

DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**DATA MINING AND MACHINE LEARNING IN
ENVIRONMENTAL INFORMATICS**

by
Göksu TÜYSÜZOĞLU

January, 2021
İZMİR

DATA MINING AND MACHINE LEARNING IN ENVIRONMENTAL INFORMATICS

**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of
Philosophy in Computer Engineering**

**by
Göksu TÜYSÜZOĞLU**

**January, 2021
İZMİR**

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**DATA MINING AND MACHINE LEARNING IN ENVIRONMENTAL INFORMATICS**” completed by **GÖKSU TÜYSÜZOĞLU** under supervision of **ASSOC. PROF. DR. DERYA BİRANT** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Derya BİRANT

Supervisor

Prof. Dr. Alp KUT

Thesis Committee Member

Prof. Dr. Ayşegül PALA

Thesis Committee Member

Assoc. Prof. Dr. Aytuğ ONAN

Examining Committee Member

Assoc. Prof. Dr. Tuğba ÖZACAR
ÖZTÜRK

Examining Committee Member

Prof. Dr. Özgür ÖZÇELİK

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

At the very outset of this thesis, I would like to express my immense gratitude to my supervisor, Assoc. Prof. Dr. Derya BİRANT, for having contributed me a lot throughout this study, never withholding her support for me, and always showing me the right way with her innovative ideas and knowledge. Furthermore, I am grateful to Prof. Dr. Alp KUT for his guidance, cooperation, and encouragement for the completion of this thesis. I am also ineffably indebted to my committee member, Prof. Dr. Ayşegül PALA for all the information she gave on environmental issues, on-site guidance, keen interest, and all her suggestions. I would also like to thank Volkan KIRANOĞLU, who has made great efforts in some parts of this thesis, for his contributions.

At last, I would like to express my sincere thanks to my family and friends who have always supported me and kept my morale and energy high at the stage of writing this thesis and during my whole life.

Göksu TÜYSÜZOĞLU

DATA MINING AND MACHINE LEARNING IN ENVIRONMENTAL INFORMATICS

ABSTRACT

Nowadays, environmental informatics is one of the fields where data mining and machine learning techniques for data processing & analysis are frequently used today. In this thesis, several case studies including environmental data on different subjects were conducted for different tasks (classification, regression, clustering, association rule mining, time series) under data mining and machine learning.

Examining the data obtained from air quality monitoring stations and meteorological data, determining which station type (industrial, rural, or urban) a particular air quality monitoring station belongs to, and making the most accurate clustering application according to the similarities of the stations are a certain part of the experimental studies. In addition, the relations between air pollutants and meteorological factors were examined in detail within the scope of association rule mining analysis, and the relationship rules were drawn. On the other side, regression and time series analysis for soil temperature prediction at various depths were performed.

The proposed methods to handle air and soil-related problems are as follows: (i) *Enhanced Bagging (eBagging)* which is a new ensemble learning method, (ii) *Soil Temperature Prediction via Self-training (STST)* which is a new semi-supervised learning method, (iii) *Majority Voting Based Multi-Task Clustering (MV-MTC)* which is proposed to consider multiple tasks jointly, (iv) *WARM* which is a new type of weighted association rule mining, and (v) *Multi-View Multi-Depth Soil Temperature Prediction (MV-MD-STP)* which is a new soil management framework. Considering the experimental results, it is clear that the newly proposed methods performed better than the other state-of-the-art methods they were compared to and offered effective solutions to environmental problems.

Keywords: Data mining, machine learning, environmental informatics, classification, regression, clustering, association rule mining, time series



ÇEVRESEL ENFORMATİKTE VERİ MADENCİLİĞİ VE MAKİNE ÖĞRENMESİ

ÖZ

Çevresel enformatik, günümüzde, veri işleme ve analizi için veri madenciliği ve makine öğrenimi tekniklerinin sıklıkla kullanıldığı alanlardan biridir. Bu tezde, farklı konulardaki çevresel verileri içeren birkaç vaka çalışması, veri madenciliği ve makine öğrenmesi kapsamında farklı görevlerde (sınıflandırma, regresyon, kümeleme, birliktelik kuralı madenciliği, zaman serileri) incelenmiştir.

Hava kalitesi izleme istasyonlarından elde edilen veriler ve meteorolojik veriler incelenerek belirli bir hava kalitesi izleme istasyonunun hangi istasyon tipine (endüstriyel, kırsal veya kentsel) ait olduğunun belirlenmesi, istasyonların benzerliklerine göre en doğru şekilde kümeleme uygulamasının yapılması deneysel çalışmaların belirli bir kısmını oluşturmaktadır. Ayrıca, hava kirletici maddeler ve meteorolojik faktörler arasındaki ilişkiler birliktelik analizi kapsamında detaylı bir şekilde incelenerek ilişki kuralları çıkarılmıştır. Diğer yandan, çeşitli derinliklerde toprak sıcaklığı tahmini için regresyon ve zaman serisi analizleri yapılmıştır.

Hava ve toprak kaynaklı problemleri çözmek için önerilen yöntemler şu şekildedir: (i) yeni bir topluluk öğrenmesi yöntemi olan *Geliştirilmiş Torbalama (eBagging)*, (ii) yarı denetimli yeni bir yöntem olan *Kendi Kendine Eğitim Yoluyla Toprak Sıcaklığı Tahmini (STST)*, (iii) birden fazla görevi birlikte ele almak için önerilen *Çoğunluk Oylamasına Dayalı Çok Görevli Kümeleme (MV-MTC)*, (iv) yeni bir ağırlıklı birliktelik kuralı madenciliği türü olan *WARM* ve (v) yeni bir toprak yönetim uygulaması olan *Çok Görünümlü Çok Derinlikli Toprak Sıcaklığı Tahmini (MV-MD-STP)*'dir. Deneysel sonuçlara bakıldığında, önerilen yeni yöntemlerin karşılaştırıldıkları son teknoloji yöntemlerden daha iyi performans gösterdiği ve çevre sorunlarına etkili çözümler sunduğu açıktır.

Anahtar Kelimeler: Veri madenciliđi, makine öğrenmesi, çevresel enformatik, sınıflandırma, regresyon, kümeleme, birliktelik kuralı madenciliđi, zaman serileri



CONTENTS

	Page
Ph.D. THESIS EXAMINATION RESULT FORM	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT.....	iv
ÖZ	vi
LIST OF FIGURES	xiii
LIST OF TABLES	xv
 CHAPTER ONE – INTRODUCTION	 1
1.1 General	1
1.2 Purpose of the Thesis	2
1.3 Main Contributions of the Thesis.....	3
1.4 Outline of the Thesis	5
 CHAPTER TWO – LITERATURE REVIEW	 7
2.1 Recent Studies on Environmental Informatics	7
2.1.1 According to Environmental Subjects	7
2.1.1.1 Air	9
2.1.1.2 Water.....	16
2.1.1.3 Soil	17
2.1.1.4 Energy	19
2.1.1.5 Other Environmental Subjects	20
2.1.2 According to the Data Mining and Machine Learning Task	28
2.1.2.1 Classification.....	28
2.1.2.2 Regression.....	28
2.1.2.3 Clustering.....	29
2.1.2.4 Association Rule Mining	30
2.1.2.5 Time Series Analysis	31

CHAPTER THREE – CLASSIFICATION STUDIES IN ENVIRONMENTAL INFORMATICS..... 32

3.1 Material and Methods.....	32
3.1.1 Classification Algorithms	32
3.1.1.1 Support Vector Machines	33
3.1.1.2 Decision Tree	33
3.1.1.3 K-Nearest Neighbours	34
3.1.1.4 Naïve Bayes	35
3.1.1.5 Neural Networks	35
3.1.2 Ensemble Learning	36
3.1.2.1 Elements of an Ensemble Classifier	36
3.1.2.2 Ensemble Strategies	37
3.1.2.2.1 Bagging	40
3.1.2.2.2 Boosting	41
3.1.2.2.3 Voting.....	41
3.1.2.3 The Proposed Method: Enhanced Bagging (eBagging).....	41
3.1.2.3.1 Error-based Bootstrapping	42
3.1.2.3.2 Enhanced Bagging (eBagging).....	45
3.1.2.4 Advantages of Ensemble Learning in Environmental Informatics..	48
3.1.2.5 Challenges of Ensemble Learning in Environmental Informatics...	50
3.2 Case Study 1 – Ensemble Methods in Environmental Data Mining	51
3.2.1 Dataset Description.....	52
3.2.2 Experimental Study and Results.....	52
3.3 Case Study 2 – Classification of Air Quality Network based on Meteorological and Pollutant Factors	56
3.3.1 Dataset Description.....	57
3.3.2 Experimental Study and Results.....	59

CHAPTER FOUR – REGRESSION STUDIES IN ENVIRONMENTAL INFORMATICS..... 62

4.1 Material and Methods.....	62
4.1.1 Regression Algorithms	62
4.1.1.1 Support Vector Regression	63
4.1.1.2 K-Nearest Neighbours Regression.....	64
4.1.1.3 Decision Tree Regression	64
4.1.1.4 Extra Tree Regression.....	64
4.1.1.6 Extreme Gradient Boosting.....	65
4.1.1.7 Multiple Linear Regression.....	65
4.1.2 The Proposed Method: Semi-supervised Soil Temperature Prediction via Self-Training (STST)	66
4.2 Case Study - Soil Temperature Prediction using Semi-supervised Learning .	69
4.2.1 Dataset Description.....	70
4.2.2 Experimental Study and Results.....	71
 CHAPTER FIVE – CLUSTERING STUDIES IN ENVIRONMENTAL INFORMATICS.....	 75
5.1 Material and Methods.....	75
5.1.1 Clustering Algorithms	75
5.1.1.1 K-Means.....	76
5.1.1.2 Expectation Maximization	76
5.1.1.3 Canopy	77
5.1.1.4 Farthest First	78
5.1.1.5 Hierarchical Clustering	78
5.1.2 Proposed Method: Majority Voting Based Multi-Task Clustering (MV- MTC).....	79
5.2 Case Study - Majority Voting Based Multi-Task Clustering of Air Quality Monitoring Network in Turkey	88
5.2.1 Dataset Description.....	90
5.2.2 Experimental Study and Results.....	93
5.2.3 Discussions	97

CHAPTER SIX – ASSOCIATION RULE MINING STUDIES IN ENVIRONMENTAL INFORMATICS 99

6.1 Material and Methods.....	99
6.1.1 Association Rule Mining.....	99
6.1.1.1 Weighted Association Rule Mining.....	102
6.1.2 The Proposed Method: Weighted Association Rule Mining (WARM)	108
6.2 Case Study - Air Pollution Monitoring in Smart Cities using Weighted Association Rule Mining.....	112
6.2.1 Dataset Description.....	113
6.2.2 Experimental Study and Results.....	114

CHAPTER SEVEN – TIME SERIES STUDIES IN ENVIRONMENTAL INFORMATICS..... 123

7.1 Material and Methods.....	123
7.1.1 Time Series Algorithms.....	123
7.1.1.1 Autoregressive Integrated Moving Average with Explanatory Variable (ARIMAX).....	124
7.1.1.2 Seasonal Autoregressive Integrated Moving Average with Explanatory Variable (SARIMAX).....	125
7.1.2 The Proposed Method: Multi-View Multi-Depth Soil Temperature Prediction (MV-MD-STP)	126
7.2 Case Study - Comparison of Machine Learning and Time Series Methods for Multi-View Multi-Depth Soil Temperature Prediction.....	128
7.2.1 Dataset Description.....	130
7.2.2 Experimental Study and Results.....	131
7.2.3 Discussions	143

