Anomalies Detection in Radon Time Series for Earthquake Prediction Using Machine Learning Techniques

by

RAGHAV GORASIYA
202111029

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

May, 2023

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree.

ii) due acknowledgment has been made in the text to all the reference material used.

Raghav

Raghav Gorasiya

Certificate

This is to certify that the thesis work entitled ANOMALIES DETECTION IN RADON TIME SERIES FOR EARTHQUAKE PREDICTION USING MACHINE LEARNING TECHNIQUES has been carried out by RAGHAV GORASIYA for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

Prof. Bhaskar Chaudhury
Thesis Supervisor

Acknowledgments

I am immensely grateful to **Dr. Bhaksar Chaudhury** sir for his invaluable guidance and support throughout my research journey. His profound knowledge and expertise in the field of data analysis and machine learning have been instrumental in shaping my understanding and exploring various aspects of this research. His constant encouragement and mentorship have been invaluable in every step of my thesis.

I would also like to extend my sincere thanks to the **Institute of Seismological Research (ISR), Gandhinagar**, for providing me with access to official data and other crucial resources for my research work. In particular, I am grateful to **Dr. Madhusudan Rao** sir for his expert guidance and insights into the field. His assistance and explanations of the workflow have greatly contributed to the success of my research. **Dr. Madhusudan Rao** sir, with his expertise in the field, provided me with invaluable guidance and explained the research workflow, significantly enhancing my understanding and facilitating my research progress.

I would also like to extend my gratitude to all the researchers, and colleagues who supported me throughout my thesis work. Their contributions, discussions, and insights have been truly valuable in shaping the outcome of my research.

Finally, I would like to express my deepest appreciation to my family and friends for their unwavering support, understanding, and encouragement throughout this journey. Their love and encouragement have been my constant source of motivation.

I am sincerely grateful to all those mentioned above and to everyone else who has directly or indirectly contributed to the successful completion of my thesis work.

Contents

Abstract

Radioactive soil and water radon gas emission is a significant precursor to earthquakes. The meteorological parameters such as temperature, pressure, humidity, rainfall, and windspeed influence the radon gas emission from the medium such as soil and water. In this study, radioactive soil radon gas has been investigated for earthquake prediction. Before the seismic events, radon gas emission is also affected by seismic energies. These seismic energies are responsible for the changes inside the earth's crust, which causes earthquakes on earth. Our focus in this work is first to predict the radon gas concentration using Machine Learning algorithms and then identify anomalies before and after the seismic events using standard confidence interval methods. We experimented with different machine learning models for the detailed comparative study of radon concentration predictions. A dataset is divided into different settings of training and testing data. Testing data includes the seismic samples only. The models are trained on non-seismic day samples and some of the seismic day samples and tested on seismic day samples. After acceptable predictions, anomaly detection can be done on test data. A simple mean plus two standard deviations away test has been used to identify the original measured radon values, which are out of this prediction confidence interval. These values are then considered as an anomaly.

List of Tables

List of Figures

CHAPTER 1
Introduction

As we are aware, earthquakes are devastating natural occurrences that can strike unexpectedly, posing a threat to both living and non-living entities worldwide. Given the absence of reliable forewarning, it remains exceedingly challenging task to predict all three critical parameters of earthquakes: time, magnitude, and location. While the precise timing is of utmost importance, current research primarily focuses on earthquake forecasting rather than prediction. Determining the location parameter can be accomplished through the location from where our data is.

There are mainly two approaches for the prediction of seismic events. In the first approach, different earthquake precursors are characterized. In the second approach geophysical trends, or seismological patterns that precede a large earthquake are identified. Precursor methods are largely used for short-term earthquake prediction [25]. Precursor methods deal with anomalies in physical phenomena that might give accurate warning of incoming earthquake [25]. Some possible precursors that might useful for the prediction of earthquakes include: strange behaviour of animals, electromagnetic anomalies, anomalies in ionosphere and behaviour of radon in soil, water and atmosphere, etc. Several studies [27][28][17] along the globe has found the radon gas concentration as strong precursor of the earthquake [31][10][32]. Also the radon can be easily detected in the soil and water. That is why in most of the study [18][19], the radon is considered as earthquake precursor.

Radon (^{222}Rn) is a radioactive gas that is naturally produced through the decay of other radioactive elements found in the Earth's crust. Radon is a naturally present noble gas that is odorless, colorless, and radioactive. It has half-live of around 3.82 days [7]. It is present in rocks worldwide. Radon has been extensively studied as a means to track volcanic activity and potentially predict earthquakes [1]. The correlation between radon signals and tectonic events is a subject of on-

going discussion, but numerous instances of anomalous radon signals associated with tectonic activity have been documented globally [13]. These anomalies are believed to occur due to the fracturing processes in the Earth's crust that precede an earthquake. As stress is released within the crust, it facilitates the movement of fluids and gases, including radon, from deeper sources.

Radon, a chemically inert radioactive gas, is continuously produced throughout the Earth, typically in small amounts due to the presence of radium in crustal materials. Many studies have focused on understanding the emission and movement of radon within the Earth and the atmosphere [24][2]. Radon has various applications, such as identifying buried uranium deposits (which give rise to radium and radon), tracing the movement of air and groundwater, indicating fault lines, and potentially serving as a tool for earthquake prediction. In Russia and China, measurements have shown anomalous changes, mostly increases, in groundwater radon content prior to certain major earthquakes [27].

Radon gas, specifically Radon-222 (^{222}Rn), is generated as a by-product of the radioactive decay of radium-226 (^{226}Ra) which is part of the ^{238}U decay series found in the Earth's crust. Radon, a naturally occurring noble gas, exists in three isotopes: ^{222}Rn (commonly referred to as radon), ^{220}Rn (known as thoron), and ^{219}Rn (referred to as actinon). Radon primarily originates from the radioactive decay of the ^{238}U series, while thoron stems from the ^{232}Th radioactive series, and actinon originates from the ^{235}U series. The crustal abundance of these isotopes is as follows: ^{238}U (uranium) - 2.7 μ g kg^{-1}, ^{232}Th (thorium) - 8.5 μ g kg^{-1}, and ^{235}U (actinum) - 0.02 μ g kg^{-1}. Although the concentration of ^{232}Th is slightly higher than ^{238}U in the Earth's crust, the production rates of ^{222}Rn and ^{220}Rn are approximately equal. This is due to the longer half-life of ^{232}Th (14.1×109 years) compared to ^{238}U (4.5×109 years). Among the three isotopes, ^{222}Rn is more significant because it has a longer half-life of 3.825 days, whereas ^{220}Rn (55.6 s) and ^{219}Rn (actinon) have much shorter half-lives. The shorter half-lives of the latter two isotopes limit their transport through diffusion to only short distances. However, thoron manages to reach the Earth's surface, although in smaller quantities compared to radon. In our work, the focus is on radon rather than the other isotopes. It is naturally occurring and can be found in small quantities throughout the Earth's environment, including soil, groundwater, and the lower layers of the atmosphere.

In May 1975, the U.S. Geological Survey initiated the monitoring of subsurface soil gas radon levels along active faults in central california to investigate whether this parameter could provide valuable earthquake-related information. Several reasons motivated us to choice soil radon gas over groundwater. Firstly, soil gas monitoring offered distinct advantages over groundwater monitoring. Additionally, preliminary evidence suggested the potential usefulness of radon measurements. It had been observed that fault zones often exhibit radon enrichment in soil gas, and there were indications of significant radon increases in near-surface air around the time of certain earthquakes [20]. Second reason is the location from where the data has been collected. The data was gathered from the Pragpar station located in the Kutch region of Gujarat, India. Kutch is characterized by arid and desert conditions with limited groundwater availability. So soil radon gas is the potential choice for our experiments. Also the environmental factors like barometric pressure, temperature, rainfall, and wind speed can strongly influence radon levels in the air near and above the ground surface.

One promising approach to earthquake prediction involves monitoring anomalous soil radon gas concentrations, which act as precursors to seismic activity. Researchers have observed abnormal radon emissions before and after earthquake events [27]. It has been discovered that radon emission is influenced by various environmental factors such as temperature, pressure, humidity, wind speed, and rainfall during normal days [16]. However, during seismic events, radon emission experiences significant anomalies due to the introduction of seismic energy [30].

Initially, classical machine learning models like Support Vector Regressor (SVR) with linear, polynomial, and radial kernels, as well as decision trees and random forests, were employed to predict radon concentration. However, these models failed to yield accurate predictions. Consequently, the research strategy was adjusted, leading to the formulation of several important hypotheses based on previous experiments.

The first hypothesis postulated that the radon time series data contained high-frequency components that might impact the performance of machine learning models. The second hypothesis we made was the lack of seismic events in the dataset, and the third hypothesis proposed that the previously used machine learning model were unable to capture the pattern of complex radon time series

data. To validate these hypotheses, various methods were employed, such as filtering and data re-sampling techniques to remove or average out high-frequency components, synthetic data generation techniques to create additional seismic events, and subsequent training of classical models to evaluate the third hypothesis.

Through a series of experiments, it was discovered that almost all the hypotheses held true. To enhance radon prediction, the decision was made to employ more robust machine learning models, such as Gradient Boosting Machine, XGBoost. Additionally, synthetic seismic events were generated using the Geometric Mean (GM) to simulate patterns similar to those of real seismic events. The original dataset comprised samples taken at 10-minute intervals. To reduce the high-frequency components within the time series data, we performed resampling, adjusting the sample interval to one hour. This was achieved by averaging six consecutive samples. The process of resampling effectively filters out the higher frequency elements present in the data.

CHAPTER 2
Literature Survey

There are several studies that have been done in the area of earthquake prediction using radon time series data as precursor [21][14]. Several studies have used Machine Learning (ML) [18][19][3], Deep Learning (DL) [23], and Artificial Intelligence (AI) [26] for the prediction and identification of anomalies in radon concentration. Some of them are described below as a brief survey.

Adil aslam mir, Hadeel Alsolai et al. [18] has suggested Machine Learning-Based Ensemble Model technique for anomalies prediction in radon time series data for earthquake likelihood. They employed and compared the results of different individual machine learning models (KNN, Support vector machine with linear and radial kernel) and ensemble machine learning models (bagging and boosting). They performed these methods on different training and test set distributions through settings from 1 to 4. The training set is composed of different seismic activities and normal data while testing data is based upon seismic activities with its associated time window from 1 to 4. They concluded from the results, that ensemble models (boosted tree method) performs better than the all other ML models in all the dataset settings. Although they does not do anything regarding the anomaly detection, they opened this work for their future work.

Adil aslam mir, Muhammad Rafique et al. [19] has done good contribution towards the anomaly detection of the seismic activity. They used stacking technique of machine learning to classify the time series data into two class seismically active (SA) and inactive (NSA) data. Further they developed an anomaly indication function to classify the data for the whole day instead of just particular observation or sample (samples are collected at every 40 minute).

The main idea behind their methodology is the use of two layers. The first layer uses a stacking ensemble-based approach that incorporates three learners,

i.e. a generalized linear model, linear regression and K-nearest neighbors, to train on seismically active and inactive periods to predict soil radon gas (SRG) concentration. These predictions are then combined with the labeled anomaly data to train a meta-learner, i.e., support vector machine with a radial kernel, that categorizes the series into active and non-active radon time series data. In the second layer, these classifications are then passed to an automatic anomaly indication function that further labels the time series in a group of readings where the level of received indications is greater than or equal to the indication factor [19].

For experimentation, the soil radon gas concentration dataset is divided into Non Seismically Active Data (NSAD) and Seismically Active Data (SAD). Furthermore, from the mixture of NSAD and SAD, training, validation, and testing is performed according to window sizes. They used window size from 0 to 3 to perform their experiment. Window size 2 means SRG data which belongs two days before and after the seismic event.

They have proposed an automatic anomaly indication function (AAIF) that measures the percentage of anomalous samples from incoming series of anomalous and non-anomalous samples. If the percentage of anomalous samples increases, i.e., the indication factor (IF) = 0.55, the full day samples are considered to be anomalous and assigned the class label of 1 (SA) [19].

Suhrid Singh, Hari Jaishi et al. [23] studied the radon gas concentration as precursory of the seismic event and done some statistical analysis using multiple linear regression and Artificial Neural Network (ANN). In their study, they have observed that anomalous decrease in radon were correlated with seismic events. However, the decrease in radon concentration prior to the seismic event had been reported only very few times in literature.

They conclude that, out of the five radon anomalies, three anomalies (two crossing -2SD and one crossing -1SD) were correlated with relevant earthquakes. These three negative anomalies caused by admixing of radon-poor water from another aquifer through cracks created by the earthquake-related strain changes along the study region. However, They said in their conclusion that, more data recorded by various geophysical and geochemical instruments with good time resolution and rigorous statistical data analysis are needed to understand the underlying earthquake mechanism, also to discover more true earthquake precur-

sors [23].

D. Torkar, B. Zmazek et al. [26] has also employed the ANN based prediction of radon concentrations and correlated them to earthquakes. They firstly eliminated the seismic activity data from the dataset and then trained the neural network. Thus the ANN never "saw" the increased radon concentrations caused by earthquakes but just those caused by other reasons (false anomalies). And these other reasons are strongly connected to the five environmental parameters (Air and Soil Temperature, Soil and Air pressure, Rainfall) that served as ANN inputs. In this way the network captured the relationship between these parameters and the radon concentration during training. This was verified by the high correlation between measured concentration and predicted ones from the validation (cross-correlation) and test set [26].

During the performance stage when the network is fed by SA and NSA data, the output of ANN does not react to the seismic activity since the input parameters are not affected much by the earthquakes. But the Rn concentration is, and the discrepancy between the measurements and the ANN output increases, and an anomaly in the —Cm/Cp1— signal is generated [26].

Further they discovered that the anomaly detection process is as critical as the prediction itself. They identified five parameters that affect anomaly detection and by using an exhaustive search they defined the optimal values for the current dataset. They then replaced the exhaustive method by another ANN with number of particular anomalies nCA (no. of correct anomalies), nFA (False anomalies), nNA (no anomaly) in the input and detection parameters in the output. They found out that the performance of this neural network is limited to successful determination of 2–3 parameters (out of five) and that using this parameter results in proper determination of CAs and FAs, but not the NAs.

B. Zmazek, L. Todorovski et al. [26] gave one hypothesis that during the seismically active (SA) periods the prediction will be significantly worse than during seismically inactive periods. They further performed the experiments to test this hypothesis. Basically they considered "the change in predictability during SA periods" as the presence of anomaly.

For the experiment, they implemented the regression tree method to find the

relationship or correlation between the radon gas concentration and the environment parameters, since all the data are numeric in nature, regression can be the good choice to predict or forecast the radon data. Unlike classical regression approaches, which find a single equation for a given set of data, regression trees partition the space of examples into axis-parallel rectangles and fit a model to each of these partitions. A regression tree has a test in each inner node that tests the value of a certain attribute and, in each leaf a model for predicting the class. The model can be a linear equation or just a constant. Trees having linear equations in the leaves are also called model trees (MT) [33].

They have used the attributes - average daily barometric pressure, average daily air temperature, average daily soil temperature, difference between daily soil and daily air temperature, daily amount of rainfall, and difference in daily barometric pressure was selected. And the value of the dependent variable is daily radon concentration. They split the data into two parts: 1) Seismically active days (i.e They choose periods of 7 days before and after an earthquake) 2) Data for remaining day were included in Non seismically active class. They trained the model on non-seismic data (i.e. second part of dataset) and tested it on seismic data (i.e. first part of the dataset).

Based on the results they observed that, A Model can predict the radon concentration with a correlation of 0.8 during the seismically inactive period, provided that it is influenced only by the environmental parameters and not by the any other seismic activity. They also observed that during the seismically active period this correlation is very much low and concluded their hypothesis as true. These decrement in predictive accuracy appears for 1 to 7 days before the earthquake with local magnitude 0.8 to 3.3 [33].

Previous research has presented diverse approaches to predict radon concentration and detect anomalies in radon. Researchers have formulated different hypotheses based on specific study regions and previous research, contributing significantly to the field. However, despite their valuable contributions, these studies have certain limitations. For instance, Adil Aslam Mir, Hadeel Alsolai, et al. [18] proposed the concept of data division for efficient case analysis but did not provide a specific strategy for dividing the data. Although they demonstrated the effectiveness of ensemble models in predicting radon concentration, their research focused solely on this aspect and did not address the crucial task of anomaly de-

tection, which plays a vital role in earthquake prediction.

Our study is motivated by the scarcity of research that utilizes machine learning techniques to predict earthquakes in the specific Kutch region of India, which is classified as Zone-V (a region of very severe intensity). Similar to the Himalayan region, Kutch experiences frequent seismic events, emphasizing the need for a highly accurate alert system in this area. While extensive analysis of earthquakes and their precursors has been conducted in these regions using traditional signal processing methods [21] [22]. Hence, our primary motivation is to develop a complete end-to-end earthquake forecasting model utilizing machine learning algorithms. By addressing the shortcomings of prior work and focusing on the unique challenges of the Kutch region, we aim to create a real-time earthquake alert system that enhances the accuracy and reliability of earthquake prediction in this high-risk area.

CHAPTER 3
Data Description

The Institute of Seismological Research (ISR), located in Gandhinagar, provided the dataset for our experiments. ISR has collected the data over two years from the Pragpar Seismology Station of the Kutch region in Gujarat, India. The dataset contains five features: Radon concentration (Bq/m^3), Temperature (C°), Pressure (mBar), Humidity (%), Windspeed (Km/h). The environmental parameters viz. Temperature, Pressure, Humidity and Windspeed are used as independent variables to the machine learning model and we are predicting the radon concentration value using that model. Dataset has total 101692 samples. The samples are collected over two years from 22/01/2020 to 28/12/2021. Total 144 samples are collected on daily bases. Time duration between each sample is 10 minutes. Measurement for the day starts from 00:00:00 and ends at 23:50:00, except the first and last day of the dataset period. For the first day of the dataset the time period is from 11:50:00 to 23:50:00 and for the last day, the time period is 00:00:00 to 16:50:00. There was some missing data in the provided dataset e.g. at row index 18897 one timestamp was missing. Missing values are filled with the average of previous data samples.

During the data collection period total 23 earthquakes has been recorded, see Figure 3.1. The details for these earthquakes are present in Table 3.1. Our interested parameter in the dataset is radon concentration, the mean value of measured radon concentration is 2378.61 Bq/m^3 with standard deviation of $\pm$1773.79 Bq/m^3. The minimum and maximum value of measured radon is 188 Bq/m^3 and 14994 Bq/m^3 respectively. During the seismically active days the mean value of observed radon concentration is 3110.32 Bq/m^3, with the standard deviation of $\pm$2383.23 Bq/m^3. The minimum and maximum value of radon concentration during seismically active days are 294 Bq/m^3 and 14994 Bq/m^3.

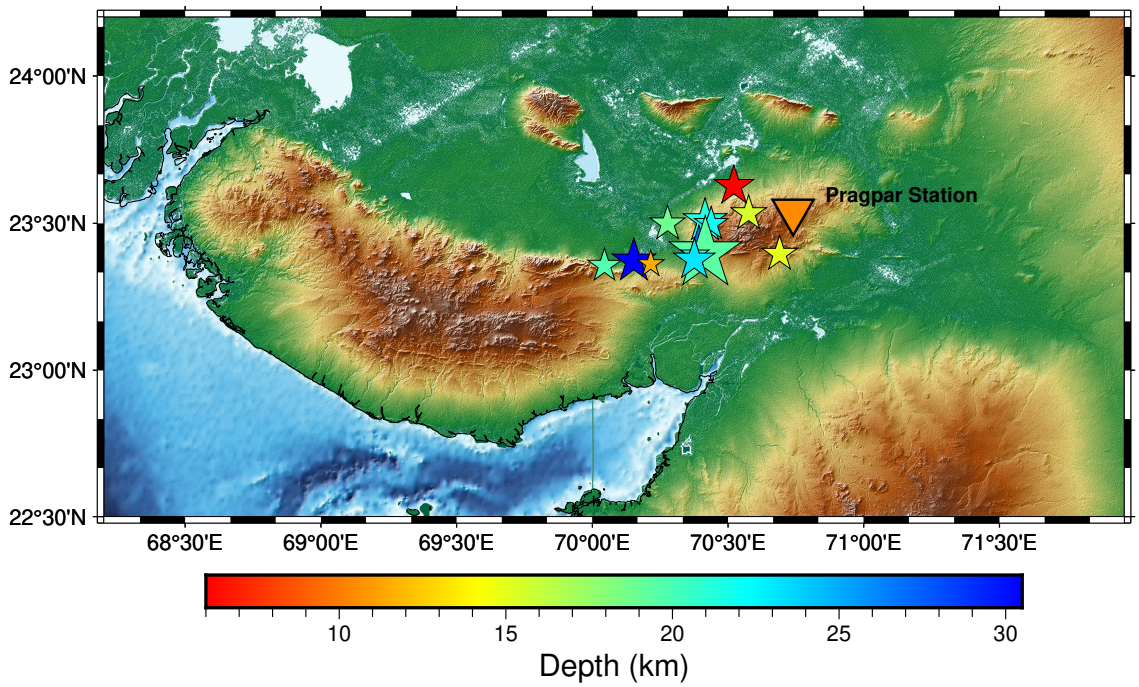


Figure 3.1: Location map of Soil radon gas concentration measuring station Pragpar, Kutch, Gujarat, India. The inverted triangle shows the location of data center; the stars shows the location of earthquakes, the size of stars describe the magnitude of earthquake (i.e. bigger size higher magnitude), color of the stars shows the depth of the earthquake epicenter

Data source: Institute of Seismological Research(ISR)

Index	Date(DD/MM/YYYY)	Time	Magnitude	Depth(Km)	Latitude	Longitude
1	02/02/2020	08:35 AM	3.3	23.394	70.378	20.8
2	04/03/2020	12:14 PM	3.2	23.362	70.215	11.5
3	14/06/2020	08:13 PM	5.3	23.397	70.416	19.4
4	05/07/2020	05:11 PM	4.2	23.412	70.393	27
5	23/07/2020	06:47 AM	3.7	23.499	70.277	18.8
6	17/08/2020	10:28 PM	3.6	23.498	70.438	22
7	23/08/2020	10:07 PM	4.1	23.509	70.415	21.3
8	02/09/2020	02:09 PM	4.1	23.372	70.153	30.5
9	07/01/2021	07:22 PM	4	23.376	70.376	23.1
10	20/01/2021	12:13 AM	3.6	23.356	70.045	19.7
11	22/01/2021	05:17 PM	3.7	23.396	70.69	15
12	04/03/2021	03:29 AM	3.9	23.627	70.522	6
13	21/03/2021	01:15 AM	3.7	23.534	70.576	15.5

Table 3.1: Earthquake details (Date, time, magnitude, depth, latitude, longitude)
Source: Institute of Seismological Research(ISR)

We calculated the Pearson correlation coefficient between each pair of the feature parameters to know the relation between environmental parameters viz—temperature, pressure, humidity, wind speed, and radon concentration. The Equation 3.1 shows the general formula for the Pearson correlation coefficient. The Table 3.2 shows the correlation between each pairs of the dataset attributes.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 (y_i - \bar{y})^2}} \tag{3.1}$$

r = correlation coefficient

x_i = values of the x variable

$\bar{x}$ = mean of the x variable

y_i = values of the y variable

$\bar{y}$ = mean of the y variable

	Radon	**Temperature**	**Pressure**	**Humidity**	**Windspeed**
Radon	1	0.222	-0.368	0.172	0.061
Temperature	0.222	1	-0.676	0.330	0.476
Pressure	-0.368	-0.676	1	-0.533	-0.339
Humidity	0.172	0.330	-0.533	1	0.108
Windspeed	0.061	0.476	-0.339	0.108	1

Table 3.2: Correlation matrix

We can see from the Table 3.2 that temperature and humidity are positively correlated with radon concentration, while pressure is negatively correlated with radon concentration. The correlation coefficient of radon and windspeed is near zero, indicating no significant correlation between radon and windspeed. temperature and pressure are negatively correlated, meaning that as temperature increases, pressure decreases, and vice versa. On the other hand, humidity and windspeed have a positive correlation with temperature, indicating that as temperature rises, humidity and windspeed also tend to increase. However, they exhibit a negative correlation with pressure, indicating that as pressure decreases, humidity and windspeed tend to rise.

We did our initial experiment on the full dataset and divided the data simply into Non-Seismically Active (NSA) and Seismically Active (SA) samples. We included the NSA data samples in the training set and SA samples in the testing set. But after the experiment, we found that this strategy was not giving acceptable prediction results, so we decided to change our approach and divide the

dataset into different settings and window sizes. Our approach to dividing data into various settings and time windows was inspired by the research conducted by Adil Aslam Mir and Hadeel Alsolai, among others, as documented in their work [18]. Since the authors' work did not provide a specific strategy for dividing the dataset into different settings, we made the decision to divide our data based on the depth and magnitude of the earthquakes. This approach was chosen in order to incorporate various scenarios into the evaluation of our machine learning models, thus ensuring a comprehensive performance testing process. Depth and magnitude serve as crucial parameters for assessing the size and destructive potential of earthquakes. To effectively analyze our seismic events, we initially created a scatter plot that plotted the events based on these two parameters. Subsequently, we divided the scatter plot into four quarters by utilizing the medians of depth and magnitude as reference lines. This division allowed us to categorize the seismic events into distinct quadrants, enabling a more detailed examination and understanding of the data. This approach will create four different settings, as shown in Figure 3.2

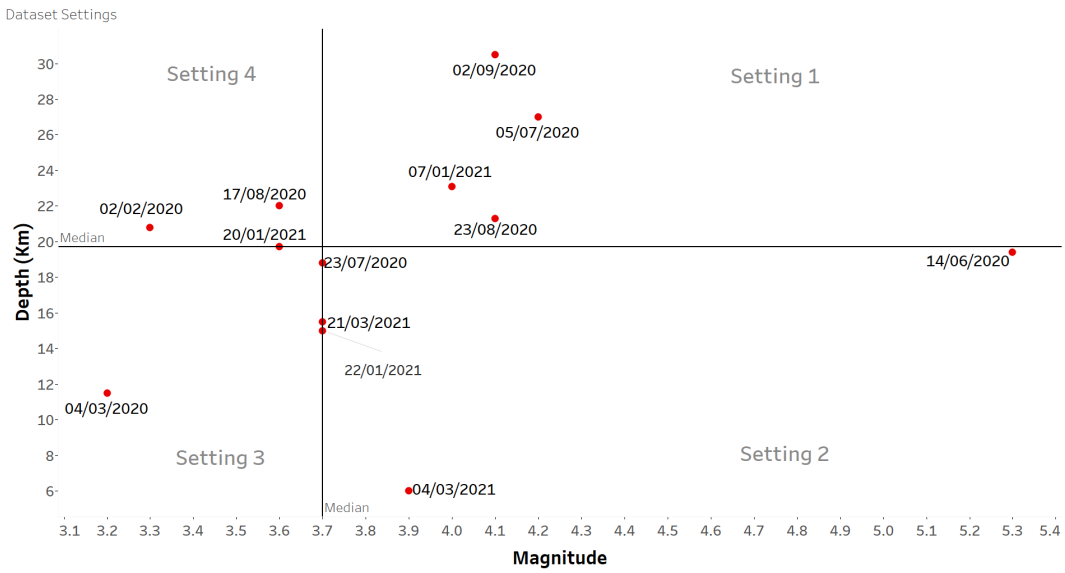


Figure 3.2: Scatter plot of seismic events using depth and magnitude parameter

The seismic events that fell within their respective quarters in the scatter plot were assigned to the corresponding settings as testing sets. And, all other seismic events, along with non-seismic events, were included in the training set. This allocation was performed to ensure that each setting contained the appropriate

seismic events for testing purposes, while the training set encompassed the remaining data points and non-seismic events. After assigning the seismic events to their respective settings and training set, we proceeded to label the samples as either "Seismically Active" (SA) or "Non-Seismically Active" (NSA) based on the window size. In our study, we used four different window sizes, namely window size 1, 2, 3, and 4. Window size 1 represented the labeling of the day prior to the seismic event, the day of the seismic event, and the day following the seismic event as SA. This same labeling approach was employed for window sizes 2, 3, and 4, whereby the labeling encompassed a range of days before and after the seismic event based on the respective window size. The incorporation of different window sizes served the purpose of identifying anomalous periods preceding or following seismic events. By employing varied window sizes, we aimed to capture the time frame during which anomalies occur, enabling us to predict earthquakes in advance. This approach allowed us to analyze and understand the patterns and signals leading up to seismic events, facilitating the development of predictive models for earthquake forecasting. As the window size increases, the number of samples in the training set decreases, while the number of samples in the testing set increases. Figure 3.8 visually illustrates the distribution of samples in different settings and windows, highlighting the varying sample counts.

In Figure 3.3, the raw radon concentration measurements are presented alongside the recorded earthquakes and their magnitudes over the data collection period. Among the seismic events, one significant event of magnitude 5.3 Mw occurred, while the remaining events ranged from magnitudes 3.1 Mw to 4.2 Mw. The time series data for the environmental parameters can be observed in Figures 3.4 to 3.7.

