

ANKARA YILDIRIM BEYAZIT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES



**RADON ANOMALY CLASSIFICATION AND MISSING DATA
IMPUTATION: A METHODOLOGY BASED ON STACKING AND
AUTOMATIC ANOMALY INDICATION FUNCTION**

Ph.D. Thesis by

Adil Aslam MIR

Department of Computer Engineering

April, 2022

ANKARA

**RADON ANOMALY CLASSIFICATION AND MISSING
DATA IMPUTATION: A METHODOLOGY BASED ON
STACKING AND AUTOMATIC ANOMALY
INDICATION FUNCTION**

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

**In Partial Fulfillment of the Requirements for the Degree of Doctor of
Philosophy in Computer Engineering, Department of Computer Engineering**

by

Adil Aslam MIR

April, 2022

ANKARA

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**Radon Anomaly Classification and Missing Data Imputation: A Methodology Based on Stacking and Automatic Anomaly Indication Function**” completed by **ADIL ASLAM MIR** under the supervision of **PROF. DR. FATİH VEHBİ ÇELEBİ** and we certify that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Fatih Vehbi ÇELEBİ

Supervisor

Assoc. Prof. Dr. Fatih NAR

Jury Member

Prof. Dr. Şahin EMRAH

Jury Member

Assist. Prof. Dr. Hilal ARSLAN

Jury Member

Assist. Prof. Dr. Bülent TUĞRUL

Jury Member

Prof. Dr. Sadettin ORHAN

Director

Graduate School of Natural and Applied Sciences

ETHICAL DECLARATION

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of the Graduate School of Natural and Applied Sciences,

- All data, information, and documents are obtained in the framework of academic and ethical rules,
- All information, documents, and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of the above statements, I accept to renounce all my legal rights.

Date:

Signature:.....

Name & Surname: Adil Aslam MIR

ACKNOWLEDGMENTS

First of all, praise is to Allah, the most compassionate, the most merciful for giving me the courage, knowledge, and persistence to accomplish this study. Darood-o-Salaam to Prophet Muhammad (P.B.U.H) chosen by Almighty Allah to guide mankind to the divine right path.

It would not have been possible to complete my thesis without the assistance and support of the wonderful individuals in my life, only a few of whom may be mentioned here.

Above all, I would like to express my deepest thanks and sincere appreciation to my supervisor **Prof. Dr. Fatih Vehbi Çelebi** whose untired support, stimulating suggestions, guidance and encouragement helped me all the time of the research and during the writing of this thesis. For me, it is a great honor to work under his supervision.

I would like to thank all the faculty members of the Department of Computer Engineering, especially **Assoc. Prof. Dr. Fatih Nar** for their support and cooperation. I am also deeply grateful to **Prof. Dr. Muhammad Rafique Mir** for his kind endless help and support during the study.

I acknowledge The University of Azad Jammu & Kashmir for providing me an opportunity for carrying my research abroad.

I submit my heartiest gratitude to my mother and my dear wife whose prayers of day and night make me able to get such success and honor. I want to take this opportunity to pay my gratitude to my late father; **Muhammad Aslam Mir** who has raised me into the person I am today.

Truthful thanks to my wife, my son, my brothers, and my family for being a great source of support and encouragement to complete this work.

I respectfully express my gratitude to all persons involved who assisted me in my endeavor.

Adil Aslam MIR

Radon Anomaly Classification and Missing Data Imputation: A Methodology Based on Stacking and Automatic Anomaly Indication Function

ABSTRACT

With the advancement of technology, several studies have been conducted across the globe that performs forecasting and anomaly detection in time-series data to capture anomalous behavior. The detection of anomalies can be utilized to make early warning systems such for earthquakes, typhoons, floods, etc. This study is primarily concerned with data preprocessing such as missing data imputation and the development of an anomaly/outlier detection methodology. We propose Imputation by feature importance (IBFI), a methodology that is capable of utilizing any machine learning algorithm as a base learner, such as Naïve Bayes or Support Vector Machine, to efficiently impute missing or intermittently sampled multivariate time series data across various missingness scenarios, such as missing not at random (MNAR), missing completely at random (MCAR), and missing at random (MAR). In the reconstruction of missing features in multivariate time series data, the proposed methodology is evaluated against commonly applied imputation methods such as predictive mean matching, median, mean, mode, and hot-deck. The proposed methodology is advantageous in scenarios where more than one feature is missing in certain samples. Employing any base learning algorithm (XGBoost in this study), the proposed methodology takes advantage of the importance of features amongst each other and reconstructs missing features in the samples in respective scans. Moreover, we also proposed an anomaly detection framework based on automatic anomaly indication function and stacking ensemble learning. The framework can accurately classify all the anomalies or outliers in the time series data. Data for the Radon Time Series (RTS) was recorded from the 1st of March 2017 to the 28th of February 2018. During the study period, there were four anomalies or earthquake events.

Keywords: Anomaly, Random Forest, Machine Learning, Stacking, XGBoost.

Radon Anomali Sınıflandırması ve Eksik Veri Tahmini: Yığınlama ve Otomatik Anomali Gösterge Fonksiyonuna Dayalı Bir Metodoloji

ÖZ

Teknolojinin ilerlemesiyle birlikte, anormal davranışları yakalamak için zaman serisi verilerinde tahmin ve anormallik tespiti yapan dünya çapında çeşitli çalışmalar yapılmıştır. Anormalliklerin tespiti, deprem, tayfun, sel vb. için erken uyarı sistemleri yapmak için kullanılabilir. Bu çalışma, öncelikle eksik veri ataması gibi veri ön işleme ve bir anormallik/aykırı değer tespit metodolojisinin geliştirilmesi ile ilgilidir. Çeşitli eksiklik senaryolarında eksik veya aralıklı olarak örneklenmiş çok değişkenli zaman serisi verilerini verimli bir şekilde atamak için Naive Bayes gibi herhangi bir makine öğrenimi algoritmasını temel öğrenici olarak kullanabilen bir metodoloji olan Özellik önemine göre Imputation'ı (IBFI) öneriyoruz. rastgele değil (MNAR), rastgele tamamen eksik (MCAR) ve rastgele eksik (MAR) gibi. Çok değişkenli zaman serisi verilerindeki eksik özelliklerin yeniden yapılandırılmasında, önerilen metodoloji, tahmine dayalı ortalama eşleştirme, medyan, ortalama, mod ve hot-deck gibi yaygın olarak uygulanan atama yöntemlerine karşı değerlendirilir. Önerilen metodoloji, belirli örneklerde birden fazla özelliğin eksik olduğu senaryolarda avantajlıdır. Herhangi bir temel öğrenme algoritması (bu çalışmada XGBoost) kullanan önerilen metodoloji, özneliliklerin birbirleri arasındaki öneminden yararlanır ve ilgili taramalarda örneklerdeki eksik özellikleri yeniden oluşturur. Ayrıca, otomatik anormallik gösterge işlevine ve kümeleme topluluğu öğrenmesine dayalı bir anormallik algılama çerçevesi de önerdik. Çerçeve, zaman serisi verilerindeki tüm anormallikleri veya aykırı değerleri doğru bir şekilde sınıflandırabilir. Radon Zaman Serileri (RTS) için veriler 1 Mart 2017 ile 28 Şubat 2018 arasında kaydedildi. Çalışma süresi boyunca dört anormallik veya deprem olayı meydana geldi.

Kelimeler: Anomali, Random Forest, Makine öğrenme, Stacking, XGBoost.

CONTENTS

Ph.D. THESIS EXAMINATION RESULT FORM	ii
ETHICAL DECLARATION	iii
ACKNOWLEDGMENTS	iv
ABSTRACT.....	v
ÖZ... ..	vi
NOMENCLATURE.....	ix
LIST OF TABLES	xi
LIST OF FIGURES	xii
 CHAPTER 1 - INTRODUCTION.....	 1
1.1 Background.....	1
1.2 Main Contributions.....	2
1.3 Outlines of Thesis.....	3
 CHAPTER 2 – LITERATURE REVIEW	 4
2.1 Machine Learning Methods for Missing Data Imputation in Radon Time Series Data	4
2.1.1 Background.....	4
2.1.2 Different Studies Regarding Missing Data Imputation	5
2.2 Missing Data Imputation Methods	6
2.2.1 Predictive Mean Matching and Hot-deck Imputation.....	6
2.2.2 Mean, Median, and Mode Imputation.....	7
2.3 Machine Learning Base Ensemble Methods for Classification and Regression	8
2.3.1 Stacking Ensemble Learning.....	12
 CHAPTER 3 – MATERIAL AND METHODS	 15
3.1 Dataset Details.....	15
3.2 Proposed Approach for the Missing Data Imputation.....	17
3.2.1 Imputation by Feature Importance	18
3.3 Proposed Methodology for Time Series Classification Using Stacking and Automatic Anomaly Indication Function	23
3.4 Performance Evaluation Metrics	26

Chapter 4 – EXPERIMENTS & RESULTS.....	29
4.1 Performance Evaluation of Imputation by Feature Importance for Missing Data Imputation	29
4.1.1 Computing the Complexity of the Imputation by Feature Importance.....	37
4.1.2 Execution time of IBFI with different base learners and other imputation methods.....	41
4.1.2.1 Justification for using the proposed methodology (IBFI).....	43
4.2 Performance Evaluation of Stacking and Automatic Anomaly Indication Function in Classification of Time Series	43
4.2.1 Performance Evaluation of Anomaly Detection at Different Window Sizes	48
Chapter 5 - CONCLUSION	52
5.1 Discussion and Conclusion.....	52
5.2 Applications of the Proposed Methods.....	53
5.3 Future Work.....	54
REFERENCES.....	55
CURRICULUM VITAE.....	62

NOMENCLATURE

AAIF	Automatic Anomaly Indication Function
AUC	Area Under the Curve
CV	Cross Validation
DLDA	Diagonal Linear Discriminate Analysis
DM	Data Matrix
EM	Expectation Maximization
FIM	Feature Importance Matrix
GLM	Generalized Linear Model
IBFI	Imputation by Feature Importance
ID	Impure Data
KNN	K-Nearest Neighbor
MAR	Missing at Random
MCAR	Missing Completely at Random
MCC	Mathews Correlation Coefficient
MICE	Multiple Imputations by Chained Equation
ML	Meta Learner
MNAR	Missing not at Random
MSE	Mean Squared Error
MSLE	Mean Squared Log Error
NN	Neural Network
NPV	Negative Predicted Value
NSAD	Non-seismic Activity Data
PB	Percentage Bias

PD	Pure Data
PDF	Probability Density Function
PMM	Predictive Mean Matching
PPV	Positive Predictive Value
RBF	Radial Basis Function
RF	Random Forest
RICA	Reconstruction Independent Component Analysis
RMSLE	Root Mean Squared Log Error
ROC	Receiver Operator Characteristics Curve
SAD	Seismically Active Data
SGR	Sequential Generalized Regression
SGRC	Soil Gas Radon Concentration
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
SVR	Support Vector Regression
TA	Total Accuracy

LIST OF TABLES

Table 4.1 For 20 percent introduction of missing data of type MNAR and MCAR and all features (RN, TH, TC, RH, and PR) examined, PB and MAPE statistics for IBFI were assessed to various imputation methods (Mean, Median, Mode, PMM, and Hotdeck).....	30
Table 4.2 For 20 percent introduction of missing data of type MAR and all features (RN, TH, TC, RH, and PR) examined, PB and MAPE statistics for IBFI were evaluated against various imputation methods	30
Table 4.3 An approximation of the total number of scans for the certain number of impure samples	40
Table 4.4 Execution time in seconds of imputation by feature importance using different base learners and other imputation methods	41
Table 4.5 The performance metrics yielded after applying the proposed methodology	44

LIST OF FIGURES

Figure 2.1 Basic working procedure of stacking based ensemble learning [82]	14
Figure 3.2 The geographic location where instrumentation was deployed for data collection.....	16
Figure 3.1 Instrumentation and Location for data collection.....	16
Figure 3.3 Simulation plan of the study regarding missing data imputation	18
Figure 3.4: Proposed approach to use base learning algorithm for imputation	19
Figure 3.5 Pseudocode for the imputation by feature importance methodology	22
Figure 3.6 In the imputation by feature importance (IBFI) technique, a proposed methodology for selecting a feature to impute first.....	23
Figure 3.7 Proposed methodology of the training phase for the classification of anomaly.....	24
Figure 3.8 The methodology for the testing phase for the classification of anomaly.....	24
Figure 4.1 The IBFI statistics compared to other imputation methods adjusted to the average statistic for all methods across different features, displaying RMSLE, RMSE, MSE, and MAPE.	32
Figure 4.2 IBFI statistics compared to other imputation methods adjusted to the average statistic for all methods aggregated across all features, revealing percent bias for 10, 20, and 30 percent MNAR, MAR, and MCAR missingness, respectively....	32
Figure 4.3 The reusability of models in respective iterations and scans during the imputation process of 10, 20, and 30 percent for missing completely at the random type of missingness, respectively.....	34
Figure 4.4 The reusability of models in respective iterations and scans during the imputation process of 10, 20, and 30 percent for missingness not at random type, respectively.	35
Figure 4.5 The reusability of models in respective iterations and scans during the imputation process of 10, 20, and 30 percent for missing at the random type of missingness, respectively.....	35
Figure 4.6 The reusability of the model in IBFI for sample features during iteration 1	36
Figure 4.7 The reusability of the model in IBFI for sample features during iterations 2 and 3.....	37
Figure 4.8 The training and testing error at window sizes ranging from 0 to 3.....	44

Figure 4.9 For split at window 0, ROC curve of assessments and outcomes	45
Figure 4.10 For split at window 1, ROC curve of assessments and outcomes	46
Figure 4.11 For split at window 2, ROC curve of assessments and outcomes	46
Figure 4.12 For split at window 3, ROC curve of assessments and outcomes	47
Figure 4.13 The seismic activity classification in the test time series at a window size of 0.....	48
Figure 4.14 The seismic activity classification in the test time series at a window size of 1	49
Figure 4.15 The seismic activity classification in the test time series at a window size of 2.....	49
Figure 4.16 The seismic activity classification in the test time series at a window size of 3.....	50

CHAPTER 1

INTRODUCTION

1.1 Background

In various sectors, especially when human lives are at stake, such as medical diagnosis, earthquake prediction, and so on, the accuracy with which decision support systems (DSSs) forecast or classify samples is of primary interest. Earthquakes are regarded as severe natural disasters because of their unpredictability, which results in the loss of human lives and infrastructure [1]. When it comes to earthquake forecasting, there are two different perspectives. The early thought was that it was an unpredictable event that could not be predicted in advance, while others invested significant resources and effort to make it predictable. Numerous research has been conducted in the past to approach this difficult job from various perspectives [2-7]. The radon time series, one of the precursors, has been shown to exhibit unusual behavior before the occurrence of multiple earthquakes. Numerous studies have been conducted across the globe that uses different precursors for anomaly detection for earthquake forecasting [3, 8-11]. As a result of advancements in computer science, many computationally sophisticated approaches such as machine learning methods for predicting radon concentration from meteorological parameters have been effectively applied [3]. Tareen et al. used three separate computational intelligence models to automatically detect outliers in the time-series data. [8]. In Vietnam, a study was undertaken to optimize the machine learning model known as artificial neural network (ANN) for accurate radon dispersion prediction, and it was shown that ANN performed very well in predicting radon dispersion with lower performance metrics [12]. A layered neural network (LNN) was used to approximate the radon values, which can map the functional link between radon concentration and other parameters [13]. For the definition of premonitory characteristics of low seismic activity, Sikder et al. used the Decision Tree algorithm, which outperformed previous regression-based algorithms [14].

1.2 Main Contributions

In summary, our main contributions are:

- **To propose an efficient methodology that reconstructs the missing data in different features regarding time-series data.**
 - To achieve this objective, a methodology is proposed that is imputation by feature importance (IBFI) that can encase several ML algorithms for filling the missing values in any time series. The methodology uses a feature importance matrix to determine which feature is to impute first in different samples. After testing the methodology for different missingness scenarios across varying missingness percentages, it is now obvious that the proposed approach by employing feature importance proficiently fills the missing values across different features in experimented time-series data.
- **To propose an automatic anomaly classification method based on a machine learning-based ensemble model**
 - To achieve this objective, we proposed a methodology that is based upon the staking ensemble learning approach. Furthermore, the function for the indication of automatic anomaly is also formulated as a post-processing technique that classifies the clumps of incoming readings into their respective categories. The basic theme of this methodology is to categorize the time series data into their respective classes based on the predicted concentration values. The methodology works in two folds; the first fold is concerned with the prediction of radon concentration from its associated predictor variables and further meta-level learning with the predicted radon concentration along with its associated class labels. To smooth the predictions by the meta-level learner, a function for automatic anomaly indication is employed to calculate the percentage of indications for seismic activity and label the clumps of reading as seismic or non-seismic. After experimentation, we have concluded that all of the anomalies are captured by the proposed methodology efficiently.

1.3 Outlines of Thesis

The general organization of our dissertation is as follows. The thesis comprises five chapters. **In chapter one**, an introduction to the importance of earthquake prediction, earthquake precursors such as radon concentration, machine learning methods for anomaly classification is provided. Moreover, the detailed objectives of this whole research work are presented along with the details of how we achieved each of these. The literature review is an important part of every research project since it establishes a link between prior research and the present one. **Chapter 2** includes a general review of the literature regarding missing data imputation in time series, earthquake precursors, classification of time series. The materials and procedures used in the study are detailed in **Chapter 3**. This chapter provides the database used in this study along with its minor statistical and seismic activity details. The mathematical formulation of the ML methods used for experimentation is also provided in detail. Furthermore, this chapter is divided into subsections concerning each of the objectives discussed above. This chapter exclusively presents the methodological details as well as simulation plans of each of the individual studies carried out in subsequent sections. **Chapter 4** includes the presentations of results to assess the efficacy of individual proposed methodologies developed to achieve the desired objectives. Compared to chapter 3, this chapter is also divided into subsections to present the results and discussion about each methodology in detail. **Chapter 5** describes the general discussion of all the results, the major findings of the research, the general conclusion, and future directions.

CHAPTER 2

LITERATURE REVIEW

The literature review is one of the most imperative parts of any research that synthesizes the concept of the study. It gives a comprehensive account of research conducted on a certain issue, highlighting the study domain's history and context. The literature review unveils a suitable theoretical edifice of the study, improves the research design and instrumentation, which were not efficacious in the previous research evidence. This chapter includes a thorough overview and literature review of the strategies established for time series categorization and missing data imputation.

2.1 Machine Learning Methods for Missing Data Imputation in Radon Time Series Data

This section provides in-depth literature regarding the background and frequently used missing data imputation methods.

2.1.1 Background

Many researchers face the challenge of missing or incomplete data in their scientific investigations, which can result in inaccurate analyses if the missing data is not adequately imputed. Human error, machine malfunctioning or failure, and routine maintenance are only a few of the causes that result in missing or lost data. [15]. The mechanism that generates the missing or incomplete data can be classified into different categories [16]. Missing not at random (MNAR), missing at random (MAR), and missing completely at random (MCAR) are the three categories of missingness. In the situations where the chance of missing data is similar in all cases, the data which is missing is designated as missing completely at random (MCAR). This suggests that the incompleteness of data is not relevant to the data itself. Missing at random (MAR) happens in situations when a data point's likelihood to be unavailable is not connected to missing data points but to the available data points.