CHAPTER EIGHT – CONCLUSION AND FUTURE WORK..... 145

8.1 Conclusion.....	145
---------------------	-----

8.2 Future Work	146
REFERENCES.....	148



LIST OF FIGURES

	Page
Figure 1.1 Interdisciplinary structure of environmental data mining and machine learning (EDML).....	2
Figure 2.1 The number of publications between the years of 2010 and 2020	8
Figure 2.2 The number of publications in terms of document type between the years of 2010 and 2020.....	9
Figure 2.3 The number of publications according to the main areas (classification & prediction, clustering, association rules and sequential patterns) in data mining.....	31
Figure 3.1 Different ensemble strategies	38
Figure 3.2 An example of eBootstrapping	44
Figure 3.3 The general framework of Enhanced Bagging (eBagging) algorithm.....	45
Figure 3.4 The pseudo-code of enhanced bagging (eBagging).....	47
Figure 3.5 Comparison of single and ensemble classifiers in terms of classification accuracies	53
Figure 3.6 Comparisons of the applied ensemble models in terms of accuracy	61
Figure 4.1 The pseudo-code of soil temperature prediction via self-training (STST).....	68
Figure 4.2 The general framework of the proposed “Soil temperature prediction via self-training (STST)” model.....	69
Figure 4.3 The comparison of the applied methods under self-training framework in terms of R ² values.....	74
Figure 5.1 The general MV-MTC clustering scheme	86
Figure 5.2 The algorithm of the MV-MTC clustering	87
Figure 5.3 Results from the implementation of multi-task clustering obtained by KM++	96
Figure 6.1 The general framework with potential impacts of the proposed intelligent environmental system.....	111
Figure 6.2 Itemset used in the transactions of air pollution dataset	114
Figure 6.3 The variations in the number of k-itemsets found by WARM model	119

Figure 6.4 The comparisons of WARM and Apriori according to the number of rules in terms of different support thresholds at 0.6 confidence.....	120
Figure 6.5 The comparisons of WARM and Apriori according to the number of rules in terms of different support thresholds at 0.9 confidence.....	121
Figure 6.6 The comparisons of WARM and Apriori according to the number of rules in terms of different support thresholds at 0.9 confidence with doubled item weights	122
Figure 7.1 The proposed multi-view multi-depth soil temperature prediction framework	127
Figure 7.2 The pseudo-code of the proposed method.....	128
Figure 7.3 The study area for soil temperature prediction.....	130
Figure 7.4 Correlations among the used parameters for three states of the U.S.	135
Figure 7.5 Comparisons of the average MSE values (a) and average MAE values (b) of the applied methods according to stations at different depths	141
Figure 7.6 Comparisons of the average RMSE values (c) and average R2 values (d) of the applied methods according to stations at different depths	142

LIST OF TABLES

	Page
Table 2.1 Several data mining and machine learning studies taking “air” as the main subject.....	21
Table 2.2 The summary table of the studies taking “air pollution in Turkey” as the main subject.....	22
Table 2.3 Several data mining and machine learning studies taking “water” as the main subject.....	23
Table 2.4 Several data mining and machine learning studies taking “soil” as the main subject.....	24
Table 2.5 Several data mining and machine learning studies taking “energy” as the main subject.....	25
Table 2.6 Several data mining and machine learning studies taking other environmental subjects as the main topic.....	26
Table 3.1 Characteristic of different ensemble classifiers	39
Table 3.2 Comparison table of eBagging, Bagging, Random Forest and Boosting techniques	48
Table 3.3 Environmental datasets and their characteristics	52
Table 3.4 Optimum classifier parameters corresponding to each dataset.....	54
Table 3.5 Precision (PRE), recall (REC) and f-measure (F-MEA) results using 10-fold cross-validation for respective algorithms in each dataset	55
Table 3.6 The selected air quality monitoring stations (AQMS) and their features ..	58
Table 3.7 The results of performance metrics of the applied algorithms	60
Table 4.1 A part of the data set used in the experiments	71
Table 4.2 The comparison of the applied methods under self-training framework in terms of MSE values	72
Table 4.3 The comparison of the mean values of the applied methods in terms of MSE values under self-training framework at different soil depths.....	74
Table 5.1 The summary table of the data mining studies taking “Multi-Task Clustering” as the main subject	82
Table 5.2 Assignments of cluster labels under different tasks.....	83

Table 5.3 An example showing MV-MTC algorithm step by step.....	84
Table 5.4 Assignments of final cluster labels under different tasks and the output of MV-MTC algorithm	85
Table 5.5 The selected air quality monitoring stations (AQMS) and their features ..	92
Table 5.6 The mean weight values of each cluster group and their intra-cluster labels obtained by KM++	94
Table 5.7 The results of the performance evaluation in terms of SSE values	95
Table 5.8 Comparisons of different clustering methods in terms of execution time (in seconds)	97
Table 6.1 An example itemset and transactions.....	101
Table 6.2 Recent studies taking WARM methods as the main methodology.....	105
Table 6.3 An example scenario using warm to find frequent itemsets	107
Table 6.4 Weight assignment to item values	110
Table 6.5 Sample dataset.....	114
Table 6.6 National air quality index breakpoints for the air pollutants (SO ₂ , NO ₂ , O ₃ and PM ₁₀).....	115
Table 6.7 Breakpoints determined for the meteorological parameters	115
Table 6.8 The variations of the k-itemsets according to different support thresholds	119
Table 7.1 Comparisons of the applied methods in different stations at the depth 40 inch	136
Table 7.2 Comparisons of the applied methods in different stations at the depth 20 inch	137
Table 7.3 Comparisons of the applied methods in different stations at the depth 8 inch	138
Table 7.4 Comparisons of the applied methods in different stations at the depth 4 inch	139
Table 7.5 Comparisons of the applied methods in different stations at the depth 2 inch	140

CHAPTER ONE

INTRODUCTION

1.1 General

The use of information as a value and power has become so important in our time as we have very large data stores. New methods to process big data are constantly being developed and implemented to take a closer look and benefit from such a large scale of data. By applying data mining and machine learning on large data stores, pattern analysis is carried out beyond simple analysis to explore trends. It is a tool for obtaining the information needed to solve existing problems, make critical decisions, or make predictions for the future.

Data coming from environmental issues are one of the major areas to handle, interpret, and analyze to get useful knowledge, as well. Environmental data mining and machine learning (EDML) is defined as the identification of interesting structures in a large amount of environmental data, where structure designates patterns, rules, predictive models, and relationships among the data. It is an interdisciplinary area of both computer and environmental sciences, including but not limited to environmental information management systems, decision support systems, environmental pattern recognition, recommender systems, environmental big data analytics, and so on. EDML can be drawn as a combination of three main areas: data mining (DM), machine learning (ML), and environmental science (Figure 1.1). ML in environmental science is learning-driven, meaning that machines teach themselves to recognize patterns by analyzing environmental big data, whereas, in contrast, DM is discovery-driven, meaning that patterns are automatically discovered from environmental big data. DM uses many ML methods, including ensemble learning methods.

Data mining and machine learning methods are applied to different environmental subjects, such as air, water, soil, plant, forests, climate, noise, rainfall, energy, as well as living organisms. In this thesis, solutions to several environmental case studies are

offered by introducing new methods under different EDML tasks such as classification, regression, clustering, association rule mining, and time series.

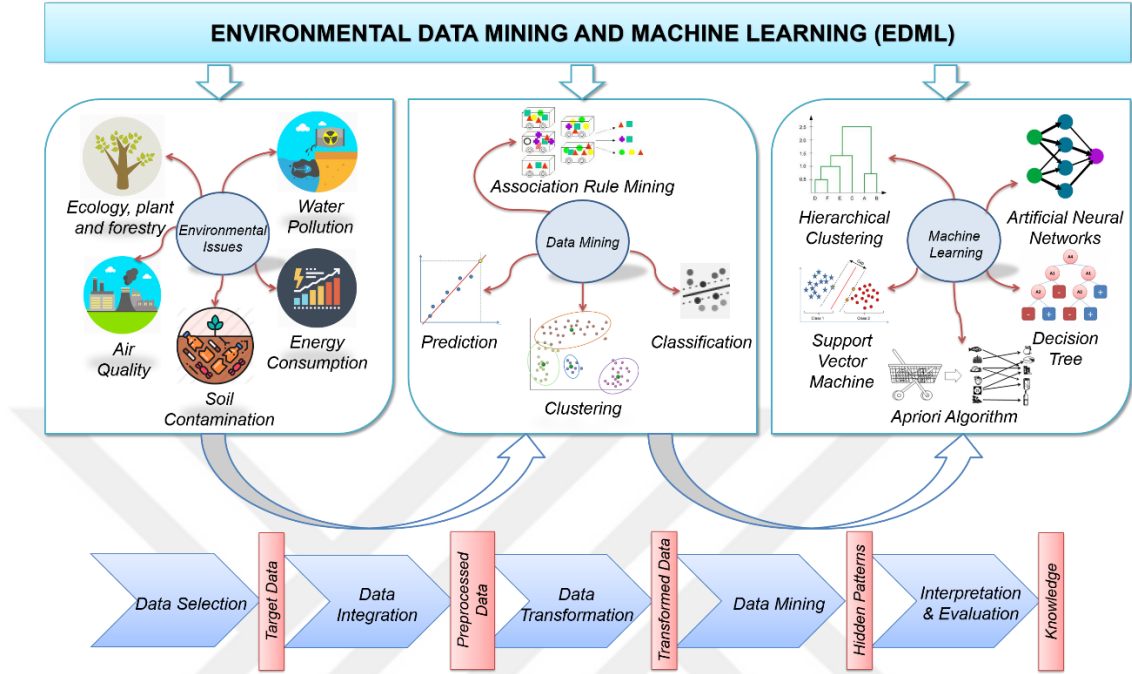


Figure 1.1 Interdisciplinary structure of environmental data mining and machine learning (EDML)

1.2 Purpose of the Thesis

The amount of data in the environmental field is increasing day by day, different methods are used and developed for their accurate analysis and processing. In this thesis, taking advantage of EDML new methodologies are designed and presented for offering solutions to different environmental problems especially including air quality control and soil management.

In this study, particularly, the following research questions are addressed:

1. What kind of environmental problems are solved using DM and ML techniques?
2. How DM and ML-based environmental studies can be grouped according to their subjects?
3. Which DM and ML methods (classification, clustering, etc.) are generally used to handle environmental problems?

4. Which DM and ML algorithms (neural network, decision tree, k-means, etc.) are generally used to deal with environmental issues?
5. What innovative data mining and machine learning methods can be developed for environmental problems?
6. Which performance criteria are generally used to measure the performance of ML models?
7. Which type of publications (article, book chapter, conference paper, etc.) were generally published in which country and when? In which subject were these publications?
8. What are the advantages of using DM and ML for environmental issues?
9. What are the main challenges of DM and ML that are faced in environmental studies?
10. What are the possible future directions for research in this field?