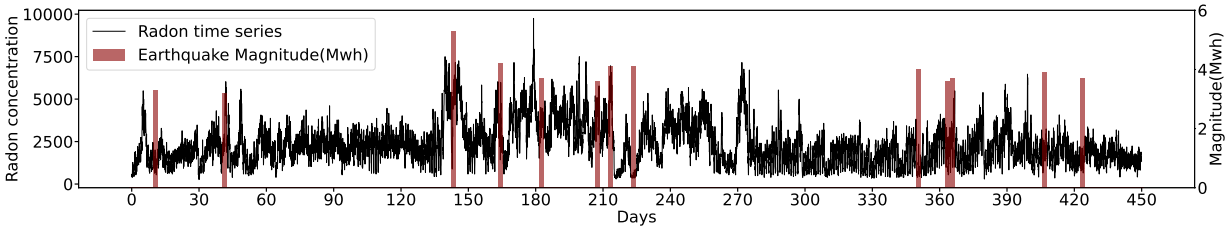


Figure 3.3: Time series data of Radon levels along with corresponding seismic events and their magnitudes during the specified time period.

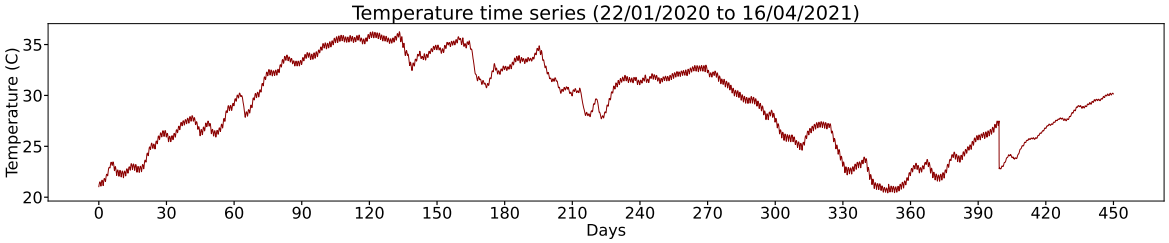


Figure 3.4: Time series data of Temperature (C°).

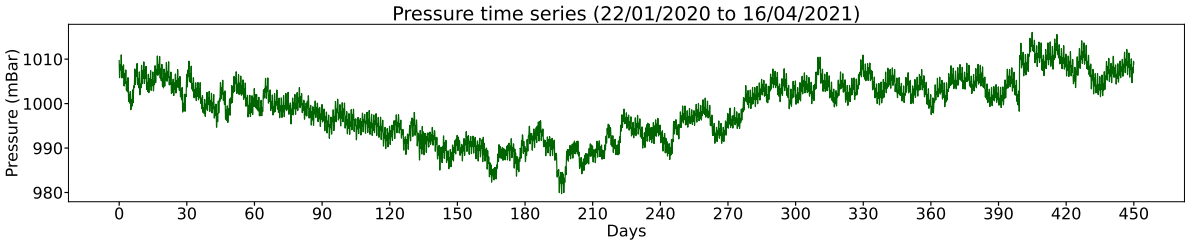


Figure 3.5: Time series data of Pressure (mBar).

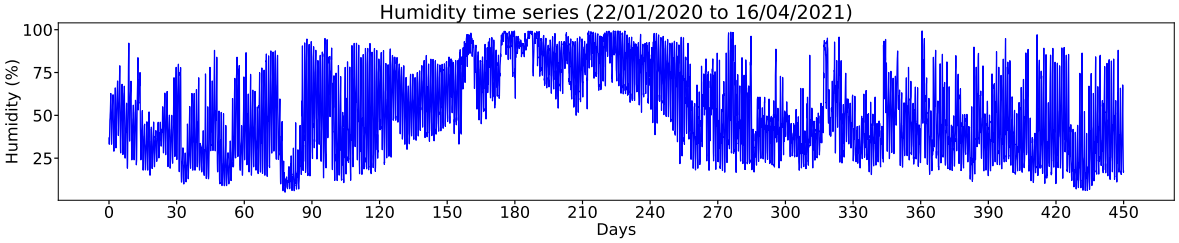


Figure 3.6: Time series data of Humidity (%).

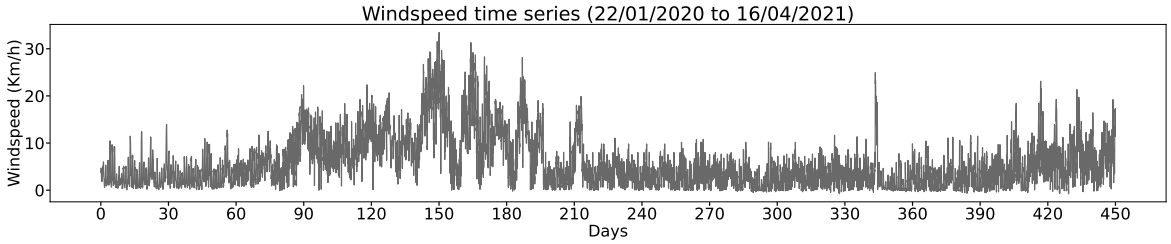


Figure 3.7: Time series data of Windspeed (Km/h).

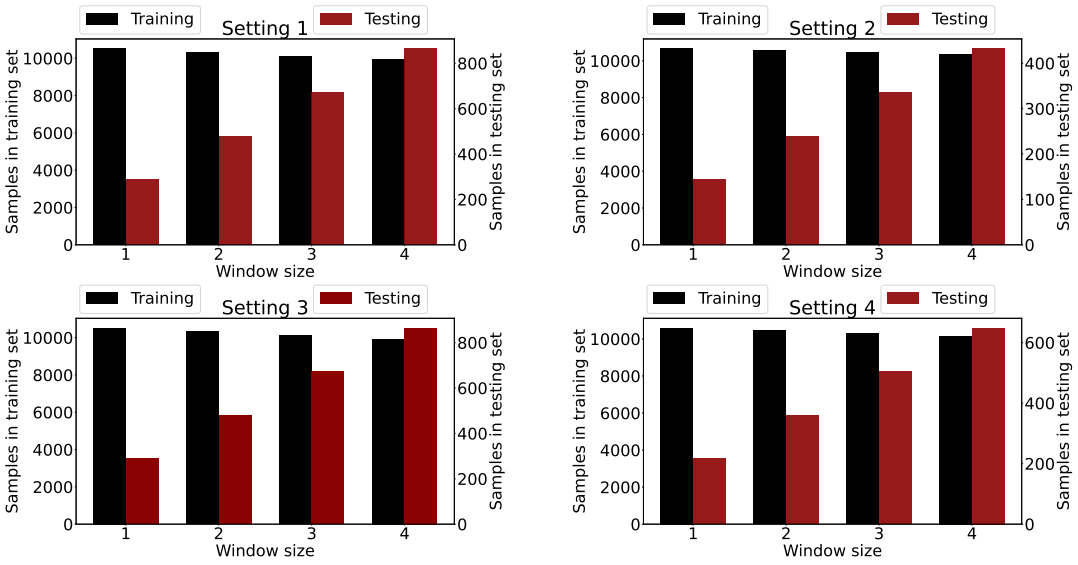


Figure 3.8: Number of samples in training and testing set of different setting with respect to window size ranging from 1 to 4

CHAPTER 4
Methods and Experiments

4.1 Data preprocessing methods

4.1.1 Empirical Mode Decomposition

The influence of environmental parameters on soil radon emission has been observed, as well as the impact of earthquake-related stress-strain changes on radon emissions preceding seismic events [15]. However, the presence of high fluctuations and higher frequency components in radon time series poses a significant challenge. To address this, it is essential to remove periodic components from the soil radon time series [21].

To achieve this, we employed the Empirical Mode Decomposition (EMD) algorithm to decompose the time series into different frequency components. These components, known as Intrinsic Mode Functions (IMFs), represent signals with distinct frequency modes. To eliminate periodicity, we applied the Fast Fourier Transform (FFT) algorithm [5] to the IMFs and removed those with higher amplitudes at 12-hour and 24-hour periods.

To identify the significant IMFs, two criteria were utilized. Firstly, we assessed the correlation of the IMFs with the raw soil radon data. Secondly, we compared the harmonic periods of all IMFs with the environmental parameters. These criteria assisted in determining which IMFs were most relevant and informative for further analysis.

Figure 4.1 displays the IMFs obtained from the radon time series decomposition using the EMD algorithm. A total of 10 IMFs were generated. Table 4.1 provides the correlation values between the radon IMFs and the raw radon series. Table 4.2 indicates whether the IMFs of the environmental parameters exhibit di-

urnal (24 hour) or semi-diurnal (12 hour) periodicity.

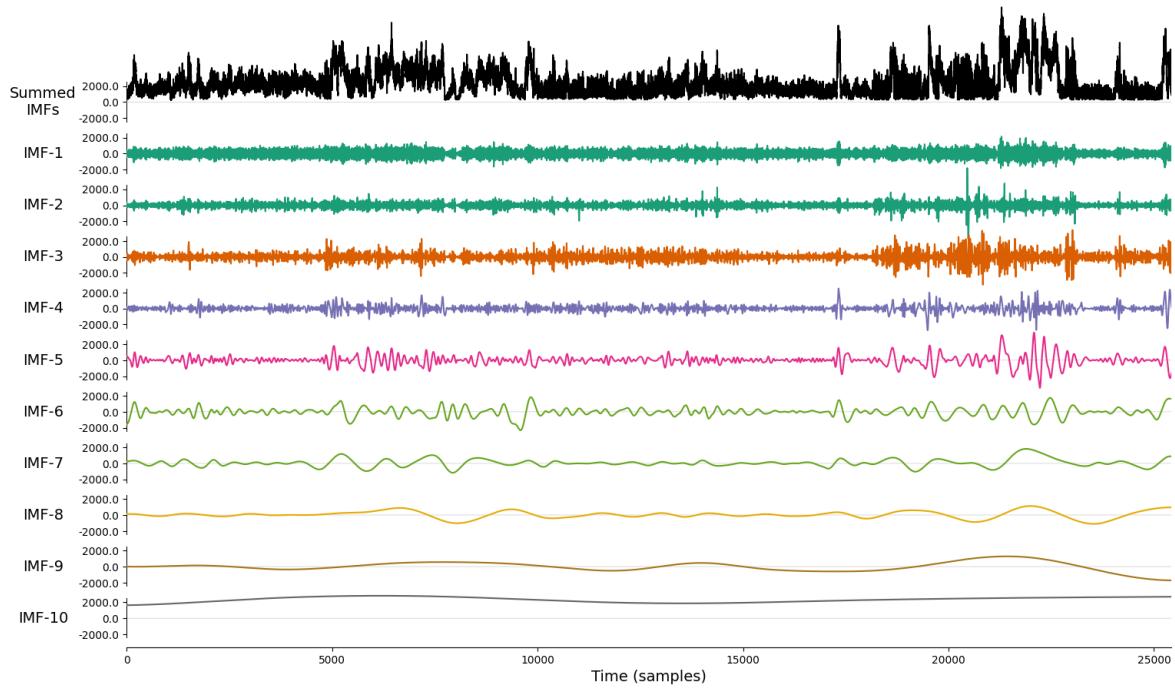


Figure 4.1: Empirical mode decomposition (EMD)-Intrinsic mode function (IMFs 1-10) of Radon time series

Subsequently, we removed the IMFs that exhibited harmonic periodicity of 12 hours or 24 hours, considering all parameters including environmental parameters and radon. Additionally, we discarded the radon IMFs with a correlation of less than 0.2 with the raw radon data. For the reconstruction of the radon time series data, we selected radon IMFs 1, 4, 5, 6, and 7, and reconstructed the data by summing these IMFs. Similarly, for the reconstruction of the environmental parameters, we removed the corresponding IMFs with 12-hour or 24-hour periodicity and summed up the remaining IMFs. Refer to Figure 4.2 for a visualization of the reconstructed time series.

IMFs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Harmonic Period	1.51 Hrs	12 Hrs	24 Hrs	3.53 Days	7.93 Days	16.05 Days	30.70 Days	88.28 Days	176.55 Days	706.22 Days
Corrleation Coefficients	0.22	0.20	0.35	0.29	0.36	0.35	0.46	0.35	0.41	0.29

Table 4.1: Harmonic periods and correlation coefficients (with raw radon data) of radon imfs

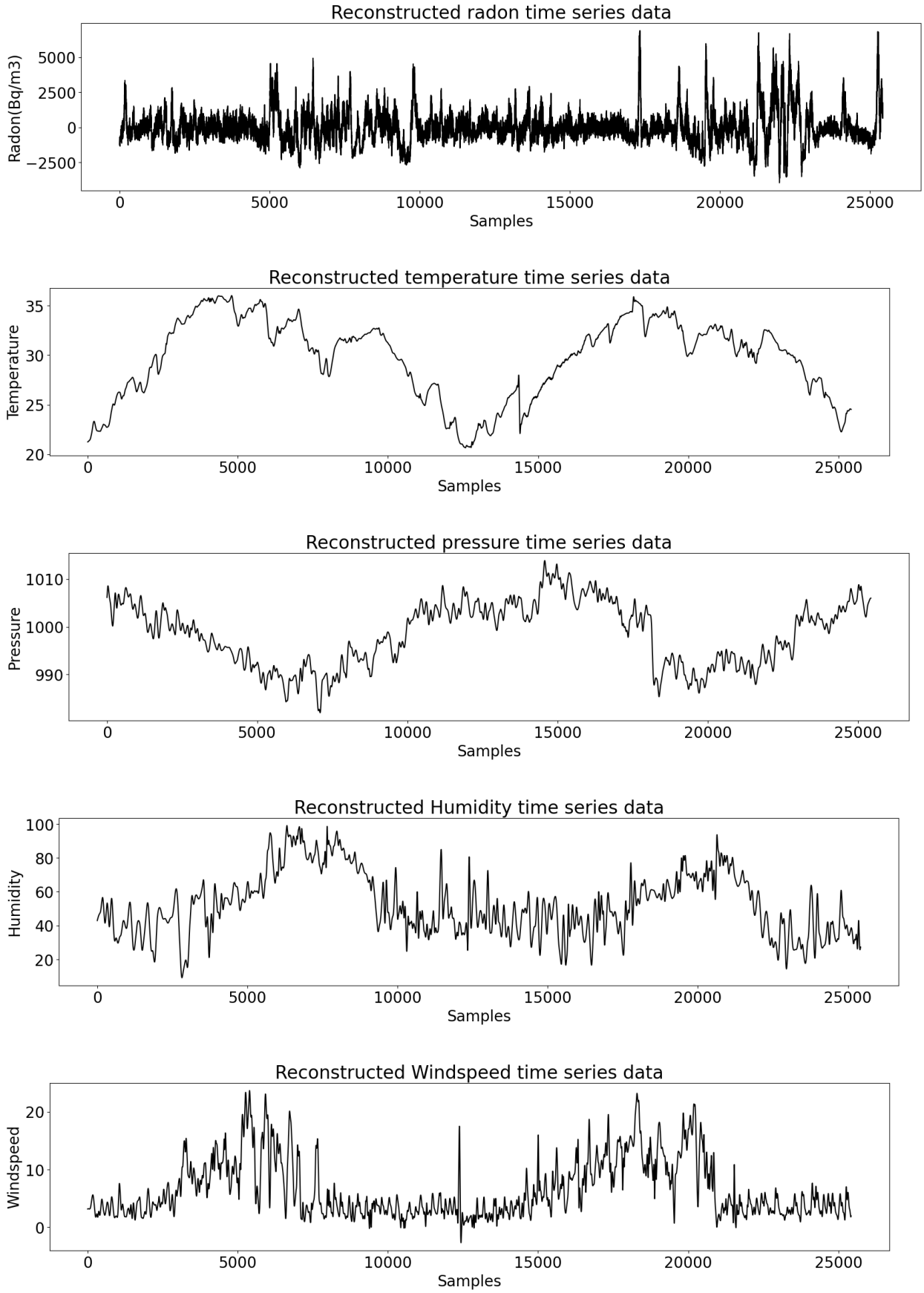


Figure 4.2: Reconstructed Radon (Bq/m^3), Temperature (C°), Pressure (mBar), Humidity (%), Windspeed (Km/h)

IMFs	1	2	3	4	5	6	7	8	9	10
Temperature	12,24 Hrs	12,24 Hrs	No	No	No	No	No	No	-	-
Pressure	12,24 Hrs	12,24 Hrs	24 Hrs	No	No	No	No	No	-	-
Humidity	12,24 Hrs	12,24 Hrs	24 Hrs	No	No	No	No	No	No	-
Windspeed	12 Hrs	12,24 Hrs	24 Hrs	No	No	No	No	No	No	No

Table 4.2: Presence of diurnal and semi diurnal periodicity in IMFs of meteorological parameters

4.1.2 Synthetic Data Generation using Geometric Averaging (GA)

Over the course of the two-year data collection period, a total of 23 seismic events occurred. Considering the approximately 730 days of data, only 23 days were identified as seismic events. This highlights a scarcity of seismic data for training a machine learning model effectively. It is crucial to have unbiased data for efficient predictions; however, in our case, the data is heavily skewed towards non-seismic days. To address this limitation and augment the available seismic events, we made the decision to employ synthetic data generation technique.

Geometric Averaging is the simple and efficient way to generate a synthetic data. The geometric mean gives a central measure for a series of number. It is the nth root of the product of n numbers [9]. A point wise weighted GA is considered for the generation of new series. Let the weight vector $W = (w_1, w_2, ..., w_n)$. The j^{th} point in each of the input time series are $x_1(j), x_2(j), ..., x_n(j)$ where j denotes the particular time instance. The j^{th} point in the synthetic time series is given by equation 4.1. To generate one seismic event time series we provided two original seismic time series to the algorithm and then it took geometric average of both input time series, which will generate new synthetic seismic event time series which follows the pattern of input time series. The dependence of input time series on synthetically generated time series is decided by the weight vector. In our case vector size of weight is 2×1.

$$x(j) = (\prod_{i=1}^{n} x_i(j)^{w_i})^{\frac{1}{\sum_{i=1}^{n} w_i}} = \exp(\frac{\sum_{i=1}^{n} w_i \ln x_i(j)}{\sum_{i=1}^{n} w_i}) \tag{4.1}$$

After some experiment we come to the conclusion that it is better to make visual cluster of seismic event time series and then generate intra-cluster synthetic data instead of all possible pair of original time series. If we go with the second scenario; which is naive approach to make pairs, it may generate non-physical data. These non-physical data can cause model to learn wrong scenarios, which

may do not occur in nature. So it is better strategy to first make clusters on original radon series and then do intra-cluster data generation. see Figure 4.3.

Figure 4.4 shows some samples of synthetically generated time series of every parameters of dataset.

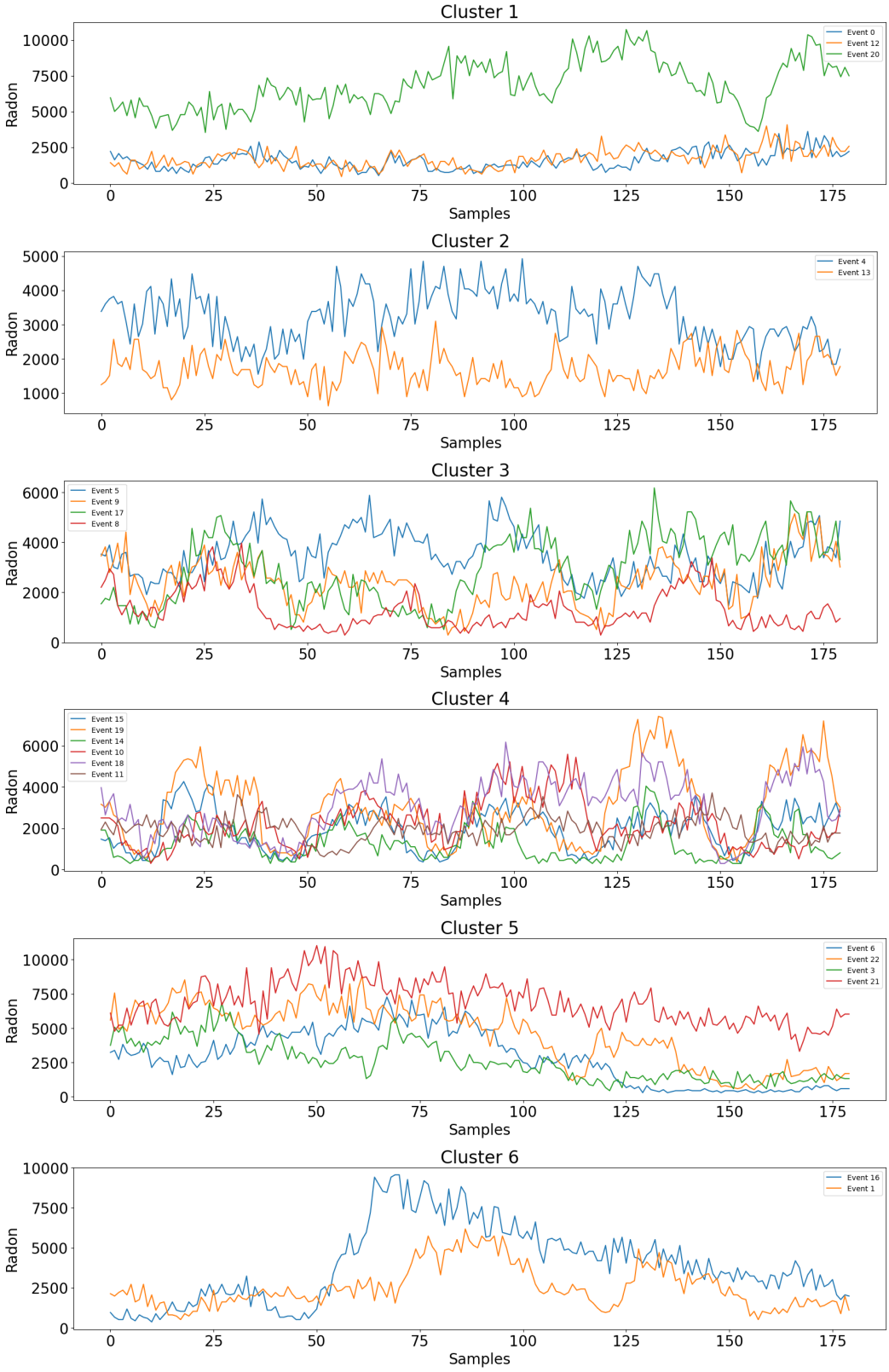


Figure 4.3: Visual clustering of seismic event radon time series

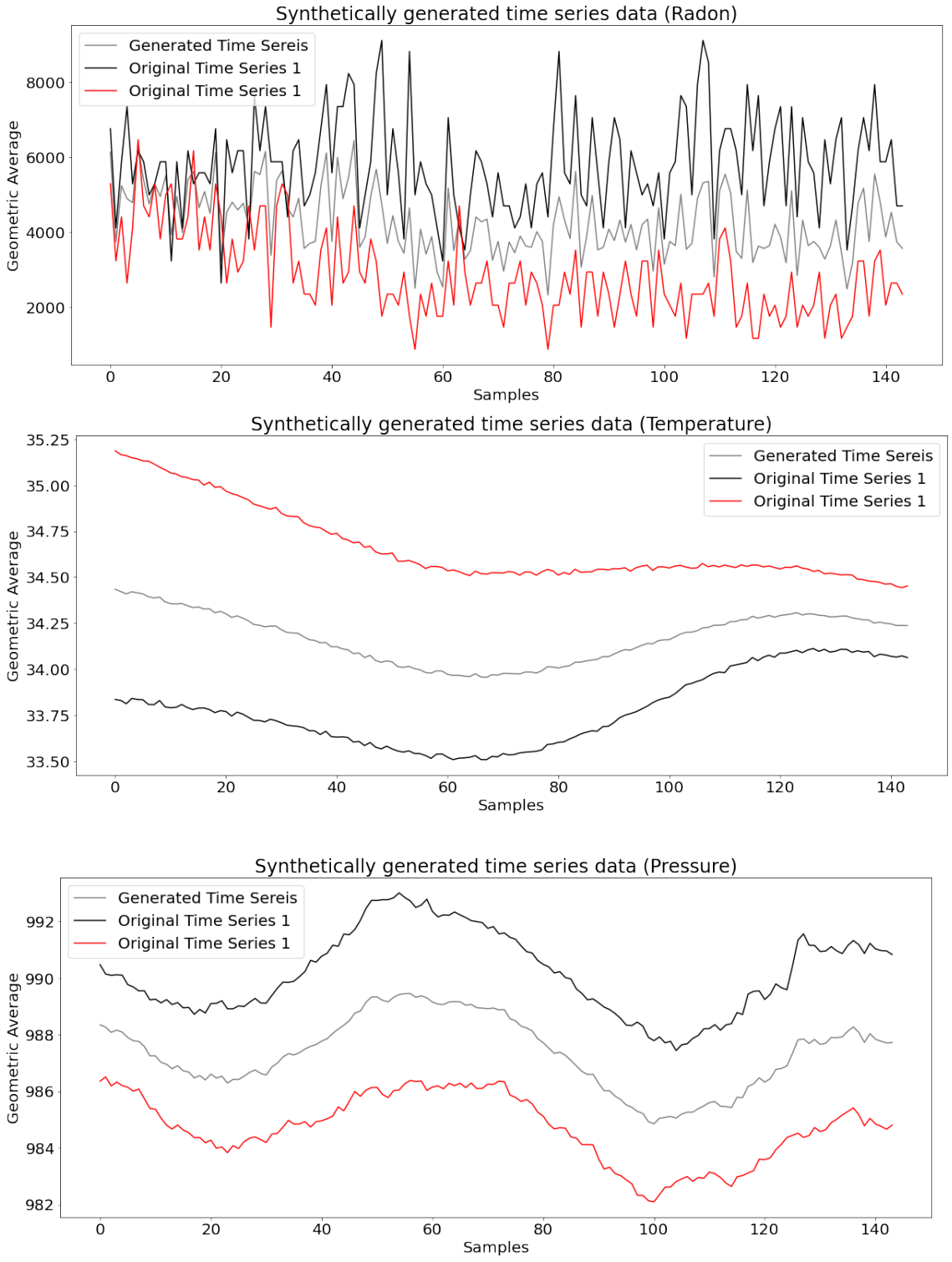


Figure 4.4: Synthetically generated Radon (Bq/m^3), Temperature (C°), Pressure (mBar)

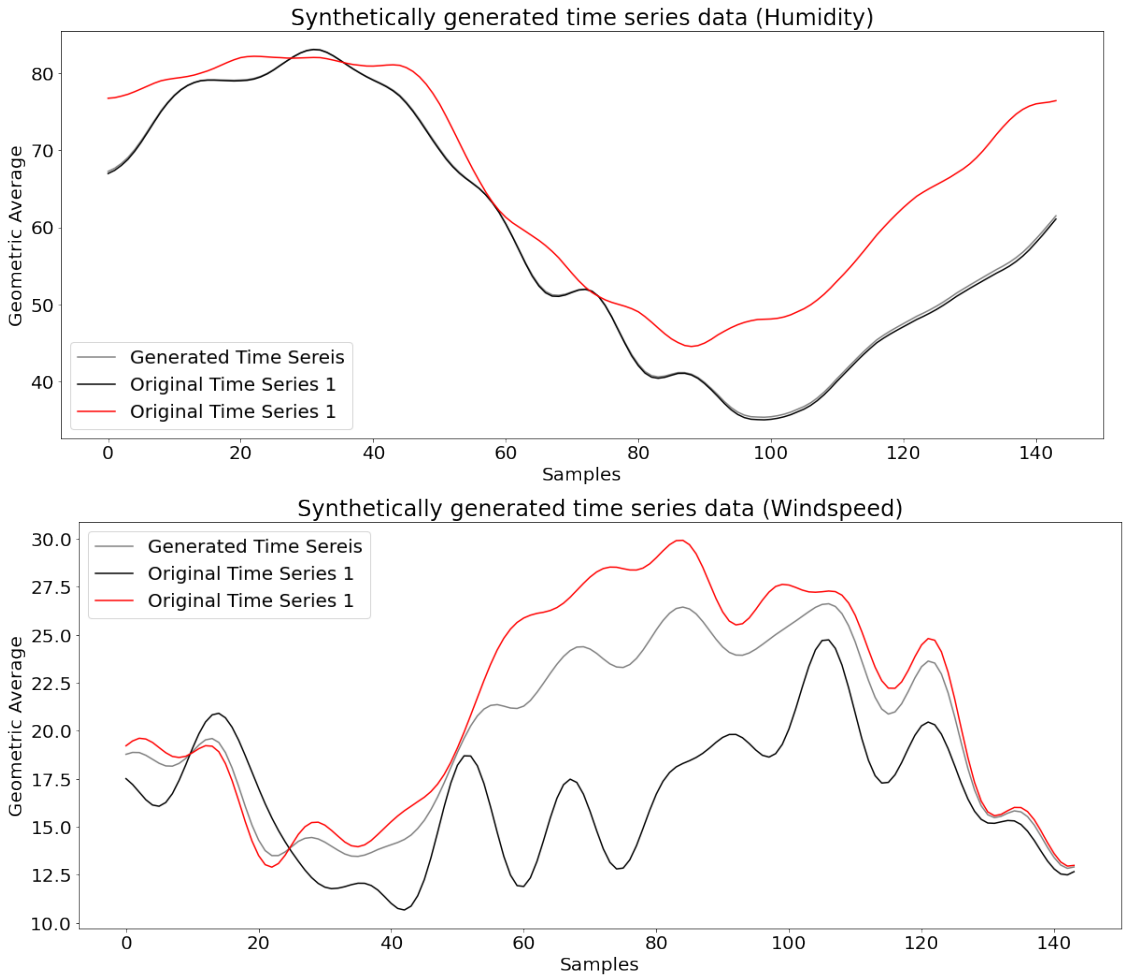


Figure 4.5: Synthetically generated Humidity (%), Windspeed (Km/h)

4.2 Machine Learning-based Prediction Models

4.2.1 Support Vector Regressor (SVR)

Support Vector Regressor (SVR) [8] is a machine learning algorithm used for regression tasks. It aims to find a function that approximates the mapping from input variables to continuous output values. The SVR approach involves mapping the input data into a higher-dimensional feature space using a kernel function, then finding the optimal hyperplane that best separates the data points while maximizing the margin.