When it comes to missing data points that are not at random, there are two possibilities. The value of the hypothetical data point determines the missing data point e.g. the employees whose bonus is very high generally do not reveal it during some conducted surveys. Furthermore, missingness is linked to other data variables, such as males' aversion to disclosing their married status, which has resulted in missingness in several surveys due to the gender variable.

2.1.2 Different Studies Regarding Missing Data Imputation

Simple procedures are often used to impute missing data. Missing indicator methods, complete case analysis, mean, median, and mode incomplete data imputation are only a few examples. Unfortunately, these methods can lead to highly skewed results and ineffective investigations [17-18]. Multiple imputation is a more advanced way of dealing with missing data that outperforms existing methods [19-22]. Multiple imputation analyses, on the other hand, have some drawbacks [23]. Multiple imputation produces implausibly low or even negative numbers when dealing with heavily skewed data. When an analysis is required to investigate the relationship between a dependent (response) and various independent (predictor) attributes, unavailable values in the dependent or response variable affect the dependent variable being ignored in the data filing method. The omission of the dependent variables in such a way erroneously diminishes the relationship between independent and independent attributes. Furthermore, the multiple imputation techniques are expensive in computation, and some methods must be run numerous times for better estimation, and the procedure's length grows concerning the increase in missing data. To recreate incomplete and unevenly recorded time series data for radon concentrations, machine learning approaches were also applied [9]. According to a comparison of conventional statistical and machine learning data imputation procedures utilizing available controls, machine learning methods excelled at statistical methods and markedly improved prognosis accuracy. [24-25]. To impute the missing data in spatiotemporally daily time series precipitation records, Mital et al. [26] suggested an imputation method based on a sequential approach. By integrating the proposed imputation approach with an interpolation (spatial) relying on a Random Forest algorithm, the authors proved that as the locations with

unavailable data rise, the suggested sequential imputation method delivers a lot of benefits. When the amount of stations with records having incomplete data decreases, however, the successive imputation strategy provides no additional spatial information. The semi-parametric regression based on stochastic for imputation was proven to be better when compared to the current imputation method based on semi-parametric regression. The stochastic approach works well for both real and simulated data [27]. A useful imputation-based approach for multivariate time series data was also presented, which makes use of an expectation-maximization (EM) methodology and assumes a normal distribution [28].

2.2 Missing Data Imputation Methods

This section provides the details of frequently used imputation methods for filling the unavailable data points in different features of the experimental dataset.

2.2.1 Predictive Mean Matching and Hot-deck Imputation

Predictive mean matching (PMM) [29-30] is an excellent methodology for multiple imputation for the reconstruction of unavailable or missing data, especially when the quantitative variables are not normally distributed. It's the most up-to-date method for multiple imputation based on a hot deck procedure [31]. Furthermore, because these estimated values are original values acquired from individuals with original data, the imputed values in the missing features have the same bounds (0 to 50 for example) or skewness (positively or negatively skewed). Although predictive mean matching (PMM) has been around for a long time [30, 32], its extensive and practical applications have only just begun. PMM's first use was constrained, in that it could only manage a particular feature with missing data. Simply put, its applicability was confined to situations with missing data having recurring patterns. The PMM methodology is now included in several software programs that use a variety of imputation techniques. The methodologies included sequential generalized regression (SGR), completely conditional specification (FCS), or multiple imputations using chained equations (MICE). Hot deck imputation's fundamental concept is based on the selection of one acceptable value of the donor from the available instances that, in some ways, are similar to the missing case [33]. As far as

PMM is of concern, to match potential donees and donors, the proximity of projected means is considered. Furthermore, within the traditional PMM method, a case is chosen from a pool of k examples with anticipated values that are closer to the missing case's anticipated value. Furthermore, the experimental value of this donor case is used to fill in the missing value. Traditionally, the term "hot deck" has been associated with the usage of punch cards for data storage in the past, as well as the donor deck available to a non-respondent. When dealing with missing data, hot-deck imputation substitutes the unavailable value with a realistic response from a "comparable" responding unit, meaning that both the intended unavailable values and those from a "comparable" responding unit are replaced. The hot deck imputation approach replaces missing values for a non-respondent with immediate values received from a responder when multiple variables are missing. The respondent is referred to as the donor, whereas the non-respondent is referred to as the recipient. In terms of features observed in both situations, the donor and the recipient are similar. The donor or respondent is chosen at random as of a pool of potential donors, known as the donor pool, and these methods are known as random hot deck methods. There are additional variations of this method in which there is only one donor and the numerical values are substituted from that case, usually, the "nearest neighbor" which is supported upon some performance measure; those types of techniques are referred to as deterministic hot-deck methods because the donor selection is not random. The hot-deck imputation, on the other hand, has a flaw in that it requires good respondent matches or donors to receivers based on the information available which is covariate. In short, the hot deck is a commonly used method to handle missing values due to its advantages like imputing realistic values, avoiding strong parametric assumptions, incorporating covariate information, and providing good inferences for both linear and nonlinear statistics if one pays proper attention to propagating imputation uncertainty.

2.2.2 Mean, Median, and Mode Imputation

The incomplete data imputation through the mean model is a technique of imputation in which the mean of all available cases of a variable is used to replace missing values in that variable. Furthermore, aside from its ease of use, the mean model

decreases data variability, ensuing in an overestimate of variance and standard deviation. Median imputation, on the other hand, is the number in the middle of the observed values for a variable when they are organized from lowest to highest. As we know, the most often occurring value is known as the mode in specific variables and serves as a substitute for missing data. Because radon concentration data is made up of continuous attributes, no two values will be precisely the same. To generate a continuous probability density function (PDF) estimate, kernel density estimation is employed for mode. Furthermore, in a continuous probability distribution, the place where the probability density function achieves its higher value has designated a mode of that continuous variable.

2.3 Machine Learning Base Ensemble Methods for Classification and Regression

The ensemble is a collection of several models in machine learning and statistics, and it is the most self-efficient method when compared to other traditional models [34]. Supervised learning algorithms are the most useful tasks in searching through given different hypothesis spaces to get predictions along with suitable hypothesis space for certain problems. Even a given hypothesis space contains well-defined different hypothesis spaces but it can be difficult to find optimal ones. The Ensemble approach aggregates several hypotheses to find the best one. With the aid of an amalgamation of weak learners, the ensemble approach is employed to produce a strong learner. While performing classification using ensemble methods, more computations are performed as compared to a prediction with a particular model so multiple models can be a way to help poor algorithms for performing well after doing extra computations. The ensemble method is also an example of supervised learning as firstly it trains and then makes predictions and it represents a single hypothesis space. Even when there is a lot of variation amongst the models, ensemble approaches offer more accurate findings in different experimentations. Machine learning has been shown in studies to help scientists in a variety of fields do everyday jobs more efficiently [35-36]. The ensemble technique has been extensively employed in research involving regression, pattern categorization, and forecasting in time series problems [37-38]. From the collection of given hypothesis space with well-defined

hypotheses, it can be difficult to find the best one. Using multiple hypothesis ensemble techniques provides the best hypothesis. When the machine learning system is accurate and at the same time makes an error on different examples then the ensemble method performs much better [39-42].

To progress the overall performance accuracy by merging many learning methods, ensemble learning is a very efficient technique that has been adopted in numerous studies [34]. To decrease the latent for overfitting the training data major benefits of ensemble techniques are to lessen the effects of the curse of dimensionality problem by incorporating and averaging over many weak learners [43]. As far as we are dealing with this technique, this is decisive for most of the biological applications having a very minimum sample size by proficiently using the training dataset. For those problems where the dimensions of the data are very high, random forest [44] is considered to be a famous ensemble method that generates many trees to make final predictions based on an aggregation of those trees or models.

The major objective of using the ensemble method is to get improved classification as well as generalization which means it works well on training as well as on unseen data. By combining the results of the group of classifiers, enhancement in classification tasks is mostly obtained by base classifiers as an ensemble and predicting unseen data in a consensus way. As far as we achieve this sometimes we have to face improvement in model complexity which means there is a decrease in interpretability of the model [45]. Using bias-variance decomposition analysis in the ensemble technique, a better generalization property is obtained [46]. Previous studies prove that bagging by decreasing variance [47] improves generalization however, boosting accomplishes this by reducing bias [48]. Numerous learners are constructed by adding some randomization to the learning process or altering input features in circumstances where only one learning algorithm is used a base learner (bagging and boosting) [49]. Stacking [50-52] varies from bagging and boosting in that it takes into account heterogeneous base learning methods and, more crucially, does meta-learning to aggregate classification results from the first-level learning methods. A collection of trained classifiers makes final predictions using weighted or majority voting in bagging and boosting; however, predictions are made from independently learned classifiers and forwarded to a second-level learner for final

classifications in stacking. The efficacy of stacked regression was demonstrated [51] by utilizing first-level learners who have used linear regression models with varying numbers of variables and a meta learner or second-level learner using the least-square linear regression model. Furthermore, all regression coefficients were non-negative in this second-level learning, which is a fundamental condition for assuring that ensemble learning via stacking beats fits the standalone learner. When performing the stacked classification, it was mentioned that the attributes that are supplied as newly constructed meta training data, as well as the choice of second learning or meta learning algorithms, are important considerations [50]. Furthermore, a study was conducted that suggested to utilize the class probabilities rather than crisp labels of class to build the meta attributes for new meta learning dataset because the probability for each class allow for accounting for the certainty of individual classifiers' classifications as well as predicted class labels, as well as it bring into play of multi-response linear regression (MLR) as a meta level learning algorithm [53]. The results of a comparison of stacking and Bayesian Model Averaging (BMA) revealed that stacking is more robust than Bayesian Model Averaging (BMA) [54]. Furthermore, empirical data shows that BMA is prone to model approximation inaccuracy. Different ensemble methods, including bagging and boosting, have been widely used in various sectors. Many studies [39, 55] conducted in the 1990s employed ensemble learning to demonstrate that accurate learning can be implemented utilizing a large number of weak learning algorithms. Furthermore, a strategy was presented that increased classification ability by considering diverse data transformations [56]. In the comparison of individual hypotheses with regression or classification-related studies, the literature demonstrates that using an ensemble of simple combinations of hypotheses to produce classifications or predictions enhances the ability toward designated tasks. Ensemble learning has also been employed for energy time series forecasting tasks: for example, an extreme learning machine (ELM) algorithm is presented and effectively implemented on Australian National Electricity Market data [57]. In wind data on natural clusters, the models built on regression parameters have also been utilized to estimate wind power using a cluster-oriented ensemble framework [58].

As a result, as computer science has progressed, many computationally sophisticated algorithms have been effectively used to forecast radon concentration from its associated predictor features [3]. Regression or model trees were used to forecast the radon concentration through environmental data and concluded that the prediction error increases a week before the earthquakes having a magnitude ranging from 0.8 to 3.3 [59]. Tareen et al. employed three different computational intelligence models to autonomously detect irregular behavior in radon time-series data by modeling the radon concentration with different statistical parameters [8]. The findings of the study reveal that the uneven behavior of radon is caused by anomalies. A study was conducted to optimize the learning model for the correct forecast of the dispersion of radon in Vietnam and concluded that ANN performed very well to predict radon dispersion with the lower values of performance metrics [12]. Machine learning entails the research and development of algorithms that can learn from and predict data. It is a branch of artificial intelligence concerned with the development and research of algorithms capable of generating models for predictions or categories [60]. With the advent of technology, machine learning methods have shown significant results in various fields of study such as medical diagnosis [61-66], banking [67-68], market basket analysis [69-70], and many more. A variety of methods are offered by machine learning such as Diagonal Linear Discriminate Analysis (DLDA) [71], k-Nearest Neighbors-kNN [72], Support Vector Machine [73], and Random Forest [74] for classification and prediction purposes. Moreover, ensemble methods were also proposed such as bagging [75], boosting [76-77], and stacking [50] where the final prediction is not made by only a single model but by aggregating the outcome of various weak learned models [34]. Ensemble methods show significant improvement in performance than individual models in classification and prediction problems [78-81]. The improvement in the performance by employing ensemble methods is based upon the premise that classification or prediction made by the ensemble or collection of classifiers is more accurate than relying on the individual classifier that constituted the ensemble [34].

2.3.1 Stacking Ensemble Learning

Stacking is a type of ensemble model in which another model is fitted using the combined predictions (labels or probability-based) of two (or more) prior models. Each consecutive layer receives the model predictions as inputs, which are then merged to generate a new set of predictions. Stacking [50-52] is considered an ensemble method where an independent learner is trained to combine the predictions of individual learners. The learner used to aggregate the predictions from individual learners is called Meta learners or second-level learners while the individual learners themselves are called first-level learners. It is a two-fold process in the first fold; individual learners are trained over the same data distribution whilst in the second fold a Meta learner is generated upon the predictions taken from first-level learners. Also, original labels are considered as labels of training data and input features are considered as outputs of the first level learners. Different approaches are used to generate first-level learners and for this purpose, often stacked ensembles and homogeneous ensembles based on the stacking approach are used. The Pseudocode of stacking is given below in figure 2.1.

Stacking can perform in multiple ways. On one side, it can be seen as a specific combinational learning method while on the other side, it behaves as a generalized framework of different ensemble methods. Using the first-level classifier, a new dataset is generated during the training phase. If we use the same data for training the first-level learners as well as second-level learners, different issues arise during the training phase of stacking like overfitting. Therefore, multiple approaches are used to overcome this problem like cross-validation or by excluding those instances for the first-level learner from training examples that are generated for a new dataset, or the leave-one-out procedure. For instance, choose k -fold cross-validation. The training dataset D is divided into k equal parts $D_1, D_2, \dots, D_k$. Define D_j and $D(-j) = D \setminus D_j$ as train and testing sets of j^{th} fold. Given L learning algorithms, including first-level learner $x_p(-j)$ is taken by appealing $D(-j)$ on the learning algorithm. For each a_i in D_j , the test set of the j th fold, assume c_{iT} represents the learner's output $x_p(-j)$ on a_i . Finally, when the whole cross-validation process is completed, using L individual learner a new dataset is generated as

$$D' = \{(c_{i1} \dots \dots c_{iT}, b_i)\}_{i=1}^n \quad (2.1)$$

upon which a second-level learning algorithm is applied, for y the function $(c_1 \dots \dots c_T)$ uses the resulting function 'h'. By training the entire training data the resulting first-level learners are re-generated after which the new dataset is generated. Stacked regression was first demonstrated by Breiman [51]. The author used different numbers of variables with linear regression models and regression trees of different sizes as the first-level learners while using the rule that the coefficients used in regression are non-negative declaring least square regression models as second-level learners. The non-negativity rule of stacked ensemble affects the performance and it was found that the performance can be improved while choosing the single best learner. Wolpert demonstrated that it is very difficult to choose the variety of instances for meta-features data and different types of methods for meta-level learning [50].

The use of class probabilities is recommended for new data rather than crisp class labels as features because it is useful not only for the confidence of individual classifiers but also for the calculation of predictions [53]. As far as the first-level classifier is discussed, that can result in class probabilities, the output of the classifier x_k on an instance a is $\left((x_{k1}(a) \dots \dots x_{k1}(a)) \right)$, over class labels $\{c_1, \dots, c_1\}$, which is considered as a probability distribution, whereas for instances taken from class c_j , the predicted probability is x_k , denoted by $x_{kj}(a)$. Although with the highest class probability $x_{kj}(a)$ as the class label, x_k predicts only class c_j , obtaining probabilities with useful information. As a result, from all first-level classifiers the probabilities on a , with true class labels generate new training examples for a second-level learner. Figure 2.1 below shows the schematic of the stacking framework.

Input: Training Dataset $D = \{(a_1, b_1), (a_2, b_2) \dots (a_n, b_n)\}$;
 Initial-level learning algorithms $L_1 \dots L_T$
 Second-level learning algorithm L .

Process:

```

1. for  $p = 1 \dots T$ :
2.    $x_p = L_p(D)$ ;
3. end
4.  $D' = NULL$ ;
5. for  $j = 1 \dots n$ :
6.   for  $p = 1 \dots T$ :
7.      $c_{jp} = X_p(a_i)$ ;
8.   end
9.    $D' = D' \cup ((c_{i1} \dots c_{iT}), b_i)$ ;
10. end
11.  $x' = L(D')$ ;
Output:  $P(a) = X'(x_1(a), x_2(a) \dots x_T(a))$ 

```

Figure 2.1: Basic working procedure of stacking based ensemble learning [82]

CHAPTER 3

Materials and Methods

In this chapter, the general structure of our proposed approach is explained. Moreover, we will talk about the detailed methodology of our proposed approaches.

3.1 Dataset Details

The dataset used in this experiment was collected (see Figure. 3.1) from Muzaffarabad, Azad Jammu & Kashmir, Pakistan (See Figure. 3.2). To record radon data, RTM 1688-2 (radon station) was installed seismically active area which is located in Muzaffarabad, a capital city of Kashmir that is under the control of Pakistan, at the latitude and longitude of 34.39621 and 73.47347 respectively. Data is collected for more than a year ranging from the 1st of March 2017 to the 11th of May 2018. Furthermore, the recorded data as well as further instrumentation information have been published elsewhere [5]. During the data collection period, the minimum and maximum observed RN values were 13743 Bq/m³ and 28085 Bq/m³ respectively. The mean concentration of 21364 Bq/m³ and median value of 21569 Bq/m³ were observed.

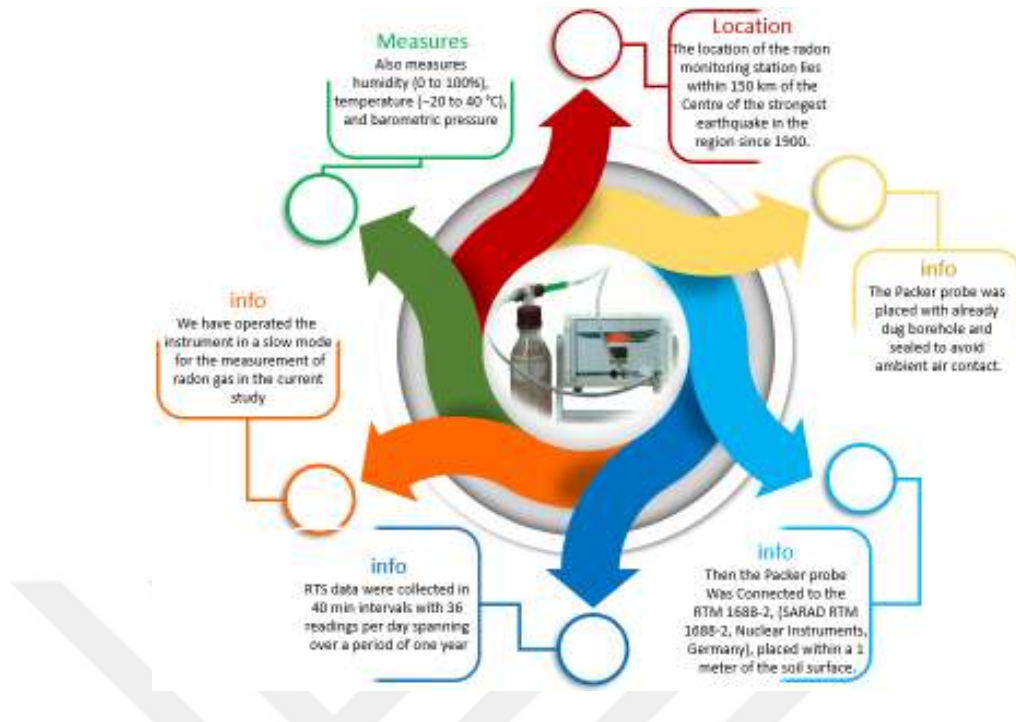


Figure 3.1: Instrumentation and Location for data collection

account when imputing missing patterns in different time series. On the other hand, in anomaly classification concerns, the RN is considered a predictor feature while others as response features.



