sensor data and the scarcity of high-quality seismic datasets. Its analysis in both the frequency domain and autocorrelation distribution further underscores the model's capability to produce reliable and stable outputs. This project highlights the broader applicability of GANs across disciplines, demonstrating their power in leveraging large volumes of unlabeled data for tasks in fields ranging from astronomy and biology to business and beyond. EQGAN's innovative architecture not only advances the state-of-the-art in seismic data augmentation and earthquake monitoring but also lays the foundation for future improvements in earthquake detection, sensor data quality, and real-time seismic monitoring systems, contributing significantly to disaster preparedness and risk mitigation efforts.

Future Scope:

The project aims to revolutionize seismic data analysis by leveraging state-of-the-art techniques, such as fine-tuning GAN models for enhanced noise removal and data augmentation, which significantly improve the quality, interpretability, and reliability of seismic datasets. It focuses on integrating sensor fusion, combining data from MEMS sensors and traditional seismic instruments, to create comprehensive datasets that provide a more holistic understanding of seismic activities. Additionally, the project seeks to optimize supervised learning models, including SVM and LSTM networks, to ensure greater accuracy, robustness, and reliability in predicting earthquakes and detecting patterns. A critical aspect involves developing real-time implementation frameworks for immediate seismic monitoring, allowing proactive disaster response and early warning systems to minimize risk. Collaborative efforts with domain experts, government agencies, and disaster management organizations will be pivotal in validating the framework and ensuring its successful deployment in real-world applications. Long-term monitoring and evaluation across diverse seismic regions will deliver valuable insights into the scalability, adaptability, and effectiveness of the system, contributing to the advancement of earthquake forecasting, disaster preparedness, and global resilience against seismic events.

References

[1]. G. Madureira and A. E. Ruano, ‘‘A neural network seismic detector,’’ IFAC Proc. vol, vol. 42, no. 19, pp. 304– 309, 2009.

[2]. S. M. Mousavi, W. Zhu, Y. Sheng, and G. C. Beroza, ‘‘CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection,’’ Sci. Rep., vol. 9, no. 1, pp. 1–14, Dec. 2019.

[3]. S. M. Mousavi, W. Zhu, Y. Sheng, and G. C. Beroza, ‘‘CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection,’’ Sci. Rep., vol. 9, no. 1, pp. 1–14, Dec. 2019.

[4]. R. M. Allen and H. Kanamori, ‘‘The potential for earthquake early warning in southern California,’’ Science, vol. 300, no. 5620, pp. 786–789, 2003.

[5]. Li, M.-A. Meier, E. Hauksson, Z. Zhan, and J. Andrews, ‘‘Machine learning seismic wave discrimination: Application to earthquake early warning,’’ Geophys. Res. Lett., vol. 45, no. 10, pp. 4773–4779, May 2018.

[6]. Botero-valencia, L. Castano-Londono, D. Marquez-Viloria, and M. Rico-Garcia, ‘‘Data reduction in a low-cost environmental monitoring system based on LoRa for WSN,’’ IEEE Internet Things J., vol. 6, no. 2, pp. 3024–3030, Apr. 2018.

[7]. D. Ciuonzo, S. H. Javadi, A. Mohammadi, and P. S. Rossi, ‘‘Bandwidth constrained decentralized detection of an unknown vector signal via multi sensor fusion,’’ IEEE Trans. Signal Inf. Process. Netw., vol. 6, pp. 744–758,2020.

[8]. M. Faulkner, M. Olson, R. Chandy, J. Krause, K. M. Chandy, and A. Krause, ‘‘The next big one: Detecting earthquakes and other rare events from community-based sensors,’’ in Proc. 10th ACM/IEEE Int. Conf. Inf.Process. Sensor Netw., Apr. 2011, pp. 13–24.

[9]. D. Li, R. Landry, and P. Lavoie, ‘‘Low-cost MEMS sensor-based attitude determination system by integration of magnetometers and GPS: A real data test and performance evaluation,’’ in Proc. IEEE/ION Position, Location Navigat.Symp., May 2008, pp. 1190–1198.