The mathematical formulation of SVR involves solving an optimization problem that seeks to minimize the empirical risk while controlling the complexity of the model. The objective function [29] is defined as follows:

$$\min_{w,b,\xi,\xi^*} \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \right) \tag{4.2}$$

subject to the constraints:

$$\begin{gathered} y_i - w \cdot \phi(x_i) - b \leq \epsilon + \xi_i^*, \\ w \cdot \phi(x_i) + b - y_i \leq \epsilon + \xi_i, \\ \xi_i, \xi_i^* \geq 0, \end{gathered}$$

where w represents the weight vector, b is the bias term, ξ_i and ξ_i^* are slack variables, x_i and y_i are the input features and corresponding output values, ϕ is the feature mapping function, and ϵ denotes the maximum allowable deviation from the true output.

The parameter C controls the trade-off between maximizing the margin and minimizing the training error. Larger values of C lead to a smaller margin and a more accurate fit to the training data, while smaller values of C result in a larger margin but may introduce more errors.

By solving the optimization problem, SVR finds the optimal hyperplane in the feature space that maximizes the margin while satisfying the given tolerance ϵ. This allows SVR to make predictions for new input data based on their position relative to the learned hyperplane.

Overall, the SVR algorithm provides an effective approach for regression tasks by leveraging support vectors and the concept of margin maximization. Further, SVR can be used with different kernels [8]. In our experiments we have used linear and radial kernel with the SVR [8].

4.2.2 K-nearest neighbors (KNN)

The k-nearest neighbor (KNN) [11] regressor is a non-parametric machine learning algorithm used for regression tasks. It is based on the idea of finding the k nearest data points in the training set to the input data point and using their target values to predict the target value of the input data point.

The KNN regressor works by calculating the distance between the input data point and every data point in the training set. The distance metric used can be any metric that measures the similarity between two data points, such as Euclidean distance or Manhattan distance. The k-nearest neighbors of the input data point are then selected based on their proximity to the input data point. Once the k-nearest neighbors are identified, the KNN regressor uses their target values to predict the target value of the input data point. This prediction is often the average or median of the target values of the k-nearest neighbors, although other methods can be used as well.

KNN regressor is simple and easy to implement, but it can be computationally expensive, especially when the training set is large. It is also sensitive to the choice of distance metric and the value of k. In general, larger values of k tend to smooth out the predictions, while smaller values of k tend to result in more localized predictions. Overall, the KNN regressor is a useful tool for regression tasks, especially when the underlying relationship between the input and target variables is complex or nonlinear. In our case relationship between environmental parameter and radon concentration is highly nonlinear, so KNN is good choice to start with.

4.2.3 Decision Tree

The decision tree regressor [3] is a supervised learning algorithm used for regression problems. It works by recursively splitting the training data into smaller

subsets based on the feature values. At each internal node of the tree, a feature and a threshold value are selected, and the data is partitioned based on whether the feature value is less than or greater than the threshold value. The splitting process is continued until a stopping criterion is met, such as reaching a maximum depth or having too few samples in a node.

The decision tree can be represented by a binary tree, where each internal node represents a splitting decision and each leaf node represents a predicted output value. Given a new input sample, the decision tree algorithm traverses the tree from the root to a leaf node based on the feature values of the input. The predicted output is then the output value associated with the reached leaf node.

The decision tree regressor can be expressed mathematically as follows:

Let $X = x_1, x_2, ..., x_n$ be the training data, where each x_i is a m-dimensional feature vector, and $y = y_1, y_2, ..., y_n$ be the corresponding target values. The decision tree regressor learns a function $f(\cdot)$ that maps a feature vector x to a predicted output value y.

The decision tree model can be represented as a set of binary decision rules that partition the feature space into rectangular regions. Let T be the binary decision tree, and t be an internal node in T. For a feature vector x, let j_t and s_t denote the feature index and threshold value at node t. Then, the binary decision rule at node t is:

$j_i \in \{1, 2, ..., m\}$ $s_t \in \mathbb{R}$ if $x_{j_t} \leq s_t$, tghen go left, else go right

Let $R_t(x)$ denote the rectangular region defined by the binary decision rules from the root of the tree to node t. Then, the predicted output value at leaf node t is given by:

$$f(x) = \sum_{t:x \in R_t} c_t \tag{4.3}$$

where c_t is the output value associated with leaf node t.

The goal of the decision tree regressor is to learn the binary decision rules and output values that minimize the mean squared error (MSE) between the predicted and true target values on the training data:

$$\text{minimize} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{4.4}$$

This optimization problem is typically solved using a top-down greedy approach called recursive binary splitting. Starting from the root node, the algorithm searches for the best feature and threshold value that minimize the MSE on the training data. The data is then split into two subsets based on the selected feature and threshold value, and the process is repeated recursively on each subset until a stopping criterion is met.

The decision tree regressor has several hyperparameters that can be tuned to control the complexity of the tree, such as the maximum depth, minimum number of samples required to split a node, and minimum number of samples required to be at a leaf node. Overfitting can be a concern with decision trees, as the model can become too complex and fit the noise in the training data.

4.2.4 Random Forest

The Random Forest Regressor [4] is a machine learning algorithm used for regression tasks. It is an ensemble learning method that combines multiple decision trees to make predictions. Each tree in the forest is trained on a random subset of the data, and the final prediction is obtained by averaging or taking the majority vote of the individual tree predictions.

The mathematical formulation of the Random Forest Regressor involves training a collection of decision trees. Each decision tree is constructed based on a subset of the training data and a random subset of the features. The prediction of an individual decision tree is denoted as $\hat{y}_i$, and the final prediction of the Random Forest Regressor is obtained by averaging the predictions of all the trees. Mathematically, the prediction of the Random Forest Regressor for a given input vector x can be represented as:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} \hat{y}_i \tag{4.5}$$

where N represents the number of trees in the forest.

The Random Forest Regressor also provides an estimate of the uncertainty as-

sociated with the predictions, often referred to as the prediction variance. This can be calculated using the variance of the individual tree predictions:

$$Var\left(\hat{y}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - \hat{y}\right)^2 \tag{4.6}$$

Overall, the Random Forest Regressor combines the predictions of multiple decision trees to provide a robust and accurate regression model [**?**]. By leveraging the diversity of the individual trees, it can effectively handle nonlinear relationships and capture complex patterns in the data.

4.2.5 Gradient Boosting Machine (GBM)

Gradient Boosting Machine (GBM) regressor [12] is a powerful machine learning algorithm used for regression tasks. It is based on the idea of combining multiple weak regression models to create a strong overall model. Each weak model is trained on the residuals of the previous model, which are the differences between the actual target values and the predicted values of the previous model. Mathematically, GBM can be expressed as follows:

Given a training set of input features x_i and target values y_i, where $i = 1, 2, ..., n$, and a loss function $L(y, f(x))$ that measures the difference between the true target value y and the predicted value $f(x)$, the objective of GBM is to find a function $F(x)$ that minimizes the following loss function:

$$L(y, F(x)) = sum(L(y_i, F(x_i)))$$

The GBM regressor works by iteratively adding new weak models to the ensemble, each of which is trained on the negative gradients of the loss function with respect to the current predictions. Specifically, at each iteration t, a new weak model $f_t(x)$ is trained to minimize the following loss function:

$$L_t = sum(L(y_i, F_{t-1}(x_i) + f_t(x_i)))$$

where $F_{t-1}(x_i)$ is the prediction of the ensemble at iteration t-1.

Once the new model $f_t(x)$ is trained, its predictions are added to the ensemble by updating the previous predictions with a learning rate parameter γ:

$$F_t(x) = F_{t-1}(x) + \gamma f_t(x)$$

This process is repeated for a specified number of iterations or until a desired level of accuracy is achieved.

GBM is known for its high accuracy and ability to handle complex nonlinear relationships between the input and target variables. However, it can be computationally expensive and prone to overfitting if the number of iterations or the learning rate is too high. In practice, cross-validation and regularization techniques are often used to optimize the hyperparameters of the GBM model.

4.2.6 Extreme Gradient Boosting (XGBoost)

XGBoost (Extreme Gradient Boosting) [6] is a popular machine learning algorithm for regression, classification, and ranking tasks. It is an ensemble method that combines multiple weak models (decision trees) to create a strong model. The mathematical expression for XGBoost regressor is as follows:

Given a training dataset with n instances and m features, let $X = x_1, x_2, ..., x_n$ denote the feature matrix, and $y = y_1, y_2, ..., y_n$ denote the corresponding target values. We aim to learn a function $f(x)$ that maps the input features to the output targets. The XGBoost regressor works by iteratively adding decision trees to the model. At each iteration, a new decision tree is fit to the negative gradient of the loss function with respect to the current predictions. The loss function is typically chosen to be the mean squared error (MSE) for regression tasks.

Let $F_t(x)$ denote the predicted target values at iteration t, and $r_{it} = y_i - F_{t-1}(x_i)$ denote the residual errors. The goal of XGBoost is to minimize the following objective function:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} (y_i - F_{t-1}(x_i) - h_t(x_i))^2 + \Omega(h_t) \tag{4.7}$$

where $h_t(x)$ is the t-th decision tree, and $\Omega(h_t)$ is a regularization term that penalizes complex models. The regularization term is typically defined as:

$$\Omega(h_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{4.8}$$

where T is the number of leaf nodes in the tree, w_j is the weight of the j-th leaf node, and γ and λ are hyperparameters that control the strength of regularization.

The XGBoost algorithm works by iteratively adding decision trees to the model. At each iteration, a new decision tree $h_t(x)$ is fit to the negative gradient of the loss function with respect to the current predictions, using a technique called gradient boosting. The final prediction is then obtained by summing the predictions of all the trees:

$$F(x) = \sum_{t=1}^{T} h_t(x) \tag{4.9}$$

where T is the total number of trees in the model. The optimal values of the hyperparameters γ and λ are typically chosen through cross-validation.

4.3 Performance metrics

To evaluate the accuracy of predictions for radon concentration (RN) based on attributes such as temperature, pressure, humidity, and windspeed, various commonly used performance metrics are computed. One frequently used metric is the Root Mean Squared Error (RMSE), which is commonly applied to prediction models in various fields. RMSE is sensitive to outliers, as a large difference between actual and predicted values has a significant impact on its value. The RMSE can be calculated using the following Equation 4.10:

$$\text{RMSE} = \sqrt{\frac{1}{\text{V}} \sum_{n=1}^{\text{V}} (y_n - \hat{y}_n)^2} \tag{4.10}$$

Here, V represents the total number of samples,
y_n represents the actual measured radon concentration value of nth sample,
$\hat{y_n}$ represents the predicted radon concentration value of nth sample.

When calculating RMSE, the presence of outliers can greatly affect the error term. In such cases, the Root Mean Squared Logarithmic Error (RMSLE) can be used to

mitigate the influence of outliers. RMSLE is calculated using the Equation 4.11.

$$\text{RMSLE} = \sqrt{\frac{1}{\text{V}} \sum_{n=1}^{\text{V}} \left(\log\left(y_n + 1\right) - \log\left(\hat{y}_n + 1\right)\right)^2} \tag{4.11}$$

Again, V represents the total number of samples.

RMSLE is particularly useful when the predicted and actual values have a wide range and when there are outliers. It scales down the effect of outliers and nullifies their impact.

Another frequently used performance metric is the Mean Absolute Percentage Error (MAPE), which assesses the accuracy of a prediction model. MAPE is computed as the average of the absolute percentage errors and is given by Equation 4.12

$$\text{MAPE} = \frac{1}{\text{V}} \sum_{n=1}^{\text{V}} \left| \frac{y_n - \hat{y_n}}{y_n} \right| \tag{4.12}$$

MAPE is popular due to its scale independence and easy interpretation. However, it has some limitations, such as producing undefined or infinite values when the actual values are close to zero or equal to zero. Magnitudes less than 1 for actual values can yield higher percentage MAPE values, while actual zero values result in infinite MAPE values.

Percentage Bias (PB) describes the tendency of predicted values to be consistently larger or smaller than their corresponding actual values. PB is computed using the following equation, based on V samples:

$$\text{PB} = 100 \times \frac{\sum_{n=1}^{\text{V}} \left(\hat{y_n} - y_n\right)}{\sum_{n=1}^{V} y_n} \tag{4.13}$$

Positive PB values indicate an overestimation bias, while negative values indicate an underestimation bias. A PB value of 0 represents an accurate model simulation.

These performance metrics provide insights into the accuracy and fit of regression models for predicting radon concentration. Each metric has its own advantages and limitations, and the choice of metric depends on the specific requirements and characteristics of the prediction problem at hand. We have used all

metrics in out experiments for better analysis of model performance.

The overall flow of our work is given in Figure 4.6

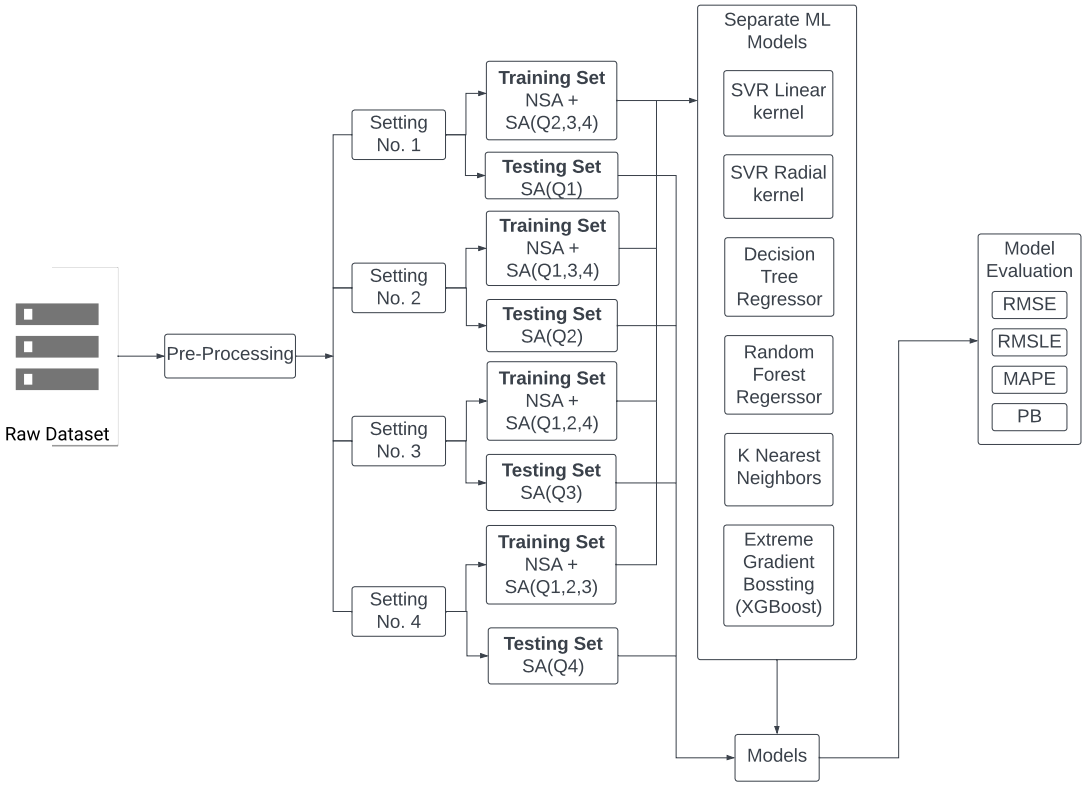


Figure 4.6: Methodology Flowchart illustrating the step-by-step process for data collection, preprocessing, feature selection, model training, and evaluation.

CHAPTER 5
Results and Discussion

The present study conducted a series of experiments to investigate the prediction of seismic events using radon time series data and machine learning models. In Experiment 1, the original dataset was utilized, and various models such as SVR linear, radial kernel, Random Forest, K-nearest neighbor, Gradient Boosting Machine, and Extreme Gradient Boosting were trained and evaluated. Experiment 2 involved augmenting the original dataset with synthetic seismic data and focused on K-nearest neighbor and Extreme Gradient Boosting models. The inclusion of gradient features in Experiment 2 further enhanced the performance of the models. Experiment 3 explored the application of Empirical Mode Decomposition (EMD) to reconstruct the radon data, but due to issues with maintaining the original value range, EMD was excluded from subsequent experiments. In Experiment 4, a segmented and resampled dataset was used, achieving satisfactory results compared to previous experiments. Additionally, anomaly detection based on mean and standard deviation was employed to identify deviations in radon measurements prior to seismic events. Figure 5.1 provides an overview of the experimental structure and summarizes the key findings of the study.

The findings revealed that the models were able to identify anomalies before several seismic events, indicating their potential as an early warning system. The analysis showed that anomalies in radon measurements were detected prior to eight out of thirteen seismic events, and the model accurately captured variations outside the confidence bounds. However, the effectiveness of the approach varied depending on the specific event and the number of anomalous measurements observed.Overall, these findings demonstrate the capability of machine learning models in predicting seismic events using radon time series data and emphasize the significance of monitoring radon measurements for earthquake detection. Figure 5.1 provides a comprehensive summary of the conducted experiments and their corresponding results, which will be presented in detail in the subsequent

sections.

5.1 Experiment 1: Original dataset trained on SVR linear, radial kernel, Random Forest, K nearest neighbor, Gradient Boosting Machine and Extreme gradient boosting algorithms

In the first experiment, the original dataset was utilized. To address the specific requirements of the study, the dataset was resampled from a 10-minute interval to a 40-minute interval. Furthermore, in order to ensure comparability and enhance the performance of the Support Vector Regression (SVR) linear and radial kernel models, all the features were normalized using z-score normalization.

For the purpose of analysis, only seismic events with a magnitude greater than 3.8 were considered as seismically active samples, while other seismic events were categorized as non-seismic events. This selection criterion aimed to focus on significant seismic events and distinguish them from less significant ones, providing a more targeted analysis.

In all of the Figures from 5.2 to 5.8 presented, the red line represents the predicted radon concentration values, while the black line represents the actual measured radon values. From the analysis of the results figures, it is evident that the SVR models with linear and radial kernels perform poorly in this specific experiment setup. However, the ensemble models such as Gradient Boosting Machine (GBM) and XGBoost demonstrate satisfactory performance. Interestingly, even the simple machine learning model K-Nearest Neighbors (KNN) yields similar results to these ensemble models. The detailed tables presenting the performance metrics can be found in the appendix section of the thesis.

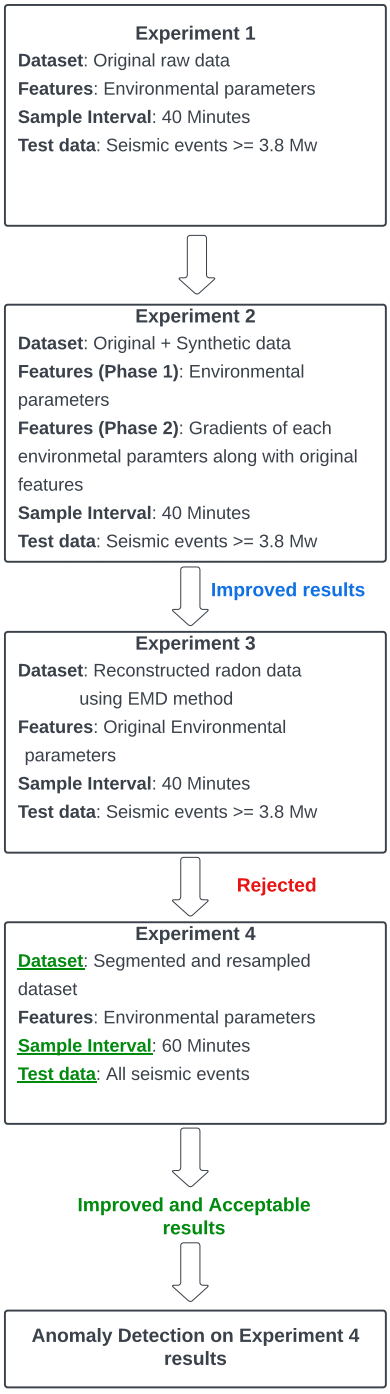
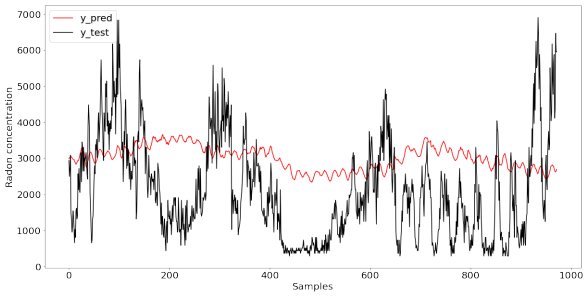
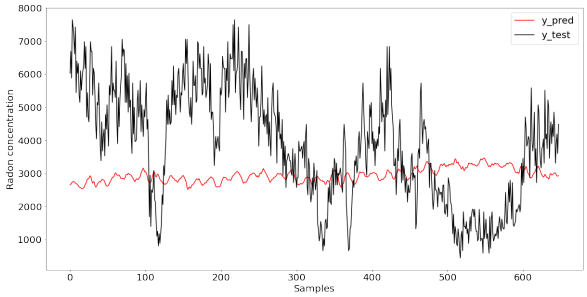


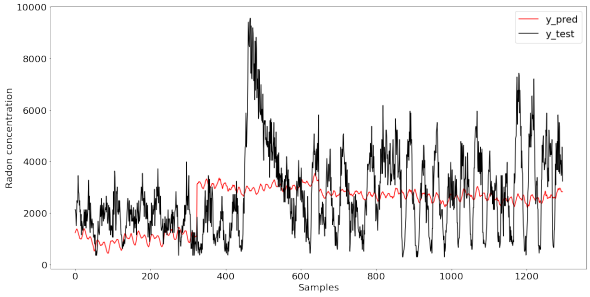
Figure 5.1: Overview of Experimental Strategy and Path for Conducting the Experiments



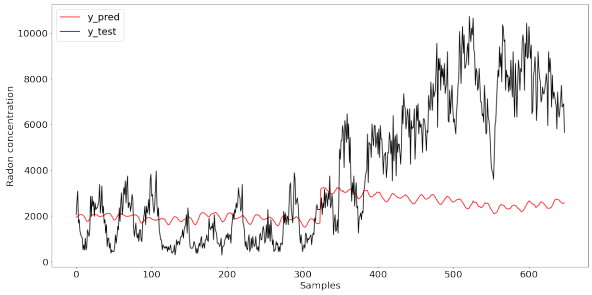
(a) Setting 1 Window 4



(b) Setting 2 Window 4

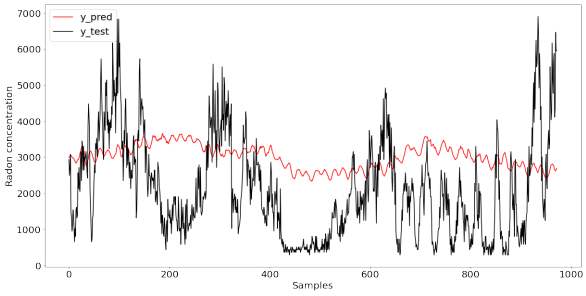


(c) Setting 3 Window 4

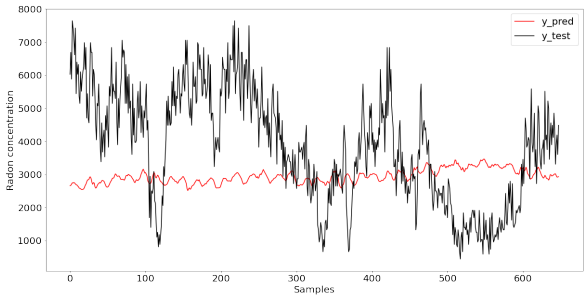


(d) Setting 4 Window 4

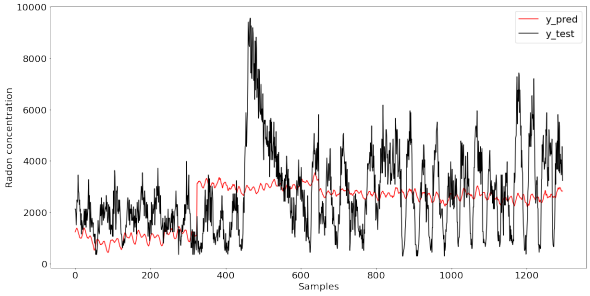
Figure 5.2: Results of SVR linear kernel on window 4



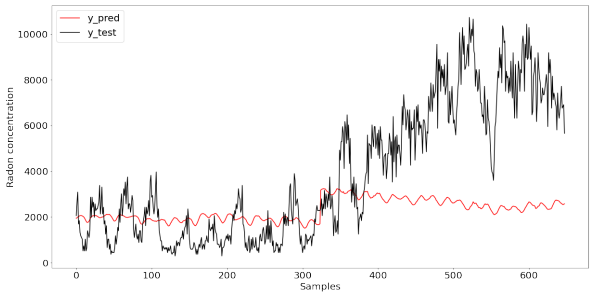
(a) Setting 1 Window 4



(b) Setting 2 Window 4

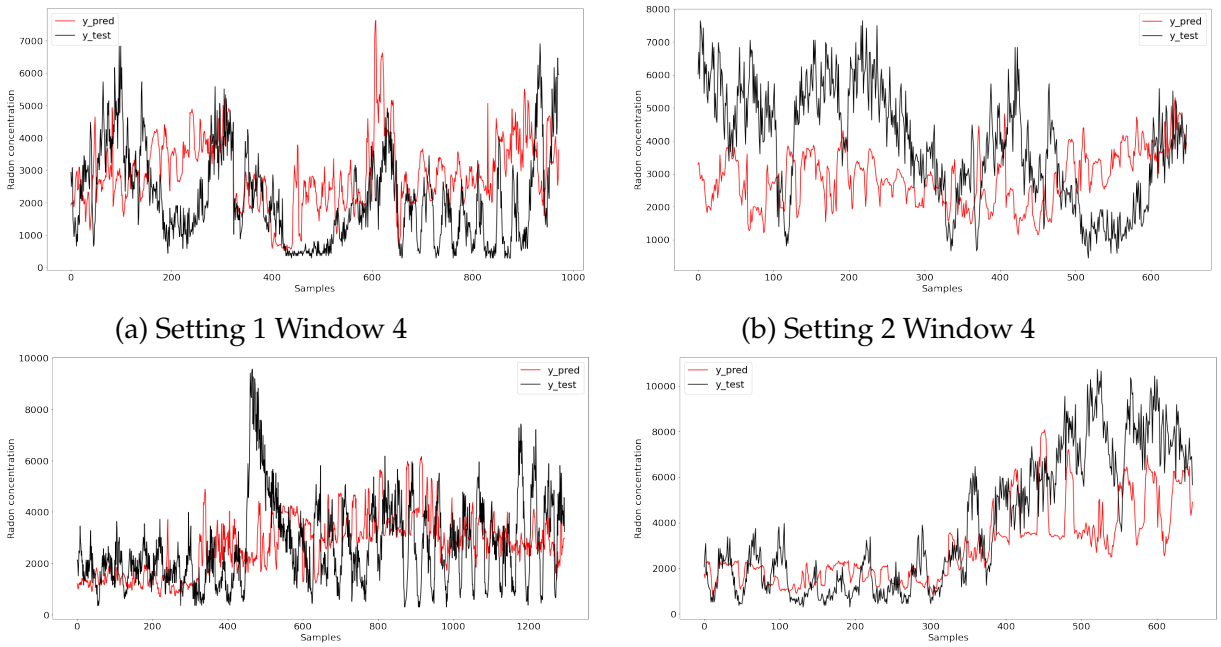


(c) Setting 3 Window 4



(d) Setting 4 Window 4

Figure 5.3: Results of SVR radial kernel on window 4



(a) Setting 1 Window 4

(b) Setting 2 Window 4

(c) Setting 3 Window 4

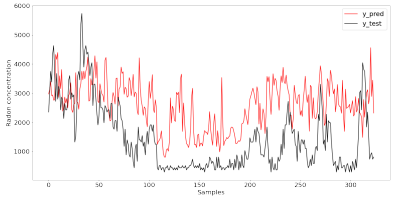
(d) Setting 4 Window 4

Figure 5.4: Results of Random Forest regressor on window 4

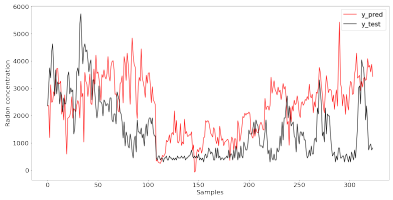
5.2 Experiment 2: Original dataset with synthetically generated dataset trained on K nearest neighbor (KNN) and Extreme gradient boosting (XGBoost) algorithms

In Experiment 2, the original dataset was augmented with synthetic seismic data, while keeping the experimental settings similar to the previous experiment. Based on the unsatisfactory results obtained from other models, we decided to focus on using only two models: K-Nearest Neighbors (KNN) and XGBoost. Among the three models tested (KNN, Gradient Boosting Machine, and XGBoost), KNN, and XGBoost demonstrated satisfactory results. The results of GBM and XGBoost were comparable, but GBM required more training time compared to XGBoost. Therefore, we opted to proceed with XGBoost as our chosen model. Consequently, the experiment was carried out using KNN and XGBoost models for further analysis.