1.3 Main Contributions of the Thesis

The main contributions are nine-folds as below:

First, recent literature studies in environmental informatics are examined and discussed by showing how data mining and machine learning are applied in the environmental area and which methodologies are mostly used.

Second, environmental subjects such as air quality control and soil management are comprehensively analyzed using various data mining and machine learning techniques by presenting various methodologies from the point of computer science.

Three, five different data mining studies (classification, regression, clustering, association rule mining, and time series) were performed on environmental data to obtain valuable knowledge to increase the quality and productivity of environmental subjects.

Four, the proposed methods in classification, regression, clustering, association rule mining, and time-series tasks can assist authorized personnel in charge of air quality control or soil management to make correct decisions when they are unsure about applying a policy. The proposed methods are as follows:

- eBagging: Enhanced Bagging (Classification)
- STST: Soil Temperature Prediction via Self-training (Regression)
- MV-MTC: Majority Voting Based Multi-Task Clustering (Clustering)
- WARM: Weighted Association Rule Mining (Association Rule Mining)
- MV-MD-STP: Multi-View Multi-Depth Soil Temperature Prediction (Time Series Analysis)

Five, a novel ensemble learning technique called eBagging that is the improved version of bagging is introduced with promising results in the classification task of air quality monitoring sites and proven to be the leading one compared to other well-known ensemble learning methods.

Six, one of the semi-supervised learning techniques called self-training is adapted to solve an environmental problem by providing an accurate prediction of the soil temperature at several depths. The general approach to increase the training set of soil-related parameters is explained when there are very few samples.

Seven, an application constructed with a new multi-task clustering method based on majority voting is proposed in the optimal design of an air quality monitoring network. Similar monitoring sites with the same characteristics on the subject of air pollutant factors are kept together in the same cluster while each cluster is very different from the other.

Eight, the noteworthy relationships among air pollutants and meteorological parameters are examined by the newly presented association rule mining technique, called WARM. The significant part is to import item weights in terms of their impact on air pollution when constructing the Apriori method.

Lastly, a new type of multi-view learning method was developed and used for soil temperature prediction at different depths. It is the first time that meteorological and soil-related parameters are separately examined in two different views, and then taking their joint decision by taking the average.

1.4 Outline of the Thesis

This thesis comprises of eight chapters and the remaining parts of the thesis are organized as follows:

In Chapter 2, the recent studies in environmental informatics are presented by examining the subject under five environmental subjects (air, water, soil, energy, and living organisms) and according to data mining and machine learning tasks (classification, regression, clustering, association rule mining, and time series analysis).

In Chapter 3, two different case studies for the classification task are given. The first study shows the applicability of ensemble learning techniques in the environmental area using various classifiers as base learners. In the second study, a novel ensemble learning method called “Enhanced Bagging (eBagging)” is introduced and applied to classify the sites of an air quality monitoring network.

In Chapter 4, a regression case study for soil temperature prediction is explained in detail. The proposed model (STST) based on self-training is stated in addition to the applied regression algorithms. Experimental study and results are given as well.

In Chapter 5, a new clustering method called “Majority voting based multi-task clustering (MV-MTC)” is proposed to cluster the sites of an air quality monitoring network optimally. Furthermore, other clustering methods used in the experimental studies are also briefly explained.

In Chapter 6, Apriori based weighted association rule mining (WARM) method is presented as an adaptation to the environmental case study in order to discover useful and outstanding association rules among air quality monitoring data and meteorological parameters.

In Chapter 7, a novel soil temperature prediction technique (MV-MD-SPT) based on “Multi-View Multi-Depth Learning” combined with time series and machine learning methods is introduced in a new case study. Two multivariate time series methods including ARIMAX and SARIMAX are also expressed comprehensively.

Finally, in Chapter 8, concluding remarks of this thesis and future directions are given.

CHAPTER TWO

LITERATURE REVIEW

In this chapter, data mining and machine learning studies presented for environmental informatics are discussed in detail. This chapter categories DM and ML studies according to the environmental subjects such as air, water, soil, ecology, and living organisms for describing the current situation more clearly. Existing solutions are comprehensively discussed according to various aspects, including tasks (classification, clustering, association rule mining, anomaly detection, sequential pattern mining, etc.), algorithms (neural network, decision tree, k-means, etc.), performance metrics (accuracy, precision, etc.) and learning types (ensemble learning, etc.). Some statistics related to recent studies between the years of 2010 and 2020 are also given from different perspectives ‘according to environmental subjects’ and ‘according to the applied data mining and machine learning task’.

2.1 Recent Studies on Environmental Informatics

Data mining and machine learning methods have been applied to different environmental subjects, such as air, water, soil, plant, forests, climate, noise, rainfall, energy, as well as living organisms. In this study, the literature review on EDML have been examined under two groups: (i) according to the environmental subjects (‘air’, ‘water’, ‘soil’, ‘energy’, ‘living organisms’, and others), and (ii) according to the data mining and machine learning tasks (‘classification’, ‘regression’, ‘clustering’, ‘association rule mining’, and ‘time series analysis’). They are to be explained in this section.

2.1.1 According to Environmental Subjects

Many publications considering environmental informatics have been investigated in this study and to demonstrate the current interest in ‘EDML’, Figure 2.1 shows the number of publications that were returned from Scopus search engine between the years of 2010 and 2020 by taking six main subjects ‘air’, ‘water’, ‘soil’, ‘energy’,

‘rainfall’, and ‘living organisms’ into consideration. The search terms were determined as ‘Machine Learning’ and ‘Data Mining’ - ‘Environmental’ and ‘Energy’ while searching the publications related to ‘Energy’ in the title/abstract/keywords sections in each year from 2010 to 2020 and it was done in the same way for other subjects. As can be seen, the most common studies have been done on the subjects of ‘water’ (1161 in total), ‘energy’ (848 in total), and ‘air’ (673 in total). The research count mostly increases as time goes by in a parallel way.

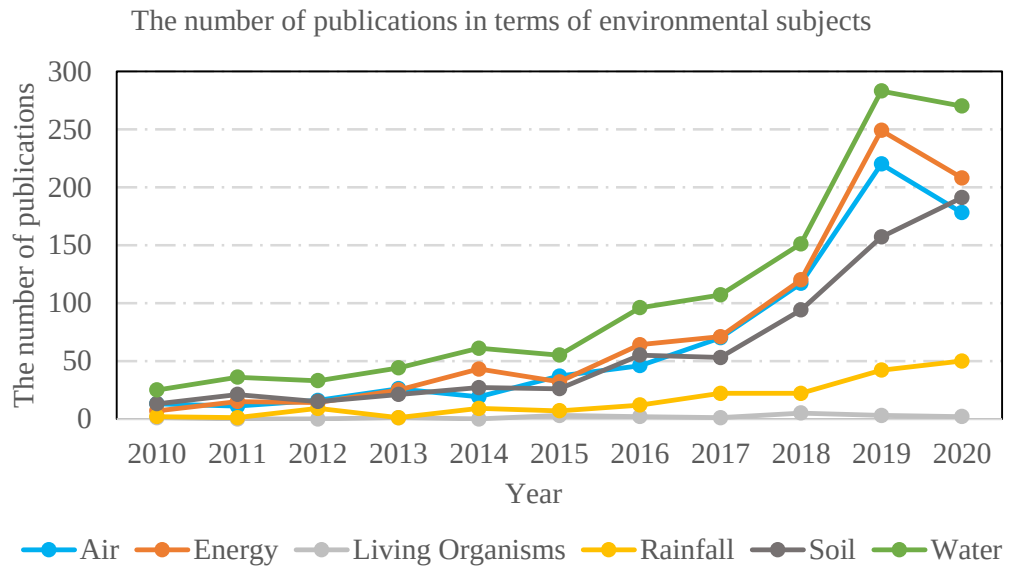


Figure 2.1 The number of publications between the years of 2010 and 2020

Figure 2.2 indicates the number of publications in terms of document type (conference paper, book chapter, article, book, etc.) in the same time interval for the same six subjects. It is clear from the findings that most of the literature studies have been done in the ‘article’ and ‘conference paper’ formats, respectively. In the following parts, each subject is taken at hand, separately.

In most of the researches about the subject of environmental informatics, ensemble learning takes place due to its significant performance improvement compared to single learners. Ensemble learning is a machine learning technique where multiple learners are trained to solve the same problem and their predictions are combined into a single output that probably has better performance on average than any individual

ensemble member. The fundamental idea is to combine weak learners into one, a strong learner, which has a better generalization error and is less sensitive to overfitting in the presence of noise or small sample size. This is because different classifiers can sometimes misclassify different patterns and accuracy can be improved by combining the decisions of complementary classifiers. Now, each environmental subject is to be handled one by one under five main categories and by indicating whether any ensemble strategy is performed.

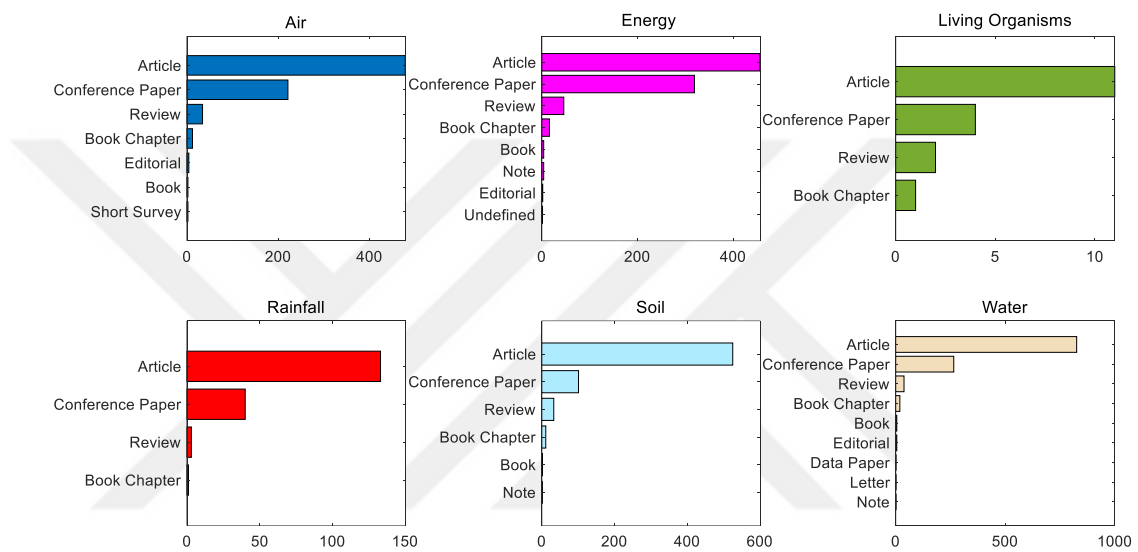


Figure 2.2 The number of publications in terms of document type between the years of 2010 and 2020

2.1.1.1 Air

One of the most significant environmental subjects influencing the life standards of living beings air. Especially, air pollution has negative impacts on human health, range of vision, materials, plants, and animal health. Smoke, a gas mixture of solid and liquid particles resulting from non-burned carbon materials such as solid fuels and fuel oil, is a variety of air pollution and has a reducing effect on the range of visibility. Air pollution has a destructive and disturbing effect on artistic and architectural structures. On plants, they can be lethal and prevent their growth. For this reason, air pollution is both damaging to the health of living things and economically damaging. Because of

the importance of the matter, many studies are performed to keep air quality at acceptable levels.