Figure 3.2: The geographic location where instrumentation was deployed for data collection

3.2 Proposed Approach for the Missing Data Imputation

The methodology employed for the imputation of missing data is shown in Figure 3.3. The three various patterns of missingness (MNAR, MAR, and MCAR) are introduced artificially into the dataset to analyze the imputation algorithms, resulting in incomplete experimental datasets with 10 to 30% of the missingness of data in different features. The introduction of missing values is deliberately being made in the dataset with the help of the R program "mice" [83]. The missing patterns, which are a mixture of different features in the dataset with missing and available values, are the basis for introducing missing values in multivariate datasets [84]. Figure 3.6 depicts the missing patterns along with their frequency. Based on n missing data patterns, the entire dataset is randomly partitioned into k subsets. As far as the frequency vector is of concern, a frequency of a specific pattern that is missing in the entire dataset, determines the subset size. The data samples in the subsets are deemed candidates for missing due to a variety of variables, including the missingness mechanism. All data samples in the subgroups enclose an equivalent chance of being missing in MCAR situations, however, in MAR and MNAR missingness scenarios, the weighted sum scores are determined. To put it another way, a linear regression equation yielded the weighted total scores, and the values of the scores determine whether or not a candidate's data rows are missing. Finally, as per the pattern of missingness in data and the chance of these being missing, the samples in the data are made incomplete by the formation of unavailable values. These subsets are merged after missing values are introduced to create a complete dataset having missing values in separate data samples.

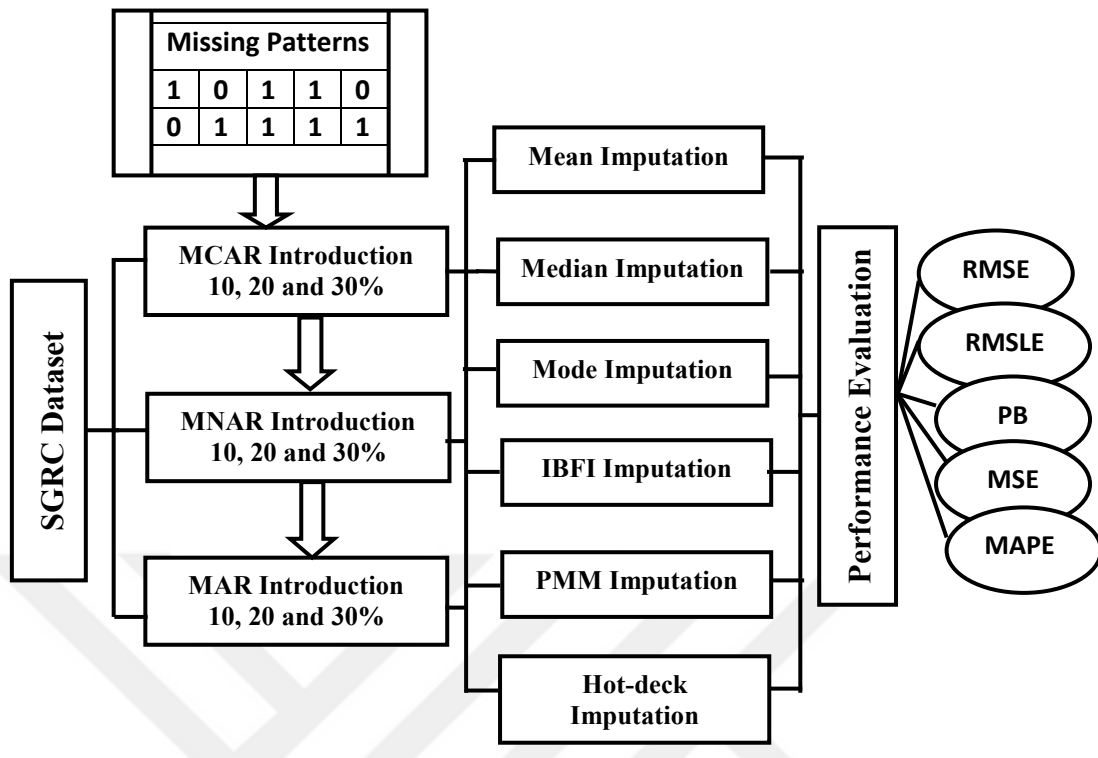


Figure 3.3: Simulation plan of the study regarding missing data imputation

Six alternative data imputation approaches are used to treat the nine modified SRGC data sets. IBFI, as well as the more prevalent mode, median, mean, hot-deck, and predictive mean matching (PMM) approaches, are among them. Root mean squared log error (RMSLE), mean squared error (MSE), root mean squared error (RMSE), percentage bias (PB), and mean absolute percentage error (MAPE) are some of the performance measures derived after using the imputation approach. The capacity of the method for imputation to fill missing values that are substantially closer to the original values for each of these criteria is critical to the approach's performance. In the end of this chapter, the description of the performance measures as well as the imputation methods.

3.2.1 Imputation by Feature Importance

Figure 3.4 shows a graphic presentation of this newly proposed imputation method for filling missing data, while Figure 3.5 shows the coded algorithm in pseudo-code configuration. The IBFI methodology begins with a data matrix (DM) including different attributes. The missingness of values of the patterns indicated in Figure 3.6

is represented in the data matrix (DM) by three categories of missing values (MCAR, MAR, and MNAR).

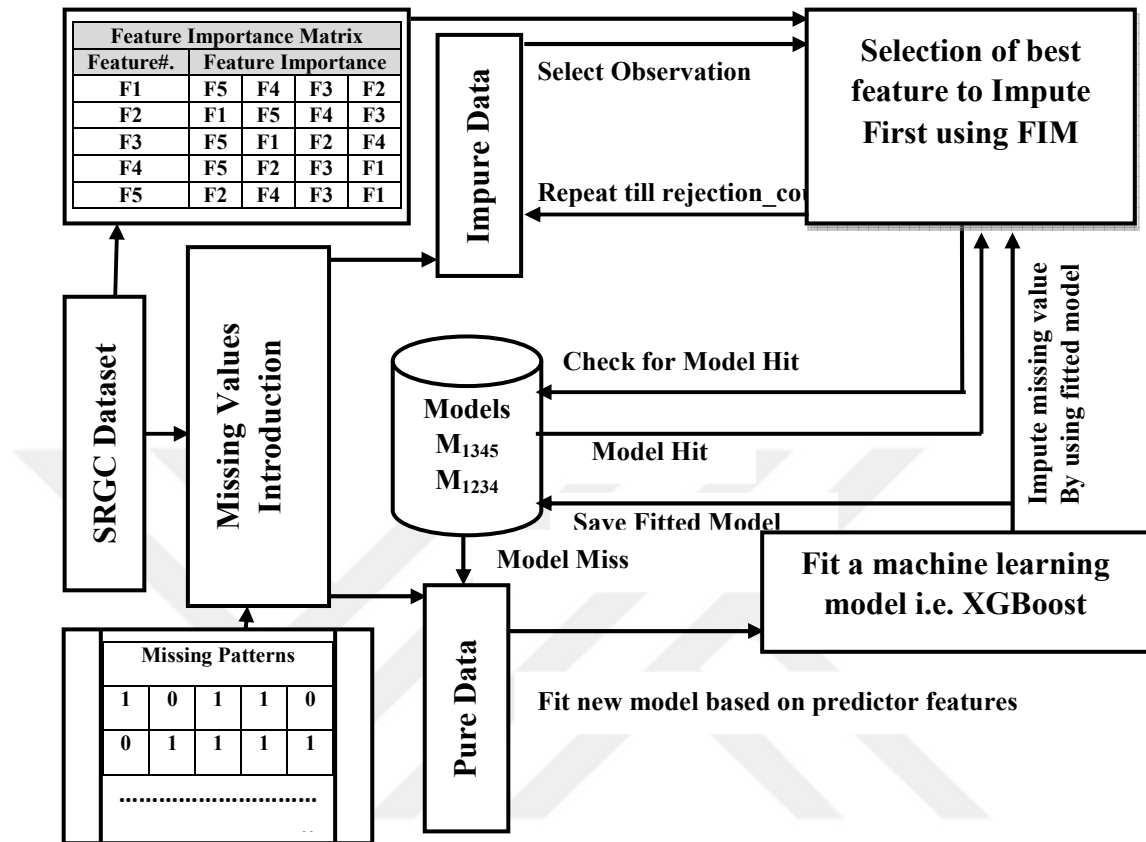


Figure 3.4: Proposed approach to use base learning algorithm for imputation

The original time series dataset is separated into two subsets in the first step: Impure Data (ID) and Pure Data (PD). The pure data is made up of the data rows from the dataset having all feature values available while an impure dataset is comprised of data samples that have one to 'n' number of missing features per sample. The missing values which get predicted are continually imputed to their relevant locations as values are predicted for certain samples during the IBFI procedure, filling the missing patterns based on the calculated importance of individual features. The R package "randomForest" is used to create the feature importance matrix [85]. For each feature in the dataset, the importance of individual features is calculated by taking a response and others as predictors. The attributes are arranged concerning their importance, first with the most significant and ending with the least important. The Impure data is utilized to impute the unavailable values for a certain attribute. This is performed by employing a machine learning-based learning algorithm built

on non-missing attributes to estimate the value of missing attributes. Take a look at the different attributes $F_1, F_2 \dots F_n$. If F_1 has a missing value, the remaining characteristics $F_2 \dots F_n$ will be utilized to train a statistical or machine learning based mathematical model. Finally, the trained model is further used to forecast the non-available values in F_1 .

If F_3 is the feature having missing values, the features $F_1, F_2, F_4 \dots F_n$ will be utilized to train any statistical or machine learning-based mathematical model, and F_3 will be forecasted from the model that has been trained upon the predictor features. The missing values get replaced by these projected values. The complexity of a sample increases when there are multiple missing values and specific features in the dataset have significant dependencies on each other. Assume that the dataset has the five(05) features F_1, F_2, F_3, F_4, F_5 , and that some samples have missing values in F_1 and F_5 . Further, the features in the DM have a high correlation with each other. Let us consider, F_1 and F_5 each contains feature importance values with additional attributes ranging from F_5, F_3, F_4, F_2 and F_2, F_4, F_1, F_3 from highest to lowest. The attributes F_2, F_3, F_4, F_5 and F_1, F_2, F_3, F_4 will be utilized to train the model for the samples where F_1 and F_5 have unavailable values, respectively. When both F_1 and F_5 contain missing values in distinct samples, the imputation process for a machine learning model gets more challenging. In this circumstance, the only features which are available for training are F_2, F_3 , and F_4 . This is because the prediction of the missing value of F_5 requires the immediate value for F_1 . Furthermore, to proficiently fill the unavailable value for F_1 and F_5 , it is necessary to decide whether F_1 or F_5 should be imputed first. The most important feature in order to predict the unavailable value of F_1 is F_5 by analyzing the correlation vector above, whereas F_2 is the essential feature to predict the unavailable value of F_5 in the correlation vector above.

The attribute F_5 must be imputed first based on feature importance, and once the value of F_5 is available, that value will be utilized to anticipate the value of F_1 . At first, there may be gaps in data for many features. The best feature to fill is selected first, and then the unavailable values for that feature are predicted and placed. After this gets accomplished, there will be one less feature with incomplete data.

Following the completion of the first procedure, the best feature to fill is determined, and the unavailable values are filled at their respective locations. The methodology is continued until all the unavailable values in different features are predicted and imputed.

By computing the variable significance for each feature in the database and arranging their importance values in descending order, the FIM determines the order in which attributes are selected for imputation. IBFI requires a termination because it is an iterative approach. The rejection threshold, also known as the maximum limit of the unavailable values per data sample, is determined for this purpose. Because several features are missing at the same time in certain data samples or rows, the threshold of rejection is the amount to the extent we desire to fill in the unavailable values in each row, e.g. $F1$ and $F4$. If the value of the rejection threshold is set to 3, that indicates that if a sample has more than three missing values, that sample will be ignored and discarded, and all other data rows having missing values will be predicted and filled in their missing places during scans. The algorithm operates in a way that, all the data samples having the number of unavailable values of three(03) gets decreased to missing at two(02) per data sample after the completion of the first iteration, as illustrated in Figure 3.6. The completion of second iteration outcomes in the missing at two per sample gets decreased to missing at one per sample. The subsequent one per sample missingness will be imputed and results in a fully imputed dataset for all data samples except those that have a count of missing values that is above the value of the threshold designated for the rejection of samples. Moreover, by keeping the trained machine learning models that are generated right through succeeding scans and iterations, the methodology makes it asymptotically superior.

```

Input:
DM( $X_1 \dots X_n$ ) = Data Matrix where  $X_1 \dots X_n$  contains the unavailable values
RejectionThreshold = The extent to which the number of features in each sample gets imputed
FIM = Feature Importance Matrix having feature importance for each feature in decreasing order
BM = Base Model
Set
ModelList = NULL
U  $\leftarrow$  sequence (1 to number of features)
PriorityMatrix  $\leftarrow$  NULL
k = 1

Procedure:
1. Divide DM in two subsets i.e. PD( $X_1 \dots X_n$ ) and ID( $X_1 \dots X_n$ ) ,
   where PD and ID are the pure and impure data respectively
2. while k <= Rejection_Threshold
   a. For scan_index = 1 through rows in ID
      i. NAsVector = which(is.na(ID[scan_index,]) == TRUE) // to find the indices containing missing value
      ii. If length(NAsVector) > Rejection_Threshold OR length(NAsVector) == 0
          1. Discard that sample
      iii. Else
          1. TrainVector = setdiff(U, NAsVector)
          2. Priority_Matrix = FIM[NAsVector,]
          3. Priority_Matrix[which(Priority_Matrix == TrainVector)] = 0
          4. Priority_Matrix[which(Priority_Matrix != 0)] = 1
          5. For n in 1 to RowCount(Priority_Matrix)
              i. if (Priority_Matrix[n,] == 1)
                  1. Loc_Vector[l_c] = which(Priority_Matrix[n,] == 1)
                  2. l_c = l_c + 1
          6. IF length(Loc_Vector) == 0
              a. indice_to_train = NA_vect
          7. Else
              a. Priority_Mat = concatenate (Priority_Mat, Loc_Vector)
              b. max_ind = which (Loc_Vector == max(Loc_Vector))
              c. max_ind = max_ind[1]
              d. indice_to_train = NAsVector[max_ind]
          8. End if
          9. Trainingdata = PD[, TrainVector]
          10. Trainclass = PD[, indice_to_train]
          11. trainF = Concatenate Column(Trainingdata, class = trainclass)
          12. ModelName = concatenate(indice_to_train , TrainVector)
          13. Flag = Check_Model(ModelList, ModelName)
          14. IF flag == -1
              a. Train a ML model i.e. BMMName and insert model to ModelList
              b. Testdata = ID[scan_index, TrainVector]
              c. Val = predict(model, testdata)
              d. ID[scan_index, indice_to_train] = val
          15. Else
              a. print("Saved Model is Found")
              b. testdata = ID[scan_index, TrainVector]
              c. val = predict(Model_List[[flag]], testdata)
              d. ID[scan_index, indice_to_train] = val
          16. End if
      iv. End if
   3. k = k + 1
   4. End

Function: Check_Model (ModelsList, ModelNames):
1. For i in 1 to length(ModelLists)
   a. IF ModelLists[i] == ModelNames {
       i. Flags = i
   }
2. Return flags

```

Figure 3.5: Pseudocode for the imputation by feature importance methodology

Models are saved in memory in a way that if $F1$ is a response variable and $F2$ and $F3$ are predictor variables, the machine learning learned model is saved as M_{123} . Afterward, for example, incomplete at three variables are reduced to missing at two variables, and $F2$ must be trained again using $F2$ and $F3$; instead of fitting a new model, the already stored model M_{123} will be utilized to fill the missing data for $F1$.

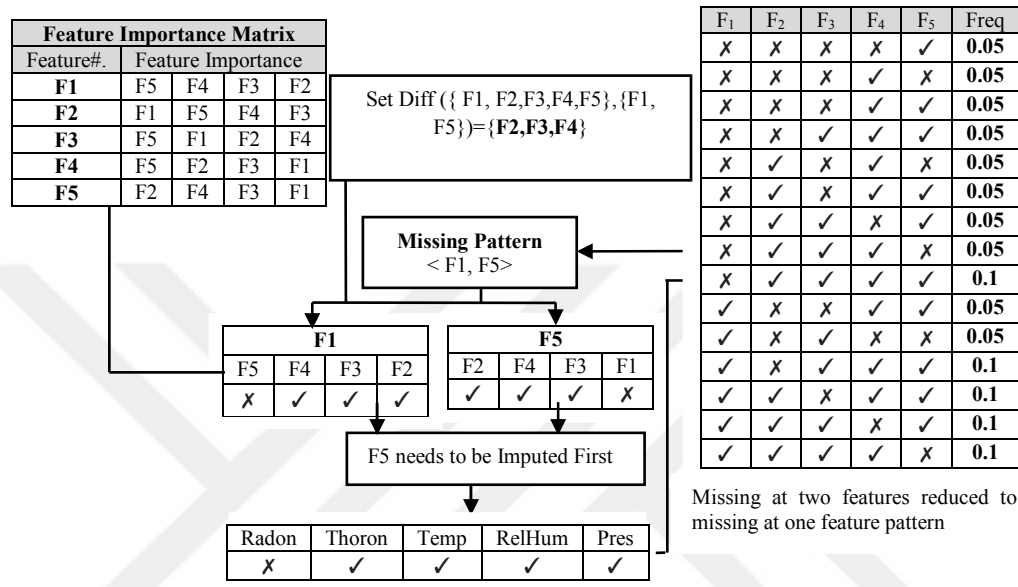


Figure 3.6: In the imputation by feature importance (IBFI) technique, a proposed methodology

3.3 Proposed Methodology for Time Series Classification Using Stacking and Automatic Anomaly Indication Function

The first phase involved manually classifying the time series data into respective classes (anomalous and normal) based on the recorded anomalous events. Figures 3.7 (Training phase) and 3.8 (Testing phase) demonstrate the whole simulation design for anomaly classification.