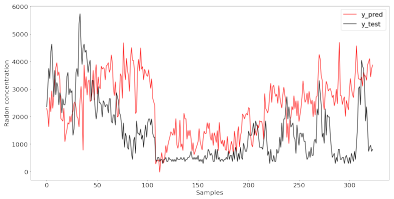
In Phase 1 of Experiment 2, the machine learning models were trained using only the environmental parameters as input. However, in Phase 2 of Experiment 2, we extended the feature set by including the gradients (first-order derivatives)



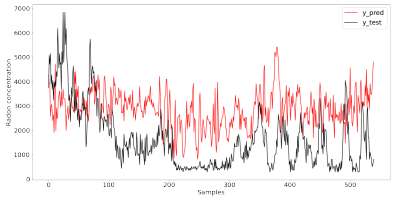
(a) KNN on Setting 1 Window 1



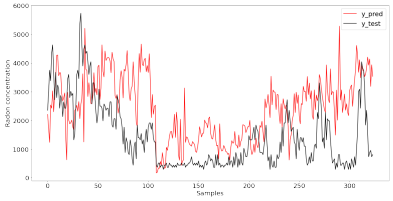
(b) GBM on Setting 1 Window 1



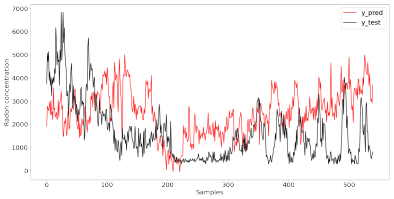
(c) XGB on Setting 1 Window 1



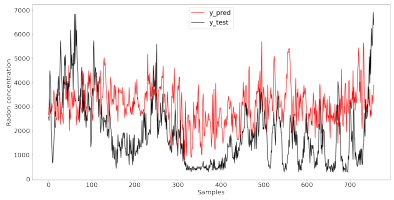
(d) KNN on Setting 1 Window 2



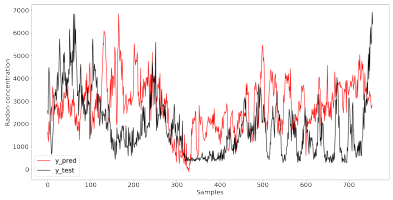
(e) GBM on Setting 1 Window 2



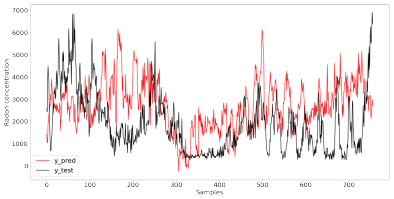
(f) XGB on Setting 1 Window 2



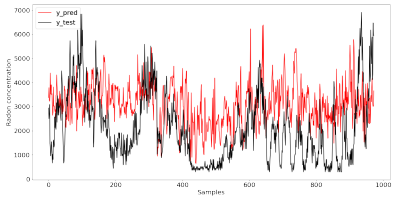
(g) KNN on Setting 1 Window 3



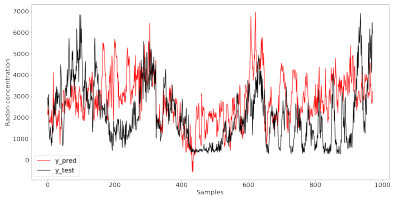
(h) GBM on Setting 1 Window 3



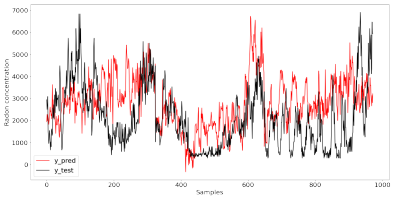
(i) XGB on Setting 1 Window 3



(j) KNN on Setting 1 Window 4

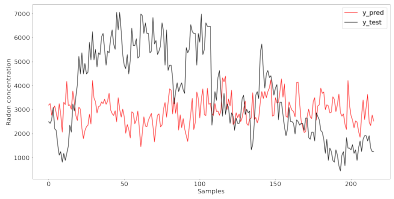


(k) GBM on Setting 1 Window 4

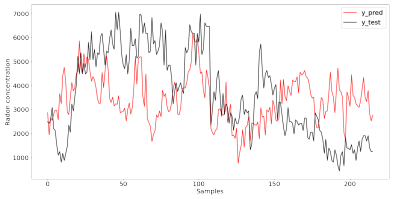


(l) XGB on Setting 1 Window 4

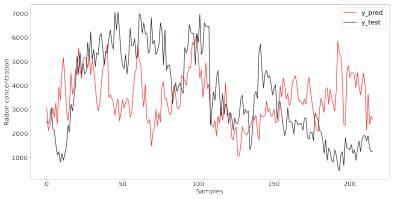
Figure 5.5: Results of KNN, GBM and XGBoost on setting 1 and all windows



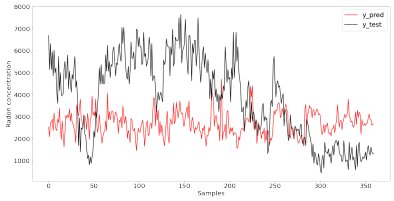
(a) KNN on Setting 2 Window 1



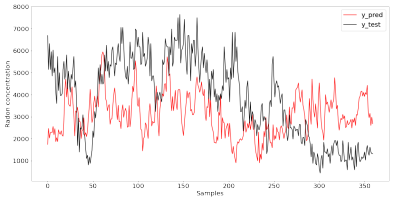
(b) GBM on Setting 2 Window 1



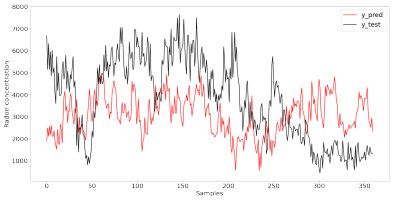
(c) XGB on Setting 2 Window 1



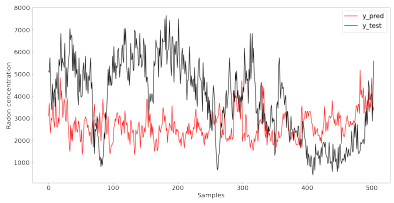
(d) KNN on Setting 2 Window 2



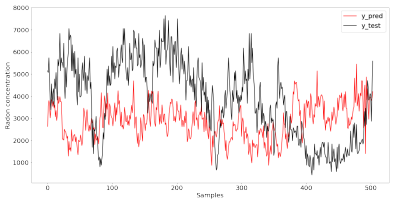
(e) GBM on Setting 2 Window 2



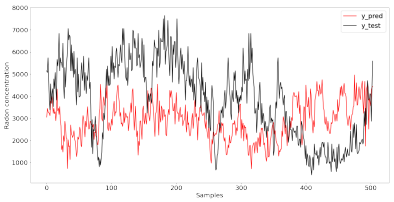
(f) XGB on Setting 2 Window 2



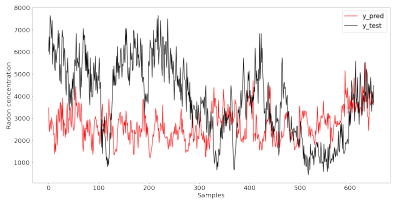
(g) KNN on Setting 2 Window 3



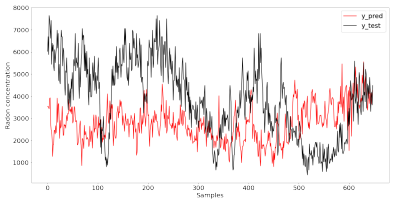
(h) GBM on Setting 2 Window 3



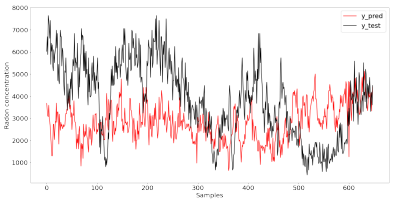
(i) XGB on Setting 2 Window 3



(j) KNN on Setting 2 Window 4

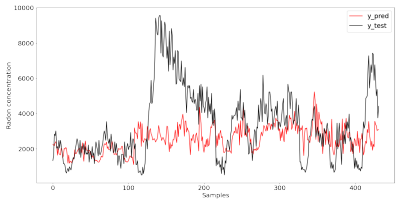


(k) GBM on Setting 2 Window 4

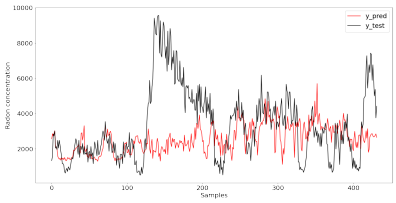


(l) XGB on Setting 2 Window 4

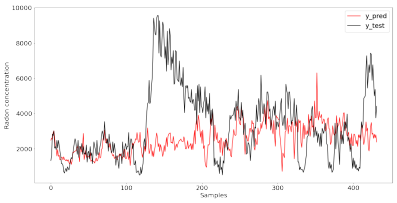
Figure 5.6: Results of KNN, GBM and XGBoost on setting 2 and all windows



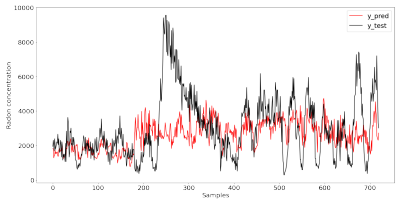
(a) KNN on Setting 3 Window 1



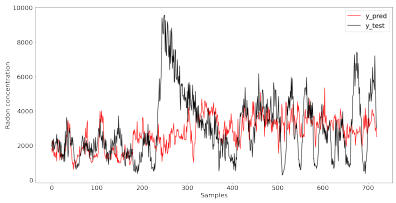
(b) GBM on Setting 3 Window 1



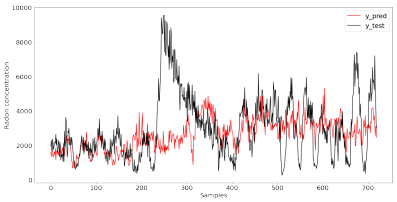
(c) XGB on Setting 3 Window 1



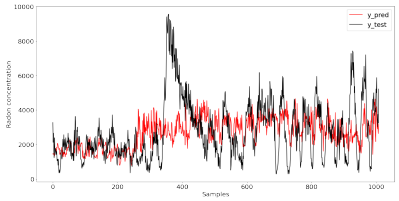
(d) KNN on Setting 3 Window 2



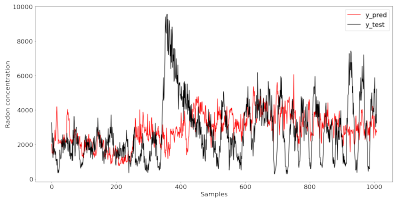
(e) GBM on Setting 3 Window 2



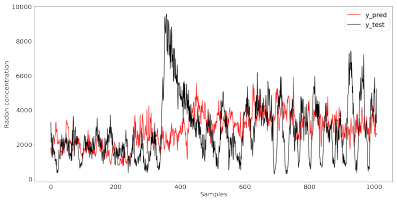
(f) XGB on Setting 3 Window 2



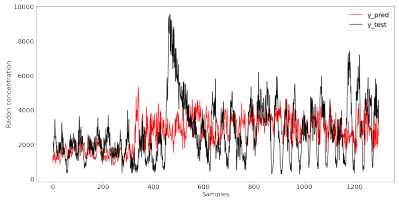
(g) KNN on Setting 3 Window 3



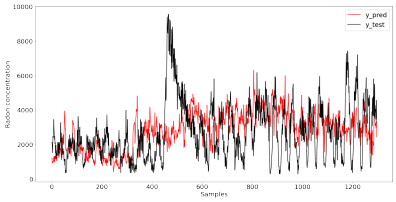
(h) GBM on Setting 3 Window 3



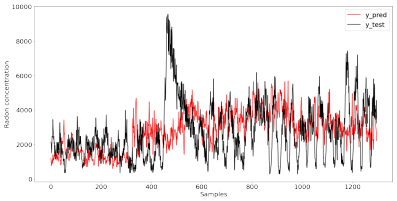
(i) XGB on Setting 3 Window 3



(j) KNN on Setting 3 Window 4

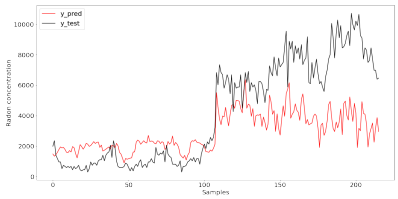


(k) GBM on Setting 3 Window 4

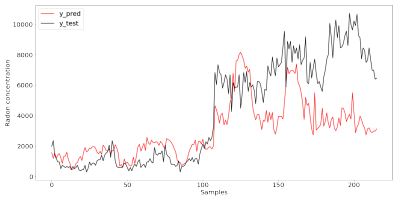


(l) XGB on Setting 3 Window 4

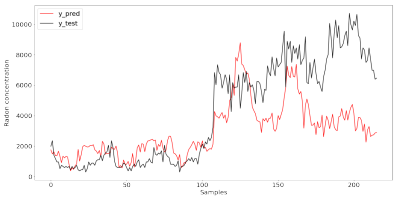
Figure 5.7: Results of KNN, GBM and XGBoost on setting 3 and all windows



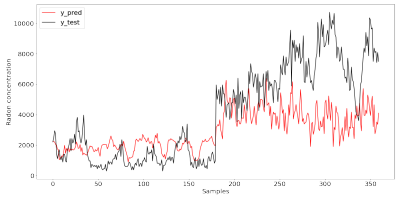
(a) KNN on Setting 4 Window 1



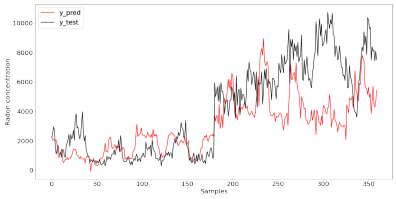
(b) GBM on Setting 4 Window 1



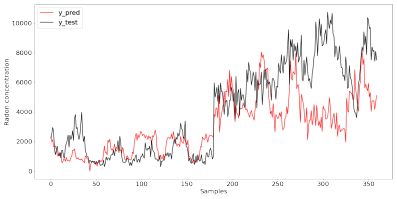
(c) XGB on Setting 4 Window 1



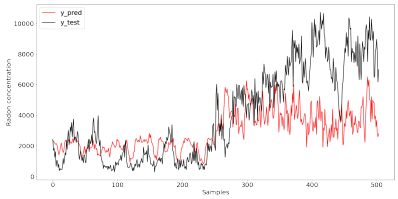
(d) KNN on Setting 4 Window 2



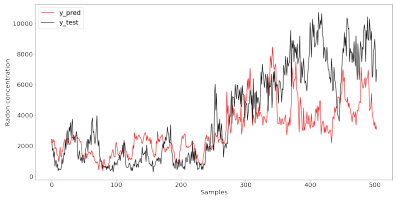
(e) GBM on Setting 4 Window 2



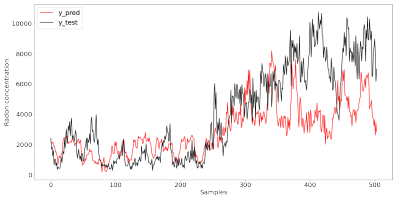
(f) XGB on Setting 4 Window 2



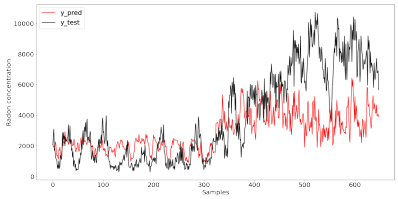
(g) KNN on Setting 4 Window 3



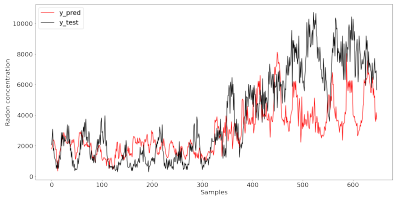
(h) GBM on Setting 4 Window 3



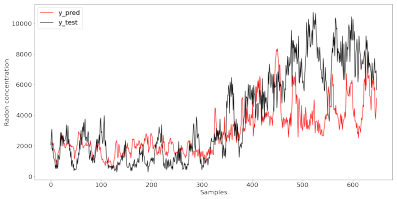
(i) XGB on Setting 4 Window 3



(j) KNN on Setting 4 Window 4



(k) GBM on Setting 4 Window 4



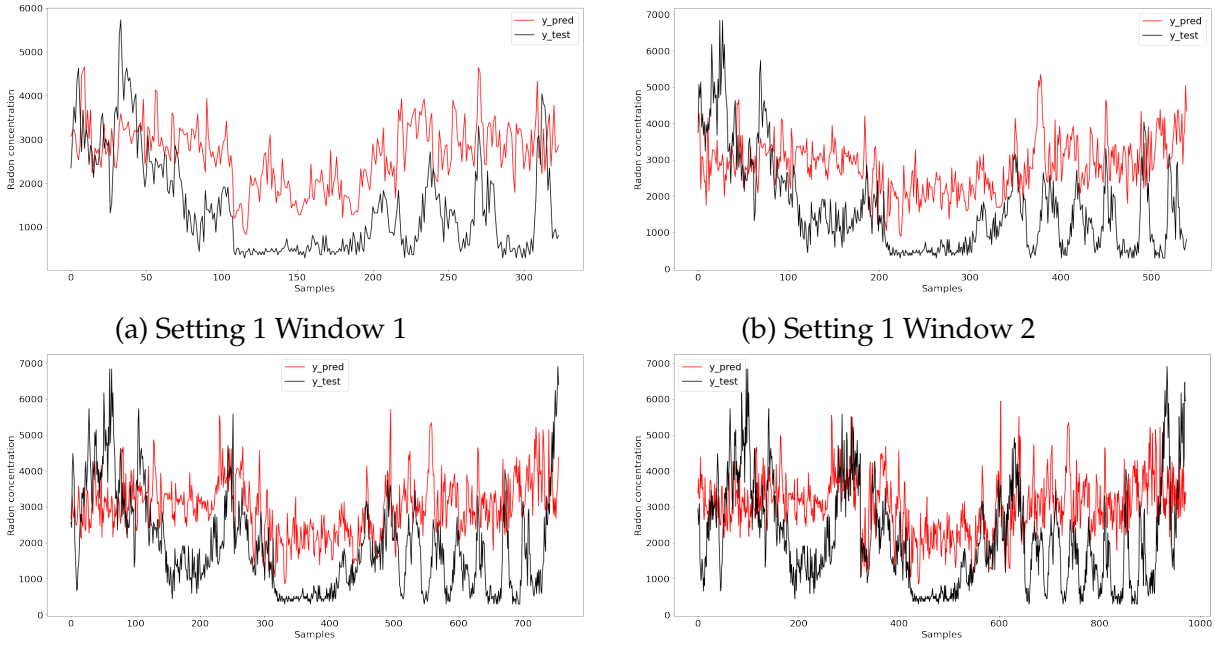
(l) XGB on Setting 4 Window 4

Figure 5.8: Results of KNN, GBM and XGBoost on setting 4 and all windows

of these environmental parameters along with the original environmental parameters to train the models.

The inclusion of gradients aimed to incorporate the sequential characteristics inherent in time series data. It is well-known that sequential features play a crucial role in the analysis of time series data. The performance of the XGBoost model showed improvement when incorporating gradient features in the Setting 1 dataset. Similarly, both the KNN and XGBoost models demonstrated enhanced performance in Setting 2 and Setting 3 when using window sizes of 2, 3, and 4, along with gradient features. Please refer to Tables B.1 to B.16 in appendix section for detailed information and results.

5.2.1 Model trained on environmental parameters



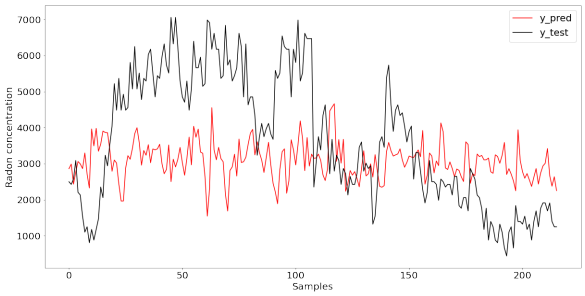
(a) Setting 1 Window 1

(b) Setting 1 Window 2

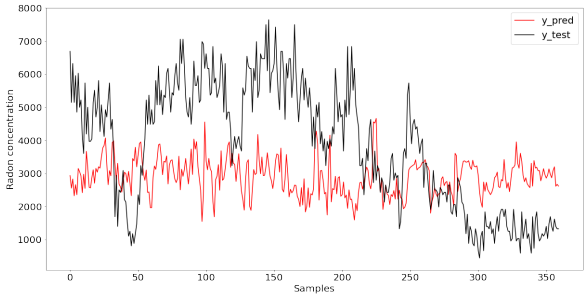
(c) Setting 1 Window 3

(d) Setting 1 Window 4

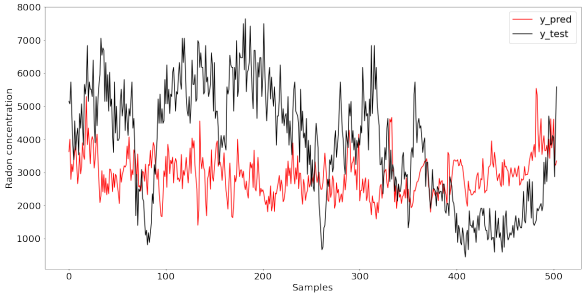
Figure 5.9: Results of KNN on Setting 1 and all Windows



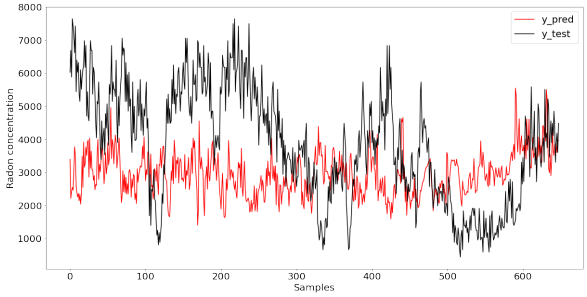
(a) Setting 2 Window 1



(b) Setting 2 Window 2

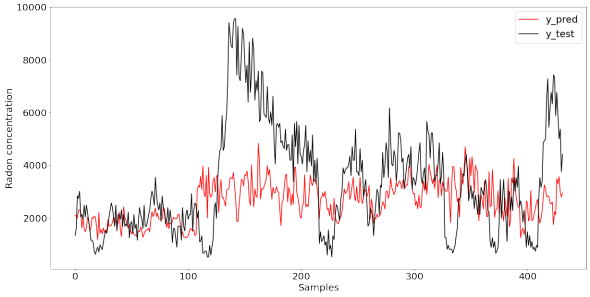


(c) Setting 2 Window 4

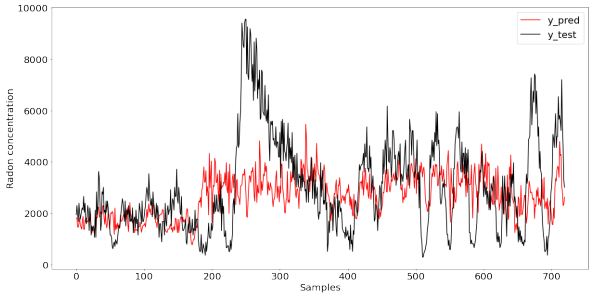


(d) Setting 2 Window 4

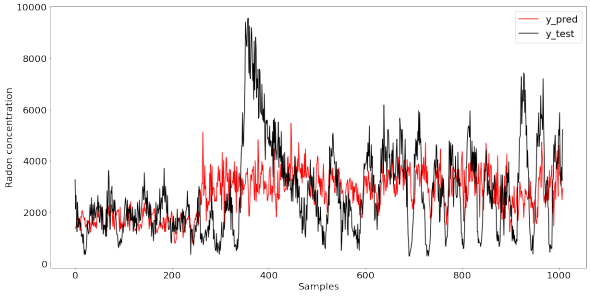
Figure 5.10: Results of KNN on Setting 2 and all Windows



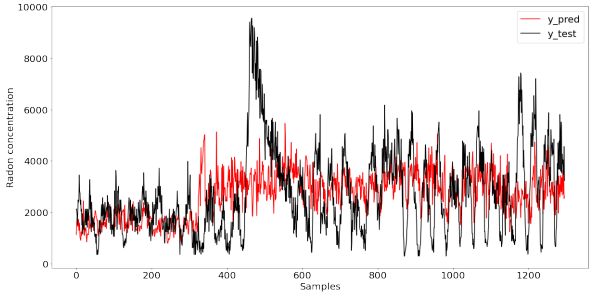
(a) Setting 3 Window 1



(b) Setting 3 Window 2

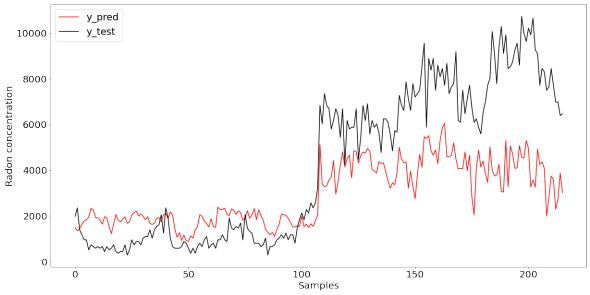


(c) Setting 3 Window 3

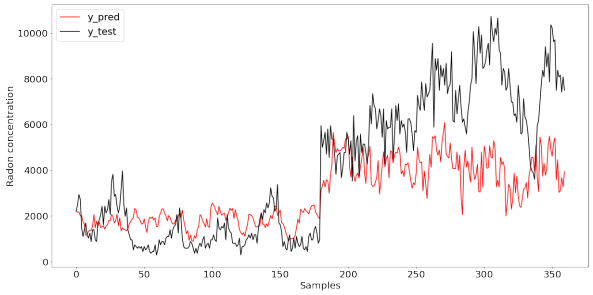


(d) Setting 3 Window 4

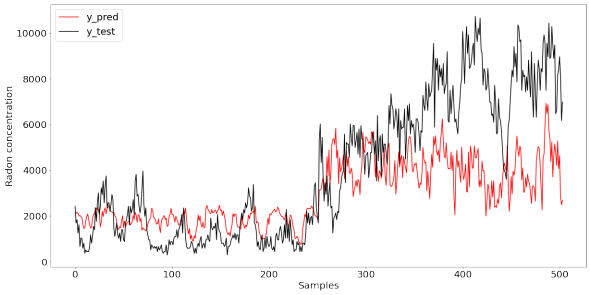
Figure 5.11: Results of KNN on Setting 3 and all Windows



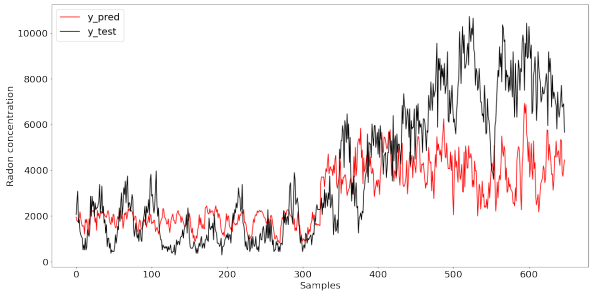
(a) Setting 4 Window 1



(b) Setting 4 Window 2

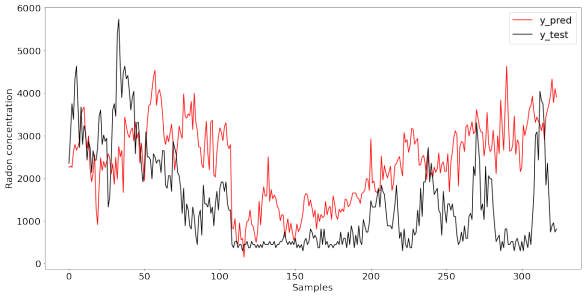


(c) Setting 4 Window 3

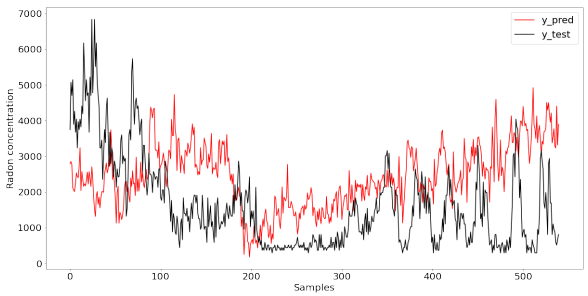


(d) Setting 4 Window 4

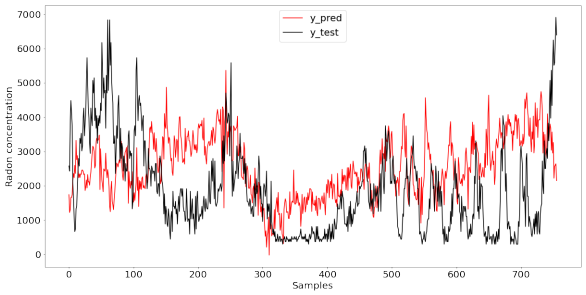
Figure 5.12: Results of KNN on Setting 4 and all Windows