Several environmental data mining studies mentioned so far considering ‘air’ are compared in Table 2.1. In this table, the description of the studies, the year they were performed, algorithms that were used in the studies, the type of data mining task, performance measure to evaluate the applied methods, and ensemble strategy if used are listed. Besides, if more than one algorithm is presented and compared with each other, the most successful one is indicated with bold font.

Four major data mining tasks ‘classification & prediction’, ‘anomaly detection’, ‘clustering’, and ‘association rules & sequential patterns’ have been implemented in environmental studies. The most work considering air is done by either classifying (Miskell, Pattinson, Weissert, & Williams, 2019; Reid et al., 2019; Pucer, Pirš, & Štrumbelj, 2018; Mad Saad et al., 2017; Chen et al., 2016), predicting (Pucer et al., 2018; Xu, Wang, Schmidt, Adams, & Hatzopoulou, 2020; Li, Cui, Meng, Zhao, & Fu, 2019; Franceschi, Cobo, & Figueredo, 2018; Lei & Wan, 2012) and clustering (Xu et al., 2020; Franceschi et al., 2018; Bougoudis, Demertzis, & Iliadis, 2016) air pollutants according to their similarities or considering labeled historical data, or finding associations between them (Zhang, Yang, & Li, 2020; Yang, Huang, & Li, 2018). Some studies also deal with discovering outliers using anomaly detection methods (Chen et al., 2018; Harrou, Dairi, Sun, & Senouci, 2018).

Ensemble strategies are represented with four numbers in Table 2.1: ‘1 – Bagging, ‘2 – Random forest’, ‘3 – Boosting’ and ‘4 – Voting, stacking, or tiering’. Bagging and random forest are the mostly applied ensemble learning techniques in the current studies to boost the performance of single learners. Bagging is well-adapted to develop models for solving environmental problems. For example, it has been utilized to forecast the air pollution level of a region (Chen et al., 2016; Lei & Wan, 2012). As the second type of ensemble learning strategy, RF algorithm has also been applied for classifying environmental data e.g. to fit models for both air pollutants and

atmospheric parameters (Pucer et al., 2018; Li et al., 2019; Bougoudis et al., 2016; Lima, Cannon, & Hsieh, 2013).

Joly & Peuch (2012) proposed a study for the classification of European air quality monitoring sites using the past time series of O₃, NO₂, NO, SO₂, and PM₁₀ observation values over the period 2002-2009. Rural and urban sites were homogeneously discriminated against by applying a Linear Discriminant Analysis method. When the classes obtained for different pollutants were compared, an interesting consistency was found between the separate classifications.

Clustering applications were applied in various studies to model air quality monitoring networks (AQMN). In the study of Barrero, Orza, Cabello, & Cantón (2015), the air quality monitoring stations (AQMSs) of Basque country were categorized according to their hidden common characteristics using the variations of PM₁₀ concentrations from 2005 to 2012. Instead of only classifying based on station type (urban/suburban/rural or industrial/traffic/background), they aimed to discover other common characteristics that are shared by the stations so that these stations can be classified in the same group. They used the autocorrelation functions of daily and hourly measured air pollution data for cluster analysis and according to the analyzed average diurnal-weekly patterns, the stations were grouped using the k-means clustering algorithm. The resulting clusters gave significant information about the dispersion and emission dynamics of the stations in the same group.

In the study of Gómez-Losada (2017), hidden Markov models were first implemented to model the AQMN, and then Smacof multidimensional scaling was applied using background and ambient pollution data. Another study (Alvarez-Guerra, Molina, Viguri, & Alvarez-Guerra, 2011) proposed a SOM based artificial neural networks method to cluster 517 monitoring stations in Spain using 11 air pollutants (NO₂, NO_x, SO₂, PM₁₀, O₃, CO, As, Cd, Ni, Pb, C₆H₆) according to their similarities.

Nguyen & Kim (2006) used data from four different types of air quality monitoring stations in Korea to discover spatiotemporal distribution patterns of NO₂. They focused

on the stations whose categories are among urban background, suburban background, urban traffic, or rural background. The resulting concentrations of NO₂ largely reflected the basic characteristics of its surrounding environment i.e. in association with such factors as the intensity of anthropogenic activity or population density.

Classification of air quality data into different regimes was studied in the paper of Flemming, Stern, and Yamartino using the data taken from observation sites of four air pollutants SO₂, O₃, NO₂, and PM₁₀ in Germany (Flemming, Stern, & Yamartino, 2005). They used a hierarchical clustering approach to determine a quick classification among “suburban”, “rural”, “mountain/coastal”, “urban”, “polluted urban”, “severely polluted street”, “urban-street” and “street” regimes for each pollutant factor. Similarly, in (Soares, Makar, Aklilu, & Akingunola, 2018a), hierarchical clustering combined with Kolmogorov–Zurbenko filtering was implemented to design the monitoring stations of Alberta, Canada based on the hourly to daily concentrations of NO₂ and SO₂. This study was then expanded for the same region using two other air pollutants PM_{2.5} and O₃ in addition to NO₂ and SO₂ (Soares, Makar, Aklilu, & Akingunola, 2018b). As a result, they represented air quality characteristics of an entire region using a single station and generated maps of sub-regions according to similarity levels.

Ignaccolo, Ghigo, & Giovenali (2008) classified the air quality monitoring network in Piemonte (Northern Italy) using functional cluster analysis based on Partitioning around Medoids algorithm and considering three air pollutants, namely NO₂, PM₁₀, and O₃, to classify sites in homogeneous clusters and identify the representative ones.

Lu, He, & Dong (2011) used principal component analysis and cluster analysis for the management of the air quality monitoring network of Hong Kong and the reduction of associated expenses.

Singh, Gupta, & Rai (2013) used ensemble learning methods which are decision tree forest (DTF), and decision tree boost (DTB) in their study to predict urban air

quality of Lucknow region in India, and by applying PCA they identified major air pollution sources.

Maji, Dikshit, & Deshpande (2016) presented a methodology based on the fuzzy compositional rule of inference (using cosine amplitude method and max-min method) to classify AQMSs of Delhi, India in order to acquire the optimum number of stations which provide the maximum information about air quality by reducing the expenses spent for their maintenance. Similarly, Wang, Zhao, Sun, Xue, & Xie (2018) studied for arranging AQMN of China to determine redundant monitoring stations using Pearson correlation analysis, PCA, assignment method (based on rotation factor matrix), cluster analysis using hierarchical agglomeration algorithm, and correspondence analysis.

In the area of environmental science, some studies using association rule mining techniques have been done. Among them, one of the most studied topics is air pollution or air pollutant factors. The major aim is either finding association rules between air pollutant factors and a specific disease (Toti et al., 2016; Sadat, Nikaein, & Karimipour; 2015; Karimipour & Kanani-Sadat, 2016) or detecting the significant relationship between air pollutants and industrialization (Li et. al., 2019; Ran, Wang, Liu, & Li, 2016). The most applied techniques are Apriori (Othman, Ismail, Bakar, Latif, & Abdullah, 2018; Zeng & Xiang, 2017; Toti et al., 2016; Karimipour & Kanani-Sadat, 2016; Isikli, Ustundag, & Cevikcan, 2015; Qin, Liu, Wang, Song, & Qu, 2015) and FP-Growth (Li et al., 2019; Wu, Wang, Liu, & Li, 2017) to discover the relations.

Li et al. (2019) presented their related study on the subject of the relationship between the industrialization level of 31 regions of China and a total of annual NO₂, SO₂, and PM₁₀ concentrations by applying the FP-Growth method. They detected ranges of six factors for industrialization related to low or high air quality.

Othman et al. (2018) applied both association rule mining and clustering techniques to explore the main factors which influence O₃ formation. In the first step, k-means clustering was implemented to discover the appropriate ranges for the selected

variables, and then Apriori was used to find the hidden patterns among these chosen variables.

Zeng & Xiang (2017) explored the pattern rules of air quality from the historical data by applying the Apriori algorithm and then they selected the sampling parameter based on Q-learning to develop an adaptive sampling framework through participatory sensing for urban air quality. According to the experimental results, their system predicted air quality with acceptable accuracy.

Wu et al. (2017) implemented both the Voronoi-based clustering method and association rule mining technique based on FP-Growth to develop a data-driven urban air pollution partitioning method. They carried out experiments on the hourly concentrations of air pollutants such as PM_{2.5}, NO₂, SO₂, PM₁₀, O₃, and CO and meteorological conditions such as wind, temperature, and relative humidity. Association rule mining was utilized to investigate the relationship between weather conditions and air quality index. They grouped 348 cities of China in terms of air pollution levels into 28 clusters three of which included the largest number of cities.

A fuzzy association rule mining method was applied for the detection of the specific correlations between SO₂ emission concentration patterns and operation characteristics of coal-fired power plants (Ran et al., 2016). An optimal strategy was developed by reducing the coal consumption rate and SO₂ emission concentration.

Karimipour & Kanani-Sadat (2016) searched for the relationship between air pollution factors such as PM₁₀, CO, NO₂, SO₂, PM_{2.5}, and O₃ and spatial distribution of allergy prevalence by deploying the fuzzy spatial association rule mining method based on Apriori. They also considered the patients' distance to the parks and roads. They experimented for two months. In December, they detected that SO₂ and O₃ did not affect allergy prevalence while the distance to roads and parks in addition to CO₂, NO₂, PM₁₀, and PM_{2.5} have a direct relation with it. On the other side, in June, CO, SO₂, and O₃ were ineffective.

In Turkey, the importance of environmental issues has also gained much attention and studies related to air quality increasingly continue in this direction. In this thesis, the air quality data used in the case studies also belongs to Turkey's air quality monitoring stations. New insights and methodologies are presented for the management of air quality in Turkey in the following chapters. Several of the environmental data mining studies mentioned so far on "air quality in Turkey" are compared in Table 2.2 by displaying the year of the publication, the target pollutants of the study, dataset content used in the experiments, the aim of the study, which data mining task was applied and which algorithms/methods were implemented as well as performance metrics to evaluate the results of the applied methodology. The bold notation in the Algorithms/Methods column shows the algorithm which performs the best among the others.

According to the findings, most of the experiments are done using the measurements of PM₁₀ concentrations (Kaya & Ögüdücü, 2018; Onal et al., 2017; Güler & İşçi, 2016; Taşpınar & Bozkurt, 2014; Şahin, Ucan, Bayat, & Tolluoglu, 2011; Kurt & Oktay, 2010) and prediction of air pollutant amount is the main goal. In addition to pollution data, some of the studies also integrate meteorological data such as temperature, wind speed, wind direction, pressure, and humidity into the problem domain (Kaya & Ögüdücü, 2018; Taşpınar & Bozkurt, 2014; Şahin et al., 2011; Kurt & Oktay, 2010). Isikli et al. (2015) conducted a study for finding the impacts of environmental risk indicators on the city life cycle. They implemented association rules based on Apriori, classification trees, MANOVA, and LDA to data including air quality, demographical factors, and socioeconomic development taken from the cities of Turkey. The general purpose was to determine the direct relationship between the well-being of human-being and air quality. The state of city development, suicide rate, and the number of people who migrate to and from a city were employed as city life cycle parameters while air quality factors were used as environmental risk indicators.