[10]. E. Yoon, O. O’Reilly, K. J. Bergen, and G. C. Beroza, ‘‘Earthquake detection through computationally efficient similarity search,’’ Sci. Adv.,vol. 1, no. 11, Dec. 2015, Art. no. e1501057.

[11]. C. E. Yoon, O. O’Reilly, K. J. Bergen, and G. C. Beroza, ‘‘Earthquake detection through computationally efficient similarity search,’’ Sci. Adv.,vol. 1, no. 11, Dec. 2015, Art. no. e1501057. K. Rong, C. E. Yoon, K. J. Bergen, H. Elezabi, P. Bailis, P. Levis, and G. C. Beroza, ‘‘Locality-sensitive hashing for earthquake detection: A case study of scaling datadriven science,’’ 2018, arXiv:1803.09835.

[12]. X. Huang, J. Lee, Y.-W. Kwon, and C.-H. Lee, ‘‘CrowdQuake: A net worked system of low-cost sensors for earthquake detection via deep learning,’’ in Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2020, pp. 3261 – 3271.

[13]. Y. Chen, ‘‘Automatic microseismic event picking via unsupervised machine learning,’’ Geophys. J. Int., vol. 222, no. 3, pp. 1750–1764, 2020.

[14]. Y. Yu, J. Lin, L. Zhang, G. Liu, J. Hu, Y. Tan, and H. Zhang, ‘‘Identification of seismic wave first arrivals from earthquake records via deep learning,’’in Proc. Int. Conf. Knowl. Sci., Eng. Manage. Cham, Switzerland: Springer,2018, pp. 274– 282.

[15]. W. Zhu and G. C. Beroza, ‘‘PhaseNet: A deep-neural-network-based seismic arrival-time picking method,’’ Geophys. J. Int., vol. 216, no. 1,pp. 261–273, 2019.

[16]. J. Wang, Z. Xiao, C. Liu, D. Zhao, and Z. Yao, ‘‘Deep learning for picking seismic

arrival times,'' J. Geophys. Res., Solid Earth, vol. 124, no. 7,pp. 6612–6624, 2019.

[17]. L. Küperkoch, T. Meier, J. Lee, and W. Friederich, ''Automated deter mination of P-phase arrival times at regional and local distances using higher order statistics,'' Geophys. J. Int., vol. 181, no. 2, pp. 1159–1170, May 2010.

[18]. S. M. Mousavi and G. C. Beroza, ''Bayesian-deep-learning estima tion of earthquake location from single-station observations,'' 2019,arXiv:1912.01144.

[19]. I. Khan, S. Choi, and Y.-W. Kwon, ''Earthquake detection in a static and dynamic environment using supervised machine learning and a novel feature extraction method,'' Sensors, vol. 20, no. 3, p. 800, Feb. 2020.

[20]. J. Wang and T.-L. Teng, ''Identification and picking of s phase using an artificial network,'' Bull. Seismol. Soc. Amer., vol. 87, no. 5,pp. 1140 –1149, Oct. 1997.

[21]. Q. Kong, R. M. Allen, L. Schreier, and Y.-W. Kwon, ''MyShake: A smart phone seismic network for earthquakeearly warning and beyond,'' Sci. Adv., vol. 2, no. 2, Feb. 2016, Art. no. e1501055.

[22]. Z. E. Ross, M.-A. Meier, E. Hauksson, and T. H. Heaton, ''Generalized seismic phase detection with deep learning,'' Bull. Seismol. Soc. Amer.,vol. 108, no. 5A, pp. 2894–2901, 2018.

[23]. Y. Wu, Y. Lin, Z. Zhou, D. C. Bolton, J. Liu, and P. Johnson, ''DeepDetect: A cascaded region-based densely connected network for seismic event detection,'' IEEE Trans. Geosci. Remote Sens., vol.

[24] R. M. Dokht, H. Kao, R. Visser, and B. Smith, ``Seismic event and phase detection using timefrequency representation and convolutional neural networks,'' Seismol. Res. Lett., vol. 90, no. 2A, pp. 481490, 2019.