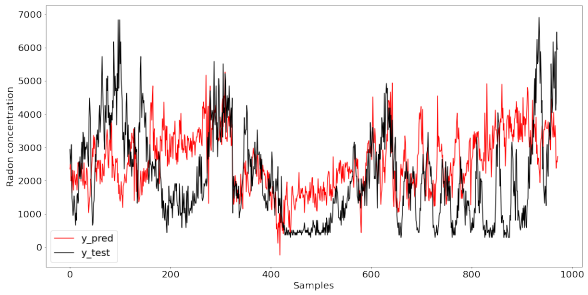
(a) Setting 1 Window 1



(b) Setting 1 Window 2

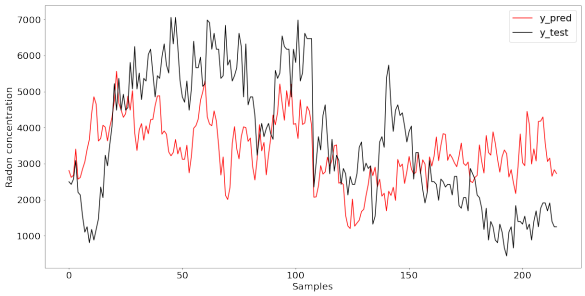


(c) Setting 1 Window 3

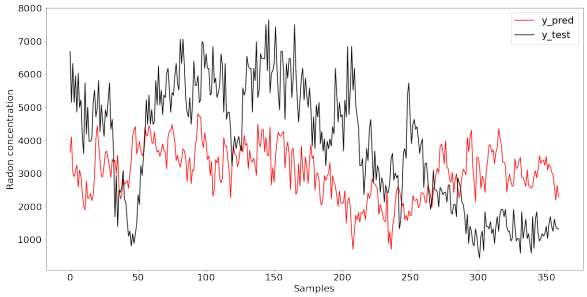


(d) Setting 1 Window 4

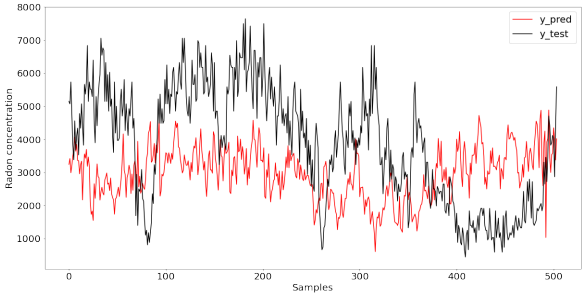
Figure 5.13: Results of XGB on Setting 1 and all Windows



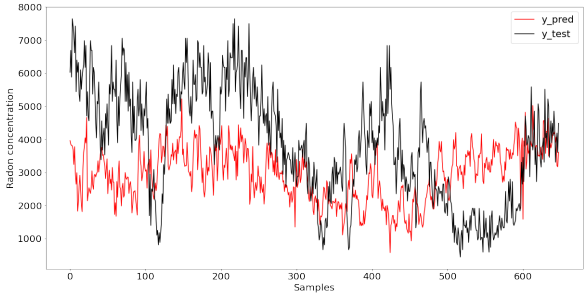
(a) Setting 2 Window 1



(b) Setting 2 Window 2

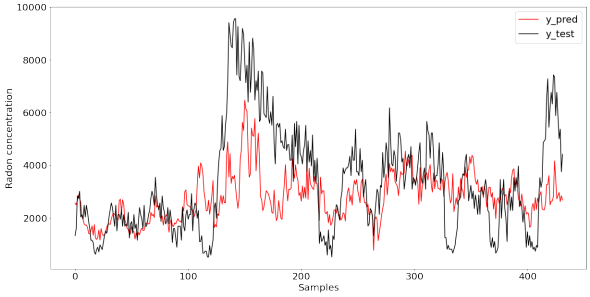


(c) Setting 2 Window 4

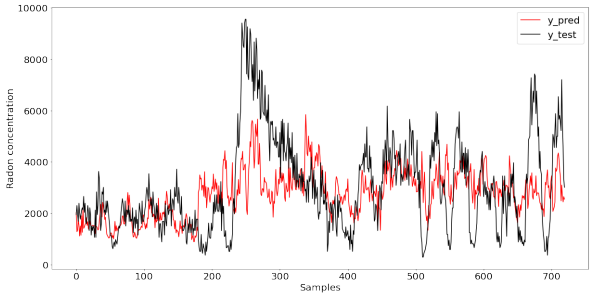


(d) Setting 2 Window 4

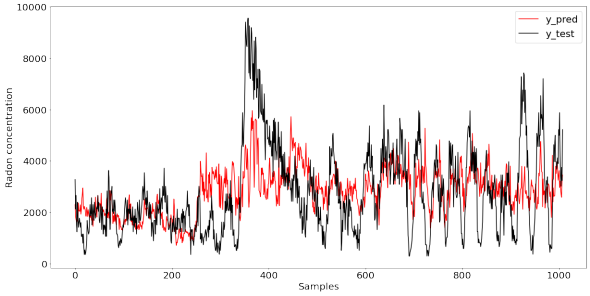
Figure 5.14: Results of XGB on Setting 2 and all Windows



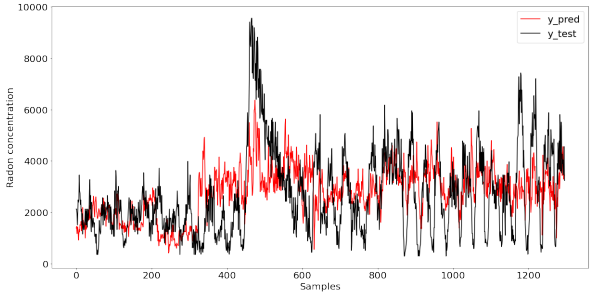
(a) Setting 3 Window 1



(b) Setting 3 Window 2

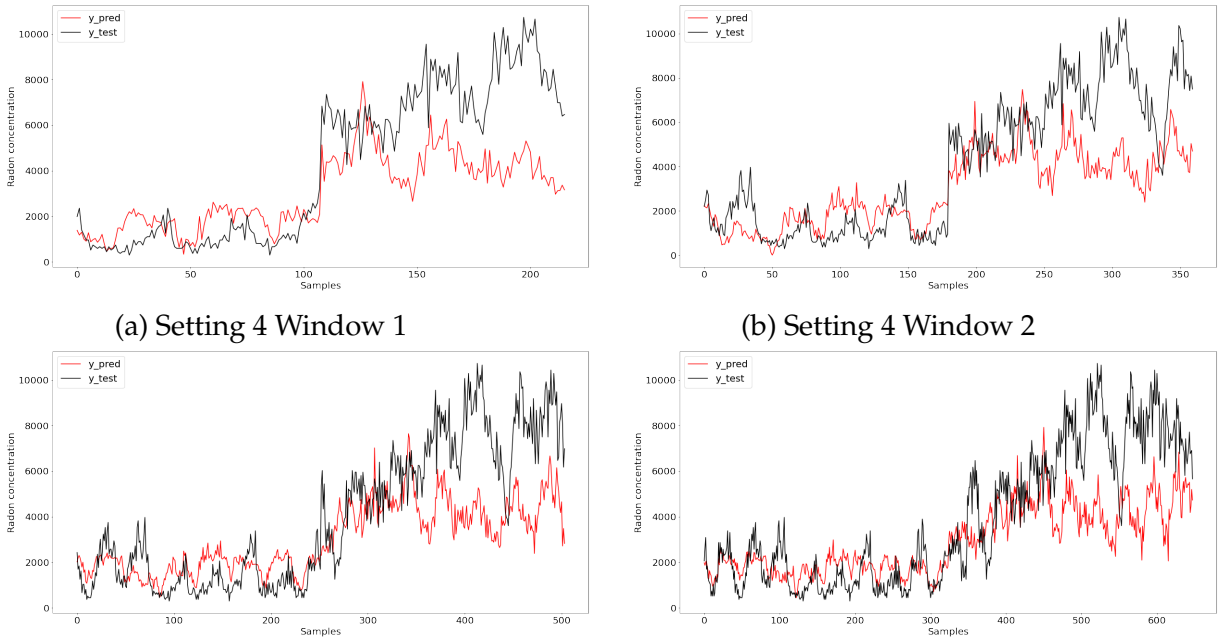


(c) Setting 3 Window 3



(d) Setting 3 Window 4

Figure 5.15: Results of XGB on Setting 3 and all Windows



(a) Setting 4 Window 1

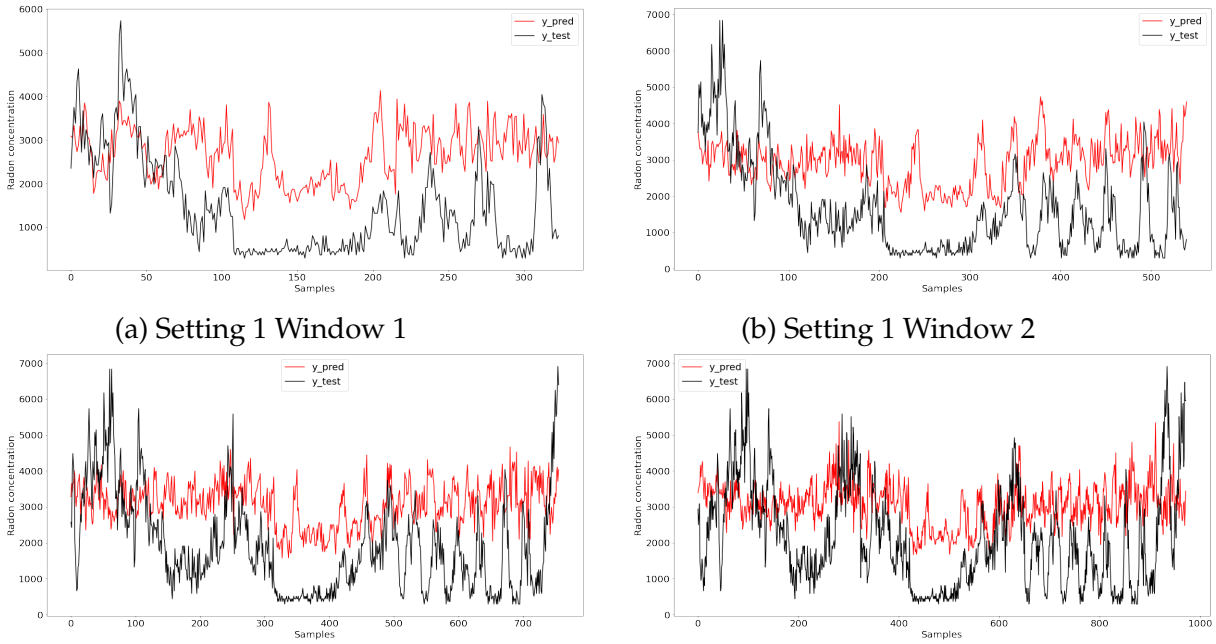
(b) Setting 4 Window 2

(c) Setting 4 Window 3

(d) Setting 4 Window 4

Figure 5.16: Results of XGB on Setting 4 and all Windows

5.2.2 Model trained on environmental parameters and gradient of these parameters



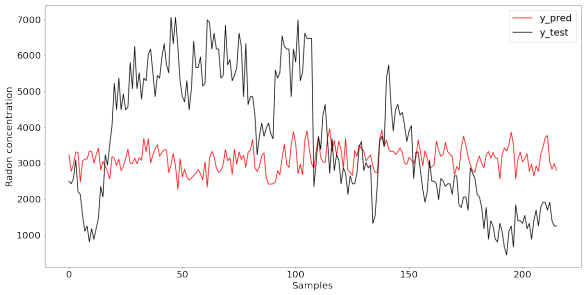
(a) Setting 1 Window 1

(b) Setting 1 Window 2

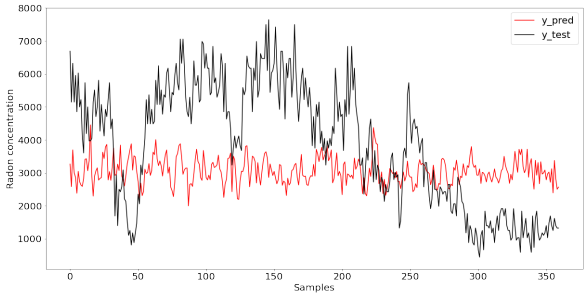
(c) Setting 1 Window 3

(d) Setting 1 Window 4

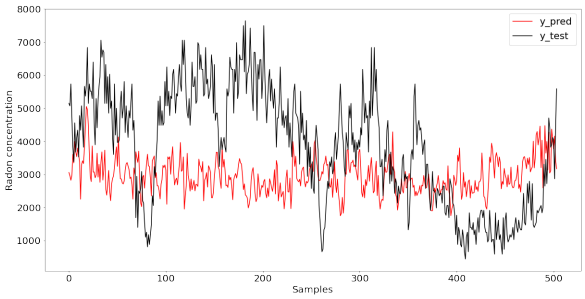
Figure 5.17: Results of KNN on Setting 1 and all Windows (Gradient features included)



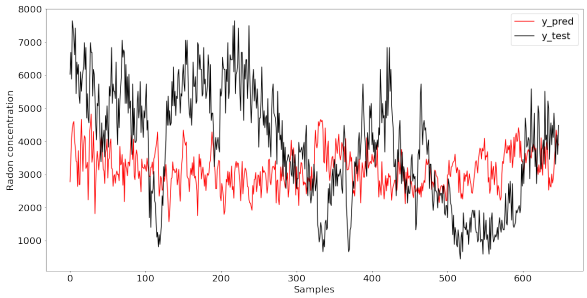
(a) Setting 2 Window 1



(b) Setting 2 Window 2

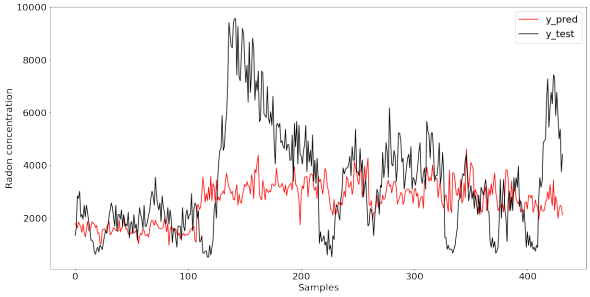


(c) Setting 2 Window 4

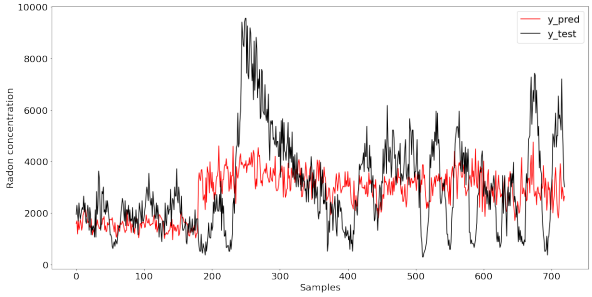


(d) Setting 2 Window 4

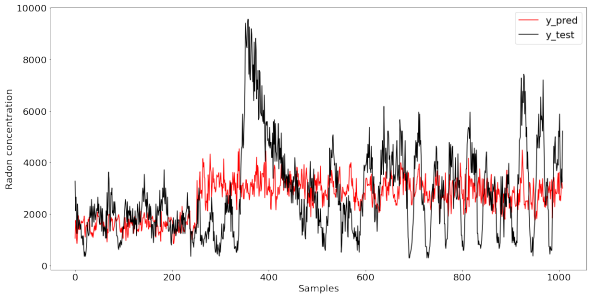
Figure 5.18: Results of KNN on Setting 2 and all Windows (Gradient features included)



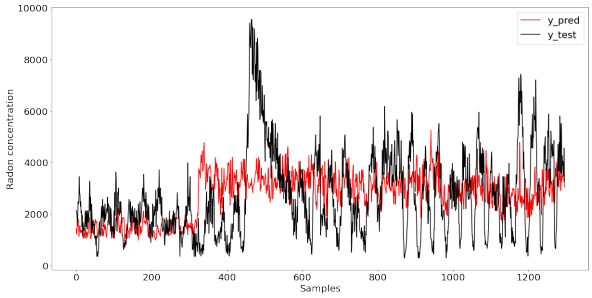
(a) Setting 3 Window 1



(b) Setting 3 Window 2

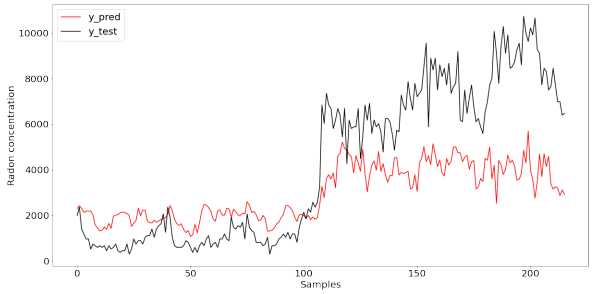


(c) Setting 3 Window 3

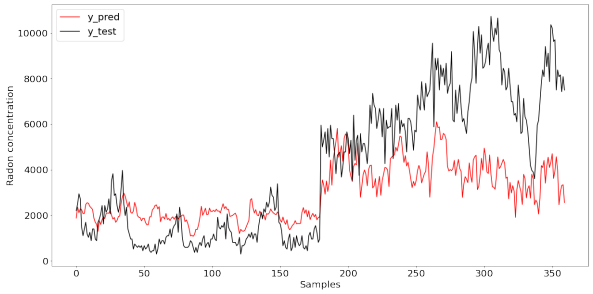


(d) Setting 3 Window 4

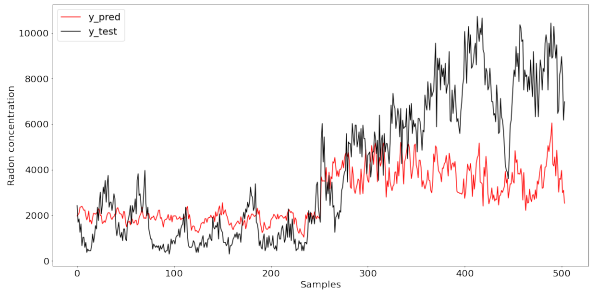
Figure 5.19: Results of KNN on Setting 3 and all Windows (Gradient features included)



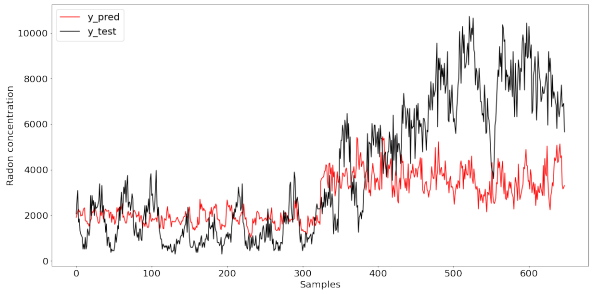
(a) Setting 4 Window 1



(b) Setting 4 Window 2

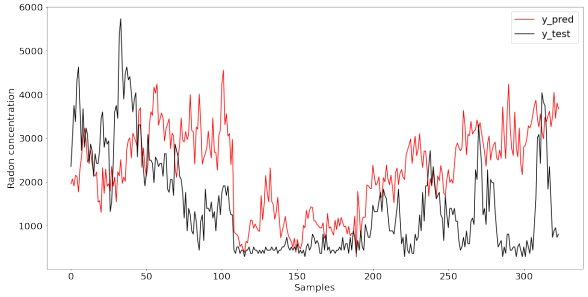


(c) Setting 4 Window 3

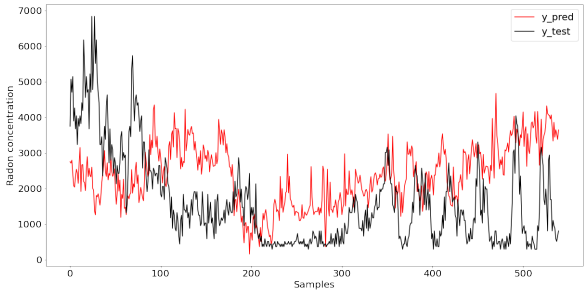


(d) Setting 4 Window 4

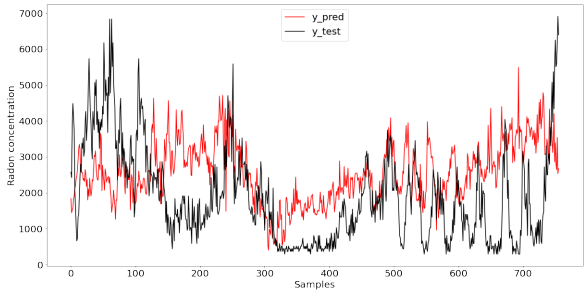
Figure 5.20: Results of KNN on Setting 4 and all Windows (Gradient features included)



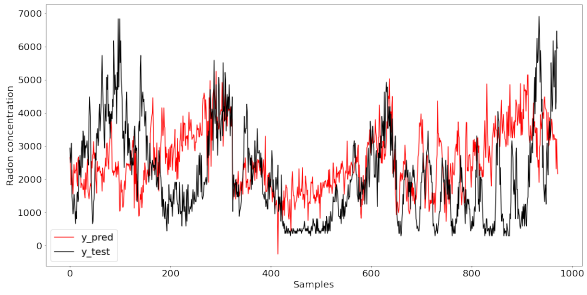
(a) Setting 1 Window 1



(b) Setting 1 Window 2

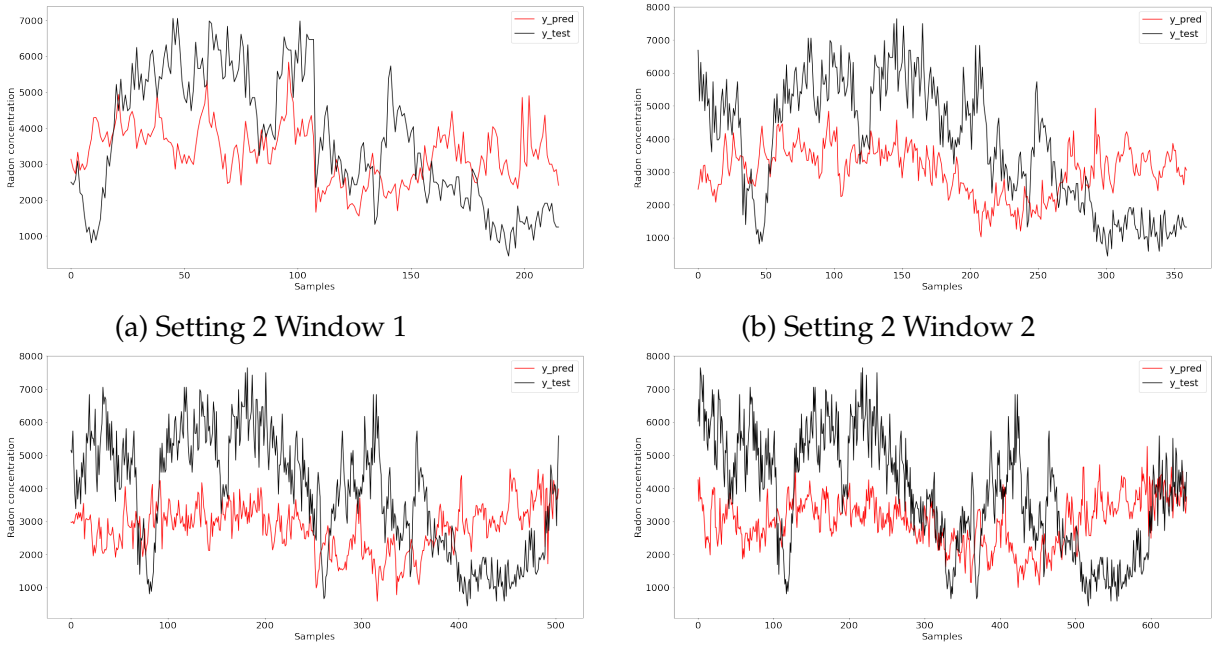


(c) Setting 1 Window 3



(d) Setting 1 Window 4

Figure 5.21: Results of XGBoost on Setting 1 and all Windows (Gradient features included)



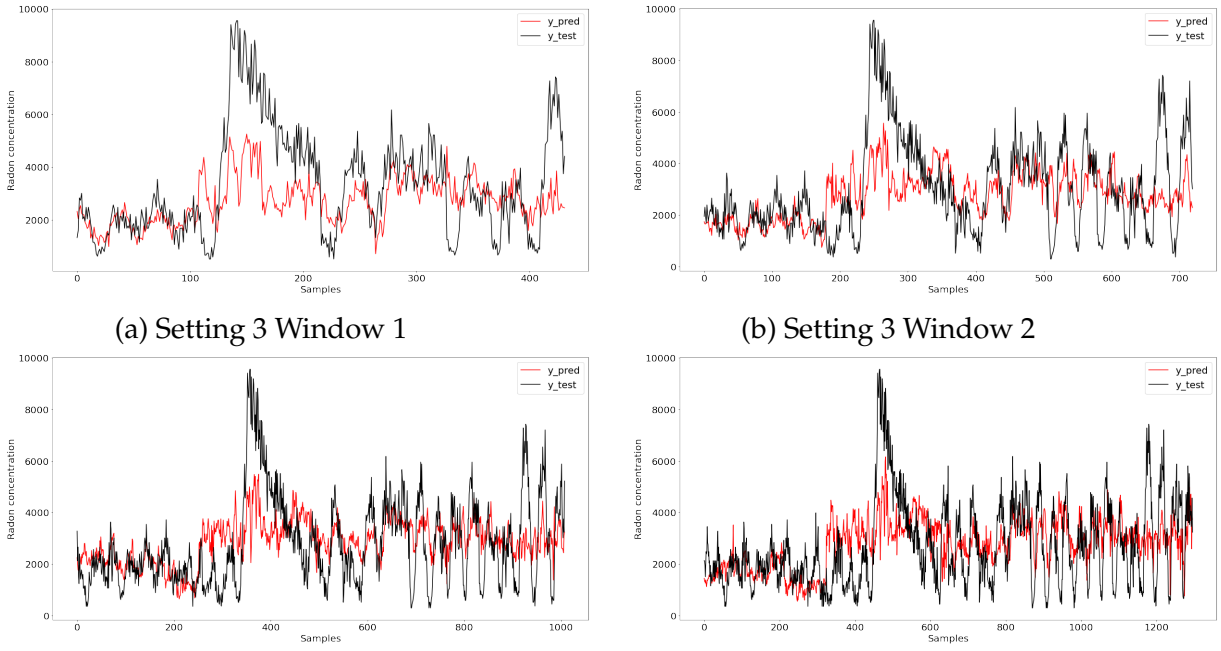
(a) Setting 2 Window 1

(b) Setting 2 Window 2

(c) Setting 2 Window 3

(d) Setting 2 Window 4

Figure 5.22: Results of XGBoost on Setting 2 and all Windows (Gradient features included)



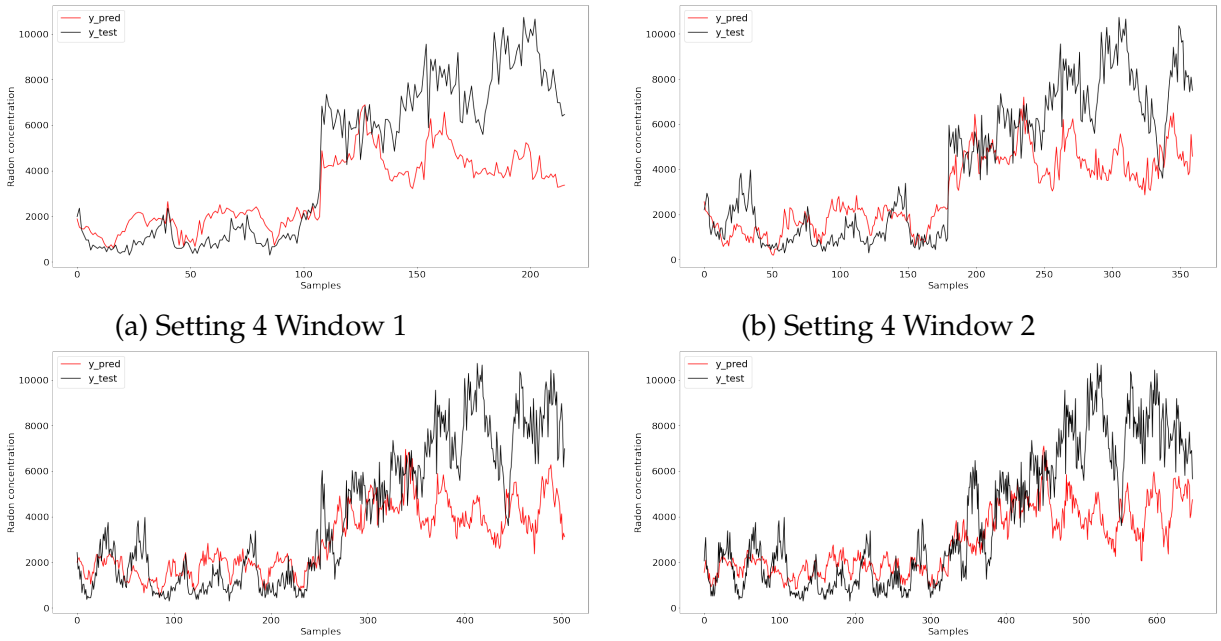
(a) Setting 3 Window 1

(b) Setting 3 Window 2

(c) Setting 3 Window 3

(d) Setting 3 Window 4

Figure 5.23: Results of XGB on Setting 3 and all Windows (Gradient features included)



(a) Setting 4 Window 1

(b) Setting 4 Window 2

(c) Setting 4 Window 3

(d) Setting 4 Window 4

Figure 5.24: Results of XGB on Setting 4 and all Windows (Gradient features included)

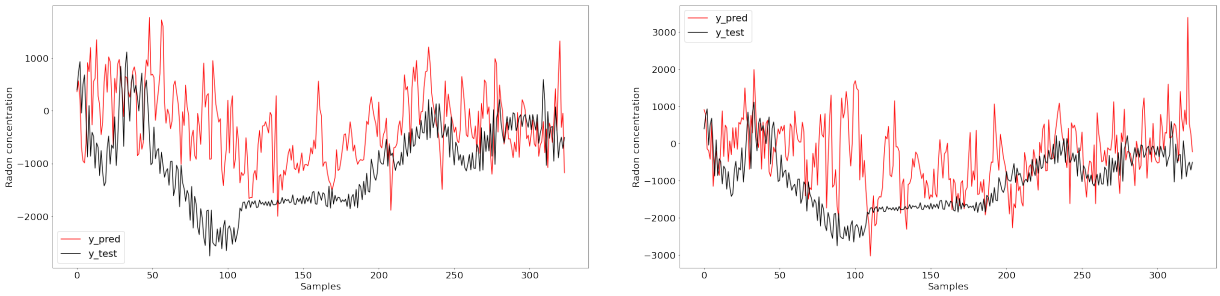
5.3 Experiment 3: Reconstructed radon data and original environmental parameters trained on K nearest neighbor and Extreme Gradient Boosting algorithms

In Experiment 3, we applied the Empirical Mode Decomposition (EMD) method as discussed in Section 4.1.1 to reconstruct our radon data. The purpose of reconstruction was to eliminate the higher frequency components present in the radon time series data. It is generally necessary to remove high frequency components from time series data to facilitate better analysis. With this objective in mind, we opted to use the EMD method, which decomposes the time series into various Intrinsic Mode Functions (IMFs) ranging from higher to lower frequencies.