2.1.1.2 Water

Water is one of the most basic necessities for all living beings as well as an essential part of the ecological environment and its quality is one of the most crucial issues in environmental science due to its strong and potential effects on the natural environment and human life. In addition to drinking purposes and personal hygiene, the ecological balance is provided as well as agricultural production, industrial and manufacturing processes using water resources. A wide range of water sources such as lakes, rivers, groundwater, and dam reservoirs are considered to measure the water quality of a specific region.

Some of the studies considering the subject of water are available in Table 2.3. Data mining tasks have been applied to predict the occurrences of water pollutants (Zhao, Guo, Han, Tang, & Li, 2019; Rodriguez-Galiano, Mendes, Garcia-Soldado, Chica-Olmo, & Ribeiro, 2014; Budka, Gabrys, & Ravagnan, 2010), to analyse environmental data towards a better water quality (Di, Chang, Guo, Li, & Chang, 2019; Ho et al., 2019; Ratolojanahary et al., 2019; Deng & Wang, 2017; Granata & de Marinis, 2017; Alatrasta-Salas et al., 2015) and to develop a model to manage water resources effectively.

In several studies, a combination of data mining tasks was applied together e.g. to find latent factors in river water by grouping similar ones into the same clusters at first and finding outliers using anomaly detection in the latter step (Deng & Wang, 2017). On the other side, after clustering, associations among the parameters of the hydrological network were analyzed in order to cope with water quality issues (Alatrasta-Salas et al., 2015).

Ensemble learning methods have also been applied to data related to water for prediction purposes in general (Zhao et al., 2019; Rodriguez-Galiano et al., 2014; Ratolojanahary et al., 2019; Chou, Ho, & Hoang, 2018). Different models pick up different patterns in data. By pooling all these predictions together, as long as they are reasonably independent, informed, and diverse, the outcomes tend to be better as in

(Rodriguez-Galiano et al., 2014; Ratolojanahary et al., 2019; Chou, Ho, & Hoang, 2018).

Many studies applied clustering models to group similar water pollutants into the same cluster and to discover the hidden patterns from historical data (Di et al., 2019; Deng & Wang, 2017; Alatrasta-Salas et al., 2015; Tiwari, Babbar, & Kaur, 2018).

2.1.1.3 Soil

Soil is an indispensable living environment for all biological beings. As in the cases of air and water, unfortunately, soil also undergoes deformation over time due to external factors or the unconscious behavior of people. Soil contamination cannot be removed in nature for very long years and it remains in there without transformation. Agriculture in contaminated soils cannot yield fruitfulness, this kind of soil remains infertile and erosion or drought may arise as a result. Due to the seriousness of the topic and to deal with soil management efficiently, many studies have been canalized to soil-related fields.

Activities in plant life such as germination of seed, uptake operation by roots, seedling emergence are realized by the suitable soil temperature (Hillel, 2005; Yang et al., 2019). Besides, high temperature in the soil causes a dramatic reduction of soil resistance to physical events such as subsidence and erosion (Peerzada & Chauhan, 2018). If the soil temperature is suitable, chemical activities in the soil continue smoothly and regularly. These activities stop when the temperature drops dramatically and the soil freezes. Because, a significant decrease in temperature changes the rate of organic matter decomposition and mineralization of different organic substances in the soil (Waring & Schlesinger, 1985; Dwevedi et al., 2017).

Table 2.4 shows the survey results of studies related to the soil as the major topic. Searching for heavy metal concentrations and detecting their outlier values using cluster analysis and anomaly detection (Jia et al., 2020; Barkett & Akün, 2018; Yang et al., 2018; Yu, Bedig, Montalto, & Quigley, 2018), classifying land-cover categories

(Halmy & Gessler, 2015), creating a classification model to control minerals found in soil resources effectively (Jia et al., 2019; Chou, Yang, & Lin, 2016), or soil temperature/humidity prediction (Zeynoddin, Ebtehaj, & Bonakdari, 2020; Ji, Liu, & Zhen, 2020) are a few of them. As can be seen, Random Forest and AdaBoost have commonly applied ensemble learning strategies by resulting in good classification performance (Halmy & Gessler, 2015; Arabameri et al., 2019; Zhang, Ma, Xu, Huang, & Li, 2015).

Many studies in agriculture are canalized into this field, especially in terms of the effects of soil moisture and soil temperature on crop yields and plant growth and due to their impact on organic and chemical substances found in soil. Different techniques were proposed considering machine learning such as a hybrid neuro-fuzzy inference system (Penghui et al., 2020), classification and regression trees (CART) and group method of data handling (GMDH) (Alizamir et al., 2020) and extreme learning machine, artificial neural network and M5 Model Tree (Sanikhani, Deo, Yaseen, Eray, & Kisi, 2018) to estimate soil temperature, and extreme learning machine combined with particle swarm optimization (Ji et al., 2020), decision tree regression (Pekel, 2020) and the least-squares support vector machine (Ren, Liang, Lu, & Yan, 2019) to predict soil moisture and collaborative data mining using the algorithms of local polynomial regression, neural networks, k-nearest neighbor, support vector machine (Anton, Matei, & Avram, 2019) to estimate both soil temperature and soil moisture.

In Chapter 4 and Chapter 7, two case studies for soil temperature prediction are given. If we predict further changes in soil temperature, we can develop new strategies in different application areas. Many studies were done on this subject in recent years. In the study of Yilmaz, Özbek, Yilmaz, & Büyükalaca (2009), it is shown that the prediction of the variation in ground surface temperatures at different soil depths makes the applicability of ground source heat pump easier. Also, approximate values of heat losses in buildings can be forecasted using the ground surface temperature (Yilmaz et al., 2009; Bilgili, 2010). According to the study of Yang et al., the degree of climate change in a region can be inferred from the soil profile (Yang, Nelson, Shiklomanov, Guo, & Wan; 2010). Taking this into consideration, environmental

measures can be taken and applied more comprehensively against significant climate warming. Depending on the characteristics of the product grown, an automatic irrigation system or cooling system, which is triggered by the changes in soil temperature, can be established if the temperature reaches undesirable levels for this product. Furthermore, in this way, the planting date of temperature-sensitive crops can be easily determined.

2.1.1.4 Energy

There are many kinds of energy sources available today such as solar, wind, biomass, geothermal, hydrogen, tidal, wave, hydroelectric, fossil fuels (coal, oil, and natural gas), and nuclear power. Even though the number of the resource type is assumed high enough, the energy crisis is present for limited natural resources as the world's demands increase. These natural resources are limited in supply. Governments and interested parties are working to prioritize the use of renewable resources and diminish the irresponsible use of natural resources through greater protection. In a parallel way, both data mining and machine learning studies take place to canalize the decision-makers into making correct decisions and developing management strategies on restricted energy supplies.

The data mining studies related to energy as the main topic are compared in Table 2.5. Some of the practices are done in analyzing electricity consumption patterns using both clustering and association rule mining methods (Wang et al., 2018; Reder, Yürüşen, & Melero, 2018), while some others compare energy performance through predicting building age (Tooke, Coops, & Webster, 2014) using a regression model. Generally, regression problems considering the prediction of the amount of energy consumption/concentration (Al-Hajj, Assi, & Fouad, 2019; Araújo, Palha, Martins, Silva, & Oliveira, 2019; Biswal & Dash, 2013) or fuel flow rate (Chati & Balakrishnan, 2016) were taken into consideration by machine learning models. Ensemble strategies (Moosa, Shabir, Ali, Darwade, & Gite, 2018; Araya, Grolinger, ElYamany, Capretz, & Bitsuamlak, 2017; Srivastava, 2012) can also be observed in these research materials.

2.1.1.5 Other Environmental Subjects

All animals and living things react and give a response to changes in their environment if they are to survive. Many animals such as ant colony, fish flock, and the herd of birds move towards (or away from) the smell of different chemicals. This may allow them to find food, avoid danger, or find potential mates. They construct their habitat models according to the environmental factors and environmental stimulus.

Table 2.6 presents a few of the publications related to living organisms under a data mining scenario. Some of them handle the direct relationships between the quality of water and specific fish communities (Coutinho et al., 2019; Guo et al., 2019; Tsai, Huang, Cheng, Shao, & Chang, 2017; Fernandes et al., 2010) using classification and clustering techniques. Inherent associations among abiotic and biotic factors reign in a plankton species are detected using association rule mining methods (Silva et al., 2013), as well.

Other studies on environmental subjects such as ecology (Sha, Tan, & Bai, 2014), climate (Kwon, Lee, & Lee, 2018; Muñoz-Mas, Lopez-Nicolas, Martínez-Capel, & Pulido-Velazquez, 2016), noise (Alsouda, Pllana, & Kurti, 2019; Tanweer, Mobin, & Alam, 2016) and plant (Chougule, Jha, & Mukhopadhyay, 2019; Dos Reis et al., 2018) are given in Table 2.7. Many of them (Muñoz-Mas et al., 2016; Chougule et al., 2019; Dos Reis et al., 2018) apply random forest as the ensemble learning strategy. Clustering (Kwon et al., 2018; Chougule et al., 2019) prediction (Muñoz-Mas et al., 2016; Tanweer et al., 2016; Dos Reis et al., 2018) and also association rule mining (Sha, Tan, & Bai, 2014) studies are exemplarily given in detail by showing the applied algorithms and used performance measures.

Table 2.1 Several data mining and machine learning studies taking “air” as the main subject

Ref.	Description	Data Mining Task	Ensemble Strategy	Algorithms	Performance Measure
Zhang et al., 2020	Identification of the spatial-temporal patterns of PM _{2.5} pollution	Association Rule Mining	-	CM-SPAM	Support, Chi-square test, p-value and the Pearson contingency coefficient
Xu et al., 2020	Prediction of near-road ultrafine particle concentrations	Prediction/Clustering	3	XGBoost and k-means clustering	R ² and RMSE
Li et al., 2019	The spatiotemporal estimation and health effect assessment of SO ₂	Prediction	2	MLR, STK, MLR-STK, RF, RF-STK	R ² and RMSE
Miskell et al., 2019	Prediction of short-term peak episodes of PM _{2.5}	Classification	3	Gradient Boosted Machine	Accuracy, Kappa, median squared error, the standard deviation of MSE, sensitivity, specificity, and ROC area
Franceschi et al., 2018	Estimation of PM ₁₀ and PM _{2.5}	Clustering/Prediction	-	Simple ANN-MLP, ANN-MLP-1 based on RPROP , k-means clustering	RMSE, MAE, CC and a nonparametric test
Pucer et al., 2018	Prediction of PM ₁₀ and O ₃ levels	Classification/Prediction	2	Bayesian Lasso, RF, GP , Lasso L ₁ -regularised regression	Man Whitney U test
Yang et al., 2018	Discovering the underlying associative linkages related to pollution	Association Rule Mining	-	GSP	MAE, accuracy, RPS cost matrix, pairwise t-test
Harrou et al., 2018	Detection of abnormal ground-level O ₃ concentrations	Anomaly Detection	-	DBN-OCSVM	Minimum support value
Chen et al., 2018	Identification of outliers in PM _{2.5} monitoring system, as well as infer anomalous events	Anomaly Detection	-	Anomaly detection framework	-
Mad Saad et al., 2017	Classification of air pollutants	Classification	-	MLP , KNN and LDA	R ²
Chen et al., 2016	Estimation of spatially fine-grained urban air quality for PM _{2.5} and PM ₁₀	Classification	1	KNN, semi-supervised methods (SemiBoost, MSAB, Co-training, Tri-training, and Semi-EP) using several base classifiers, and air quality estimation methods (GPR, U-Air, U-Air(-Fc), U-Air(+kNN))	MSE, Accuracy
Bougoudis et al., 2016	Forecasting of air pollutants values	Clustering	2,4	HISYCOL (a three-fold model: SOM for clustering, FFANN and RF ANN for regression, FIS to obtain fuzzy values)	Accuracy, pairwise t-test
Lei & Wan, 2012	Prediction of air pollution index	Prediction	1	ANFIS with random sampling and ANFIS with bootstrap sampling	RMSE and R ²
					RMSE