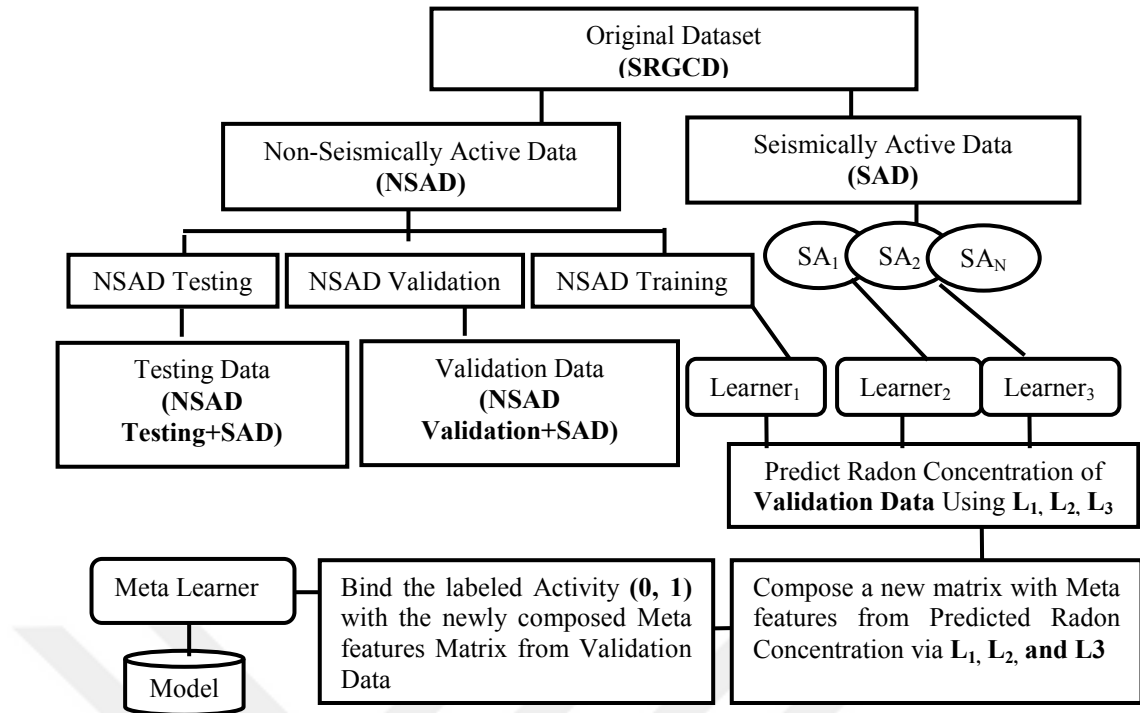


Figure 3.7: Proposed methodology of the training phase for the classification of anomaly

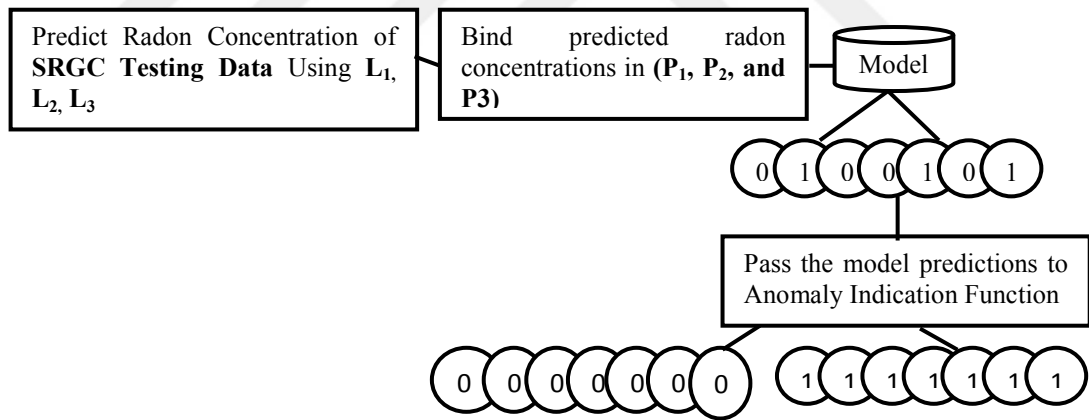


Figure 3.8: The methodology for the testing phase for the classification of anomaly

Figures 3.7 and 3.8 show the proposed method for automatic classification of an anomaly in time series data. The SRG concentration data is manually labeled to respective categories, as previously stated. The data is separated into two subsets once it has been processed: normal or non-seismic data (NSAD) and anomalous or seismically active data (SAD). Non-seismic data rows are those in which no earthquake or anomaly was reported, whereas seismic or anomalous samples are those in which an

earthquake was reported. The categorization of time series data into seismic (anomalous) and non-seismic (normal) in the current study is solely based on radon concentration. There are two phases to the current study methodological approach: training and testing. Non-seismically active data (NSAD) is additionally separated into validation, training, and testing to proficiently partition both classes into the training and testing phases. validation data (SAD+ NSAD validation), Training data (SAD + NSAD training), and testing data (SAD + NSAD testing) make up the entire training and testing data.

The stacking strategy has been utilized to learn from both normal and anomalous patterns, considering the efficiency of the value of the ensemble-based approach for its improvement of ability towards classification and regression problems. Learner₁ (Generalized Linear Model (GLM)), Learner₂ (Linear regression), and Learner₃ (*K*-Nearest Neighbors (*k*-NN)) have been used in this study as a base learner (BL) to learning algorithms to train from anomalous and normal activity data using 5 times 5 fold cross-validation (5×5 CV).

$$\mathbf{BL}_1 = L_1(\mathbf{NSAD}), \mathbf{BL}_2 = L_2(\mathbf{SAD}_1), \mathbf{BL}_3 = L_3(\mathbf{SAD}_2) \quad (3.1)$$

The same learned models are employed to predict the values for validation data after the base-level learning. As Meta features, the forecasted radon concentration is merged into a new data frame. Furthermore, the newly built matrix is coupled to the seismic activity labeled feature. Further, the meta-learner model (ML) has been trained upon meta feature matrix (MF) to categorize radon concentrations into seismic (anomalous) and non-seismic (normal) categories.

$$mf_{it} = BL_{1:t}(\text{ValidationSet}_i) \quad i = 1:n, t = 1:3 \quad (3.2)$$

$$\mathbf{MF}' = \mathbf{MF}' \cup ((\mathbf{mf}_{i1}, \mathbf{mf}_{i2}, \mathbf{mf}_{i3}), \mathbf{y}_i) \quad (3.3)$$

$$\mathbf{ML} = \mathbf{SVMR}(\mathbf{MF}') \quad (3.4)$$

In the categorization of testing data, every learner from 1 through 3 predicts the concentration for all data samples. The estimated concentration values are aggregated (P1, P2, P3) and categorized as 0 (NSA) or 1 (SA) by the Meta learner.

$$H(\text{TestSet}_i) = ML(BL_1(\text{TstSet}_i), BL_2(\text{TstSet}_i), BL_3(\text{TstSet}_i)) \quad (3.5)$$

Furthermore, because many samples containing varied measured radon concentrations are obtained each day, complete dependence on the classification labels allocated by the Meta learner is not feasible. The Meta learner may have been assigned $\{0,1,1,0,1,0,0,1,0 \dots \dots 1\}$ class labels to data samples in a clump of one-day readings, but that type of result cannot be used to determine whether the following series is seismic or not. We've presented an automatic anomaly indication function (AAIF) for this purpose, which calculates the proportion of samples having anomalies in a series of non-anomalous and anomalous data samples. The complete day samples are classified as seismic and assigned the class label of '1' (SA) if the percentage of indication reaches the indication factor (IF=0.55). As a result, the AAIF is solely responsible for assigning the final categorization labels to the following time series.

$$\text{Anomaly Classification} = \text{AAIF}(H(\text{TstSet}_i)), i = 1:n \quad (3.6)$$

To classify the received predicted values at various time windows before and after the earthquake, the time series data is put to the test in a variety of scenarios using a time window that spans the days leading up to and after the anomalous event. One day earlier than and subsequent to the seismic event is defined as a window size of one. Samples taken two and three days before and after earthquake activity are referred to as window sizes 2 and 3, respectively.

3.4 Performance Evaluation Metrics

The five statistical parameters listed below are computed to evaluate the efficacy of the models for the purpose of reconstruction of unavailable values in different features as per missing patterns: The root mean squared log error (RMSLE), root

mean square error (RMSE), mean squared error (MSE), mean absolute percentage error (MAPE), and percentage bias. The root mean square error is a widely used measure to assess the performance of forecasting models in a variety of fields, including air pollution [86-87]. Since every partial error influences root mean square error that is proportionate to the error which is squared, this method is susceptible to outliers [88]. As a result, a higher difference between the anticipated value and the actual value outcomes is an unreasonably bigger end product of it. It is the square root of averaged squared errors calculated for T values formulated as:

$$RMSE = \sqrt{\frac{1}{T} \sum_{k=1}^T (Predicted_k - Actual_k)^2} \quad (3.7)$$

The RMSLE is calculated using the log of anticipated and observed values, which is as follows:

$$RMSLE = \sqrt{\frac{1}{T} \sum_{k=1}^T (\log(Preds_k + 1) - (\log(Act_k + 1)))^2} \quad (3.8)$$

The root mean squared log error is used in the circumstances when it's important to get rid of the prevailing large disparities between forecasted and real values, especially when those differences are large. While root mean square error is susceptible to outliers and raises the error value very high when they occur, Root mean squared log error minimizes the impact of outliers greatly. It's worth noting that root mean squared log error punishes underestimating of real values more strongly than it punishes overestimation. The mean absolute percentage error is a commonly used performance evaluation measure of prediction system efficacy, calculated from:

$$MAPE = \frac{1}{T} \sum_{k=1}^T \left| \frac{Actual_k - Predicted_k}{Actual_k} \right| \quad (3.9)$$

The main benefit of calculating the MAPE as a percent value rather than merely providing the error which is mean absolute and is easier to comprehend for investigators. With dataset values of 0, the MAPE becomes undefined as a result of normalization, which is a weakness.

The MSE is a metric that determines how much the proximity of anticipated and experimental values are. It is calculated as follows:

$$MSE = \frac{1}{T} \sum_{k=1}^T (Actual_k - Predicted_k)^2 \quad (3.10)$$

The distance between each predicted value and the associated actual value is measured, and the final value is squared. This measure is the mean of the squares of errors, to put it another way. The PB performance metric captures the average affinity of the projected value to be bigger or smaller than its real value, which is defined as:

$$PB = 100 \times \frac{\sum_{k=1}^T (Predicted_k - Actual_k)}{\sum_{k=1}^T Actual_k} \quad (3.11)$$

A percent bias with the value of 0 is thought to be the best value, suggesting correct model simulation with low magnitude values. Overestimation and underestimation bias is shown by larger positive and negative values, respectively.

Chapter 4

EXPERIMENTS & RESULTS

The detailed experimentation results for individual studies performed are presented in this chapter. For each simulation plan discussed in chapter 3, the performance is evaluated and concrete details are discussed in the subsequent sections.

4.1 Performance Evaluation of Imputation by Feature Importance for Missing Data Imputation

Tables 4.1 and 4.2 show the mean absolute percentage error and percent bias information for all features (RN, TH, TC, RH, and PR), for 20 percent MAR, MNAR, and MCAR datasets. All approaches had a mean absolute percentage error of less than 0.03% for PR, which is to be expected given that PR differences are typically minimal and consistent over time. For any given approach, TH and RH statistics are similar, with IBFI outperforming all other methods. The mean absolute percentage error for IBFI for RN has a minimum and maximum of 0.50 and 0.53 percent for 20 percent missingness, as indicated in this table, herewith statistical value higher for hot-deck which is approximately 1.8 times. IBFI outperforms every other approach for calculating TC values. The average mean absolute percentage error for 20% MCAR was observed to be 0.8%, whereas other approaches had MAPE of 1.3-3.5%. IBFI had a MAPE of 0.5% for TC and 20% MNAR, compared to 0.8 to 2.9% for other approaches. The results of the 20% MAR data are similar, with IBFI scoring 1% compared to 1.4-3.9% MAPE for other approaches.

When considering MAR, MCAR, and MNAR scenarios with 20% missingness, the lower value of absolute percent bias was observed for imputation by feature importance once weigh against other missing data filling methods across all the features.

Table 4.1: For 20 percent introduction of missing data of type MNAR and MCAR and all features (RN, TH, TC, RH, and PR) examined, PB and MAPE statistics for IBFI were assessed to various imputation methods (Mean, Median, Mode, PMM, and Hotdeck)

Scheme	Statistics	MCAR 20%					MNAR 20%				
		RN	TH	TC	RH	PR	RN	TH	TC	RH	PR
IBFI	MAPE	0.530%	0.480%	0.830%	0.240%	0.010%	0.520%	0.460%	0.54%	0.18%	0.01%
IBFI	PB	-0.040%	-0.050%	-0.180%	-0.010%	0.00%	0.310%	0.21%	0.12%	0.07%	0.01%
Mean	MAPE	0.690%	0.720%	2.780%	0.740%	0.02%	0.670%	0.750%	1.71%	0.47%	0.02%
Mean	PB	-0.050%	-0.110%	-1.380%	-0.180%	0.00%	0.420%	0.420%	0.46%	0.34%	0.01%
Median	MAPE	0.690%	0.710%	2.840%	0.750%	0.02%	0.630%	0.770%	1.70%	0.37%	0.02%
Median	PB	-0.130%	-0.050%	-1.560%	-0.360%	0.00%	0.350%	0.470%	0.38%	0.18%	0.01%
Mode	MAPE	0.720%	1.070%	3.010%	0.750%	0.03%	0.860%	1.42%	2.91%	0.30%	0.03%
Mode	PB	0.170%	0.980%	2.610%	-0.180%	0.02%	0.730%	1.40%	2.71%	-0.14%	0.03%
PMM	MAPE	0.810%	0.710%	1.280%	0.330%	0.02%	0.770%	0.67%	0.82%	0.26%	0.02%
PMM	PB	-0.090%	-0.110%	-0.240%	-0.010%	0.00%	0.320%	0.210%	0.10%	0.08%	0.01%
Hotdeck	MAPE	0.950%	0.980%	3.470%	0.960%	0.03%	0.880%	0.940%	2.24%	0.64%	0.03%
Hotdeck	PB	-0.020%	-0.140%	-1.340%	-0.130%	0.00%	0.450%	0.420%	0.40%	0.34%	0.01%

Table 4.2: For 20 percent introduction of missing data of type MAR and all features (RN, TH, TC, RH, and PR) examined, PB and MAPE statistics for IBFI were evaluated against various imputation methods

Method	Statistics	MAR 20%				
		RN	TH	TC	RH	PR
IBFI	MAPE	0.500%	0.480%	0.970%	0.230%	0.010%
IBFI	PB	0.010%	-0.030%	-0.160%	-0.010%	0.00%
Mean	MAPE	0.650%	0.740%	3.080%	0.630%	0.020%
Mean	PB	0.170%	-0.150%	-2.280%	-0.240%	-0.010%
Median	MAPE	0.630%	0.730%	3.140%	0.660%	0.020%
Median	PB	0.090%	-0.090%	-2.370%	-0.410%	-0.010%
Mode	MAPE	0.750%	0.710%	2.820%	0.930%	0.020%
Mode	PB	0.440%	0.100%	2.380%	-0.900%	0.010%
PMM	MAPE	0.780%	0.680%	1.390%	0.360%	0.020%
PMM	PB	-0.020%	-0.060%	-0.280%	-0.030%	0.0%
Hotdeck	MAPE	0.920%	1.00%	3.930%	0.830%	0.030%
Hotdeck	PB	0.190%	-0.140%	-2.340%	-0.260%	-0.010%

The resultant values of normalizing the data to the same statistical values which are mean values across all imputation methods, then averaging over all the features are shown in Fig 4.1. For all variables, average statistics are computed across several missingness scenarios such as MAR, MCAR, and MNAR. In the very first step, for all the variables, the average value of all performance metrics for all imputation methods are computed across various missingness scenarios and their related percentage of missingness. Secondly, every value in various missingness types, such as MNAR 10 percent, is standardized to various performance measures by dividing

by the equivalent average value computed in the first step above. This yields standardized values of each performance evaluation parameter for various missingness percentages and their types. Finally, using the imputation method, the averages of performance metrics for all the features are computed. The RMSLE, RMSE, MSE, and MAPE statistics show small variation as a function of data missingness degree and type, as shown in Fig 4.1. For these statistics, IBFI is superior in all circumstances. As seen in Fig. 4.2, the percent bias (PB) appears to be to some extent reliant on both the degree and type of data missingness. For missing completely at random data, the percent bias for imputation by feature importance is comparable to predictive mean matching and Hotdeck for missingness of 10% or less but superior for missingness of greater degrees (20% and 30%). PB only displays positive bias when using MNAR data. For all degrees of missingness, IBFI is similar to PMM, but it outperforms the other approaches. When missingness is MAR, we see a mix of positive and negative percent bias. At 10% MAR, imputation by feature importance has a percent bias that is identical to Mean and Hotdeck, as well as PMM at 10% and 20% MAR. IBFI has a lower PB than all other approaches for Mode, greater than 10 percent missing at the random type of missingness.

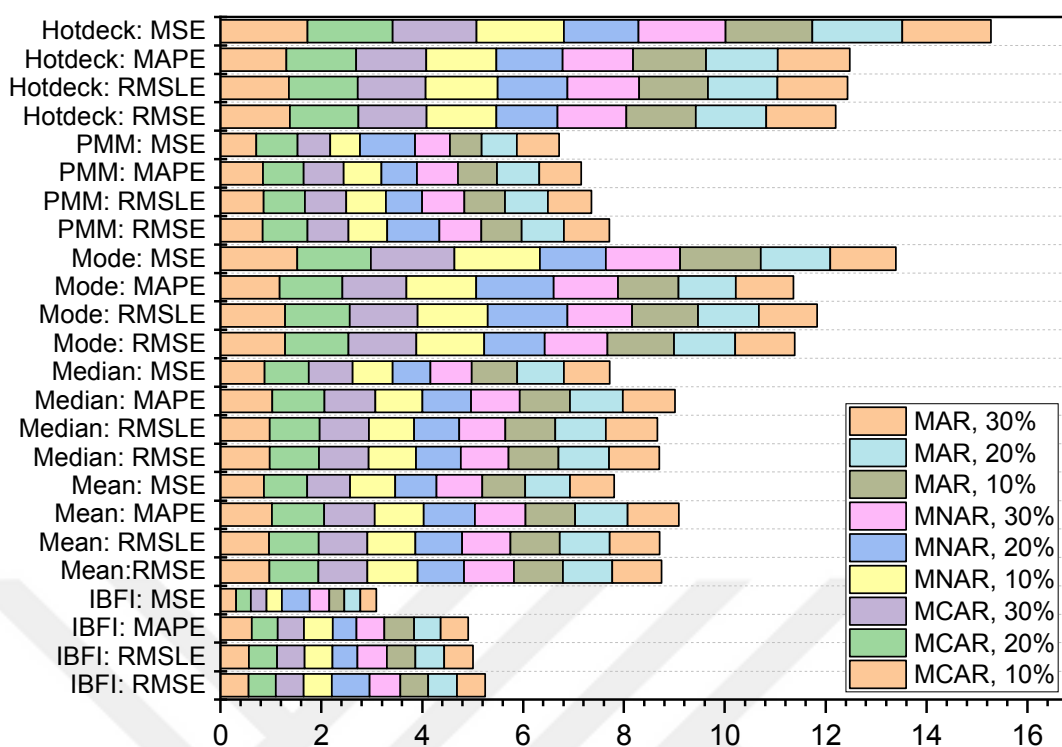


Figure 4.1: The IBFI statistics compared to other imputation methods adjusted to the average statistic for all methods across different features, displaying RMSLE, RMSE, MSE, and MAPE.