To reconstruct the radon time series, we removed the IMFs containing diurnal and/or semidiurnal periodicity and then summed up the remaining IMFs. However, we encountered an issue with this approach: the original value range of the radon data was lost. Maintaining the original range of values is important for accurate predictions and analysis. Figure 5.25 clearly illustrates a notable change in

the prediction range of radon concentration after applying the EMD-based reconstruction. Initially, the minimum value was 0; however, post-reconstruction, the minimum value ranged from approximately -2000 to -3000. Such values cannot be directly compared with the original radon measurements during real-time earthquake prediction system. Consequently, we made the decision to exclude EMD as a preprocessing step for subsequent experiments.

Nevertheless, in future research, efforts could be directed towards addressing this issue and devising a solution to maintain the original value range. Once resolved, incorporating EMD as a preprocessing strategy may prove beneficial for enhancing the efficiency of radon prediction.



(a) Setting 1 Window 4 (KNN)

(b) Setting 1 Window 4 (XGBoost)

Figure 5.25: Results of KNN and XGBoost on reconstructed radon time series (setting 1 window 4)

5.4 Experiment 4: Original segmented, resampled dataset trained on SVR linear, radial kernel, Random Forest, K nearest neighbor and Extreme gradient boosting algorithms

The results obtained from the previous experiments showed gradual improvement as we made changes to various experimental parameters. However, these results did not meet the acceptance criteria set by domain expert. In contrast, Experiment 4 yielded satisfactory outcomes, which were considered acceptable by the domain expert. And, the performance metrics of Experiment 4 outperformed those of previous experiments.

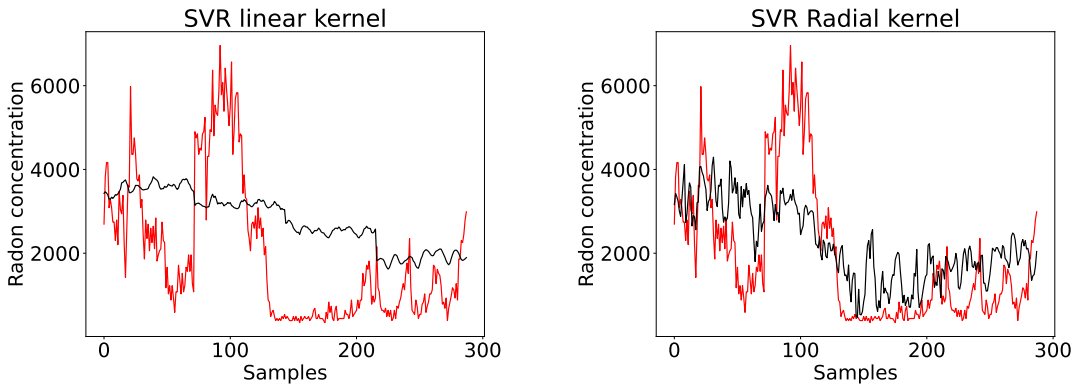
For Experiment 4, we made the decision to segment the data and solely utilize the samples recorded from January 22, 2020, to April 16, 2021. Initially, the dataset encompassed samples from January 22, 2020, to December 28, 2021. However, based on the advice of domain experts, we excluded the latter portion of the dataset due to its random nature, characterized by high random fluctuations and sudden changes. By removing this segment, we obtained a segmented dataset for further analysis and investigation.

As part of our data preprocessing in Experiment 4, we opted to resample the original data into 60-minute intervals. This involved converting the original 10-minute sample interval data into 60-minute samples by averaging six consecutive rows. The purpose of this resampling was to remove the impact of high random fluctuations and achieve a smoother dataset for improved analysis. Subsequently, we evaluated both the 40-minute and 60-minute resampled data using various machine learning models. Remarkably, the 60-minute resampled data outperformed the 40-minute resampled data in terms of model performance.

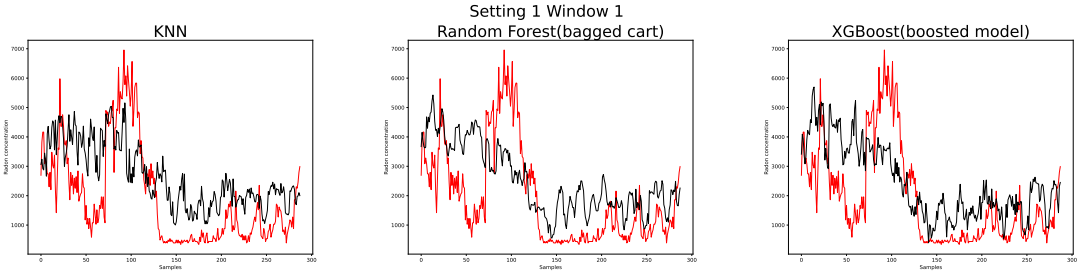
A significant change incorporated in our final experiment was the inclusion of all earthquakes in the test data. Previously, only seismic events with a magnitude greater than or equal to 3.8 were considered. However, after including smaller earthquakes (with magnitudes ranging from 3 to 3.5), the results for the testing data improved noticeably. This demonstrates that machine learning models are more capable of predicting radon concentrations during small earthquakes rather than large earthquakes.

Notably, the results for Setting 3 and Setting 4 exhibited better performance compared to Setting 1 and Setting 2. This is primarily because Settings 3 and 4 contains radon samples of low magnitude earthquakes. Figure 3.2 illustrates that Settings 3 and 4 primarily consisted of seismic events with magnitudes ranging from 3.1 to 3.7 Mw.

Figures 5.26 to 5.41 demonstrate the excellent performance of Support Vector Regression (SVR) with radial kernel, K Nearest Neighbors (KNN), and XGBoost models across various settings and window sizes. These models consistently yielded accurate predictions for radon concentrations. Particularly, the prediction results for Setting 3 and Setting 4 datasets were highly efficient. Figures 5.34 to 5.41 shows the overlapping between the measured radon concentration and the corresponding predicted radon concentration. The red line represents the measured radon values, while the black line represents the predicted radon values. This visual representation effectively illustrates the accuracy of the predictions, as the measured and predicted radon concentrations align closely.

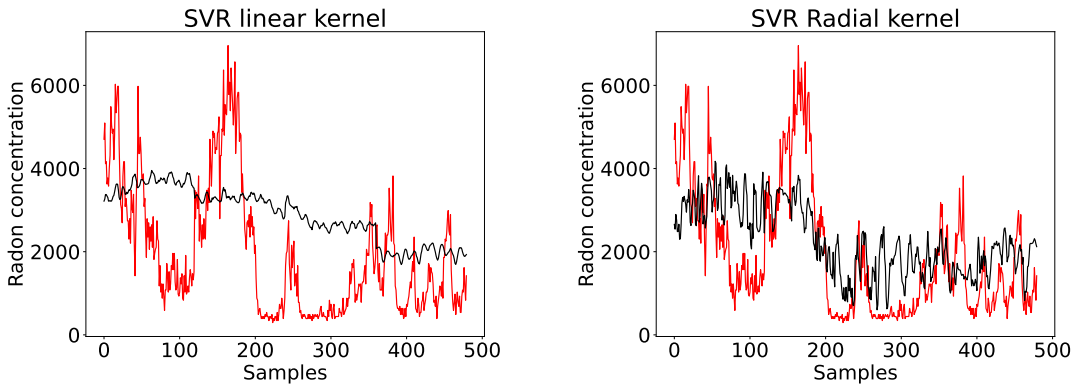


(a) SVR linear and radial kernel on Setting 1 and Window 1

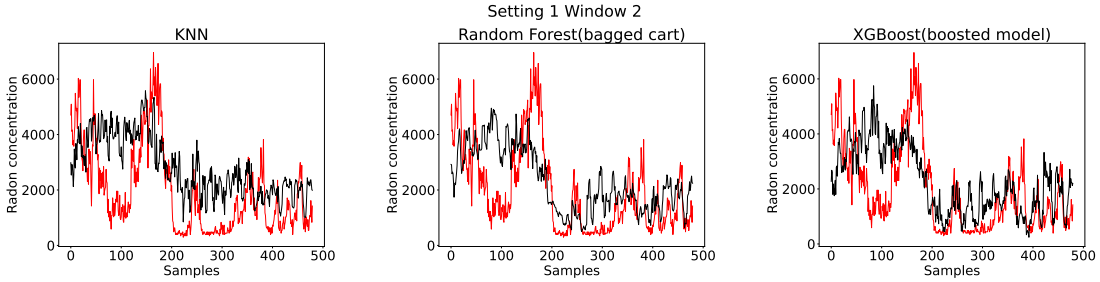


(b) KNN, Random Forest and XGBoost on Setting 1 and Window 1

Figure 5.26: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 1 window 1)

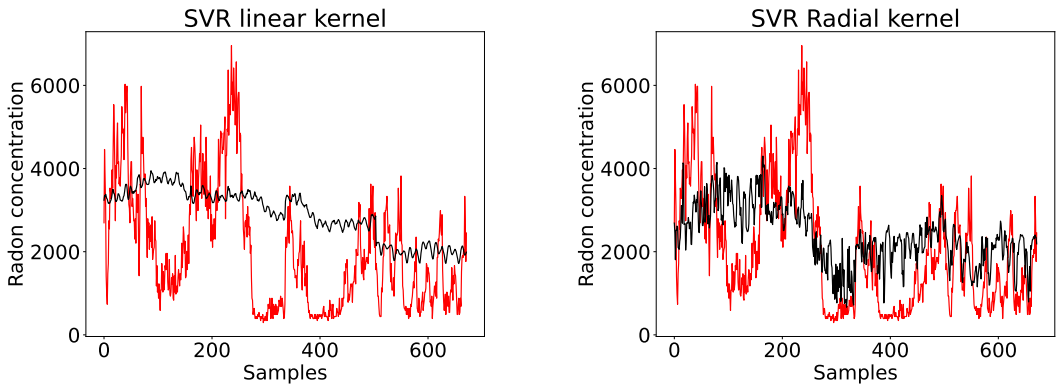


(a) SVR linear and radial kernel on Setting 1 and Window 2

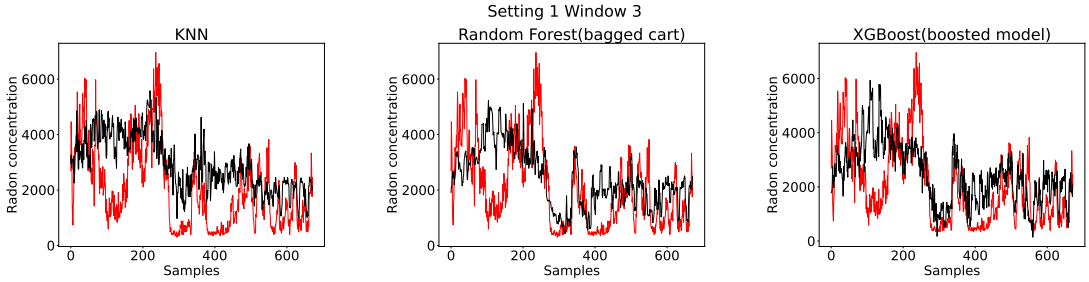


(b) KNN, Random Forest and XGBoost on Setting 1 and Window 2

Figure 5.27: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 1 window 2)

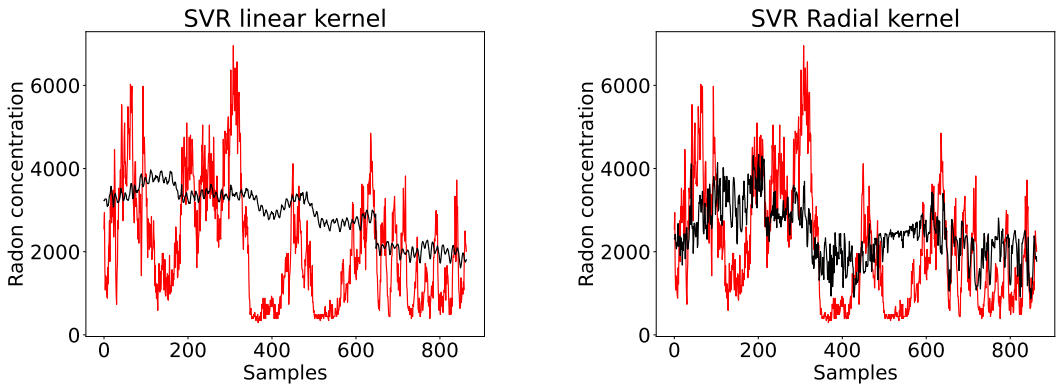


(a) SVR linear and radial kernel on Setting 1 and Window 3

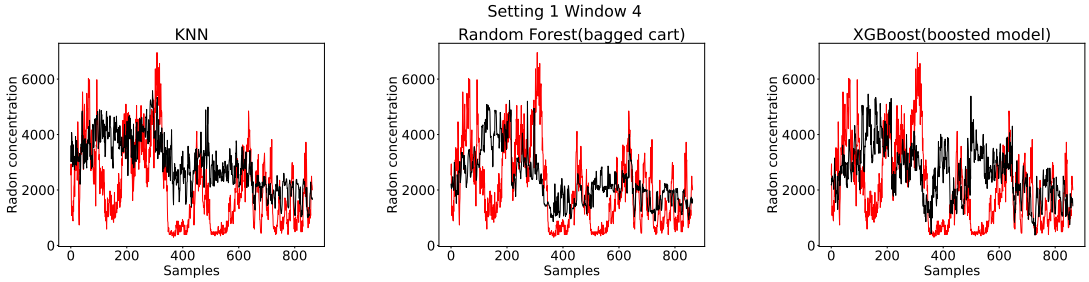


(b) KNN, Random Forest and XGBoost on Setting 1 and Window 3

Figure 5.28: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 1 window 3)

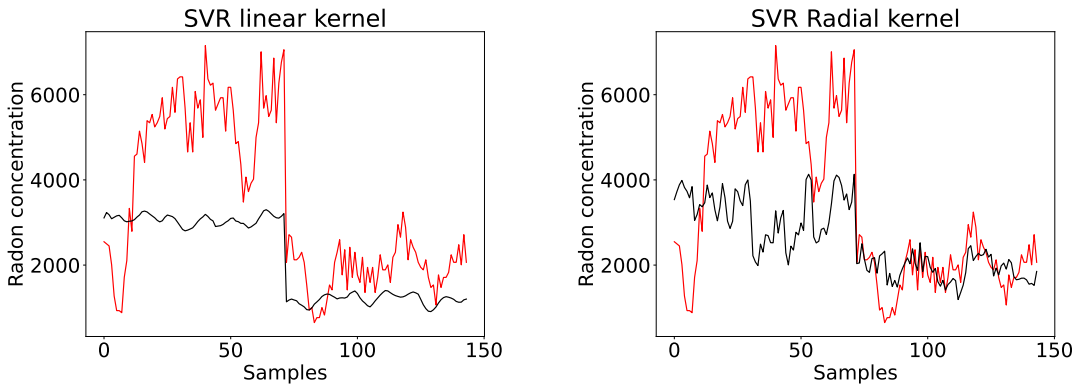


(a) SVR linear and radial kernel on Setting 1 and Window 4

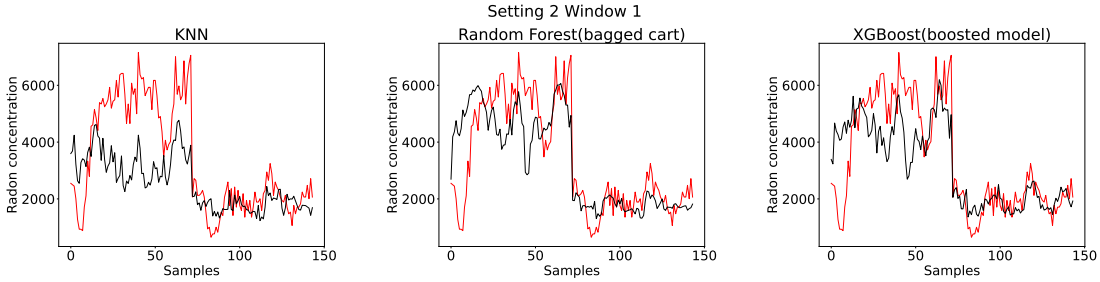


(b) KNN, Random Forest and XGBoost on Setting 1 and Window 4

Figure 5.29: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 1 window 4

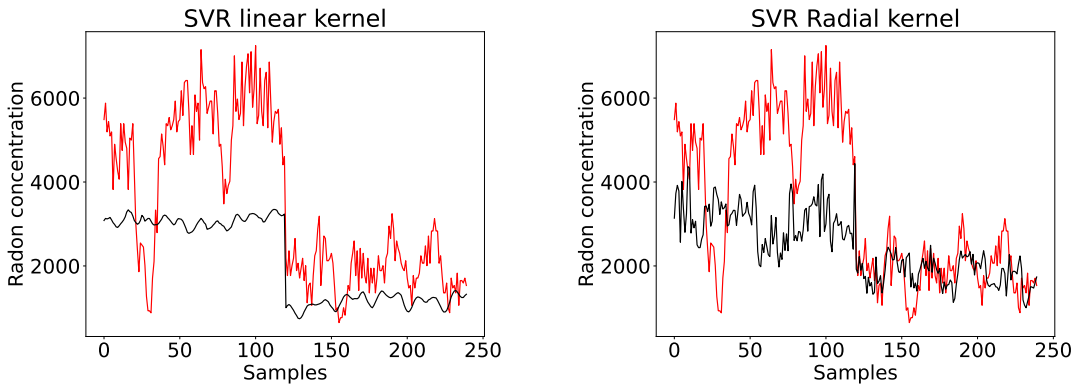


(a) SVR linear and radial kernel on Setting 2 and Window 1

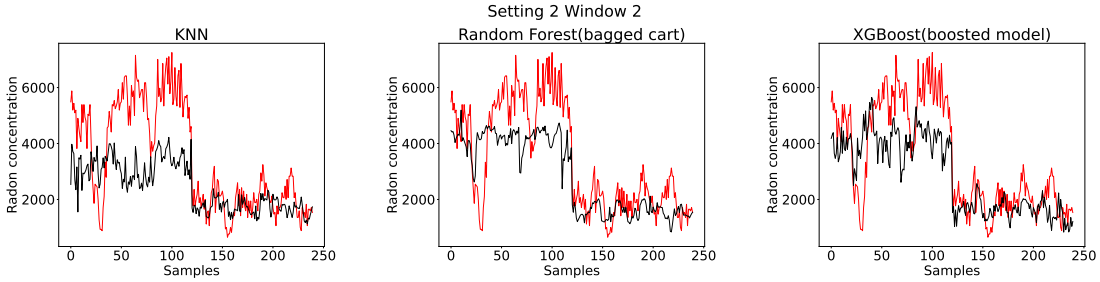


(b) KNN, Random Forest and XGBoost on Setting 2 and Window 1

Figure 5.30: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 2 window 1)

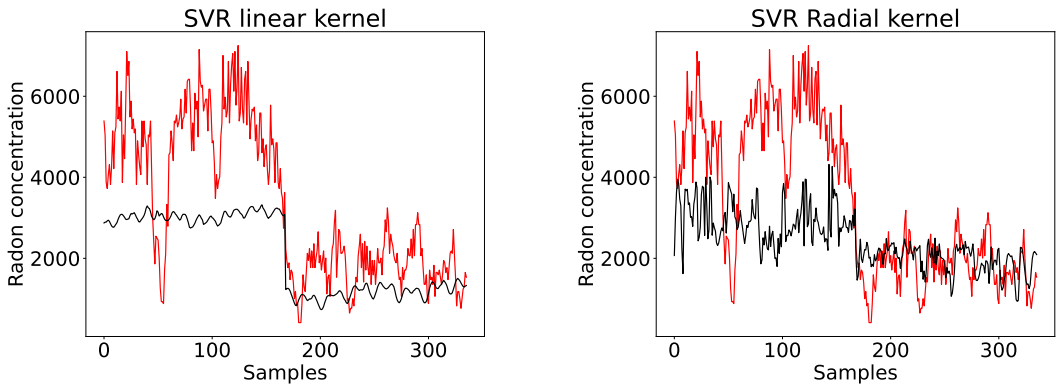


(a) SVR linear and radial kernel on Setting 2 and Window 2

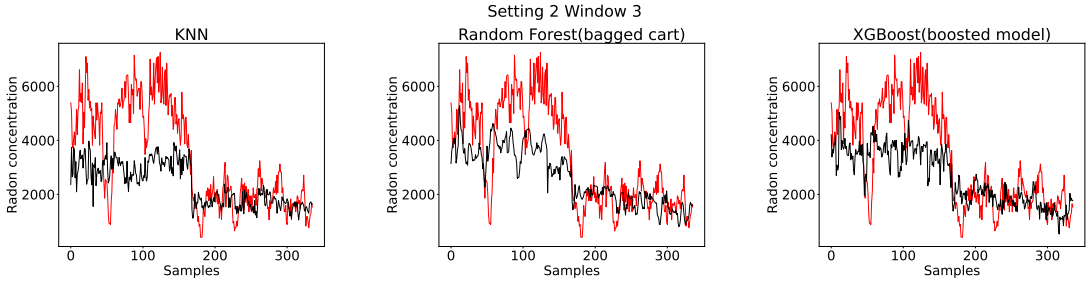


(b) KNN, Random Forest and XGBoost on Setting 2 and Window 2

Figure 5.31: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 2 window 2)

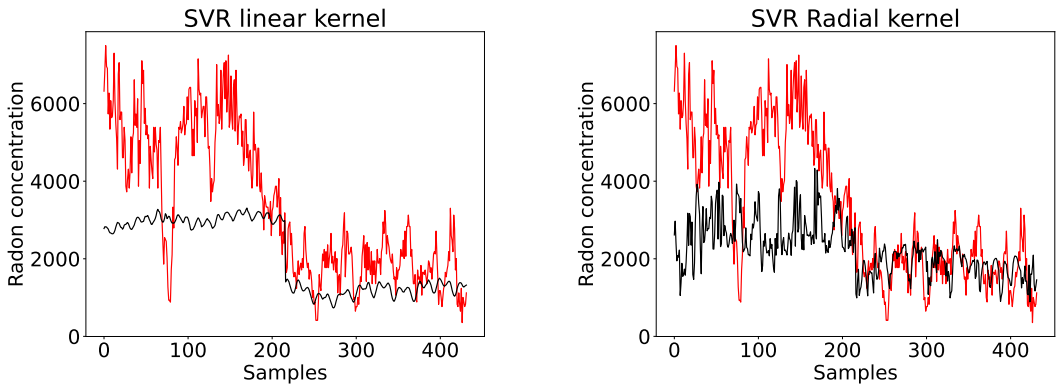


(a) SVR linear and radial kernel on Setting 2 and Window 3

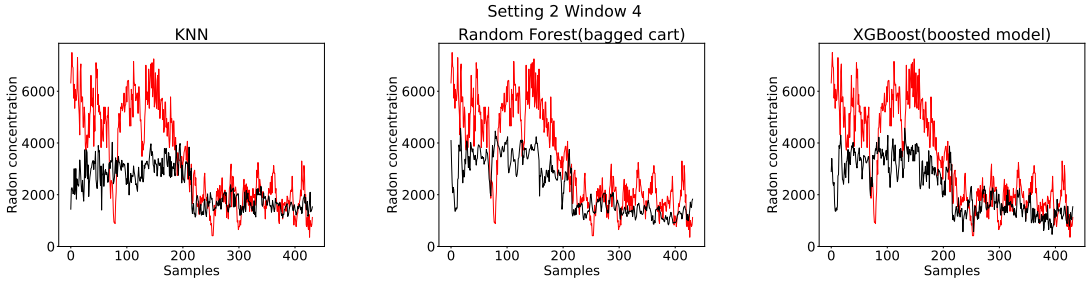


(b) KNN, Random Forest and XGBoost on Setting 2 and Window 3

Figure 5.32: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 2 window 3)