RPROP: resilient back propagation algorithm, MAE: mean absolute error, CC: correlation coefficient, ANN-MLP: multi-layer perceptron neural network, RPS: ranked probability score, Semi-EP: semi-supervised learning and pruning, RMSE: root mean square error, DBN: deep belief network, MSAB: multi-semi adaBoost, GP: Gaussian process, SVM: support vector machines, SemiBoost: boosting for semi-supervised learning, MLP: multilayer perceptron, KNN: k nearest neighbour, LDA: linear discrimination analysis, RF: random forest, MSE: mean squared error, U-Air: urban air quality estimation method, SOM: self organizing maps, FIS: fuzzy inference system, XGBoost: extreme gradient boosting, ROC: receiver operating characteristic, GPR: gaussian process regression, ANFIS: adaptive neuro-fuzzy inference system, R²: coefficient of determination, NRMSE: normalized root mean squared error, GSP: the generalized sequential pattern, FFANN: feedforward artificial neural networks, STK: spatiotemporal Kriging, MLR: multiple linear regression, CM-SPAM: co-occurrence map sequential pattern mining, DBN-OCSVM: an approach integrating DBN and one-class SVM, RF-STK: random forest coupled with spatiotemporal Kriging, HISYCOL: the hybrid system of combined machine learning algorithm

Table 2.2 The summary table of the studies taking “air pollution in Turkey” as the main subject

Ref.	The Target Pollutants	The Dataset Content	Aim	Task	Algorithms / Methods	Performance Metrics
Kaya & Ögüdücü, 2018	PM ₁₀	PD: hourly densities of CO, NO, NO ₂ , NO _x , O ₃ and SO ₂ ; M: T, WD, WS, P, RH, min. T, max. T, max. WD and max. WS	Estimation of the density of PM ₁₀ at Istanbul using datasets with imbalanced class distribution	Prediction Classification	LRC, RFC , ETC, and GBC	Accuracy, AUROC
Dincer & Akkuş, 2018	SO ₂	The time series of weekly SO ₂ concentrations	Forecast air pollution in 65 monitoring stations	Clustering Prediction	FTS based on FKM , FCMF and GKF	RMSE and PB
Onal et al., 2017	PM ₁₀ , SO ₂	Four months SO ₂ and PM ₁₀ concentrations in Istanbul	Evaluation of the results of regular measurements of PM ₁₀ and SO ₂ concentrations in the city of Istanbul	Prediction	ANOVA	SSE and MSE, Kolmogorov-Smirnov Test, t- test
Atacak, Arici & Guner, 2017	AQI (monthly basis)	SO ₂ , NO ₂ , CO, O ₃ and PM ₁₀ (hourly and daily basis)	Determination of the AQI in Ankara	Classification	FLA	-
Güler & İşçi, 2016	PM ₁₀	Weekly PM ₁₀ concentrations in numerous stations in Turkey Hourly observations consisted of: M: max. T, avg. T, std. T, max. WS, avg. WS, std. WS, max. WD, avg. WD, std. WD, degree, max. RH, avg. RH, std. RH; PD: max. PM ₁₀ , avg. PM ₁₀ , std. PM ₁₀	Predicting a model to estimate PM ₁₀ concentrations for 130 monitoring stations	Clustering Prediction	FCARM , AR	MAPE, Dickey-Fuller test, p-value
Taşpınar & Bozkurt, 2014	PM ₁₀	Weekly PM ₁₀ concentrations in numerous stations in Turkey Hourly observations consisted of: M: max. T, avg. T, std. T, max. WS, avg. WS, std. WS, max. WD, avg. WD, std. WD, degree, max. RH, avg. RH, std. RH; PD: max. PM ₁₀ , avg. PM ₁₀ , std. PM ₁₀	Forecast maximum Daily PM ₁₀ concentrations one day ahead in Duzce	Prediction	ANN (MLP) , SWR, MLR	IA, FMB, RMSE, R ²
Cagcag, Yolcu, Egrioglu, & Aladag, 2013	SO ₂	The amount of SO ₂ in Ankara	Seasonal fuzzy time series forecasting in Ankara	Clustering Prediction	Fuzzy C-means combined with ANN and SARIMA	RMSE, MAPE
Şahin et al., 2011	PM ₁₀ , SO ₂	M: T, WS and WD, RH, P, S, C, R	Prediction of the daily and hourly mean concentrations of PM ₁₀ and SO ₂ pollutants in the regions of Istanbul	Prediction	CNN , PER	r, d, the Mean Bias Error, MAE and RMSE, t-test (p-value)
Kurt & Oktay, 2010	PM ₁₀ , SO ₂ , CO	M: P, Day T, Night T, H, WS, WD; PD: SO ₂ , CO, PM ₁₀ ; GC, Day of Week, Date	Forecasting SO ₂ , CO and PM ₁₀ levels 3 days in advance for the Besiktas district of Istanbul	Prediction	GFM_NN	Band Error

LRC: logistic regression classifier, T: temperature, FKM: fuzzy k-medoid, RFC: random forest classifier, RH: relative humidity (H), P: pressure, ETC: extra trees classifier, AQI: air quality index, FTS: fuzzy time series, FMB: fractional mean bias, GBC: gradient boosting classifier, IA: index-of-agreement, MLR: multiple linear regression, PD: pollution data, FCMF: FTS models based on fuzzy c-means, FCARM: fuzzy c-auto regressive model, SWR: stepwise regression, M: meteorological data, GKF: Gustafson-Kessel, AR: autoregressive model, WS: wind speed, PB: percent bias, MAPE: mean absolute percentage error, GC: general condition, WD: wind direction, ANOVA: analysis of variance, FLA: fuzzy logic algorithm, ANN: artificial neural network, r: correlation coefficient, S: sunshine, R: rainfall, C: cloudiness, d: index of agreement, PER: statistical persistence method, CNN: cellular neural network, AUROC: the area under the receiver operating characteristic, SARIMA: seasonal autoregressive integrated moving average, GFM_NN: geographic forecasting models using neural networks

Table 2.3 Several data mining and machine learning studies taking “water” as the main subject

Ref.	Description	Data Mining Task	Ensemble Strategy	Algorithms	Performance Measure
Chen, Tang, Rakstad, Patrick, & Zhou, 2020	Water leak detection	Classification	-	PGNN , Generic deep learning	Recall, accuracy, precision
Di et al., 2019	Identification of surface water quality and wastewater discharges	Clustering	-	EM-PAM , DBSCAN	Welch t-test, Wilcoxon test and Spearman correlation
Zhao et al., 2019	Sulfate content prediction in lakes	Prediction	1, 3	GPR , Linear regression, Decision tree, Linear-SVR, Quadratic-SVR, RBF-SVR, Boosting trees and Bagging trees	MAPE, MAE, MSE, RMSE, R ²
Ho et al., 2019	Prediction of WQI and classification according to one of the labels among five water quality classes	Classification	-	Decision tree	Accuracy, precision, recall
Ratolojanahary et al., 2019	Water quality analysis to address the issue of data imputation	Prediction	2, 3	MICE-LR, MICE-RF, MICE-BRT, MICE-KNN and MICE-SVR	Processing time, MSE
Chou et al., 2018	Forecasting Carlson's Trophic State Index, which is a metric of water quality	Prediction	1, 4	ANN, SVR, CART and LR, hybrid metaheuristic regression, FA and LSSVR, voting ensemble (CART+SVR+LR)	R, RMSE, MAE, MAPE, SI
Tiwari et al., 2018	Estimation of the most sensitive water quality parameter that can cause a change in the predicted water quality	Clustering	-	FCM-ANFIS and SC-ANFIS	RMSE, MSE, and R ²
Ji, Shang, Dahlgren, & Zhang, 2017	Dissolved oxygen prediction of hypoxic river systems	Prediction	-	MLR, backpropagation neural network, general regression neural network, and SVM	MSE, R ² , and NS model efficiency
Deng & Wang, 2017	Discovering the latent information in water quality data	Anomaly Detection/ Clustering	-	HDA-AD, hierarchical cluster analysis, ARIMA	-
Alatrasta-Salas et al., 2015	Analysis of water quality	Clustering/ Association Rule Mining	-	K-medoids, PrefixSpan, S ² MP, SPMF	SSE, minsupp, LTC
Rodriguez-Galiano et al., 2014	Estimation of groundwater nitrate pollution	Prediction	2	RF regression , LR	ROC, AUC

NS: Nash-Sutcliffe, LTC: the least temporal contradiction, SSE: sum of squared errors, MICE: multivariate imputations by chained equations, FA: the conventional firefly algorithm, SPMF: sequential pattern mining framework, CART: classification and regression trees, MAPE: mean absolute percentage error, SI: synthesis index, FCM: fuzzy c-means, SC: subtractive clustering, R: linear correlation coefficient, LR: linear regression, AUC: area under curve, minsupp: minimum support value, S²MP: Similarity measure for sequential patterns, WQI: water quality index, SVR: support vector regression, BRT: Boosted Regression Trees, PGNN: Physics-guided neural networks, EM: expectation maximization, PAM: partitioning around medoids, RBF: radial basis function, GPR: Gaussian process regression, ARIMA: Autoregressive integrated moving average model, DBSCAN: density-based spatial clustering of applications with noise, HDA-DA: heuristic density attractor-based anomaly detection, LC: least contradiction

Table 2.4 Several data mining and machine learning studies taking “soil” as the main subject