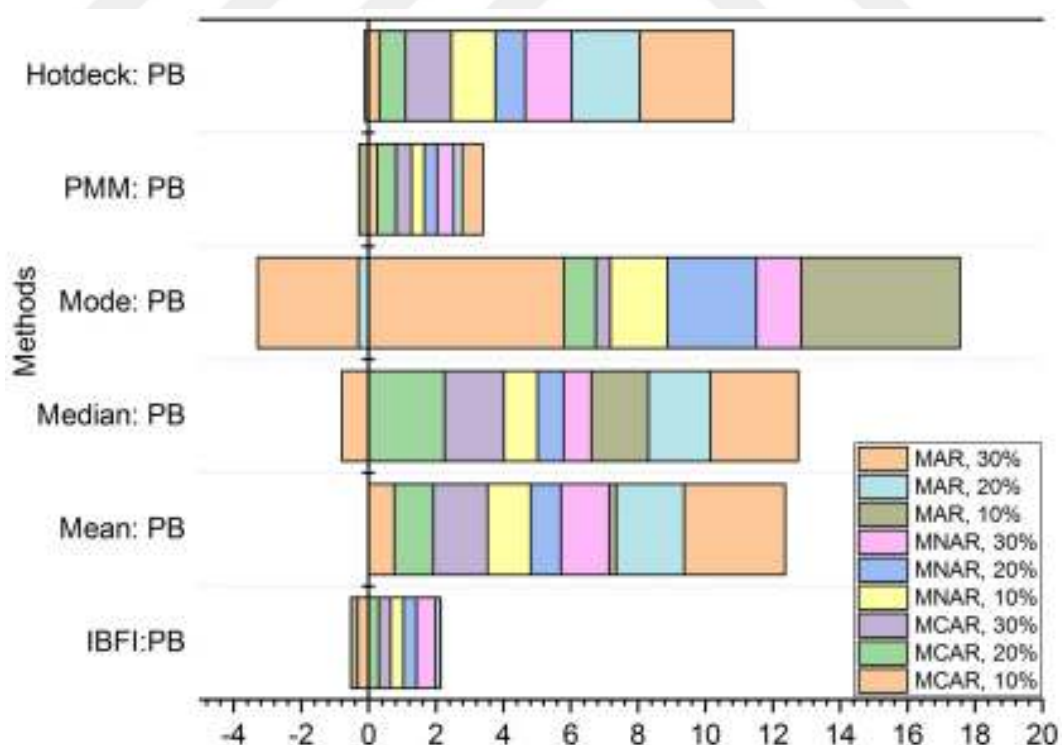


Figure 4.2: IBFI statistics compared to other imputation methods adjusted to the average statistic for all methods aggregated across all features, revealing percent bias for 10, 20, and 30 percent MNAR, MAR, and MCAR missingness, respectively.

The reusability of the previously trained model on different features in subsequent iterations and scanning of samples at missingness type of completely at random with varying percentages of missingness across 10 to 30% is shown in Figure 4.3(a-c). A sample number in the following iterations is depicted on the X-axis, while the previously trained model reusability and new model generation are depicted on the Y-axis. Model reusability is used in the suggested methodology, which keeps the models that are fitted throughout following iterations for future patterns. Fitted models are saved throughout sample processing, as shown in Figure 4.3a, and those models are employed in the following samples, as indicated by the black dotted line. Figure 4.3(a-c) illustrates that the model hit rate grows rapidly as the number of samples increases, as illustrated by the black dotted line. The reusability of the prior model is reflected in the model hit. Before creating a model, the directed model which is comprised of the features determined by the FIM is looked for in the model list while imputing each value. If it is located, the model that has already been fitted should be able to forecast the missing value at this point. When a model hit isn't found, a new model is trained for the current predictor attributes directed by the feature importance values. Furthermore, the newly trained model is added to a list of models that can be used for future formulations. Blue bubbles indicate the location where the new model is fitted. As it can be seen, the previously trained model's hit rate of the suggested approach is increasing as sample processing progresses. The construction of models is only seen during the first few measurements, and the models are then utilized to anticipate missing values in subsequent data. The suggested methodology benefits from the reusability of models in this way because it enables the imputation process efficient in terms of time and space complexity. The fitted model's reusability statistics were shown in Figures 4.3 to 4.5, which demonstrate the same pattern during successive iterations. The new model training may be seen in blue bubbles at the very beginning of the process, but the previously fitted model's reusability improves with the progress of imputation of the missing values.

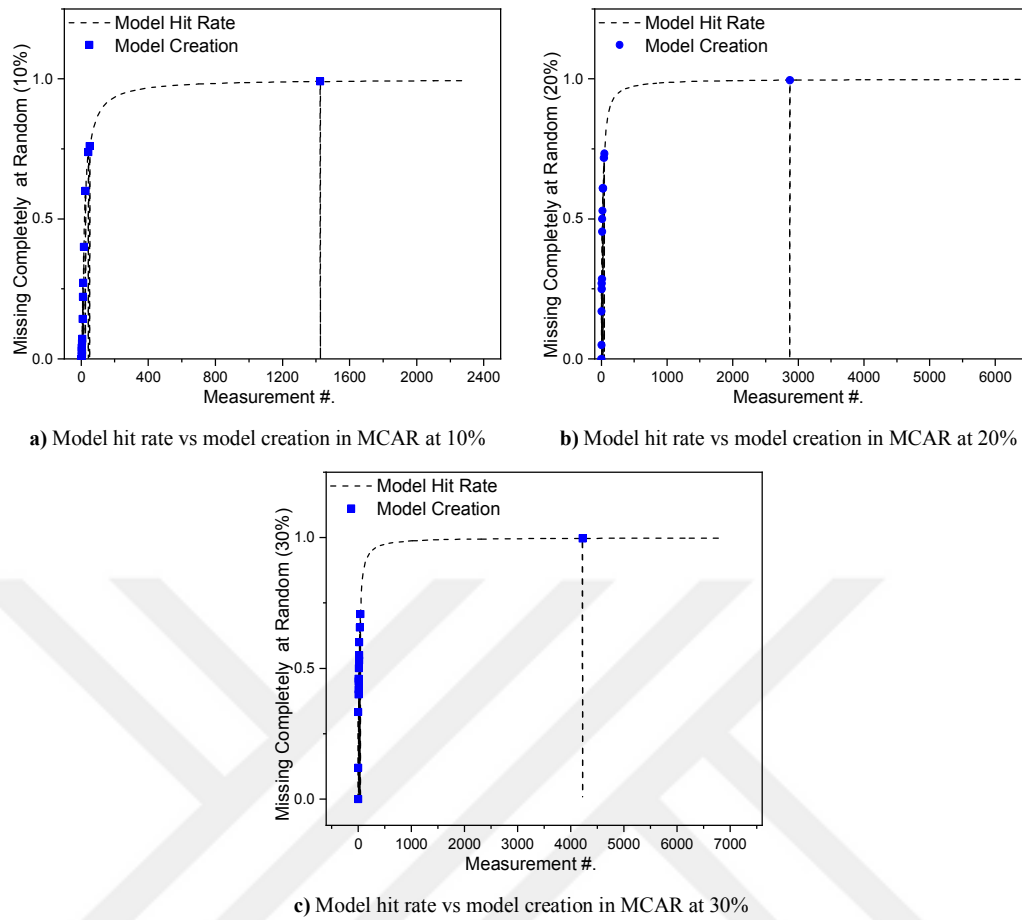
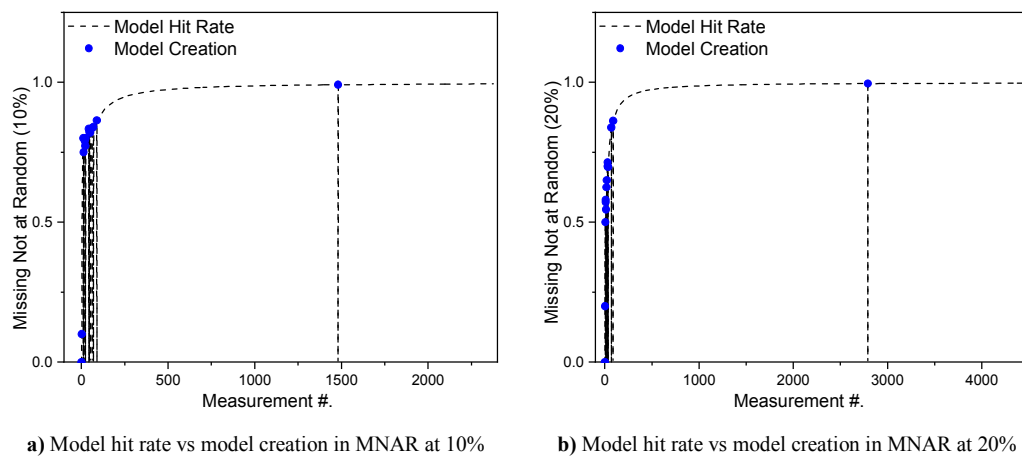
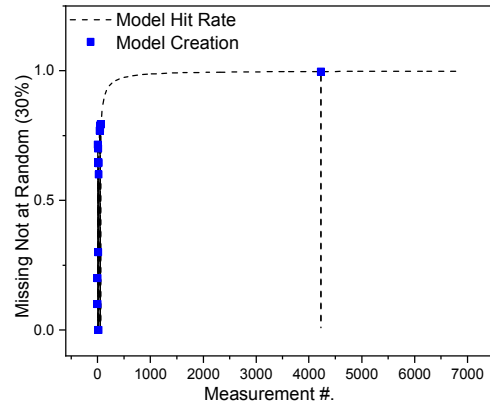


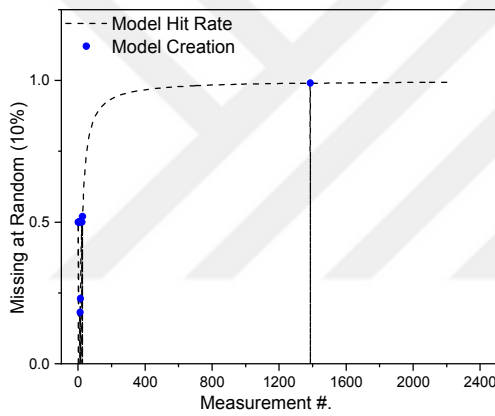
Figure 4.3: The reusability of models in respective iterations and scans during the imputation process of 10, 20, and 30 percent for missing completely at the random type of missingness, respectively.



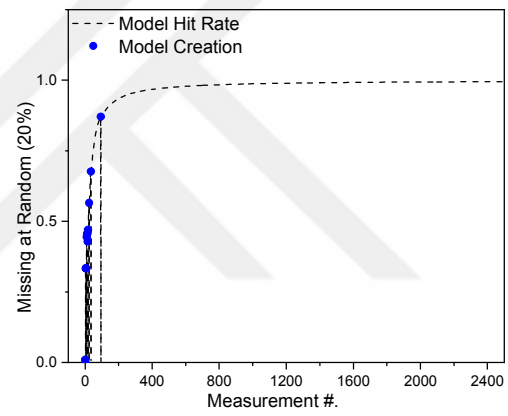


c) Model hit rate vs model creation in MNAR at 30%

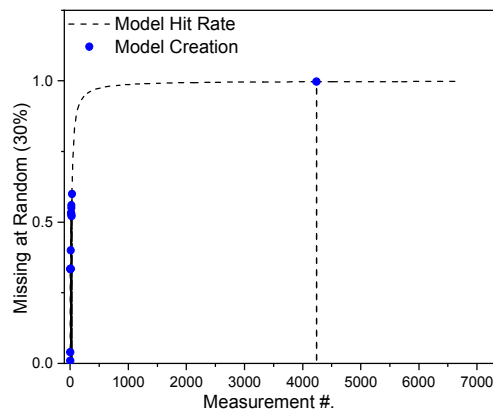
Figure 4.4: The reusability of models in respective iterations and scans during the imputation process of 10, 20, and 30 percent for missingness not at random type, respectively.



a) Model hit rate vs model creation in MAR at 10%



b) Model hit rate vs model creation in MAR at 20%



c) Model hit rate vs model creation in MAR at 30%

Figure 4.5: The reusability of models in respective iterations and scans during the imputation process of 10, 20, and 30 percent for missing at the random type of missingness, respectively.

As a result of this reusability, effective reconstruction of missing values for all types of missing patterns and types is achieved with better time and space complexity. The model reusability is further depicted in figures 4.6 and 4.7 below.

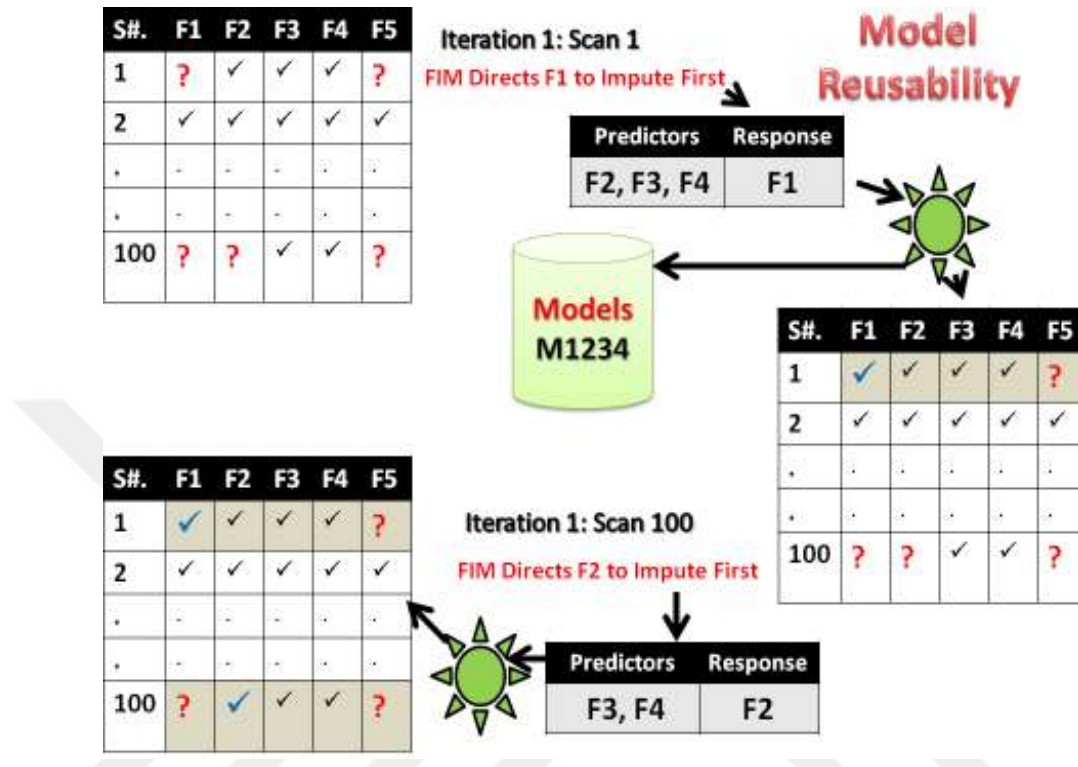


Figure 4.6: The reusability of the model in IBFI for sample features during iteration 1

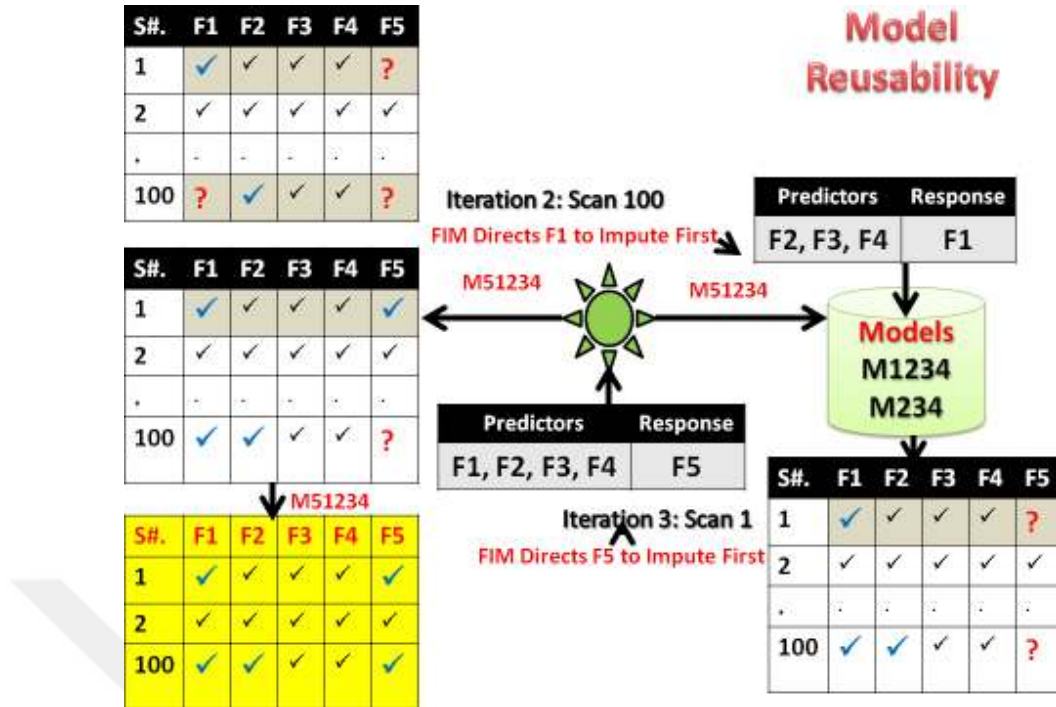


Figure 4.7: The reusability of the model in IBFI for sample features during iterations 2 and 3

4.1.1 Computing the Complexity of the Imputation by Feature Importance

This section provides the step-by-step time complexity of the methodology which is proposed to impute the missing data in time series across different types of patterns of missingness. As the proposed imputation methodology is the pre-processing method which is a one-time effort to best fill the missing values for further analysis because the biased imputation results in the poor fitting of any statistical or machine learning models.