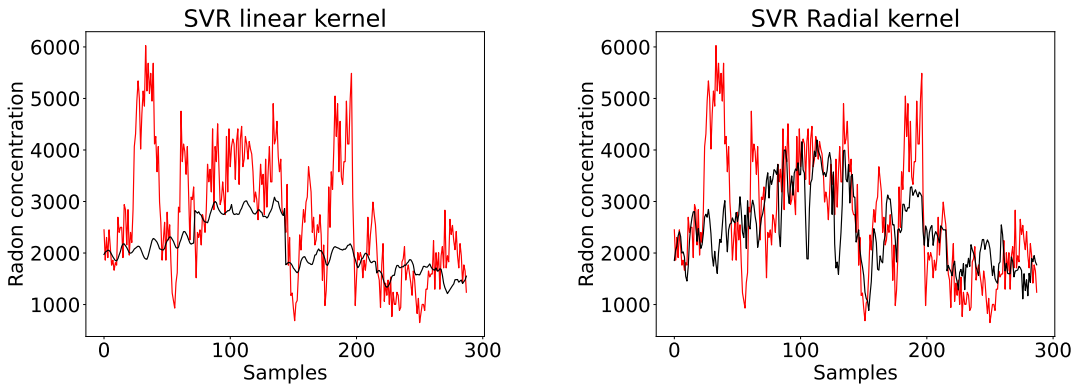


(a) SVR linear and radial kernel on Setting 2 and Window 4

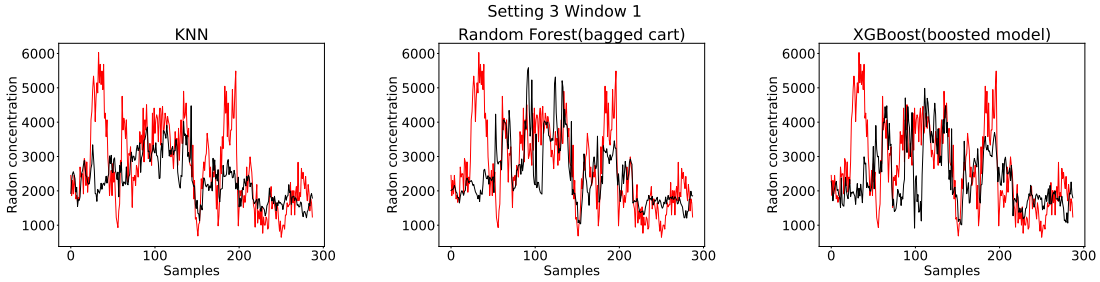


(b) KNN, Random Forest and XGBoost on Setting 2 and Window 4

Figure 5.33: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 2 window 4

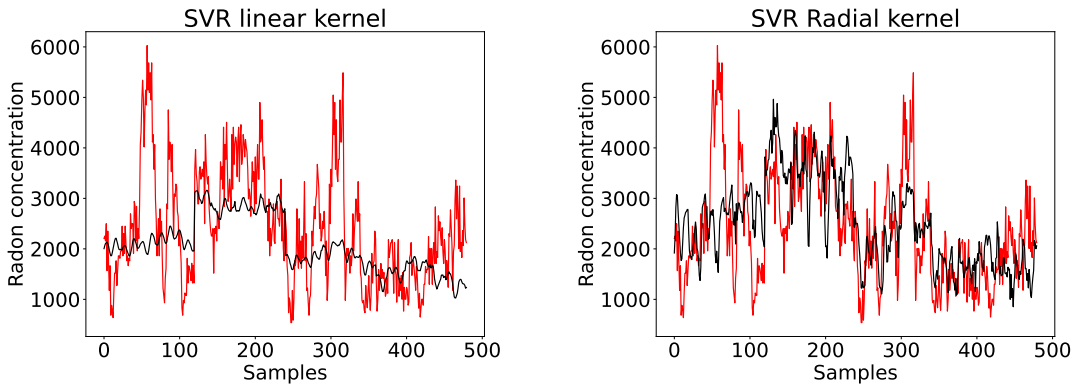


(a) SVR linear and radial kernel on Setting 3 and Window 1

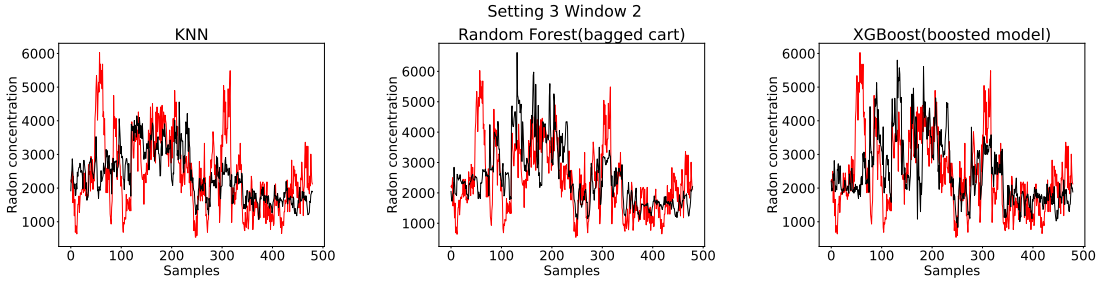


(b) KNN, Random Forest and XGBoost on Setting 3 and Window 1

Figure 5.34: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 3 window 1

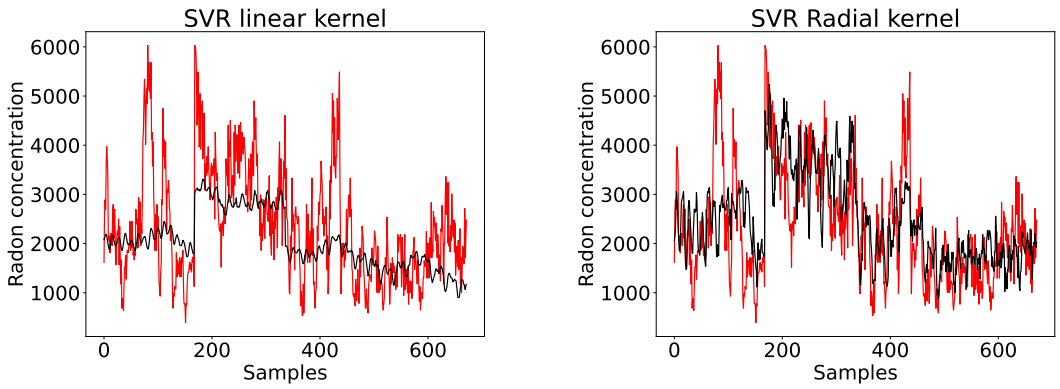


(a) SVR linear and radial kernel on Setting 3 and Window 2

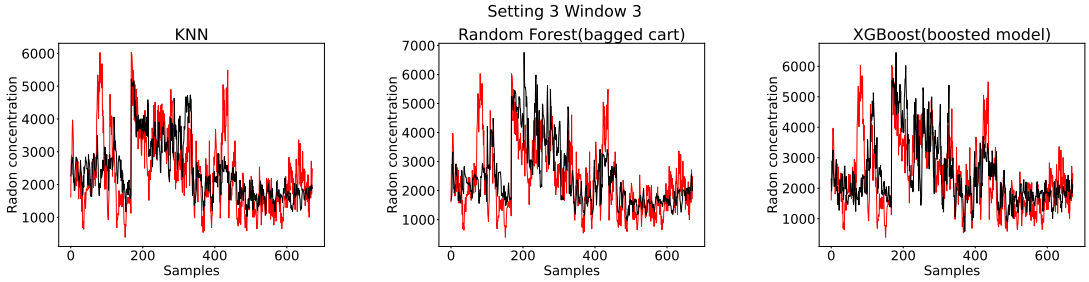


(b) KNN, Random Forest and XGBoost on Setting 3 and Window 2

Figure 5.35: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 3 window 2

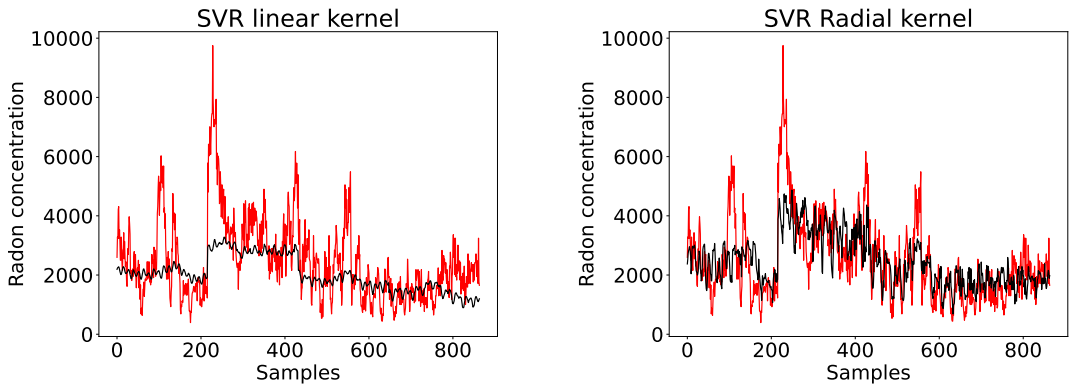


(a) SVR linear and radial kernel on Setting 3 and Window 3

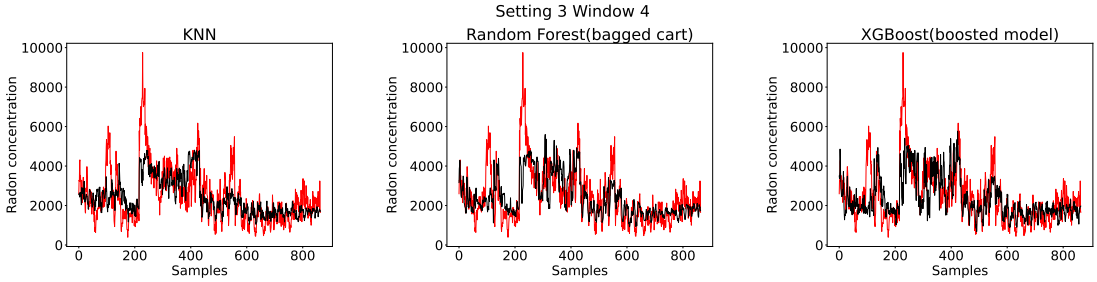


(b) KNN, Random Forest and XGBoost on Setting 3 and Window 3

Figure 5.36: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 3 window 3

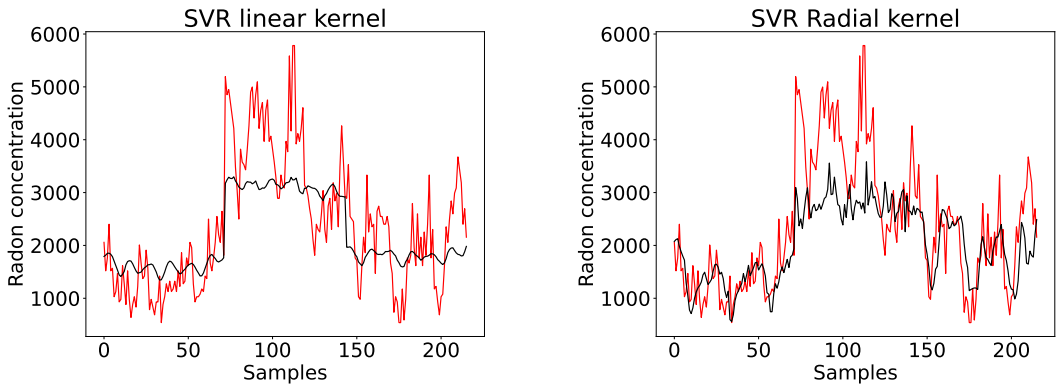


(a) SVR linear and radial kernel on Setting 3 and Window 4

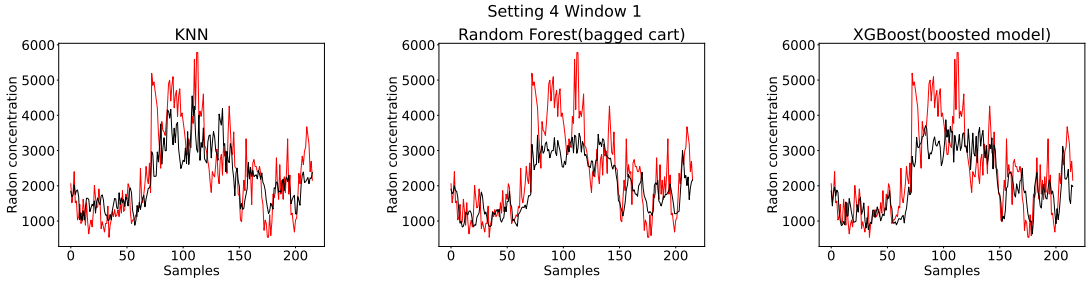


(b) KNN, Random Forest and XGBoost on Setting 3 and Window 4

Figure 5.37: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 3 window 4

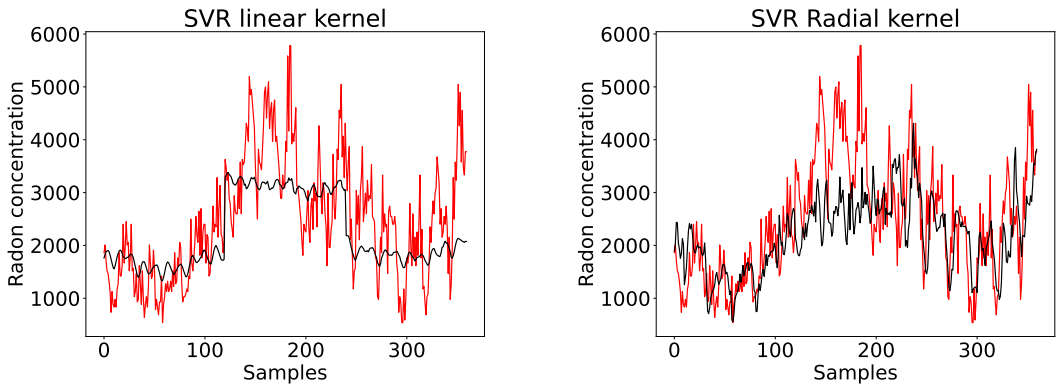


(a) SVR linear and radial kernel on Setting 4 and Window 1

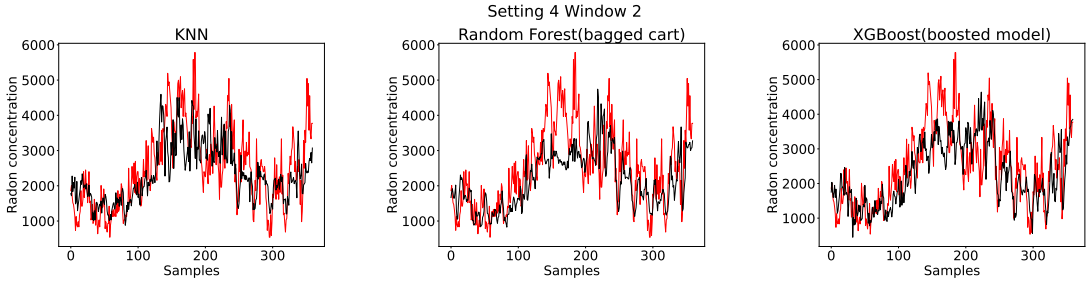


(b) KNN, Random Forest and XGBoost on Setting 4 and Window 1

Figure 5.38: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 4 window 1

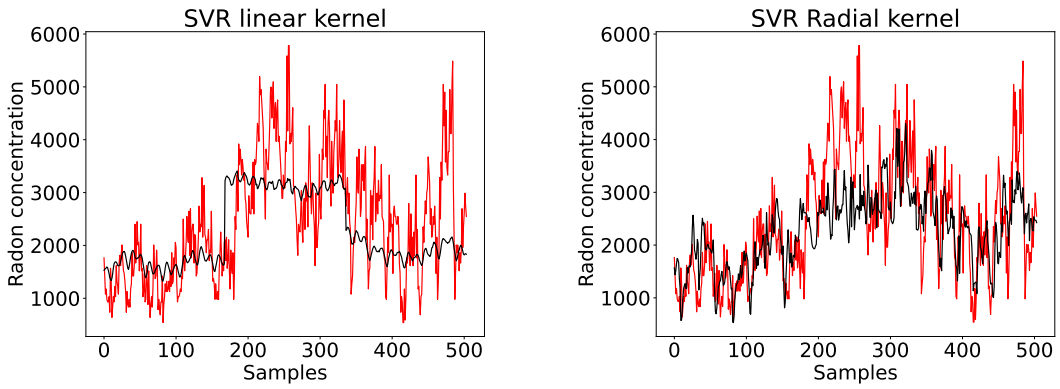


(a) SVR linear and radial kernel on Setting 4 and Window 2

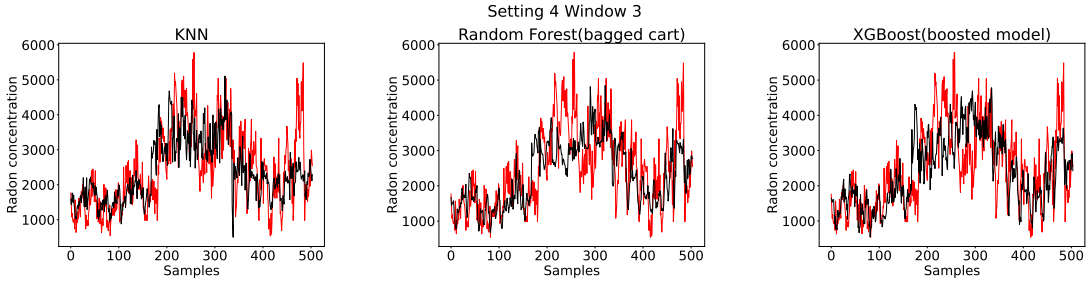


(b) KNN, Random Forest and XGBoost on Setting 4 and Window 2

Figure 5.39: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 4 window 2

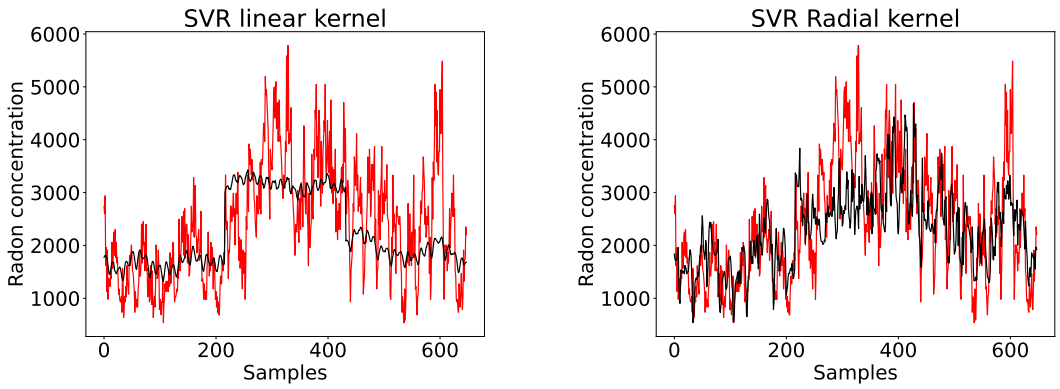


(a) SVR linear and radial kernel on Setting 4 and Window 3

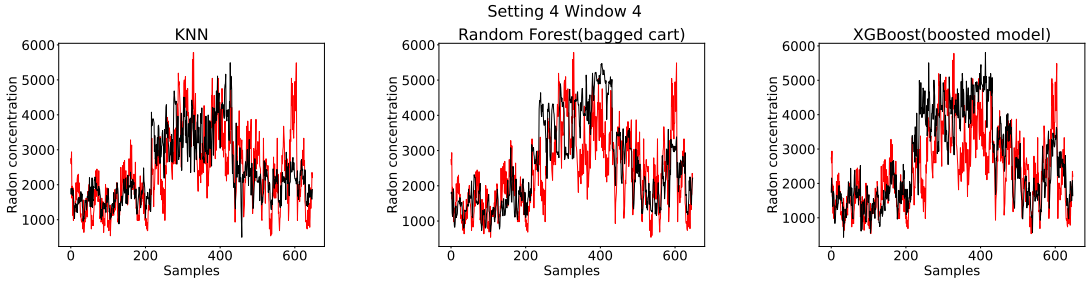


(b) KNN, Random Forest and XGBoost on Setting 4 and Window 3

Figure 5.40: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 4 window 3



(a) SVR linear and radial kernel on Setting 4 and Window 4



(b) KNN, Random Forest and XGBoost on Setting 4 and Window 4

Figure 5.41: Results of SVR linear, rbf kernel, Random Forest, KNN and XGBoost on setting 4 window 4

5.5 Anomaly Detection in Radon Time Series Data: A Comparison of Predicted and Original Data using Simple Mean and Standard Deviation Method

To establish confidence bounds for our predictions, we employed the simple mean and standard deviation method. Initially, the mean and standard deviation of the predicted radon values were calculated. Subsequently, we determined the upper bound of the confidence interval as $\mu + 2\sigma$ and the lower bound as $\mu - 2\sigma$.

Any measured radon value that fell outside this confidence bound was identified as an anomaly. This approach allowed us to detect and flag anomalous radon measurements based on their deviation from the expected range determined by the mean and standard deviation of the predicted values. The presence of anomalies in radon measurements occurring several days prior to seismic events can serve as an early warning indication. Such anomalous readings can alert us to an impending seismic event. This finding highlights the potential utility of monitoring radon measurements as an effective alarm system for identifying seismic activity in advance.

The analysis of Figures 5.42 to 5.45 reveals that our model successfully identified anomalies preceding 8 out of 13 seismic events. Figure 5.42 illustrates that for all seismic events, the radon measurements exhibited anomalous values prior to the earthquakes. In the case of the first two seismic events, the radon measurements were notably higher, surpassing the upper bound of the confidence interval. Conversely, the last two events displayed lower radon measurements, falling below the lower bound of the confidence interval.

Figure 5.43 demonstrates our ability to detect the anomaly before a significant seismic event, which was of magnitude 5.3. In the third figure, we observed an anomaly preceding only one out of four seismic events. Figure 5.45 displayed anomalies prior to two out of three seismic events. However, for the last seismic event, only a single anomalous value was detected beforehand, which may not provide full confidence in predicting future earthquake. Furthermore, it is important to note that some seismic events displayed anomalous radon values even after the event. This may be due to the lasting effects of the earthquake, which influenced the radon measurements in the surrounding environment.

These findings highlight the model's capability to identify anomalies preceding seismic events, although the effectiveness may vary depending on the specific event and the number of anomalous measurements detected.

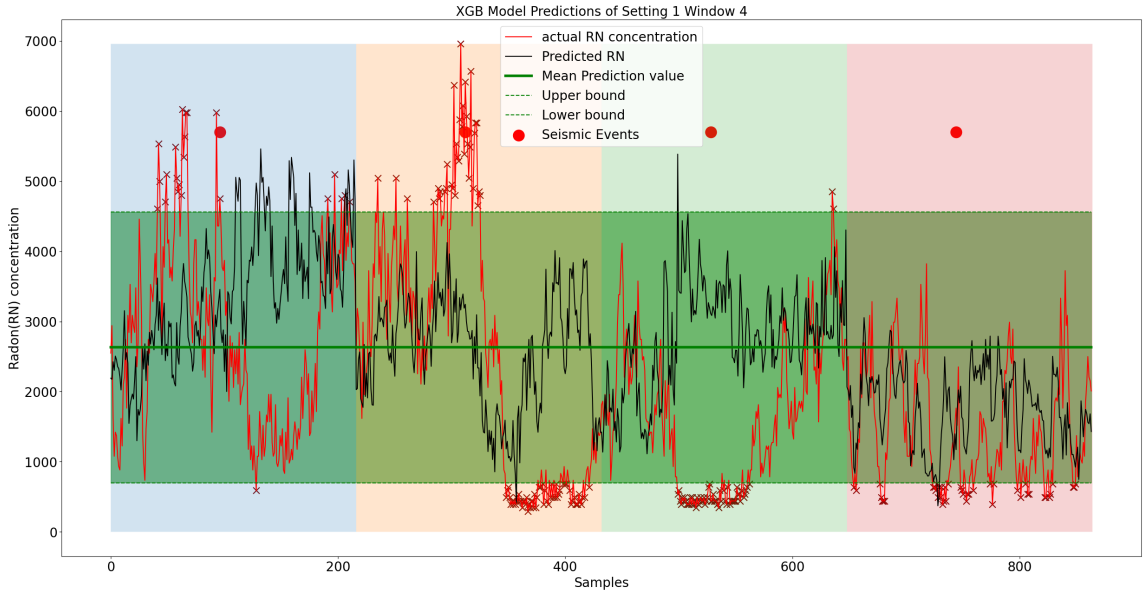


Figure 5.42: Anomaly Detection: Setting 1 and Window 4

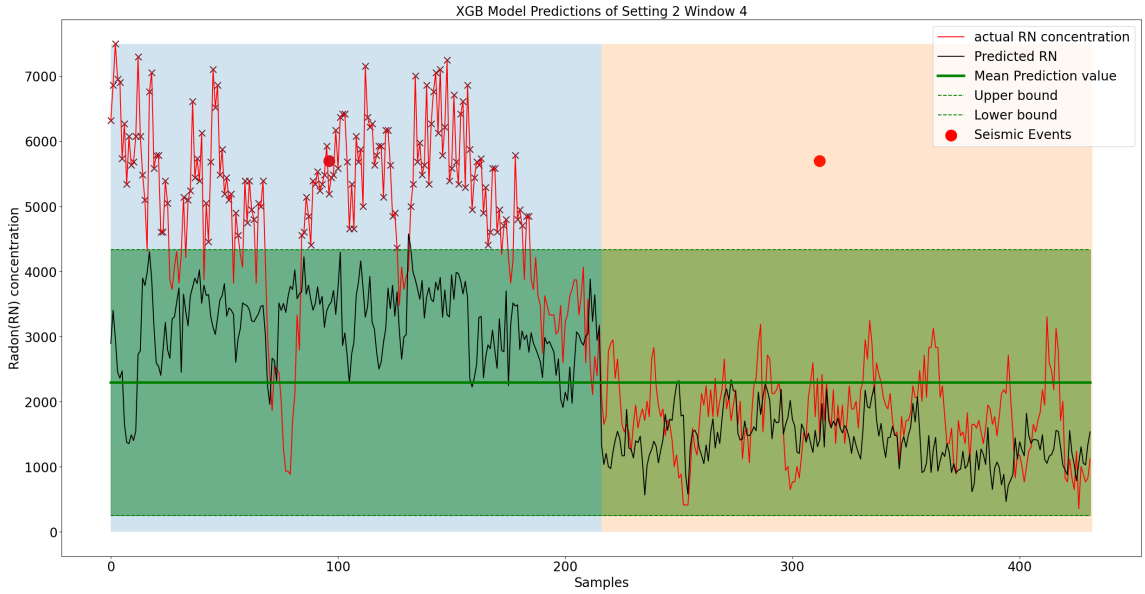


Figure 5.43: Anomaly Detection: Setting 2 and Window 4

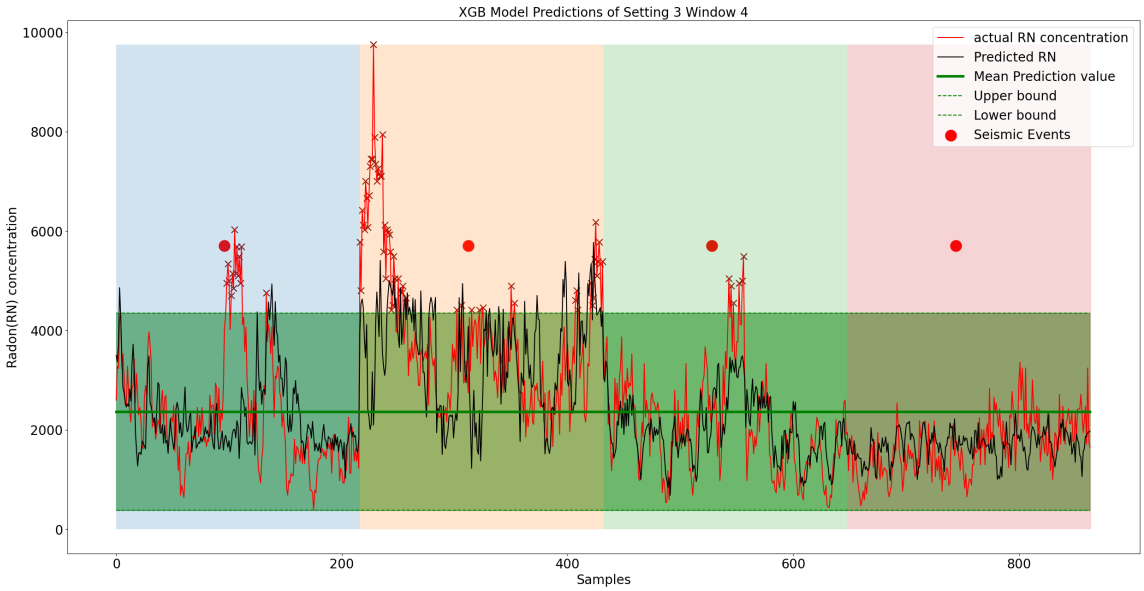


Figure 5.44: Anomaly Detection: Setting 3 and Window 4

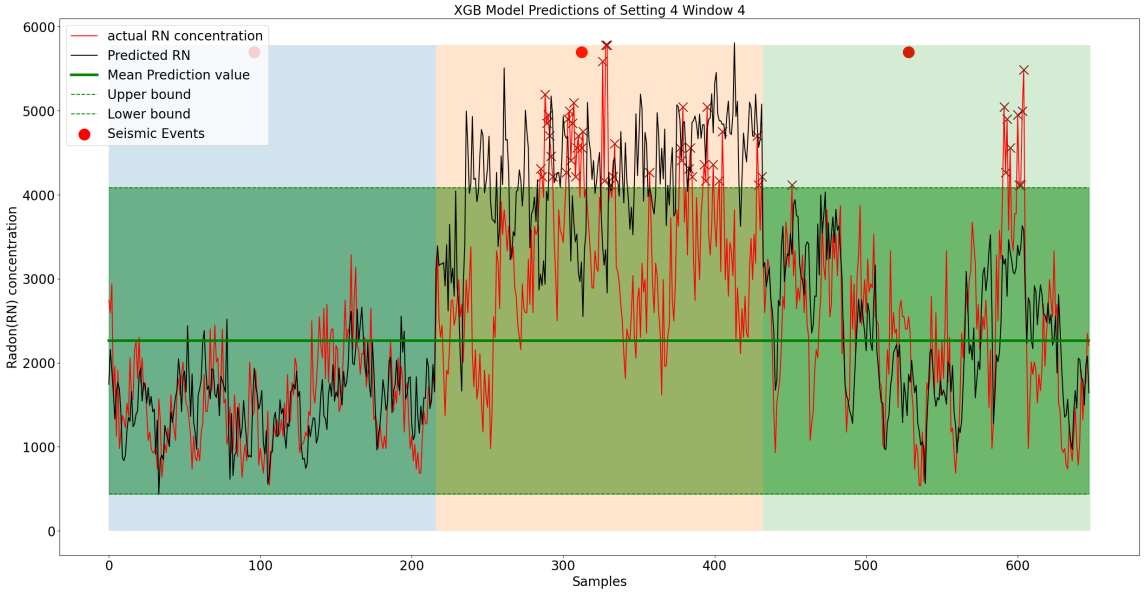


Figure 5.45: Anomaly Detection: Setting 4 and Window 4

CHAPTER 6
Conclusion

In conclusion, this thesis aimed to investigate the relationship between radon concentration and seismic events for the purpose of earthquake prediction. We conducted a series of experiments using various machine learning models and explored different preprocessing techniques to enhance the accuracy of radon predictions.

Through our experiments, we observed several key findings. Firstly, incorporating synthetically generated seismic data with the original dataset has improved the results significantly. Secondly, inclusions of extra features, such as the first-order derivatives(gradients) of environmental parameters, proved beneficial in capturing the sequential nature of time series data. This led to improved performance in radon prediction models.

Furthermore, we examined the impact of data segmentation and resampling on model performance. Segmentation allowed us to focus on a specific time period of the dataset, excluding potentially noisy or erratic data. Resampling the data into larger time intervals, such as 60 minutes, effectively smoothed out random fluctuations, resulting in enhanced model performance.

Moreover, we investigated the use of anomaly detection techniques to identify abnormal radon measurements prior to seismic events. Our results showed promising potential in using these anomalies as early warning signs of impending earthquakes. This highlights the importance of radon monitoring as a potential alarm system for seismic activity.

In terms of model selection, we found that Support Vector Regression with radial kernel, K-nearest neighbor, Gradient Boosting Machine and XGBoost consistently yielded excellent results across various settings and window sizes. These

models demonstrated their capability to accurately predict radon concentrations, particularly during seismic events.

However, it is important to note that further improvements can be made. Future research could focus on refining the anomaly detection algorithms and exploring additional preprocessing techniques to enhance the accuracy and reliability of radon predictions. Additionally, the incorporation of more diverse environmental parameters and the consideration of other external factors could provide valuable insights into the complex relationship between radon concentration and seismic events. One can extend this work further by incorporating more efficient deep learning models like Artificial Neural Network (ANN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), Long Short Term Model (LSTM) for the better radon time series prediction.

In conclusion, this thesis contributes to the field of earthquake prediction by demonstrating the effectiveness of machine learning models and preprocessing techniques in forecasting radon concentrations. The results obtained provide valuable insights and lay the foundation for future advancements in the field. By continuing to explore and refine these methods, we can improve our understanding of radon dynamics and further enhance our ability to predict seismic events with greater accuracy and reliability.

References

[1] S. Barbosa, R. Donner, and G. Steinitz. Radon applications in geosciences – progress perspectives. 224:597–603, 2015.

[2] S. Baykut, T. Akgül, S. İnan, and C. Seyis. Observation and removal of daily quasi-periodic components in soil radon data. *Radiation Measurements*, 45(7):872–879, 2010.

[3] L. Breiman. *Classification and Regression Trees*. Routledge, 1984.

[4] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[5] E. O. Brigham and R. E. Morrow. The fast fourier transform. *IEEE Spectrum*, 4(12):63–70, 1967.

[6] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[7] Z. Chen, Y. Li, Z. Liu, J. Wang, X. Zhou, and J. Du. Radon emission from soil gases in the active fault zones in the capital of china and its environmental effects. 8, 2018.