Ref.	Description	Data Mining Task	Ensemble Strategy	Algorithms	Performance Measure
Zeynoddin et al., 2020	Soil temperature prediction	Prediction	-	SARIMA , ELM, SaE-ELM, ANFIS	R ² , RMSE, MAPE, AICc
Ji et al., 2020	Estimation of soil humidity	Prediction	-	ELM, RandWPSO-ELM	Absolute error, relative error, MSE
Jia et al., 2020	Determination of the potential risk areas for soil heavy metal pollution	Clustering	2	FKM-RF , PLSR	RMSE, RPIQ, R ²
Arabameri et al., 2019	Spatial prediction of gully erosion	Prediction	2, 3, 4	COPRAS-FR-RF , COPRAS-FR-BRT, and COPRAS-FR-LR	AUC-SRC and AUC-PRC
Jia et al., 2019	Identification of the sources of soil heavy metal pollution	Classification	-	SVM, NB, Multinomial NB and ANN	Accuracy and Kappa coefficient
Barkett & Akün, 2018	Determination of the total concentration of heavy metals in soil	Clustering	-	Hierarchical cluster analysis	-
Yang et al., 2018	Finding the outliers in the heavy metals of the soil	Anomaly Detection	-	The spatial autocorrelation, the range methods (mathematical statistical and background value) and a combined method based on spatial autocorrelation and the probability theory	MRE, MSE, ME, RMSE, p-value
Yu et al., 2018	Detection of anomalous responses from soil moisture probes	Association Rule Mining/ Anomaly Detection	-	Apriori algorithm	Sup, Conf, Lift
Chou et al., 2016	Prediction of the shear strength of fiber-reinforced soil	Classification/ Prediction	1, 4	REG, CART, GENLIN, and CHAID; ANN and SVM and SVR; and their voting, bagging, stacking, and tiering versions, Tiering SVM-(SVR/SVR)	R, MAPE, RMSE, MAE, SI
Halmy & Gessler, 2015	Classification of the land-use/land-cover categories of desert landscapes	Classification	2, 3	RF and boosted ANNs	Mean class user's accuracy
Zhang et al., 2015	Rare classes' classification for dust storm forecasting	Classification	2, 3	SARF , SVM, Fuzzy ANN	Accuracy

MRE: the mean relative error, ME: mean error, REG: linear regression analysis, GENLIN: generalized linear model, Conf: confidence value, SRC: success rate curve, SMOTE: the synthetic minority oversampling technique, CHAID: chi-squared automatic interaction detection, PRC: prediction rate curve, LR: Logistic regression, COPRAS: complex proportional assessment of alternatives, FR: frequency ratio, SARF: SMOTE with AdaBoost and RF, SUP: support, ELM: extreme learning machine, SaE-ELM: self-adaptive evolutionary ELM, AICc: Akaike's information criterion, SARIMA: Seasonal ARIMA, PLSR: partial least squares regression, FKM: fuzzy k-means, RandWPSO-ELM: random weight Particle Swarm Optimized ELM, RPIQ: ratio of performance to interquartile distance

Table 2.5 Several data mining and machine learning studies taking “energy” as the main subject

Ref.	Description	Data Mining Task	Ensemble Strategy	Algorithms	Performance Measure
Al-Hajj et al., 2019	Prediction of solar radiation intensity	Prediction	4	Stacking of SVR with one of the methods (MLP , DTR, KNN)	RMSE, MAE, MBE, MAPE, R
Araújo et al., 2019	Estimation of the energy consumption	Prediction	-	ANN , SVM	R ² , RMSE, MAD
He et al., 2019	Determination of the coal and gangue areas	Classification	1, 2	C-SVM, BagRF, ML-KELM, TR, ELM, TR-ELM, MR-ELM, TR-ML-ELM	Accuracy
Reder et al., 2018	Analysis of failure data and environmental conditions ahead of wind turbine component failures	Association Rule Mining/ Clustering	-	Apriori algorithm, k-means clustering	Support, confidence, lift
Moosa et al., 2018	Forecasting of solar radiation	Prediction	2, 3	ANN, RF, XGB	R ² , RMSE, MAE
Wang et al., 2018	Identification of the most significant household characteristics affecting residential electricity consumption patterns	Association Rule Mining/ Clustering	-	DBSCAN, k-means, enhanced Apriori algorithm	Chi-squared test, CC, Lift, Sup, Conf, Lift, Imp, p-value
Araya et al., 2017	Identification of anomalous consumption patterns in building energy consumption	Anomaly Detection	2, 4	RF, SVR, CCAD-SW using autoencoder and PCA, EAD	TPR, FPR
Chati & Balakrishnan, 2016	Modeling of fuel flow rate	Prediction	3	CART, LSB	PC, ME
Tooke et al., 2014	Prediction of building age and energy performance	Prediction	2	RF regression model	RMSE, R ²
Biswal & Dash, 2013	Determination of energy concentration of the time-frequency distribution	Classification	-	FDST , DTC, DWT, WPT, ST, ANN, neuro-fuzzy system, MSVM, rule-based expert system, PNN	Gini's diversity index, Accuracy
Srivastava, 2012	Detection of the overconsumption of fuel in aircrafts	Anomaly Detection	1	Bootstrap sampling on each of the regression tree (tree), Elastic network, GP, and stable GP regression method	ROC, NRMSE

EAD: ensemble anomaly detection, CC: contingency coefficient, TPR: true positive rate, ST: S-Transform, Imp: improvement value, MSVM: multiclass support vector machine, PNN: probabilistic neural network, DWT: discrete wavelet transform, DTC: decision tree classifier, WPT: wavelet packet transform, LSB: least-squares boosting algorithm, PC: percentage coverage, DDPG: Deep deterministic policy gradients, CCAD-SW: the collective contextual anomaly detection using sliding window framework, SCN: structured control net, DQN: deep Q network, XGB: eXtreme gradient boosting, TR-ML-ELM: tree root multi-layer ELM, ML-KELM: multi-layer kernel ELM, MAD: mean absolute deviation, TR-ELM: tree root ELM, BagRF: bagging RF, DTR: decision tree regressor, FDST: fast variants of the discrete S-transform algorithm

Table 2.6 Several data mining and machine learning studies taking other environmental subjects as the main topic

Ref.	Description	Data Mining Task	Ensemble Strategy	Algorithms	Performance Measure
Coutinho et al., 2019	Prediction of the response of the microbial community to environmental variables	Prediction	-	ANN	PCC, RMSE
Guo et al., 2019	Modeling fish community indices with water quality indicators	Prediction	2	KNN, SVM, PLS, CART, ANN, GLM, RF	MSE, RMSE, R ²
Tsai et al., 2017	Extraction of the complex multi-relation between water quality factors and fish community	Clustering	-	SOM , ANN	QE, TE, SI
Salman et al., 2016	Learning species-dependent visual features of fish	Classification	-	CNN, KNN, SVM, PCA, SRC, CNN-KNN , CNN-SVM	AC, AP, AR, Accuracy
Kocev & Džeroski, 2013)	Construction of habitat models for living species	Clustering	1, 2	RF and Bagged versions of multi-target PCT and single-target DT	RRMSE
Silva et al., 2013	Finding relationships between biotic and abiotic factors in ichthyoplankton samples from a freshwater reservoir	Association Rule Mining	-	Apriori Algorithm, BestFirst and CfsSubsetEval algorithms	Supp, Conf, Lift
Pais, Henrique, Batista, Costa, & Cabral, 2013	Delimitation of reef fish assemblage types	Classification/Clustering	4	MPCK-Means, KNN with IDW , KNN with MR, PERMANOVA	Accuracy, Precision, Recall
Fernandes et al., 2010	Fish recruitment prediction	Classification	1	CFS, NB , TAN, J48DT, MPNN, SVM	TPR, Accuracy, CPU time, Brier Score, paired t-test

PLS: partial least squares, J48DT: J48 decision tree, MPNN: multi-layer perceptron neural network, TE: topographic error, AR: average recall, IDW: inverse-distance weighted voting, MR: majority rule voting, SI: structuring index, QE: quantization error, CNN: convolutional neural network, PCC: Pearson correlation coefficient, AP: average precision, TAN: tree augmented naive Bayes, SRC: sparse representation of features, AC: average count, GLM: generalized linear model, PCT: predictive clustering tree, PERMANOVA: analysis of variance using permutations, MPCK-Means: metric pairwise constrained k-Means, CFS: multivariate correlation-based feature selection method

Table 2.6 continues

Ref.	Type	Description	Data Mining Task	Ensemble Strategy	Algorithms	Performance Measure
Sparke, 2018	Noise	Environmental noise classification	Classification	2	SVM, LR, DT, KNN , ANN, RF	Accuracy, R ²
Alsouda et al., 2019	Noise	Urban noise identification	Classification	1, 2	SVM , KNN , Bagged DT, RF	Accuracy, Precision, Recall, F ₁ Score
Chougule et al., 2019	Plant	Prediction of the suitable crop for a field and type of soil	Clustering	2	K-means, RF	Precision
Dos Reis et al., 2018	Plant	Estimate and map the basal area and volume of Eucalyptus	Prediction	2	MLR, RF , SVM, ANN	RMSE, ME, MAE, RI
Kwon, Lee, & Lee, 2018	Climate	Establishing boundaries of greenhouse gas emission features	Clustering	-	Hierarchical cluster analysis using complete linkage technique	Elbow method
Muñoz-Mas et al., 2016	Climate	Determination of the impact of climate change on the habitat suitability for large brown trout	Prediction	1, 2, 4	Ensemble of generalized additive models, MLP with bagging ensembles, RF, SVM and fuzzy rule base systems (TSK)	Weighted MSE, Overall TSS
Tanweer et al., 2016	Noise	Environmental noise classification	Prediction	-	LPC, MFCC, LDA, QDA, ANN and ANN-MFCC	ROC, MSE
Sha, Tan, & Bai, 2014	Ecology	Prediction of the vegetation growing status	Association Rule Mining	-	Quadtree-based algorithm	Sup, Conf, Sup _{ED} , Conf _{ED}

Sup_{ED}: support of event density, Conf_{ED}: confidence of event density, LPC: linear predictive coding, MFCC: mel frequency cepstral coefficient, QDA: quadratic discriminative analysis, TSS: true skill statistics, RI: relative improvement

2.1.2 According to the Data Mining and Machine Learning Task

Environmental informatics are generally divided into five major groups: classification, regression, clustering, association rule mining, and time series analysis. In this part, the studies in each group are briefly discussed.

2.1.2.1 Classification

Classification is a data mining task that aims to assign test samples to target classes by using a classifier model by employing a training set with known class labels. Because it uses a training set with predefined labels, it is also known as supervised learning. The target attribute is categorical, and no numeric predictions are made.

Land cover classification (Halmy & Gessler, 2015), environmental noise classification (Tanweer et al., 2016; Sparke, 2018; Alsouda et al., 2019), forecasting of urban air quality spatially (Chen et al., 2016), water leak detection (Chen et al., 2020), identification of the sources of soil heavy metal pollution (Jia et al., 2019), and estimation of recruitment in fisheries management (Fernandes et al., 2010) are some of the classification problems in environmental science.

The popular classification algorithms used in the environmental studies are support vector machines (Zhang et al., 2015; Salman et al., 2016; Jia et al., 2019; Alsouda et al., 2019), decision tree (Chou et al., 2016; Ho et al., 2019), k-nearest neighbours (Pais et al., 2013; Salman et al., 2016; Sparke, 2018; Alsouda et al., 2019), naïve Bayes (Fernandes et al., 2010; Jia et al., 2019), and neural networks (Zhang et al., 2015; Mad Saad et al., 2017; Jia et al., 2019; Chen et al., 2020).

2.1.2.2 Regression

Regression is another supervised learning task where the target variables of test samples are estimated using a prediction model constructed by a training set as in the classification task, however the target variable is now numeric instead of categorical.

In environmental informatics, there are many studies that employ regression methods such as prediction of air pollutant levels (Pucer et al., 2018; Mad Saad et al., 2017; Chen et al., 2016; Xu et al., 2020; Franceschi et al., 2018; Lei & Wan, 2012), forecasting the pollutants in a water supply (Zhao et al., 2019; Rodriguez-Galiano et al., 2014; Ji et al., 2017), prediction of the response of the microbial community to environmental variables (Coutinho et al., 2019), estimation of fuel flow rate (Chati & Balakrishnan, 2016), soil temperature/humidity prediction (Zeynoddin et al., 2020; Ji et al., 2020), and forecasting energy performance (Tooke et al., 2014).