<i>ModelList</i> = NULL	(1 Time)
<i>U</i> ← sequence (1 to number of features)	(1 Time)
<i>PriorityMatrix</i> ← NULL	(1 Time)
<i>k</i> = 1	(1 Time)

The above piece of code is just the initialization of the process by giving the base value to run the imputation process. Each statement in the piece of code above took a constant amount of time.

Process:

5. Divide **Data Matrix** in to two subsets i.e. $PD(X_1 \dots X_n)$ and $ID(X_1 \dots X_n)$ (**m Times**)

When considering steps 1 and 2, step 1 splits the data into two subparts i.e. pure and impure data. The whole process just scans each row of the dataset and checks for the missing value. If there are ‘m’ number of rows in the dataset. The whole procedure would run ‘m’ times to complete the process. On the other hand, the while loop will run according to the threshold of rejection which is the degree up to which the number of features to impute in certain samples. For the sake of simplicity, we label the rejection threshold with ‘n’. Step 2 will run $n + 1$ times.

a. For scanningindex = 1 to number of samples in ID	n(m+1) times
i. NAsVector = indices(is.NA(ID[scanningindex,]==TRUE))	n(m) times
ii. If length(NAsVector)> Rejection_Threshold OR length(NAsVector)==0	n(m) times
1. Discard that sample	n(m) times
iii. Else	
1. TrainVector = setdiff(U,NAsVector)	n(m) times
2. Prior_Mat = FIM[NAsVector,]	n(m) times
3. The indexes of Prior_Mat contains TrainVector values, do Prior_Mat[indices] = 0	n(m) times
4. For all the indices where Prior_Mat do not contain values of TrainVector, Set Prior_Mat[indices] = 1	n(m) times
5. For n in 1 to RowsCount(Prior_Mat)	nm(p) times
i. if (Prior_Mat[n,j]==1)	
ii. location_vector[l_c] = which(Prior_Mat[n,]==1)	nm(p) times
iii. l c = l c + 1	nm(p) times

The above piece of code depicts the actual methodology for different iterations and scans regarding the imputation procedure. The ‘k’ variable in the previous code snippet represents the iterations that run according to the threshold whilst the scan index traverses each sample one by one and check for the missing pattern in it and selects the best attribute to impute first. Here come two different scenarios regarding time complexity. The first scenario occurs when the if condition in ‘a(ii)’ gets true always and the time complexity is the same as the previous line of code i.e. $O(nm)$ times. This is the minimal time taken to execute this piece of code. On the other hand, when this condition gets false, there is one more ‘for loop’ which simply executes according to the rows count of the priority matrix and finds the location of the maximum index of the missing value for each row. When the condition gets false this results in the maximum complexity of the imputation process which is $O(nmp)$.

6. IF length(location_vector) == 0)	n(m) times
a. indice_to_train = NA_vect	
7. Else	
a. max_value = maximum (location_vector)	n(m) times
b. max_index = indices where (location_vector== max_value)	n(m) times
c. max_index =max_index[1]	n(m) times
d. indice_to_train = NAvector[max_index]	n(m) times

In step 6 and 7 above, each statement is running according to scan index and perform a different type of processing regarding assigning different variables to one another. All the statements take similar steps to complete the procedure i.e. $O(nm)$.

8. Traindata = PD [:,TrainVector]	n(m) times
9. Trainclass = PD [:,indice_to_train]	n(m) times
10. trainF = Column bind(trainingdata,class = trainingclass)	n(m) times
11. ModelsName = ConcatString(indice_to_train , TrainVector)	n(m) times

From steps 8 to 11, these all statements execute $O(nm)$ times where 'n' represents the rejection threshold and 'm' represents the number of rows in the impure dataset.

12. Flag = CheckModel (ModelList,ModelName)	$O(nmj)$
--	----------------------------

Function: CheckModel (ModelLists, ModelNames):	$O(j)$
3. Flags = -1	1 time
4. For i in 1 to len(Model_Lists)	j times, where j is the length of items
a. IF ModelLists[i]== ModelNames	
i. Flags = 1	best 1 time, worst j time
ii. break	
5. Return flags	1 time

In step 12, a function is called which checks the name of the model and if it is found in the model list, the same model from the memory is used for the current imputation of the missing value.

```

13. IF flag == -1 // Current formulation not found
    a. Train a ML model i.e.  $BM_{Name}$  and add it to ModelList for reusability
    b. Testdata =  $ID[scanindex, TrainVector]$  n(m) times
    c. Value = predict(model, testdata)
    d.  $ID[scanindex, indice\_to\_train] = value$ 
14. Else
    a. print("Previous model found for reusability")
    b. testdata =  $ID[scanindex, TrainingVector]$  n(m) times
    c. val = predict(Model_List[[flag]], testdata)
    d.  $ID[scanindex, indice\_to\_train] = value$ 
15. End if
    a. End if
16. k += 1 n(m) times
17. End

```

By carefully examining the algorithm, we have concluded that there are two different scenarios for computing time complexity. In the first scenario, we have different conditional statements which lead to true results in lower time complexity of the proposed methodology. On the other hand, if the conditional statements lead to false, the methodology results in maximum time complexity of the proposed methodology. If we calculate the complexity in terms of theoretical concepts, it will be a minimum of $O(n^2)$ and a maximum of $O(n^3)$.

Table 4.3: An approximation of the total number of scans for the certain number of impure samples

#. of Impure Samples	Rejection Threshold	#. of Features	Iterations	Total Scans
300	1	5	1	300
500	2	5	2	1000
700	3	5	3	2100
900	4	5	4	3600
300	4	10	4	1200
500	6	10	6	3000
700	7	10	7	4900
900	8	10	8	7200

The approximate number of total scans are presented for a certain number of samples are presented in Table 4.3. It can be observed that the total number of scans is controlled by the rejection threshold and the total number of impure samples. The impure samples are those which have missing values of 1 to the count of rejection threshold.

4.1.2 Execution time of IBFI with different base learners and other imputation methods

The results in Table 4.4 below present the execution time of imputation by feature importance and other frequently used imputation methods. The imputation by feature importance is executed using different base learners such as XGBoost, *K*-Nearest Neighbors (KNN), and Linear Model (LM). The hardware specifications of the system used for experimentations are the windows10 operating system with the following hardware configuration: Lenevo P500 Thinkstation (Intel® Xeon® CPU E5-2650 v3 @ 2.3GHz, 64 GB Ram, GPU GeForce GTX 1080 Ti). The time of execution is composed of the following two points.

- The time to find the appropriate value to impute
- The time to impute the missing value to the missing location exactly.

Table 4.4: Execution time in seconds of imputation by feature importance using different base learners and other imputation methods

Type	Percentage	IBFI XGBoost	IBFI K-NN	IBFI Linear Model	Mean	Median	Mode	PMM	Hot-deck
MCAR	10%	180.09	120.47	98.48	23	23	24	86	52
	20%	198.19	133.61	114.61	24.1	26	27.2	92.91	60.67
	30%	210.06	147.78	122.36	26.03	27.02	30.03	102.2	67.07
MNAR	10%	174.47	120.64	107.45	23.1	23.5	24.0	87.44	55
	20%	198.8	129.79	112.87	24.3	26	29.1	92.99	61.20
	30%	209	143.90	122.87	26	27.5	29.78	100.07	70
MAR	10%	172.27	128.01	106.06	23.3	24	24.5	95.90	62.0
	20%	200.54	137.91	109.33	24.0	24	25.01	98.02	65.0
	30%	211.17	148.25	123.63	26.75	28.01	29.09	103.3	72.03

In table 4.4, the execution time of imputation methods is calculated in seconds for different missingness types and percentages. When analyzing the computation time of imputation by feature importance, we have analyzed that the execution time depends upon the base learner used in the methodology. If the computational complexity of the base model is high then the execution of the imputation by feature importance will take a longer time. This can be observed in the case of XGBoost and linear model. The time complexity of the linear model is much lower than the XGBoost because of its simple calculations for training the model and computing different coefficients. In the case of the K-nearest neighbor, the imputation will take a longer wait than the linear model because of the distance calculation step. This can be simply formulated as.

$$\text{TC (Imputation by Feature Importance)} \propto \text{Base Learning Model} \quad (4.1)$$

On the other hand, the mean, and the median has similar time complexity while the mode has a higher time complexity because of the calculation of its probability density function for computing the mode of continuous data. More precisely, the following relationship holds for the time complexity of different computation methods.

$$\text{TC}(\text{IBFI (Linear Model)}) < \text{TC}(\text{IBFI (KNN)}) < \text{TC}(\text{IBFI (XGBoost)}) \quad (4.2)$$

$$\text{TC}(\text{Mean}) < \text{TC}(\text{Median}) < \text{TC}(\text{Mode}) \quad (4.3)$$

$$\text{TC}(\text{Hot-deck}) < \text{TC}(\text{PMM}) \quad (4.4)$$

$$\text{TC}(\text{IBFI (Linear Model)}) \cong \text{TC}(\text{PMM}) \quad (4.5)$$

After analyzing the time complexity of the all models from the identities above, we have concluded that.

$$\begin{aligned} &\text{IBFI (XGBoost)} > \text{IBFI (KNN)} > \text{IBFI (LM)} > \text{PMM} > \text{HOT-DECK} > \text{MODE} \\ &> \text{MEDIAN} > \text{MEAN} \end{aligned} \quad (4.6)$$

4.1.2.1 Justification for using the proposed methodology (IBFI)

The imputation of missing values in the dataset plays an important role in the further design and testing of machine learning models for experimental data used for the study. The problem becomes more complicated in the scenarios where the attributes in the experimental dataset have some correlation with each other. It is a fact that any statistical computation performed on some outliers or biased values results in poor outcomes and dissatisfactory results.

Moreover, in the era of science and technology where the personal computer used for official work has a range of CPU cores and large primary and secondary memories. The data preprocessing is a one-time effort for the data analyst and developers. If we use the traditional statistical imputation method for the pre-processing of missing values, we have to compromise between the accuracy and time complexity. Also, when analyzing table 4.4, the maximum time complexity for IBFI to impute the missing values in the dataset is approximately 211 seconds and a minimum of 98.48 seconds. For purification and unbiased estimates, this is the maximum time taken by IBFI to impute missing values till 30% of missingness in whole data and results in very fewer error values than all the other imputation methods. For data analysts, IBFI is the best option to choose because of its high precision towards missing values imputation. Also, all the upcoming course of experimentation relies on the imputed dataset.

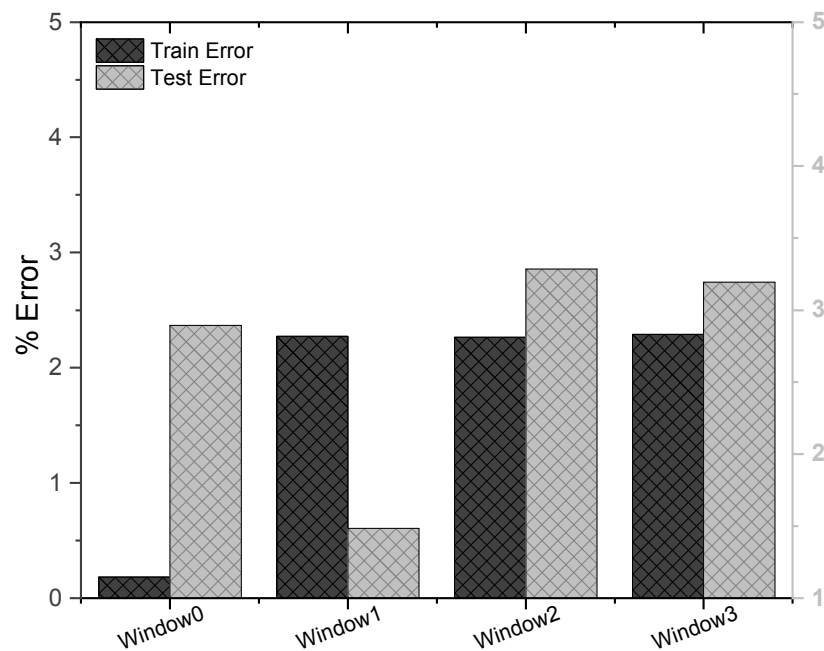
4.2 Performance Evaluation of Stacking and Automatic Anomaly Indication Function in Classification of Time Series

This section contains extensive experimental findings to assess the performance and accuracy of anomalies classification over seismic periods at different window sizes (0 to 3).

Table 4.5: The performance metrics yielded after applying the proposed methodology

Win #.	Testing Accuracy	Training Accuracy	F1	AUC	Precision	Recall	MCC
Win 0	97.1	99.8	98.5	83.4	97.4	99.6	39.6
Win 1	98.5	97.7	99.2	89.9	99.2	99.3	78
Win 2	96.7	97.7	98.2	91.5	97.4	99.1	73.5
Win 3	96.8	97.7	98.3	93.2	97.5	99	79.9

The final labels are chosen only after several predictions have been analyzed using an indicator function for automatic detection of an anomaly, as previously indicated. The automatic anomaly indication function assesses the proportion of seismic classification results for approximately 36 samples comprised 1 day and classifies all of the data for those samples as anomalous if the percentage of indication comes up to the indication factor (IF). The automatic anomaly indication function's final categorization is based primarily on the Meta Learner's classification results. Table 4.5 displays the results of the first stage, which were obtained by using the stacking technique without using the indicator function, and Figure 4.8 illustrates the error rate.

**Figure 4.8:** The training and testing error at window sizes ranging from 0 to 3

Using the stacking ensemble strategy to classify resulted in efficient classification, according to all of the statistics, with testing accuracy of 0.971, 0.985, 0.967, and 0.968. The testing and training errors with various window sizes are shown in Figure 4.6. Furthermore, the receiver operating characteristics curve (ROC) is widely regarded as an excellent graphical depiction for quickly assessing the system's classification performance. The area under the curve (AUC) is an extensively used global method for determining a ROC plot's pinpointing accuracy [89-90]. A diagram of the true positive rate (Sensitivity) adjacent to the false positive rate (1-Specificity) for various cut-off points is known as a Receiver Operating Characteristic curve (or ROC curve). Every spot on the ROC curve symbolizes a sensitivity/specificity duo analogous to a specific decided threshold. The test's accurateness is determined by how well it differentiates the groups being assessed. The area under the ROC curve is used to determine accuracy. The area of a perfect test is 1, whereas the area of a worthless test is 0.5.