[8] N. Cristianini and J. S. Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, 1999.

[9] N. Dalsania, Z. Patel, S. Purohit, and B. Chaudhury. An application of machine learning for plasma current quench studies via synthetic data generation. *Fusion Engineering and Design*, 171, 2021.

[10] N. Das, R. Bhandari, D. Ghose, P. Sen, and B. Sinha. Anomalous fluctuation of radon, gamma dose and helium emanating from a thermal spring prior to an earthquake. *Current Science*, 89, 10 2005.

[11] E. Fix and J. J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *Randolph Field, Texas, USA*, pages 21–49, 1951.

[12] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Institute of Mathematical Statistics*, 29:1189–1232, 2001.

[13] D. Ghosh, A. Deb, and R. Sengupta. Anomalous radon emission as precursor of earthquake. 69:67–81, 2009.

[14] Y. Hwa Oh and G. Kim. A radon-thoron isotope pair as a reliable earthquake precursor. *Scientific Reports*, 5, 2015.

[15] C. King. Episodic radon changes in subsurface soil gas along active faults and possible relation to earthquakes. *Journal of Geophysical Research*, 85:3065–3078, 1980.

[16] R. Klusman and J. Webster. Preliminary analysis of meteorological and seasonal influences on crustal gas emission relevant to earthquake prediction. 71, 1981.

[17] S. Maldonado, M. Monnin, N. Segovia, and J. Seidel. A radon measurement network to study radon anomalies as precursors of strong earthquakes in the guerrero seismic gap. 11 wcee. page 6, 1996.

[18] A. A. Mir, F. V. Çelebi, H. Alsolai, S. A. Qureshi, M. Rafique, J. S. Alzahrani, H. Mahgoub, and M. A. Hamza. Anomalies prediction in radon time series for earthquake likelihood using machine learning-based ensemble model. *IEEE Access*, 10:37984–37999, 2022.

[19] A. A. Mir, F. V. Çelebi, M. Rafique, M. R. I. Faruque, M. U. Khandaker, K. J. Kearfott, and P. Ahmad. Anomaly classification for earthquake prediction in radon time series data using stacking and automatic anomaly indication function. *Pure and Applied Geophysics*, 178:1593–1607, 2021.

[20] S. Okabe. Time variation of the atmospheric radon content near the ground surface with relation to some geophysical phenomena. 28:699–115, 1956.

[21] S. K. Sahoo, M. Katlamudi, C. Barman, and G. U. Lakshmi. Identification of earthquake precursors in soil radon-222 data of kutch, gujarat, india using empirical mode decomposition based hilbert huang transform. *Journal of Environmental Radioactivity*, 222, 2020.

[22] S. K. Sahoo, M. Katlamudi, K. S. Murali Krishna, and G. Udaya Lakshmi. Influence of meteorological parameters on the soil radon (rn222) emanation in kutch, gujarat, india. *Environmental Monitoring and Assessment*, 190, 2018.

[23] S. Singh, H. P. Jaishi, R. P. Tiwari, and R. C. Tiwari. Time series analysis of soil radon data using multiple linear regression and artificial neural network in seismic precursory studies. *Pure and Applied Geophysics volume*, 174:2793–2802, 2017.

[24] A. Tanner. Radon migration in the ground: a supplementary review. 1, 1980.

[25] A. D. K. Tareen, K. M. Asim, K. J. Kearfott, M. Rafique, M. S. Nadeem, T. Iqbal, and S. U. Rahman. Automated anomalous behaviour detection in soil radon gas prior to earthquakes using computational intelligence techniques. 203:48–54, 2019.

[26] D. Torkar, B. Zmazek, and J. Vaupotič. Application of artificial neural networks in simulating radon levels in soil gas. *Chemical Geology*, 270(1):1–8, 2010.

[27] V. I. Ulomov and B. Z. Mavashev. A precursor of a strong tectonic earthquake. 176:9–11, 1967.

[28] V. I. Ulomov, A. I. Zakharovc, and N. V. Ulomova. Tashkent earthquake of april 26, 1966, and its aftershocks. 177:567–570, 1967.

[29] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Information Science and Statistics (ISS). Springer, 2013.

[30] J. Vaupotič, A. Riggio, M. Santulin, B. Zmazek, and I. Kobal. A radon anomaly in soil gas at cazzaso ne italy as a precursor of an ml=5.1 earthquake. 55:507–511, 2010.

[31] V. Walia, T. Su, C. Fu, and T. Yang. Spatial variations of radon and helium concentrations in soil-gas across the shan-chiao fault, northern taiwan. *Radiation Measurements*, 40(2):513–516, 2005. Proceedings of the 22nd International Conference on Nuclear Tracks in Solids.

[32] T. Yang, V. Walia, L. Chyi, C. Fu, C.-H. Chen, T. Liu, S. Song, C. Lee, and M. Lee. Variations of soil radon and thoron concentrations in a fault zone and prospective earthquakes in sw taiwan. *Radiation Measurements*, 40(2):496–502, 2005. Proceedings of the 22nd International Conference on Nuclear Tracks in Solids.

[33] B. Zmazek, L. Todorovski, S. Džeroski, and J. Vaupotič. Application of decision trees to the analysis of soil radon data for earthquake prediction. *Applied Radiation and Isotopes*, 58(6):697–706, 2003.

CHAPTER A
Experiment 1: Performance metrics tables

A.1 Results of Setting 1 Dataset

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1763.079	1.2115	2.410	102.156
SVR with RBF kernel	1362.790	1.0393	1.7916	62.4114
Random forest regressor	1566.4525	1.0190	1.74971	72.4987
KNN regressor	1521.047	1.0447	1.8453	78.4474
Gradient Boosting Machine	1562.5684	0.9985	1.6773	73.903
XGBoost	1562.5684	0.9985	1.6773	73.903

Table A.1: Performance of ML models on Setting1 and Window1

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1798.032	1.143	2.172	82.189
SVR with RBF kernel	1473.1004	1.0095	1.7000	50.7488
Random forest regressor	1809.3301	1.0753	1.8574	56.4338
KNN regressor	1782.5615	1.088	1.9878	74.7393
Gradient Boosting Machine	1562.5684	0.9985	1.6773	73.903
XGBoost	1562.5684	0.9985	1.6773	73.903

Table A.2: Performance of ML models on Setting1 and Window2

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1708.418	1.0286	1.78212	59.3874
SVR with RBF kernel	1433.0753	0.9148	1.4114	34.7324
Random forest regressor	1824.0299	0.9937	1.6242	48.7033
KNN regressor	1784.3240	1.0047	1.7131	61.1671
Gradient Boosting Machine	1562.5684	0.9985	1.6773	73.903
XGBoost	1562.5684	0.9985	1.6773	73.903

Table A.3: Performance of ML models on Setting1 and Window3

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1646.490	0.9531	1.5432	42.5179
SVR with RBF kernel	1397.1943	0.8489	1.2270	22.8572
Random forest regressor	1753.1814	0.9154	1.3921	36.9052
KNN regressor	1742.0361	0.9336	1.4916	46.9100
Gradient Boosting Machine	1562.5684	0.9985	1.6773	73.903
XGBoost	1562.5684	0.9985	1.6773	73.903

Table A.4: Performance of ML models on Setting1 and Window4

A.2 Results of Setting 2 Dataset

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	2073.461	0.6510	0.6541	-17.108
SVR with RBF kernel	2131.8123	0.6444	0.5503	-29.9984
Random forest regressor	1789.772	0.6353	0.6762	-1.0238
KNN regressor	2093.4504	0.6542	0.6063	-19.6836
Gradient Boosting Machine	1562.5684	0.9985	1.6773	73.903
XGBoost	1562.5684	0.9985	1.6773	73.903

Table A.5: Performance of ML models on Setting2 and Window1

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	2172.874	0.6807	0.7023	-21.048
SVR with RBF kernel	2321.677	0.7029	0.6185	-33.7399
Random forest regressor	2089.787	0.7053	0.6834	-19.99
KNN regressor	2285.698	0.708	0.639	-29.436
Gradient Boosting Machine	2149.712	0.716	0.698	-19.82
XGBoost	2178.504	0.729	0.710	-20.215

Table A.6: Performance of ML models on Setting2 and Window2

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	2061.104	0.6371	0.625	-22.1240
SVR with RBF kernel	2246.103	0.6794	0.5792	-33.3903
Random forest regressor	2250.601	0.735	0.681	-24.184
KNN regressor	2280.120	0.710	0.606	-30.216
Gradient Boosting Machine	2268.797	0.736	0.682	-25.077
XGBoost	2272.418	0.742	0.688	-24.213

Table A.7: Performance of ML models on Setting2 and Window3

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	2037.274	0.6158	0.5731	-23.543
SVR with RBF kernel	2200.4396	0.6588	0.5428	-33.2532
Random forest regressor	2236.737	0.713	0.630	-25.605
KNN regressor	2276.245	0.712	0.584	-29.636
Gradient Boosting Machine	2242.602	0.714	0.619	-27.306
XGBoost	2244.821	0.719	0.627	-26.737

Table A.8: Performance of ML models on Setting2 and Window4

A.3 Results of Setting 3 dataset

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	2040.004	0.6691	0.5815	-31.1512
SVR with RBF kernel	2022.704	0.6137	0.5472	-25.9081
Random forest regressor	2137.607	0.647	0.515	-26.118
KNN regressor	2031.657	0.621	0.543	-23.787
Gradient Boosting Machine	2221.759	0.681	0.558	-23.129
XGBoost	2203.462	0.683	0.560	-23.857

Table A.9: Performance of ML models on Setting 3 and Window 1

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1831.9619	0.71350	0.6804	-23.3767
SVR with RBF kernel	1772.821	0.6295	0.6272	-16.7380
Random forest regressor	1845.840	0.661	0.633	-15.148
KNN regressor	1807.304	0.653	0.665	-13.577
Gradient Boosting Machine	1913.934	0.692	0.696	-8.779
XGBoost	1898.775	0.677	0.657	-11.641

Table A.10: Performance of ML models on Setting 3 and Window 2

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1684.5796	0.70579	0.7080	-16.4619
SVR with RBF kernel	1620.1294	0.6343	0.6701	-9.2713
Random forest regressor	1799.419	0.709	0.798	0.285
KNN regressor	1672.823	0.663	0.722	-3.659
Gradient Boosting Machine	1843.053	0.726	0.830	3.802
XGBoost	1820.738	0.716	0.816	2.582

Table A.11: Performance of ML models on Setting 3 and Window 3

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1613.001	0.7099	0.7384	-12.8658
SVR with RBF kernel	1551.1507	0.6471	0.7045	-6.6269
Random forest regressor	1828.878	0.746	0.864	0.117
KNN regressor	1644.635	0.683	0.763	-2.072
Gradient Boosting Machine	1778.022	0.742	0.865	4.318
XGBoost	1802.980	0.746	0.870	3.700

Table A.12: Performance of ML models on Setting3 and Window4

A.4 Results of Setting 4 dataset

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	3548.5262	0.9365	0.9536	-45.5601
SVR with RBF kernel	3195.6427	0.8299	0.8768	-40.1138
Random forest regressor	2748.581	0.690	0.677	-28.815
KNN regressor	2777.307	0.772	0.857	-30.263
Gradient Boosting Machine	2642.056	0.645	0.610	-27.979
XGBoost	2678.717	0.669	0.645	-28.755

Table A.13: Performance of ML models on Setting 4 and Window 1

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	3281.4268	0.8718	0.8326	-43.984
SVR with RBF kernel	2995.1563	0.7810	0.7698	-39.493
Random forest regressor	2351.834	0.629	0.571	-24.201
KNN regressor	2558.415	0.710	0.731	-28.209
Gradient Boosting Machine	2367.993	0.638	0.561	-25.978
XGBoost	2359.404	0.784	0.540	-26.302

Table A.14: Performance of ML models on Setting 4 and Window 2

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	3254.8118	0.8611	0.8031	-42.6700
SVR with RBF kernel	2998.7690	0.7816	0.7597	-38.5656
Random forest regressor	2284.228	0.623	0.596	-23.411
KNN regressor	2588.837	0.705	0.719	-26.375
Gradient Boosting Machine	2337.541	0.640	0.619	-23.784
XGBoost	2354.264	0.632	0.578	-25.919

Table A.15: Performance of ML models on Setting 4 and Window 3

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	3135.115	0.8301	0.7539	-40.9715
SVR with RBF kernel	2912.004	0.7567	0.7056	-38.1760
Random forest regressor	2229.348	0.609	0.565	-24.685
KNN regressor	2525.720	0.683	0.671	-26.204
Gradient Boosting Machine	2311.208	0.641	0.612	-24.242
XGBoost	2300.017	0.641	0.624	-23.929

Table A.16: Performance of ML models on Setting 4 and Window 4

CHAPTER B
Experiment 2: Performance metrics tables

B.1 Results of Setting 1 Dataset

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	1569.860	1.080	1.956	82.752
XGBoost	1465.613	0.957	1.569	63.079
KNN regressor (with gradient features)	1616.096	1.122	2.092	83.868
XGBoost (with gradient features)	1440.941	0.920	1.456	56.039

Table B.1: Performance of ML models on Setting1 and Window1

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	1765.723	1.089	1.992	72.143
XGBoost	1685.308	1.013	1.664	47.270
KNN regressor (with gradient features)	1818.978	1.123	2.120	77.991
XGBoost (with gradient features)	1654.398	1.001	1.626	44.345

Table B.2: Performance of ML models on Setting1 and Window2

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	1732.477	0.992	1.678	56.892
XGBoost	1655.804	0.927	1.407	34.692
KNN regressor (with gradient features)	1772.231	1.026	1.788	63.033
XGBoost (with gradient features)	1620.467	0.919	1.402	35.434

Table B.3: Performance of ML models on Setting1 and Window3

B.2 Results of Setting 2 Dataset

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	1670.266	0.923	1.463	43.302
XGBoost	1565.454	0.859	1.206	21.459
KNN regressor (with gradient features)	1684.242	0.941	1.512	44.821
XGBoost (with gradient features)	1581.797	0.856	1.201	21.196

Table B.4: Performance of ML models on Setting1 and Window4

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	1930.453	0.607	0.581	-16.222
XGBoost	1730.685	0.599	0.612	-8.511
KNN regressor (with gradient features)	1950.659	0.618	0.619	-15.036
XGBoost (with gradient features)	1785.854	0.600	0.622	-10.184

Table B.5: Performance of ML models on Setting2 and Window1

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	2162.946	0.676	0.633	-24.519
XGBoost	2059.587	0.688	0.673	-20.709
KNN regressor (with gradient features)	2048.277	0.643	0.645	-20.498
XGBoost (with gradient features)	2073.945	0.687	0.690	-20.704

Table B.6: Performance of ML models on Setting2 and Window2

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	2110.825	0.655	0.589	-24.135
XGBoost	2154.978	0.707	0.672	-22.732
KNN regressor (with gradient features)	2047.914	0.629	0.578	-23.117
XGBoost (with gradient features)	2191.391	0.7125	0.6460	-27.039

Table B.7: Performance of ML models on Setting2 and Window3

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	2080.455	0.639	0.558	-23.261
XGBoost	2052.696	0.664	0.590	-23.569
KNN regressor (with gradient features)	1969.966	0.610	0.575	-18.208
XGBoost (with gradient features)	2042.590	0.650	0.601	-21.944

Table B.8: Performance of ML models on Setting2 and Window4

B.3 Results of Setting 3 dataset

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	1949.270	0.612	0.568	-21.249
XGBoost	1761.409	0.564	0.513	-16.392
KNN regressor (with gradient features)	1949.604	0.617	0.579	-21.080
XGBoost (with gradient features)	1767.043	0.565	0.515	-18.023

Table B.9: Performance of ML models on Setting 3 and Window 1

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	1752.394	0.647	0.680	-11.128
XGBoost	1648.556	0.628	0.660	-8.168
KNN regressor (with gradient features)	1676.477	0.651	0.713	-8.652
XGBoost (with gradient features)	1611.761	0.620	0.657	-8.235

Table B.10: Performance of ML models on Setting 3 and Window 2

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	1637.057	0.659	0.734	-2.878
XGBoost	1539.941	0.647	0.728	1.019
KNN regressor (with gradient features)	1625.123	0.658	0.725	-6.092
XGBoost (with gradient features)	1517.524	0.651	0.744	2.737

Table B.11: Performance of ML models on Setting 3 and Window 3

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	1619.538	0.683	0.786	-0.303
XGBoost	1532.417	0.686	0.790	3.035
KNN regressor (with gradient features)	1629.991	0.698	0.826	1.149
XGBoost (with gradient features)	1516.264	0.677	0.780	2.901

Table B.12: Performance of ML models on Setting3 and Window4

B.4 Results of Setting 4 dataset

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	2608.249	0.715	0.774	-29.806
XGBoost	2510.788	0.646	0.655	-28.000
KNN regressor (with gradient features)	2681.130	0.737	0.828	-29.350
XGBoost (with gradient features)	2448.747	0.626	0.640	-27.143

Table B.13: Performance of ML models on Setting 4 and Window 1

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	2412.385	0.662	0.667	-27.503
XGBoost	2273.783	0.676	0.577	-26.407
KNN regressor (with gradient features)	2513.046	0.705	0.751	-27.211
XGBoost (with gradient features)	2229.114	0.613	0.576	-25.147

Table B.14: Performance of ML models on Setting 4 and Window 2

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	2438.237	0.658	0.655	-25.716
XGBoost	2289.689	0.621	0.608	-24.213
KNN regressor (with gradient features)	2606.892	0.699	0.707	-29.296
XGBoost (with gradient features)	2313.418	0.612	0.584	-25.592

Table B.15: Performance of ML models on Setting 4 and Window 3

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	2379.275	0.638	0.617	-25.378
XGBoost	2224.220	0.599	0.569	-24.296
KNN regressor (with gradient features)	2618.930	0.700	0.698	-29.732
XGBoost (with gradient features)	2275.854	0.612	0.577	-25.635

Table B.16: Performance of ML models on Setting 4 and Window 4

CHAPTER C
Experiment 3: Performance metrics tables

Models	RMSE	RMSLE	MAPE	PB
KNN regressor	1121.797	1.083	2.705	-73.089
XGBoost	1218.154	1.275	2.246	-68.946

Table C.1: Performance of KNN and XGBoost models on Setting 1 and Window 4 of reconstructed radon data

CHAPTER D
Experiment 4: Performance metrics tables

D.1 Results of Setting 1 Dataset

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1675.153 ± 4.106	1.041 ± 0.002	1.794 ± 0.007	43.747 ± 0.487
SVR with RBF kernel	1266.694 ± 12.871	0.748 ± 0.007	0.996 ± 0.015	14.965 ± 0.881
Random forest regressor	1504.743 ± 17.203	0.833 ± 0.010	1.200 ± 0.026	29.918 ± 1.010
KNN regressor	1453.810 ± 13.572	0.886 ± 0.007	1.348 ± 0.018	38.616 ± 0.689
XGBoost	1425.636 ± 34.579	0.775 ± 0.019	1.057 ± 0.040	25.946 ± 1.547

Table D.1: Performance of ML models on Setting1 and Window1

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1685.032 ± 4.592	1.021 ± 0.002	1.747 ± 0.007	47.405 ± 0.420
SVR with RBF kernel	1260.341 ± 9.237	0.758 ± 0.006	1.007 ± 0.014	15.823 ± 0.821
Random forest regressor	1440.571 ± 16.687	0.757 ± 0.007	0.988 ± 0.015	18.627 ± 0.719
KNN regressor	1555.096 ± 9.474	0.913 ± 0.004	1.418 ± 0.011	44.399 ± 0.503
XGBoost	1447.739 ± 26.569	0.730 ± 0.014	0.900 ± 0.025	16.174 ± 1.243

Table D.2: Performance of ML models on Setting1 and Window2

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1591.935 ± 5.367	0.958 ± 0.002	1.559 ± 0.007	42.722 ± 0.494
SVR with RBF kernel	1276.489 ± 10.431	0.764 ± 0.006	1.018 ± 0.015	17.281 ± 0.660
Random forest regressor	1410.108 ± 14.336	0.736 ± 0.008	0.948 ± 0.018	19.168 ± 0.887
KNN regressor	1503.911 ± 9.480	0.887 ± 0.005	1.357 ± 0.013	44.474 ± 0.544
XGBoost	1401.926 ± 25.832	0.729 ± 0.020	0.923 ± 0.043	19.879 ± 1.676

Table D.3: Performance of ML models on Setting1 and Window3

D.2 Results of Setting 2 Dataset

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1523.800 ± 4.899	0.907 ± 0.002	1.411 ± 0.006	37.351 ± 0.449
SVR with RBF kernel	1275.861 ± 7.009	0.769 ± 0.004	1.014 ± 0.009	13.903 ± 0.620
Random forest regressor	1385.973 ± 12.021	0.743 ± 0.009	0.941 ± 0.020	11.238 ± 0.875
KNN regressor	1486.361 ± 8.207	0.864 ± 0.004	1.290 ± 0.010	40.166 ± 0.521
XGBoost	1558.656 ± 50.066	0.858 ± 0.023	1.216 ± 0.065	19.099 ± 2.088

Table D.4: Performance of ML models on Setting1 and Window4

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1817.051 ± 7.155	0.576 ± 0.002	0.450 ± 0.001	-37.482 ± 0.215
SVR with RBF kernel	1769.745 ± 24.322	0.524 ± 0.007	0.424 ± 0.004	-24.869 ± 0.753
Random forest regressor	1224.620 ± 12.099	0.412 ± 0.004	0.359 ± 0.004	-3.774 ± 0.783
KNN regressor	1655.044 ± 13.438	0.483 ± 0.004	0.388 ± 0.004	-25.171 ± 0.351
XGBoost	1245.129 ± 20.986	0.409 ± 0.006	0.353 ± 0.007	-6.459 ± 1.123

Table D.5: Performance of ML models on Setting2 and Window1

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1806.712 ± 5.672	0.584 ± 0.002	0.430 ± 0.001	-38.980 ± 0.134
SVR with RBF kernel	1803.951 ± 15.598	0.517 ± 0.005	0.391 ± 0.004	-28.999 ± 0.483
Random forest regressor	1263.030 ± 17.053	0.406 ± 0.007	0.324 ± 0.004	-17.532 ± 0.687
KNN regressor	1691.291 ± 12.912	0.477 ± 0.004	0.360 ± 0.003	-29.271 ± 0.356
XGBoost	1312.230 ± 24.688	0.412 ± 0.010	0.330 ± 0.006	-18.447 ± 1.011

Table D.6: Performance of ML models on Setting2 and Window2

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1733.833 ± 7.662	0.558 ± 0.003	0.411 ± 0.001	-37.684 ± 0.200
SVR with RBF kernel	1822.966 ± 16.930	0.552 ± 0.006	0.437 ± 0.005	-27.671 ± 0.529
Random forest regressor	1432.741 ± 26.043	0.456 ± 0.007	0.374 ± 0.006	-20.629 ± 0.730
KNN regressor	1686.968 ± 9.767	0.494 ± 0.003	0.380 ± 0.002	-29.491 ± 0.312
XGBoost	1449.707 ± 34.108	0.472 ± 0.011	0.393 ± 0.009	-21.616 ± 1.160

Table D.7: Performance of ML models on Setting2 and Window3

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1754.626 ± 5.556	0.559 ± 0.002	0.407 ± 0.001	-37.188 ± 0.170
SVR with RBF kernel	1960.545 ± 11.275	0.601 ± 0.004	0.429 ± 0.003	-32.667 ± 0.478
Random forest regressor	1624.719 ± 22.304	0.520 ± 0.008	0.388 ± 0.004	-30.031 ± 0.782
KNN regressor	1791.879 ± 9.166	0.530 ± 0.003	0.388 ± 0.003	-31.665 ± 0.310
XGBoost	1623.635 ± 26.630	0.536 ± 0.012	0.390 ± 0.008	-30.180 ± 1.098

Table D.8: Performance of ML models on Setting2 and Window4

D.3 Results of Setting 3 dataset

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1163.396 ± 2.436	0.422 ± 0.001	0.309 ± 0.001	-22.173 ± 0.138
SVR with RBF kernel	1049.105 ± 12.598	0.402 ± 0.005	0.315 ± 0.004	-9.251 ± 0.598
Random forest regressor	1095.825 ± 14.553	0.397 ± 0.004	0.299 ± 0.003	-8.918 ± 0.661
KNN regressor	1067.870 ± 5.950	0.387 ± 0.003	0.287 ± 0.003	-13.865 ± 0.312
XGBoost	1157.040 ± 16.396	0.445 ± 0.009	0.329 ± 0.006	-11.857 ± 0.914

Table D.9: Performance of ML models on Setting 3 and Window 1

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1003.835 ± 1.443	0.425 ± 0.000	0.338 ± 0.001	-14.102 ± 0.162
SVR with RBF kernel	976.498 ± 7.378	0.423 ± 0.003	0.378 ± 0.003	2.639 ± 0.476
Random forest regressor	1010.883 ± 11.862	0.409 ± 0.003	0.365 ± 0.004	3.966 ± 0.784
KNN regressor	958.384 ± 5.257	0.405 ± 0.002	0.351 ± 0.003	-1.904 ± 0.315
XGBoost	1034.254 ± 16.228	0.434 ± 0.007	0.376 ± 0.006	0.405 ± 0.877

Table D.10: Performance of ML models on Setting 3 and Window 2

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	964.549 ± 1.462	0.420 ± 0.000	0.339 ± 0.001	-13.539 ± 0.157
SVR with RBF kernel	903.212 ± 5.921	0.398 ± 0.002	0.360 ± 0.002	4.700 ± 0.417
Random forest regressor	958.488 ± 11.598	0.398 ± 0.003	0.355 ± 0.004	4.402 ± 0.597
KNN regressor	911.911 ± 4.321	0.392 ± 0.001	0.347 ± 0.002	1.554 ± 0.291
XGBoost	960.675 ± 15.065	0.415 ± 0.005	0.362 ± 0.006	1.826 ± 0.861

Table D.11: Performance of ML models on Setting 3 and Window 3

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	1167.565 ± 2.313	0.449 ± 0.001	0.358 ± 0.001	-18.090 ± 0.187
SVR with RBF kernel	988.932 ± 6.302	0.395 ± 0.002	0.350 ± 0.003	-1.432 ± 0.486
Random forest regressor	1030.605 ± 11.240	0.398 ± 0.003	0.340 ± 0.003	-2.611 ± 0.443
KNN regressor	1004.551 ± 4.659	0.396 ± 0.001	0.344 ± 0.002	-3.082 ± 0.269
XGBoost	1077.445 ± 14.536	0.427 ± 0.006	0.356 ± 0.005	-4.495 ± 0.758

Table D.12: Performance of ML models on Setting3 and Window4

D.4 Results of Setting 4 dataset

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	799.430 ± 2.866	0.366 ± 0.001	0.325 ± 0.002	-7.836 ± 0.290
SVR with RBF kernel	883.409 ± 16.093	0.355 ± 0.005	0.275 ± 0.004	-14.205 ± 0.669
Random forest regressor	820.725 ± 11.004	0.340 ± 0.004	0.268 ± 0.003	-14.503 ± 0.593
KNN regressor	821.089 ± 7.673	0.350 ± 0.003	0.301 ± 0.004	-6.256 ± 0.425
XGBoost	813.663 ± 17.767	0.350 ± 0.007	0.278 ± 0.005	-11.141 ± 0.885

Table D.13: Performance of ML models on Setting 4 and Window 1

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	843.725 ± 2.459	0.361 ± 0.001	0.306 ± 0.001	-10.058 ± 0.253
SVR with RBF kernel	924.168 ± 12.107	0.370 ± 0.005	0.296 ± 0.004	-11.331 ± 0.588
Random forest regressor	882.463 ± 10.462	0.355 ± 0.004	0.284 ± 0.003	-12.750 ± 0.668
KNN regressor	848.230 ± 7.604	0.348 ± 0.003	0.297 ± 0.003	-6.377 ± 0.389
XGBoost	856.988 ± 16.883	0.355 ± 0.006	0.286 ± 0.006	-8.983 ± 0.788

Table D.14: Performance of ML models on Setting 4 and Window 2

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	871.275 ± 1.807	0.370 ± 0.001	0.317 ± 0.001	-8.293 ± 0.260
SVR with RBF kernel	877.931 ± 10.470	0.365 ± 0.004	0.302 ± 0.003	-7.384 ± 0.527
Random forest regressor	880.119 ± 8.523	0.362 ± 0.003	0.306 ± 0.003	-5.311 ± 0.657
KNN regressor	848.903 ± 6.439	0.363 ± 0.002	0.304 ± 0.003	-2.879 ± 0.331
XGBoost	889.043 ± 18.206	0.365 ± 0.005	0.313 ± 0.006	-1.502 ± 1.097

Table D.15: Performance of ML models on Setting 4 and Window 3

Models	RMSE	RMSLE	MAPE	PB
SVR with linear kernel	832.009 ± 1.445	0.365 ± 0.000	0.318 ± 0.001	-5.691 ± 0.291
SVR with RBF kernel	849.146 ± 6.916	0.363 ± 0.003	0.310 ± 0.004	-3.967 ± 0.656
Random forest regressor	998.946 ± 24.774	0.391 ± 0.005	0.363 ± 0.006	10.250 ± 0.942
KNN regressor	881.389 ± 5.870	0.376 ± 0.002	0.330 ± 0.002	1.736 ± 0.349
XGBoost	1031.347 ± 35.306	0.395 ± 0.007	0.370 ± 0.010	12.709 ± 1.690

Table D.16: Performance of ML models on Setting 4 and Window 4