[25] O. M. Saad and Y. Chen, ``Automatic waveform-based source-location imaging using deep learning extracted microseismic signals,'' Geophysics, vol. 85, no. 6, pp. KS171KS183, Nov. 2020.

[26] O. M. Saad, G. Huang, Y. Chen, A. Savvaidis, S. Fomel, N. Pham, and Y. Chen, ``SCALODEEP: A highly generalized deep learning framework for real-time earthquake detection,'' J. Geophys. Res., Solid Earth, vol. 126, no. 4, Apr. 2021, Art. no. e2020JB021473.

[27] L. Zhu, Z. Peng, J. McClellan, C. Li, D. Yao, Z. Li, and L. Fang, ``Deep learning for seismic phase detection and picking in the aftershock zone of 2008 Mw7. 9 Wenchuan earthquake,'' Phys. Earth Planet. Interiors,vol. 293, Aug. 2019, Art. no. 106261.

[28] O. M. Saad and Y. Chen, ``Earthquake detection and P-wave arrival time picking using capsule neural network,'' IEEE Trans. Geosci. Remote Sens.,vol. 59, no. 7, pp. 62346243, Jul. 2021.

[29] , and B. Smith, ``Seismic event and phase detection using timefrequency representation and convolutional neural networks,'' Seismol. Res. Lett., vol. 90, no. 2A, pp. 481490, 2019.

[30] O. M. Saad and Y. Chen, ``Automatic waveform-based source-location imaging using deep learning extracted microseismic signals,'' Geophysics, vol. 85, no. 6, pp. KS171KS183, Nov. 2020.

[31] O. M. Saad, G. Huang, Y. Chen, A. Savvaidis, S. Fomel, N. Pham, and Y. Chen, ``SCALODEEP: A highly generalized deep learning framework for real-time earthquake detection,'' J. Geophys. Res., Solid Earth, vol. 126, no. 4, Apr. 2021, Art. no. e2020JB021473.

[32] L. Zhu, Z. Peng, J. McClellan, C. Li, D. Yao, Z. Li, and L. Fang, ``Deep learning for seismic phase detection and picking in the aftershock zone of 2008 Mw7. 9 Wenchuan earthquake,'' Phys. Earth Planet. Interiors,vol. 293, Aug. 2019, Art. no.

106261.

[33] O. M. Saad and Y. Chen, ``Earthquake detection and P-wave arrival time picking using capsule neural network,'' IEEE Trans. Geosci. Remote Sens.,vol. 59, no. 7, pp. 62346243, Jul. 2021.

[34] Y. Li, B. Ku, S. Zhang, J.-K. Ahn, and H. Ko, ``Seismic data augmentation based on conditional generative adversarial networks,'' Sensors, vol. 20,no. 23, p. 6850, Nov. 2020.

[35] *T. Wang, Z. Zhang, and Y. Li, ``EarthquakeGen: Earthquake generator using generative adversarial networks,'' in Proc. SPIE, SEG Tech. Program Expanded Abstr.,2019, pp. 267426*

APPENDICES

DETAILS OF THE STUDENTS

BATCH – 2024-2025

NAME:	SPARSH BISEN	SURYA M	TEJASWINI P	U JEEVAN HARI
USN:	1DB21CS148	1DB21CS155	1DB21CS158	1DB21CS161
PHONE NUMBER:	7619225222	8971449324	8660186264	7892925004
EMAIL-ID:	sparshbisen @gmail.com	suryasuria077 @gmail.com	tejaswinipcs @gmail.com	iamjeeva2k@gmail.com
PLACED-IN:	-----	-----	----	----
PERMANENT ADDRESS:	P2A-453, Sector 27, Atal Nagar, Chattisgarh - 492101	#144, 3rd main, Panchasheelanagara Bangalore	G8, Sri Pearl Park Apartment, Dubasipalya, Bangalore - 560059	#112, ITI Layout, 2nd main, 2nd Cross, Nayandahalli Post, Bangalore - 560039