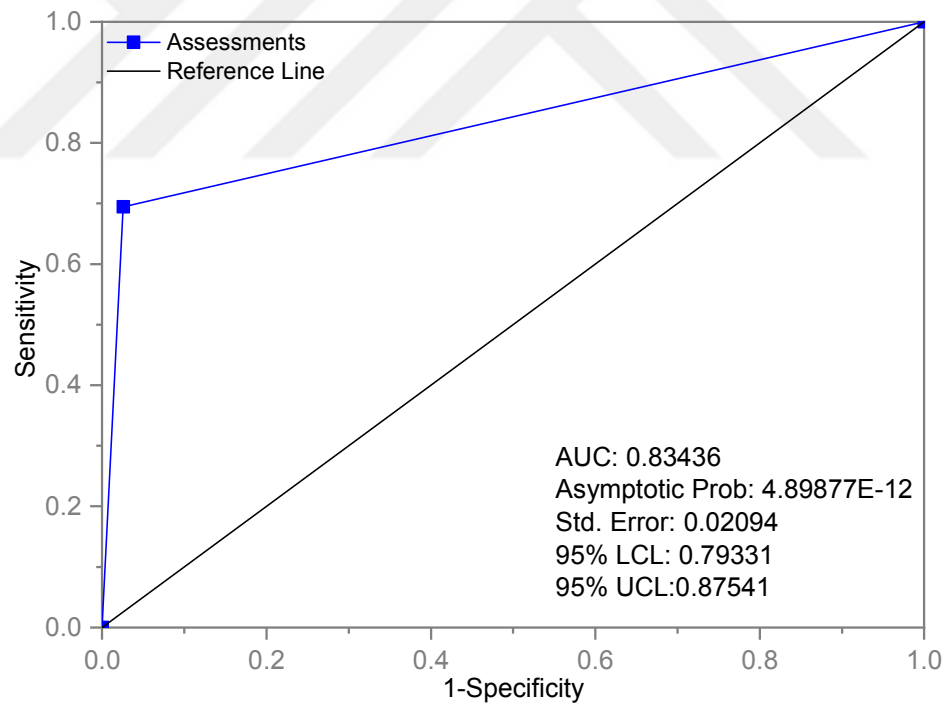


Figure 4.9: For split at window 0, ROC curve of assessments and outcomes

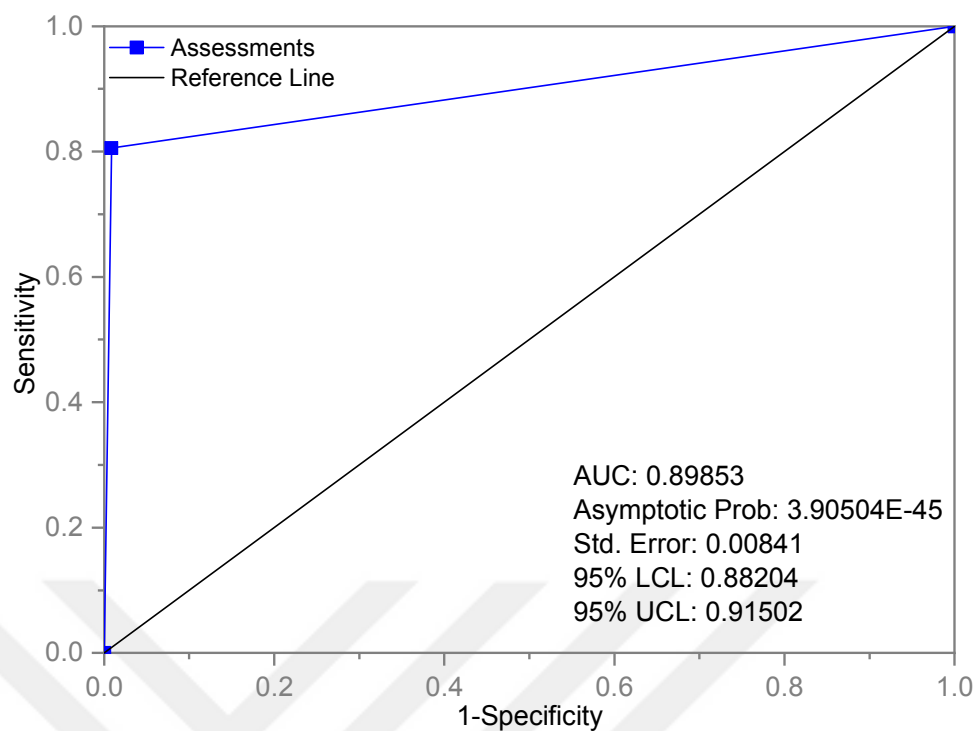


Figure 4.10: For split at window 1, ROC curve of assessments and outcomes

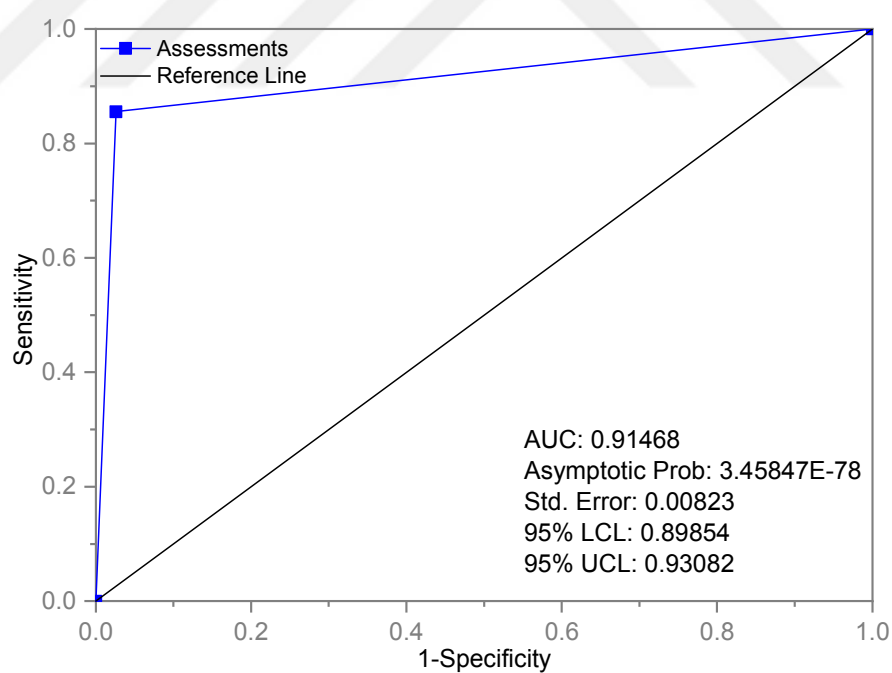


Figure 4.11: For split at window 2, ROC curve of assessments and outcomes

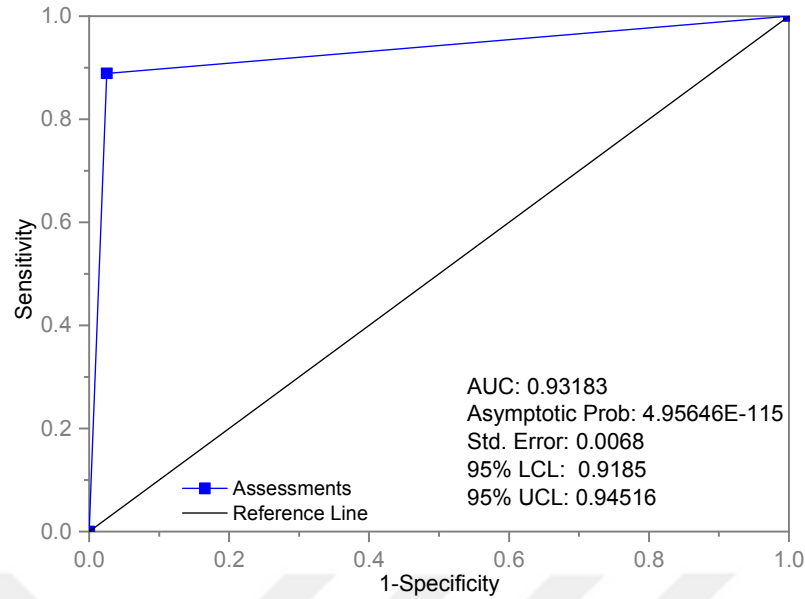


Figure 4.12: For split at window 3, ROC curve of assessments and outcomes

Figures 4.9 through 4.12 illustrate the receiver operating characteristic (ROC) curve at various window sizes, without using the automatic anomaly detection function, to evaluate the classification model's performance. Figures 4.9 and 4.10 demonstrate the true positive rate (Sensitivity) vs the false positive rate (100-Specificity) at various cut-off positions for Windows sizes of zero and one. The window size of zero is comprised of original samples of anomaly whilst one presents the samples a day before and after the occurrence of anomaly. zero and one day before and after the anomaly. The suggested methodology's ability to distinguish the samples into their respective classes results in AUC values of 83.436 and 89.853 percent, respectively, with the lower and upper confidence limits (95 percent LCL and 95 percent UCL) of 0.79331, 0.88204, and 0.87541, 0.91502. Figure 4.11 and Figure 4.12 indicate the ability of the suggested technique on the same grounds as Figure 4.9 and Figure 4.10 for the window size of two and three. With an area under the curve (AUC) value of 91.468, 93.183, and an inferior confidence limit (95% LCL) and superior confidence limit (95% UCL) of 89.854%, 91.85%, and 93.082%, 94.516%, respectively, these time windows classify the observations into anomalous or normal one's. As stated above, the classification done by the meta learner is not smooth enough to assign final class labels. The resultant class labels may be in the form:

anomalous $\{1, 0, 1, 0, 1, 0, \dots\}$. and normal $\{0, 1, 1, 1, 0, 0, \dots\}$, instead of $\{1, 1, 1, 1, 1, 1, \dots\}$ and $\{0, 0, 0, 0, 0, 0, \dots\}$

Although the accuracy of classification is persuasive, classification accuracy may vary depending on seismic activity. This is shown in Table 4.5, where all of the performance evaluation metrics for the window size of zero to three show considerable values, excluding Mathews Correlation Coefficient (MCC), which demonstrates efficacy towards the anomalous class is poor, particularly at window 0. These uneven classifications towards testing data are submitted to a function for automatic detection of anomalies, which refines the predictions for a full day into anomalous or normal classes. The results depict that all anomalous samples are accurately identified and labeled after supplying the expected class label to the automated anomaly detection function.

4.2.1 Performance Evaluation of Anomaly Detection at Different Window Sizes

After the anomaly indication function has processed the incoming radon series class labels, this portion graphically shows the final anomaly classification and indication. The results reveal that anomalies are efficiently recognized and classified into 0 (Normal) and 1 (Seismic) following post-processing of early crisp labels using the stacking approach.

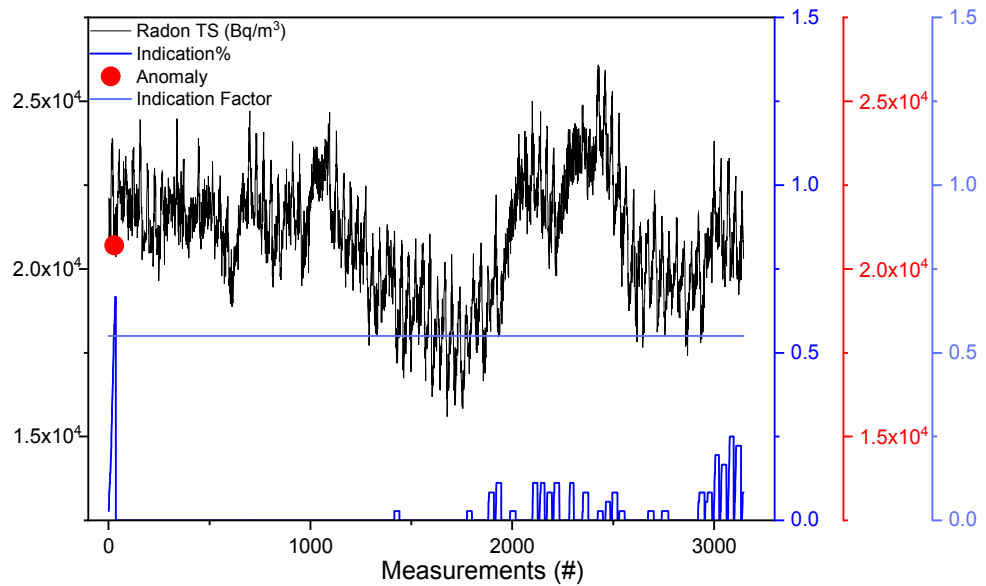


Figure 4.13: The seismic activity classification in the test time series at a window size of 0

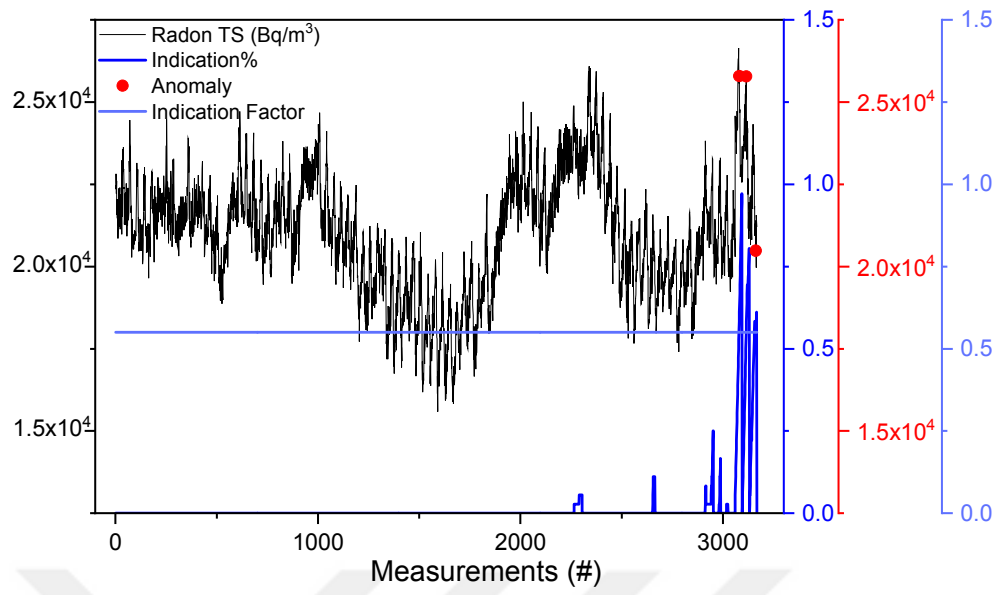


Figure 4.14: The seismic activity classification in the test time series at a window size of 1

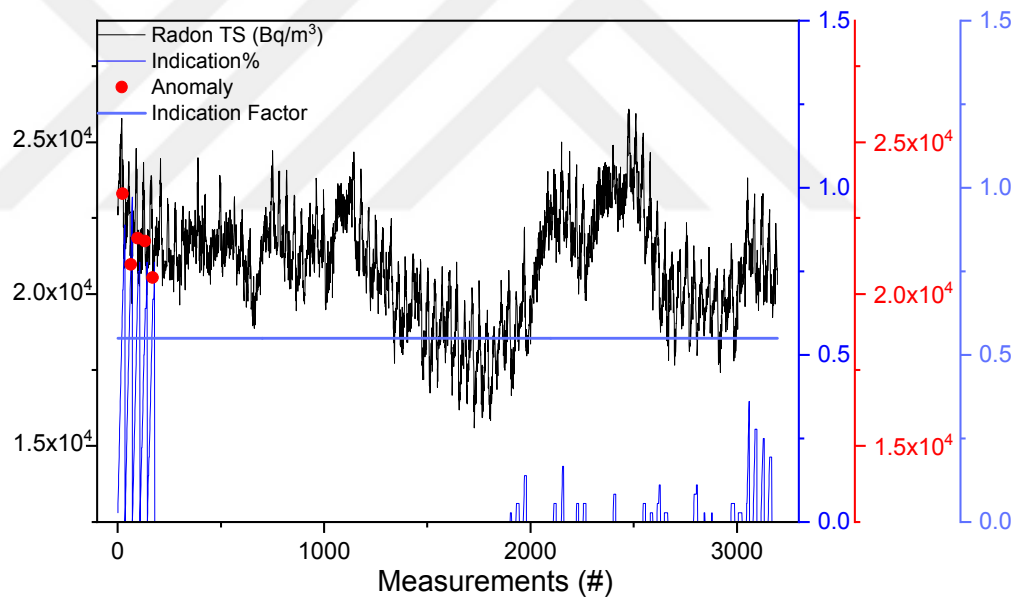


Figure 4.15: The seismic activity classification in the test time series at a window size of 2

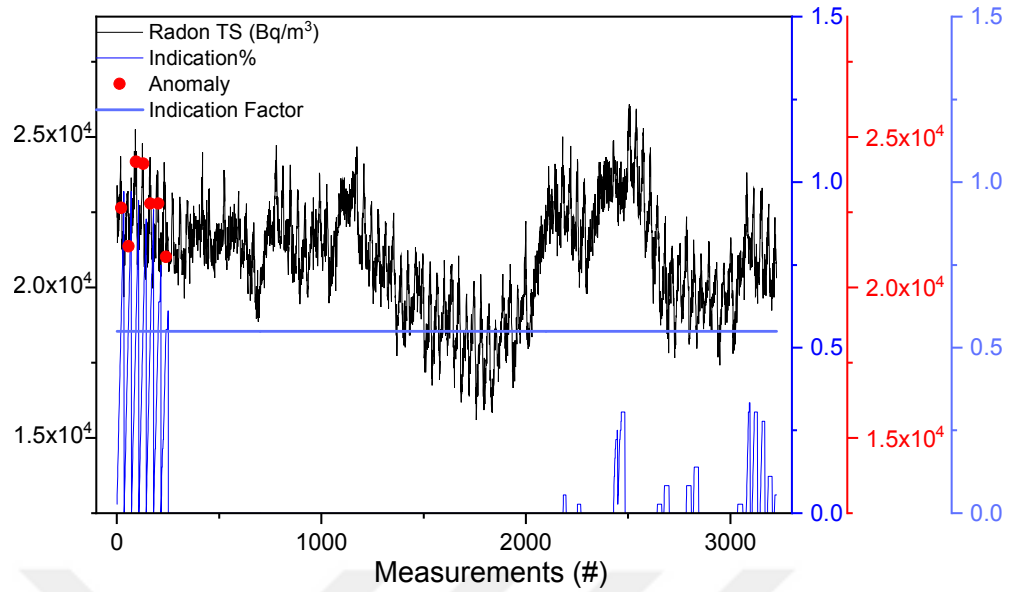


Figure 4.16: The seismic activity classification in the test time series at a window size of 3

The outcomes achieved after applying the suggested approach to the data at various windows from 0 to 3 are shown in Figures 4.13 through 4.16. We divided the training step into two phases to make it easier to understand. The main task in the first step is to post crisp class information to arriving time series data, whereas, in the second step, each classification made by step 1 is forwarded to the automatic anomaly indication function, which calculates the percentage of anomalous events classified in the step before it. The concept of the indication factor is used here, which states that if the percentage of arriving seismic events categorized by step one is equal to or more than a specific limit or threshold, then clumps of readings are classed as seismic time series. As previously noted, the variances in step 1 data are not enough for final anomaly classification. Although the performance measurements for window 0 appear to be sufficient for labeling the anomaly, putting the data through the automated anomaly indication tool improves the sensitivity for detecting anomalies.

Figure 4.13 shows how to classify anomalies using many criteria, including indication factor and its percentage, for window 0, the initial SRGC. The indication factor (value of 0.55) is indicated by the royal blue horizontal line, while the original soil radon gas concentration for that period of time is represented by the black curve. The red circle depicts the actual anomaly that occurred within that time period, while

the blue color depicts the percentage indication. The indication percentage is the percentage of indication received, and clumps of reading will be regarded as anomalous if the indication percentage is more than or equal to the indication factor. Where radon concentration is abnormal on certain indices, as illustrated by the black curve in figure 4.13, the percentage of indications got increased above the threshold for indication. The indication percentage that grew above the indication factor at all three sites where the red circle indicates an anomaly is illustrated in Figure 4.13. As illustrated in figures 4.15 and 4.16, the same patterns were seen in windows 2 and 3, two and three days before and after the anomaly. Figure 4.15 shows that the indication percentage is increased over the indication factor at all five periods where there is an anomaly, resulting in the series being classified as anomalous.

Furthermore, in figure 4.16, the indication percentage follows the pattern every time the red circle indicates an abnormality. The proposed methodology for classification based on stacking and automatic anomaly indication function efficiently classifies the incoming time series into seismic and non-seismic ones and achieves a Mathews Correlation Coefficient (MCC) value of +1 (after step 2) for the complete windows ranging from 0 to 3 respectively, as shown by all of the experimental results discussed above.

Chapter 5

CONCLUSION

This chapter highlights a summary of the whole dissertation along with the conclusions drawn from the individual. It presents the novice contributions of the study regarding earthquake prediction and forecasting from pre-processing to post-processing. This chapter also illustrates the prominence of the results from an implementation perspective. Limitations of the current study and future research directions are also provided to apply research in different domains.

5.1 Discussion and Conclusion

The novel contributions of the research reported in this dissertation can be gauged from the perspectives of machine learning experts in the field of earthquake prediction and forecasting arena. This study helps the researchers to get in-depth background knowledge and new methodologies for missing data reconstruction and anomaly classification.

- Missing values are common in real-time series, and they can occur for a variety of reasons. When missing data causes bias in the forecasting model, the situation becomes critical. In this work, a method is proposed that makes use of the importance of features and imputes the missing values using any ML algorithm, such as XGBoost, like an encased learning algorithm. The suggested method infers various complicated patterns of missing values in different samples and establishes a rejection count that involuntarily excludes samples with a matching count of missing data. Through artificial means, missing value patterns in many features have been introduced in this approach. The anticipated methodology does better than other commonly used methods such as median, mode, mean, hot-deck, and predictive mean matching imputation when imputing the same incomplete data. On different missing scenarios of MNAR, MCAR, and MAR with the missing percentages of 10, 20, and 30%, different statistical parameters, such as MAPE, RMSE,

MSE, and RMSLE have been computed, indicating that the imputation by feature importance yielded best values for each performance parameter compared to other methods of imputation. The study's findings reveal that the proposed methodology's efficiency is based on the choice of the optimal training features for various patterns of missing values and the utilization of the trained models that had been stored. Utilization of the proposed methodology in other sectors of research, such as electric load prediction and medical including gene expression and clinical databases, could be of interest in the future. While imputing supervised classification datasets, Imputation by feature importance (IBFI) can be expanded to include labeled category information.

- This work presented a system for classifying time series data into its respective categories using a stacking ensemble learning approach and an automatic anomaly indication function as a post-processing method. The concentration is forecasted for validation data using these three learners, and a new matrix of meta-features is created, with predicted radon concentration acting as Meta features alongside class labels. The newly generated metadata is trained using a different machine learning method (Support Vector Machine with the radial kernel), and that Meta learner is used for the initial labeling of time series data. Finally, the function for automatic detection of anomalies takes the indications acquired by the meta-learner and smoothes out the labeling process by estimating the count of indications and comparing them to the indication percentage to generate final class labels. We determined that the proposed methodology correctly identifies all anomalies in the time series data based on all of the results reported.

5.2 Applications of the Proposed Methods

- The imputation methodology can be employed on any time series such as electric load forecasting, financial time series data, etc. where the attributes have correlations among each other.

- The automatic anomaly indication function can be utilized as a post-processing technique with any time series classification where the predictions are not smooth enough.
- The methodology described in the prediction of radon concentration can be utilized in any time series data for better distribution of samples for experimentation.
- This study provides a basic understanding of time series classification and windowing mechanisms.
- The proposed meta-learning approach in stacking ensemble can be useful for the classification of any multivariate time series data

5.3 Future Work

This study can be further extended in the following scenarios:

- The machine learning-based reject option classification can be further employed to get better results.
- Bagging-based ensemble classification can be employed for the classification of RTS time series data.
- The bagging-based ensemble can be empirically analyzed as can the stacking ensemble for the classification of time series.
- The imputation by feature importance can be further improved by incorporating class information in missing data reconstruction

REFERENCES

- [1] B. Adhikari, S. R. Mishra, S. B. Marahatta, N. Kaehler, K. Paudel, J. Adhikari, and S. Raut, "Earthquakes, fuel crisis, power outages, and health Care in Nepal: implications for the future," *Disaster medicine and public health preparedness*, vol. 11, pp. 625-632, 2017.
- [2] V. G. Gitis and A. B. Derendyaev, "Machine learning methods for seismic hazards forecast," *Geosciences*, vol. 9, p. 308, 2019.
- [3] M. Rafique, A. D. K. Tareen, A. A. Mir, M. S. A. Nadeem, K. M. Asim, and K. J. Kearfott, "Delegated Regressor, A Robust Approach for Automated Anomaly Detection in the Soil Radon Time Series Data," *Scientific Reports*, vol. 10, pp. 1-11, 2020.
- [4] M. Awais, A. Barkat, A. Ali, K. Rehman, W. A. Zafar, and T. Iqbal, "Satellite thermal IR and atmospheric radon anomalies associated with the Haripur earthquake (Oct 2010; Mw 5.2), Pakistan," *Advances in Space Research*, vol. 60, pp. 2333-2344, 2017.
- [5] Z. Jilani, T. Mehmood, A. Alam, M. Awais, and T. Iqbal, "Monitoring and descriptive analysis of radon in relation to seismic activity of Northern Pakistan," *Journal of environmental radioactivity*, vol. 172, pp. 43-51, 2017.
- [6] B. Aslam, A. Zafar, U. A. Qureshi, and U. Khalil, "Seismic investigation of the northern part of Pakistan using the statistical and neural network algorithms," *Environmental Earth Sciences*, vol. 80, pp. 1-18, 2021.
- [7] A. D. K. Tareen, M. S. A. Nadeem, K. J. Kearfott, K. Abbas, M. A. Khawaja, and M. Rafique, "Descriptive analysis and earthquake prediction using boxplot interpretation of soil radon time series data," *Applied radiation and isotopes*, vol. 154, p. 108861, 2019.
- [8] A. D. K. Tareen, K. M. Asim, K. J. Kearfott, M. Rafique, M. S. A. Nadeem, T. Iqbal, and S. U. Rahman, "Automated anomalous behaviour detection in soil radon gas prior to earthquakes using computational intelligence techniques," *Journal of environmental radioactivity*, vol. 203, pp. 48-54, 2019.
- [9] M. Janik, P. Bossew, and O. Kurihara, "Machine learning methods as a tool to analyse incomplete or irregularly sampled radon time series data," *Science of The Total Environment*, vol. 630, pp. 1155-1167, 2018.
- [10] R. Törnqvist, J. Jarsjö, J. Pietroni, A. Bring, P. Rogberg, S. M. Asokan, and G. Destouni, "Evolution of the hydro-climate system in the Lake Baikal basin," *Journal of Hydrology*, vol. 519, pp. 1953-1962, 2014.
- [11] 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*, vol. 222, p. 106353, 2020.

[12] H. D. Van, H.-B. Ly, T. D. Huan, S. N. Thai, and B. T. Pham, "Development of Artificial Neural Network for Prediction of Radon Dispersion Released from Sinquyen Mine, Vietnam," *Environmental Pollution*, p. 116973, 2021.

[13] A. Negarestani, S. Setayeshi, M. Ghannadi-Maragheh, and B. Akashe, "Layered neural networks based analysis of radon concentration and environmental parameters in earthquake prediction," *Journal of environmental radioactivity*, vol. 62, pp. 225-233, 2002.

[14] I. U. Sikder and T. Munakata, "Application of rough set and decision tree for characterization of premonitory factors of low seismic activity," *Expert Systems with Applications*, vol. 36, pp. 102-110, 2009.

[15] N. A. Zakaria and N. M. Noor, "Imputation methods for filling missing data in urban air pollution data formalyasia," *Urbanism. Arhitectura. Constructii*, vol. 9, p. 159, 2018.

[16] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, pp. 581-592, 1976.

[17] R. J. Little, "Regression with missing X's: a review," *Journal of the American statistical association*, vol. 87, pp. 1227-1237, 1992.

[18] S. Greenland and W. D. Finkle, "A critical look at methods for handling missing covariates in epidemiologic regression analyses," *American journal of epidemiology*, vol. 142, pp. 1255-1264, 1995.

[19] Y. C. Yuan, "Multiple imputation for missing data: Concepts and new development," in *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*, 2000.

[20] M. Fichman and J. N. Cummings, "Multiple imputation for missing data: Making the most of what you know," *Organizational Research Methods*, vol. 6, pp. 282-308, 2003.

[21] P. Li, E. A. Stuart, and D. B. Allison, "Multiple imputation: a flexible tool for handling missing data," *Jama*, vol. 314, pp. 1966-1967, 2015.

[22] G. L. Schlomer, S. Bauman, and N. A. Card, "Best practices for missing data management in counseling psychology," *Journal of Counseling psychology*, vol. 57, p. 1, 2010.

[23] J. A. Sterne, I. R. White, J. B. Carlin, M. Spratt, P. Royston, M. G. Kenward, A. M. Wood, and J. R. Carpenter, "Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls," *Bmj*, vol. 338, 2009.

- [24] J. M. Jerez, I. Molina, P. J. García-Laencina, E. Alba, N. Ribelles, M. Martín, and L. Franco, "Missing data imputation using statistical and machine learning methods in a real breast cancer problem," *Artificial intelligence in medicine*, vol. 50, pp. 105-115, 2010.
- [25] K. Lakshminarayan, S. A. Harp, R. P. Goldman, and T. Samad, "Imputation of Missing Data Using Machine Learning Techniques," in *KDD*, 1996, pp. 140-145.
- [26] U. Mital, D. Dwivedi, J. B. Brown, B. Faybishenko, S. L. Painter, and C. I. Steefel, "Sequential imputation of missing spatio-temporal precipitation data using random forests," *Frontiers in Water*, vol. 2, p. 20, 2020.
- [27] Y. Qin, S. Zhang, X. Zhu, J. Zhang, and C. Zhang, "Semi-parametric optimization for missing data imputation," *Applied Intelligence*, vol. 27, pp. 79-88, 2007.
- [28] W. Junger and A. P. De Leon, "Imputation of missing data in time series for air pollutants," *Atmospheric Environment*, vol. 102, pp. 96-104, 2015.
- [29] D. B. Rubin, "Multiple imputation for survey nonresponse," ed: New York: Wiley, 1987.
- [30] D. B. Rubin, "Statistical matching using file concatenation with adjusted weights and multiple imputations," *Journal of Business & Economic Statistics*, vol. 4, pp. 87-94, 1986.
- [31] K. Kleinke, "Multiple imputation by predictive mean matching when sample size is small," *Methodology*, 2018.
- [32] R. J. Little, "A test of missing completely at random for multivariate data with missing values," *Journal of the American statistical association*, vol. 83, pp. 1198-1202, 1988.
- [33] R. R. Andridge and R. J. Little, "A review of hot deck imputation for survey non-response," *International statistical review*, vol. 78, pp. 40-64, 2010.
- [34] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, 2000, pp. 1-15.
- [35] E. Z. Naeini and K. Prindle, "Machine learning and learning from machines," *The Leading Edge*, vol. 37, pp. 886-893, 2018.
- [36] W. Li, W. Hu, and A. Abubakar, "Machine learning and data analytics for geoscience applications—Introduction," *Geophysics*, vol. 85, pp. 1-4, 2020.
- [37] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *2014 IEEE symposium on computational intelligence in ensemble learning (CIEL)*, 2014, pp. 1-6.

- [38] L.-p. Jin and J. Dong, "Ensemble deep learning for biomedical time series classification," *Computational intelligence and neuroscience*, vol. 2016, 2016.
- [39] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, pp. 993-1001, 1990.
- [40] S. Hashem, "Optimal linear combinations of neural networks," *Neural networks*, vol. 10, pp. 599-614, 1997.
- [41] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Advances in neural information processing systems*, 1995, pp. 231-238.
- [42] D. W. Opitz and J. W. Shavlik, "Actively searching for an effective neural network ensemble," *Connection Science*, vol. 8, pp. 337-354, 1996.
- [43] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, pp. 139-157, 2000.
- [44] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5-32, 2001.
- [45] L. I. Kuncheva, *Combining pattern classifiers: methods and algorithms*: John Wiley & Sons, 2014.
- [46] G. I. Webb and Z. Zheng, "Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 980-991, 2004.
- [47] L. Breiman, "Arcing classifier (with discussion and a rejoinder by the author)," *The annals of statistics*, vol. 26, pp. 801-849, 1998.
- [48] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *The annals of statistics*, vol. 26, pp. 1651-1686, 1998.
- [49] S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?," *Machine learning*, vol. 54, pp. 255-273, 2004.
- [50] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, pp. 241-259, 1992.
- [51] L. Breiman, "Stacked regressions," *Machine learning*, vol. 24, pp. 49-64, 1996.
- [52] P. Smyth and D. H. Wolpert, *An evaluation of linearly combining density estimators via stacking*: Information and Computer Science, University of California, Irvine, 1998.

- [53] K. M. Ting and I. H. Witten, "Issues in stacked generalization," *Journal of artificial intelligence research*, vol. 10, pp. 271-289, 1999.
- [54] B. Clarke, "Comparing Bayes model averaging and stacking when model approximation error cannot be ignored," *Journal of Machine Learning Research*, vol. 4, pp. 683-712, 2003.
- [55] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," BROWN UNIV PROVIDENCE RI INST FOR BRAIN AND NEURAL SYSTEMS1992.
- [56] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with COTE: the collective of transformation-based ensembles," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, pp. 2522-2535, 2015.
- [57] R. Zhang, Z. Y. Dong, Y. Xu, K. Meng, and K. P. Wong, "Short-term load forecasting of Australian National Electricity Market by an ensemble model of extreme learning machine," *IET Generation, Transmission & Distribution*, vol. 7, pp. 391-397, 2013.
- [58] S. Tasnim, A. Rahman, A. M. T. Oo, and M. E. Haque, "Wind power prediction using cluster based ensemble regression," *International Journal of Computational Intelligence and Applications*, vol. 16, p. 1750026, 2017.
- [59] S. Džeroski, L. Todorovski, B. Zmazek, J. Vaupotic, and I. Kobal, "Modelling soil radon concentration for earthquake prediction," in *International Conference on Discovery Science*, 2003, pp. 87-99.
- [60] E. Alpaydin, *Introduction to machine learning*: MIT press, 2020.
- [61] I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective," *Artificial intelligence in medicine*, vol. 23, pp. 89-109, 2001.
- [62] I. Kononenko, I. Bratko, and M. Kukar, "Application of machine learning to medical diagnosis," *Machine Learning and Data Mining: Methods and Applications*, vol. 389, p. 408, 1997.
- [63] B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline, "Machine learning for medical imaging," *Radiographics*, vol. 37, pp. 505-515, 2017.
- [64] J. G. Richens, C. M. Lee, and S. Johri, "Improving the accuracy of medical diagnosis with causal machine learning," *Nature communications*, vol. 11, pp. 1-9, 2020.
- [65] M. Amrane, S. Oukid, I. Gagaoua, and T. Ensari, "Breast cancer classification using machine learning," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, 2018, pp. 1-4.
- [66] A. C. Tan and D. Gilbert, "Ensemble machine learning on gene expression data for cancer classification," 2003.

- [67] M. Leo, S. Sharma, and K. Maddulety, "Machine learning in banking risk management: A literature review," *Risks*, vol. 7, p. 29, 2019.
- [68] P. S. Patil and N. V. Dharwadkar, "Analysis of banking data using machine learning," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2017, pp. 876-881.
- [69] Y.-L. Chen, K. Tang, R.-J. Shen, and Y.-H. Hu, "Market basket analysis in a multiple store environment," *Decision Support Systems*, vol. 40, pp. 339-354, 2005.
- [70] R. Gangurde, B. Kumar, and D. S. Gore, "Building prediction model using market basket analysis," *Int. J. Innov. Res. Comput. Commun. Eng*, vol. 5, pp. 1302-1309, 2017.
- [71] C. Park, Park, and H., "A relationship between LDA and the generalized minimum squared error solution," *Journal on Matrix Analysis and Applications*, vol. 27, pp. 474-492, 2005.
- [72] Cover, T.M., and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform.Theory*, vol. IT-13, pp. 21-27, 1967.
- [73] V. N. Vapnik, "Introduction to Statistical Learning Theory," *IEEE transactions on neural networks*, vol. 10, pp. 988-999, 1979.
- [74] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, pp. 5-32, 2001.
- [75] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123-140, 1996.
- [76] R. Schapire, Freund, Bartlett, P., and Lee, "Boosting the margin: a new explanation for the effectiveness of voting methods," *Annals of Statistics* vol. 26, pp. 1651-1686, 1998.
- [77] R. E. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, pp. 197-227, 1990.
- [78] S.-T. Luo and B.-W. Cheng, "Diagnosing breast masses in digital mammography using feature selection and ensemble methods," *Journal of medical systems*, vol. 36, pp. 569-577, 2012.
- [79] D. Lavanya and K. U. Rani, "Ensemble decision tree classifier for breast cancer data," *International Journal of Information Technology Convergence and Services*, vol. 2, p. 17, 2012.
- [80] T. Gneiting and A. E. Raftery, "Weather forecasting with ensemble methods," *science*, vol. 310, pp. 248-249, 2005.
- [81] I. Maqsood, M. R. Khan, and A. Abraham, "An ensemble of neural networks for weather forecasting," *Neural Computing & Applications*, vol. 13, pp. 112-122, 2004.

- [82] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*: CRC press, 2012.
- [83] S. v. Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in R," *Journal of statistical software*, pp. 1-68, 2010.
- [84] R. M. Schouten, P. Lugtig, and G. Vink, "Generating missing values for simulation purposes: a multivariate amputation procedure," *Journal of Statistical Computation and Simulation*, vol. 88, pp. 2909-2930, 2018.
- [85] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R news*, vol. 2, pp. 18-22, 2002.
- [86] U. Kumar and V. Jain, "ARIMA forecasting of ambient air pollutants (O₃, NO, NO₂ and CO)," *Stochastic Environmental Research and Risk Assessment*, vol. 24, pp. 751-760, 2010.
- [87] Y. A. Ayturan, Z. C. Ayturan, and H. O. Altun, "Air pollution modelling with deep learning: a review," *International Journal of Environmental Pollution and Environmental Modelling*, vol. 1, pp. 58-62, 2018.
- [88] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature," *Geoscientific model development*, vol. 7, pp. 1247-1250, 2014.
- [89] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, pp. 1145-1159, 1997.
- [90] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, pp. 29-36, 1982.

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Adil Aslam MIR
Date of Birth :
Phone :
E-mail :

EDUCATION

High School : Government Pilot High School No.1, Pakistan
Bachelor : The University of Azad Jammu and Kashmir, Pakistan
Master Degree : The University of Azad Jammu and Kashmir, Pakistan

WORK EXPERIENCE

Lecturer : From May, 2016 to date (The University of AJ&K)

TOPICS OF INTEREST

- | | |
|----------------------------|-------------------------|
| - Machine Learning | - Data Mining |
| - Artificial Intelligenece | - Computer Architecture |

SKILLS AND STRENGTHS

- Expertise in programming languages (R language, C/C++, Visual C++, Asp.Net, C#, Assembly Language).
- Machine Learning and Data Mining based Decision Support Systems
- Command on Advanced level MS Office tools
- Strong analytical skills
- Ability to work independently
- An ambitious, motivated, hardworking, and well-disciplined involved having the capacity of working in a brilliant environment.
- Leadership and Management Skills

RESEARCH PUBLICATIONS JOURNAL

▪ Journal Publications

- Mir AA, Kearfott KJ, Çelebi FV, Rafique M. Imputation by feature importance (IBFI): A methodology to envelop machine learning method for imputing missing patterns in time series data. PLoS One. 2022 Jan 13;17(1):e0262131. doi: 10.1371/journal.pone.0262131. PMID: 35025953; PMCID: PMC8758196.
- Mir, Adil Aslam, et al. "Anomaly Classification for Earthquake Prediction in Radon Time Series Data Using Stacking and Automatic Anomaly Indication Function." Pure and Applied Geophysics (2021): 1-15. (Impact Factor: 2.335)
- Rafique, Muhammad, Aleem Dad Khan Tareen, Adil Aslam Mir, Malik Sajjad Ahmed Nadeem, Khawaja M. Asim, and Kimberlee Jane Kearfott. "Delegated Regressor, A Robust Approach for Automated Anomaly Detection in the Soil Radon Time Series Data." Scientific Reports 10, no. 1 (2020): 1-11. (Impact Factor: 5.133)
- Shah, Syed Taimoor Hussain, Shahzad Ahmad Qureshi, Syed Adil Hussain Shah, Arslan Amjad, Adil Aslam Mir, Amal Alqahtani, David A. Bradley, Mayeen Uddin Khandaker, Mohammad Rashed Iqbal Faruque, and Muhammad Rafique. "A Novel Hybrid Learning System Using Modified Breaking Ties Algorithm and Multinomial Logistic Regression for Classification and Segmentation of Hyperspectral Images." Applied Sciences 11, no. 16 (2021): 7614. (Impact Factor: 2.679)
- Syed Ali Abbas, Adil Aslam Mir, Aqeel Ur Rehman ; Wajid Arshad Abbasi ; Saeed Arif ; Zaki Hassan, "K-means and K-medoids: Cluster Analysis on Birth Data Collected in City Muzaffarabad, Kashmir," in *IEEE Access*, doi: 10.1109/ACCESS.2020.3014021. (Impact Factor: 3.745)
- Lone, Kashif Javed, Lal Hussain, Sharjil Saeed, Adil Aslam, Asim Maqbool, and Faisal Mehmood Butt. "Detecting basic human activities and postural transition using robust machine learning techniques

by applying dimensionality reduction methods." *Waves in Random and Complex Media* (2021): 1-26. (Impact Factor: 4.853)

- Khan, Abdul Razzaq, Adil Aslam Mir, Sharjil Saeed, Muhammad Rafique, Khawaja M. Asim, Talat Iqbal, Abdul Jabbar, and Saeed Ur Rahman. "Classification of rocks radionuclide data using machine learning techniques." *Acta Geophysica* (2018): 1-7. (Impact Factor: 2.054)
- Mir, Adil Aslam, Rabia Riaz, Sobia Kazmi, Sajjad Ahmed Nadeem, and Sharjil Saeed. "Comparative Analysis of Boosting with other Machine Learning Methods for classifying Labeled Wireless Sensor Network Data." *International Journal of Computer Science and Information Security* 14, no. 9 (2016): 917.