Support vector regression (Chou et al., 2018; Zhao et al., 2019; Ratolojanahary et al., 2019), linear regression (Rodriguez-Galiano et al., 2014; Taşpınar & Bozkurt, 2014; Ji et al., 2017; Chou et al., 2018; Zhao et al., 2019; Ratolojanahary et al., 2019), neural networks (Kurt & Oktay, 2010; Taşpınar & Bozkurt, 2014; Ji et al., 2017), decision tree regression (Al-Hajj et al., 2019; Chou et al., 2018; Zhao et al., 2019), and logistic regression (Kaya & Öğüdücü, 2018) are some of the well-known regression methods used in the environmental informatics.

2.1.2.3 Clustering

Clustering is another data mining task that aims to assign similar samples into the same group, where each group is as different from the other as possible, according to a similarity criterion (i.e. distance measures). It is different from classification and regression tasks that there is no known labels and training instances are not found. Therefore, it is known as unsupervised learning.

In addition to classification and regression, clustering also takes part in environmental studies such as grouping similar water quality parameters together to find out the most effective parameters causing a significant change in the quality of water (Tiwari et al., 2018), determination of boundaries among greenhouse gas emission features (Kwon et al., 2018), defining the most important household characteristics affecting residential electricity consumption patterns (Wang et al., 2018), predicting the concentrations of heavy metals found in soil (Barkett & Akün,

2018) and dealing with habitat models for living species (Kocev & Džeroski, 2013; Pais et al., 2013).

The most common clustering methods are k-means (Wang et al., 2018; Chougule et al., 2019; Xu et al., 2020), self-organizing map (Tsai et al., 2017), expectation maximization (Di et al., 2019), density-based spatial clustering of applications with noise (Wang et al., 2018; Di et al., 2019) and hierarchical clustering (Deng & Wang, 2017; Barkett & Akün, 2018; Kwon et al., 2018).

2.1.2.4 Association Rule Mining

Association rule mining (ARM), which is one of the main data mining tasks, aims to observe frequently occurring patterns (co-occurrences), correlations, or interesting relations between variables in a large collection of data using some measures of interestingness. As a result, strong rules are determined at the minimum thresholds of support and confidence.

Several association rule mining studies in environmental area are as follows: detection of the associative linkages of air pollution (Pucer et al., 2018; Zhang et al., 2020), analysis of water quality data (Alatrasta-Salas et al., 2015) and finding a relationship between biotic and abiotic factors in ichthyoplankton samples (Silva et al., 2013).

ARM methods used in most of the environmental studies are as follows: Apriori (Yu et al., 2018; Reder et al., 2018; Wang et al., 2018), FP-Growth (Wu et al., 2017; Li et al., 2019), generalized sequential pattern (Yang et al., 2018), Quadtree-based algorithm (Sha et al., 2014), and PrefixSpan (Alatrasta-Salas et al., 2015).

Figure 2.3 demonstrates the percentages in a pie chart and it is very clear that most of the environmental studies considering data mining tasks have been done in the area of classification & prediction (regression). According to the findings from the Scopus search engine, the number of publications in the subject of classification & prediction,

clustering, and association rule mining & sequential patterns are 3434, 457, and 127, respectively between the years of 2010 and 2020. “Water” is the most studied topic for the subjects of clustering and classification & prediction. On the other hand, “air” is the first topic in association rule mining & sequential patterns studies.



Figure 2.3 The number of publications according to the main areas (classification & prediction (regression), clustering, association rules & sequential patterns) in data mining

2.1.2.5 Time Series Analysis

Time series analysis considers data, which are taken from a series of particular time periods or intervals, to extract meaningful statistics and other characteristics of the data. Future values are predicted based on previously observed values. It is a statistical technique, and not a data mining task.

Some environmental problems solved by time series methods are as follows: soil temperature prediction (Bonakdari et al., 2019; Mehdizadeh, Fathian, Safari, & Khosravi, 2020; Zeynoddin et al., 2020), forecasting air pollution (Cagcag et al., 2013), discovering the latent information in water quality data (Deng & Wang, 2017), estimation of the concentrations of air pollutants (Güler & İşçi, 2016).

Time series methods generally used in environmental subjects are AR (Güler & İşçi, 2016), ARMA (Bonakdari et al., 2019), ARIMA (Deng & Wang, 2017), SARIMA (Cagcag et al., 2013; Zeynoddin et al., 2020), and FARIMA (Mehdizadeh et al., 2020).

CHAPTER THREE

CLASSIFICATION STUDIES IN ENVIRONMENTAL INFORMATICS

This chapter presents two classification case studies in the environmental area by (i) introducing a novel ensemble learning method called “Enhanced Bagging (eBagging)”, (ii) providing an efficient solution to classify monitoring stations in an air quality network, and (iii) comparing four ensemble learning strategies through testing them for different environmental subjects. Furthermore, experimental study applied methodologies, and the obtained results are discussed and presented in detail.

3.1 Material and Methods

In this section, the applied methods in experimental studies are explained. A new ensemble learning technique called “Enhanced Bagging (eBagging)” is introduced. Five classifiers including support vector machines, decision tree, k-nearest neighbor, naïve Bayes, neural networks, and random forest, and six ensemble learning techniques containing bagging, boosting, random forest, random subspace, and voting are talked about in-depth.

3.1.1 Classification Algorithms

Classification is one of the data mining tasks that is used to classify each sample in a data set into one of a predefined set of classes or groups. After a classifier model is constructed using the samples in the training set, the class labels of new coming instances in the test set are estimated by employing this trained model. In this part, six classification algorithms, which are support vector machines, decision tree, k-nearest neighbor, naïve Bayes, and neural networks, were performed in the experimental studies.

3.1.1.1 Support Vector Machines

It is one of the most applied and powerful classification methods for both linear and non-linear data. The aim is to find the maximum marginal hyperplane that splits the data points in a way that the samples belonging to different classes are placed as far away as possible to other classes' samples. The shortest distance from the optimal hyperplane to the closest training sample (i.e. support vector) of either class is known as margin. Using a non-linear mapping, the training data is transformed into a higher dimension.

When the data is linearly inseparable, kernel functions are used. In the experimental studies, the *polynomial kernel* and *radial basis function kernel* are applied. After the data is transformed into a new higher space using kernel functions, a linear separating hyperplane in the new space is searched.

3.1.1.2 Decision Tree

It is one of the supervised learning methods where the classification scenario is represented as a tree-based system in which each branch points out a possible outcome. The depth-first strategy in a top-down recursive and divide-and-conquer manner is applied to predict unknown class attribute values for test cases. Each node of the tree refers to a specific attribute as the branches show their values. The leaves carry the results (the value of the class label). There are several different algorithms to implement decision trees such as C4.5, ID3, CHAID, CART, etc. In the case studies, the C4.5 algorithm was applied, therefore, it is explained in detail in this part.

The construction of a decision tree is started with a root node. The determination of the initial node is based on the mutual information which gives the highest benefit for learning. For this purpose, *information gain* given in (3.1), which is calculated for each attribute, is used. Information gain evaluates the gain of each feature in the context of a target variable. It is performed by taking the mutual information (i.e. the determination of the statistical dependence) between two random variables. It

measures the reduction in entropy before and after splitting a dataset according to a given value of a random variable as expressed in (3.1) where S is a dataset with a set of instances, A is an attribute, v is the all possible values of attribute A , and S_v is the subset of S where attribute A has the value v . Entropy given in (3.2), where p_i expresses the probability of output state value i in the dataset with n samples and S subsets, expresses the amount of information in a random variable. Lower probability events yield more information with high entropy and higher probability events yield less information with low entropy.

$$\text{InfoGain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v) \quad (3.1)$$

$$\text{Entropy}(S) = \sum_{i=1}^n -p_i \log_2 p_i \quad (3.2)$$

First of all, the attribute which should be considered first as a root node is decided by selecting the one which maximizes the information gain. Each node can have various possible split points to branch out the tree and the optimal split point value is found by using information gain.

3.1.1.3 K-Nearest Neighbours

It is one of the instance-based lazy learners where the method memorizes the training dataset instead of learning a discriminative function for predicting future samples. It compares a given test instance with training instances that are similar to it. The parameter k refers to the number of samples to be considered in the determination of the class label of a new sample. The optimal k value is found empirically. The most common class label among its k -nearest neighbors is selected for the test instance by applying majority voting. The nearest neighbors are calculated using one of the distance metrics such as *Manhattan*, *Euclidean*, *Chebychev*, *Cosine*, etc. In the experimental studies, Euclidean distance given in (3.3), where two points are described as $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$ for m features is used.

$$\text{Dist}(X_1, X_2) = \sqrt{\sum_{i=1}^m (x_{1i} - x_{2i})^2} \quad (3.3)$$

3.1.1.4 Naïve Bayes

It is a probabilistic classifier model based on the Bayes theorem. The general assumption is that all the features are independent in other words they do not affect each other. Another assumption is that all the predictors have an equal effect on the outcome. The Bayes theorem is given in (3.4) where y is the class variable and X is the feature set described as $X = (x_1, x_2, \dots, x_m)$ for m features.

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \quad (3.4)$$

The general formula can be written as in (3.5) after applying the chain rule and removing the denominator. For multivariate cases, we find the class y with maximum probability as in (3.6).

$$P(y|x_1, x_2, \dots, x_m) \propto P(y) \prod_{i=1}^m P(x_i|y) \quad (3.5)$$

$$y = \operatorname{argmax}_y P(y) \prod_{i=1}^m P(x_i|y) \quad (3.6)$$

3.1.1.5 Neural Networks

One of the most commonly used classification models is artificial neural networks which are constructed with a set of connected input and output units and weights between each unit that form interconnected multiple layers. The applied neural network type in the experiments is the multilayer perceptron (MLP) and it is explained in there.

In the model, the input data is transformed using a learnt non-linear transformation function to make it linearly separable. Input layer is made up of input features $(x_1, x_2, \dots, x_m)$. In each hidden layer, the values transferred from the previous layer are transformed with a weighted linear summation as $w_1x_1 + w_2x_2 + \dots + w_mx_m$ followed by a non-linear activation function such as *sigmoid*, *logistic*, *identity*, *tanh*, *relu*, etc. The output layer finally transforms the values from the last hidden layer into output values.

3.1.2 Ensemble Learning

Ensemble learning is a machine learning technique where multiple learners are trained to solve the same problem and their predictions are combined into a single output that probably has better performance on average than any individual ensemble member. The fundamental idea behind ensemble learning is to combine weak learners into one, a strong learner, that has a better generalization error and is less sensitive to overfitting in the presence of noise or small sample size. This is because different classifiers can sometimes misclassify different patterns and accuracy can be improved by combining the decisions of complementary classifiers.

3.1.2.1 Elements of an Ensemble Classifier

A typical ensemble framework for classification tasks contains four fundamental components described as follows:

- *Training Set* – A training set is a special set of labeled examples providing known information that is used for training.
- *Base Inducer(s) or Base Classifier(s)* – An inducer is a learning algorithm that is used to learn from a training set. A base inducer obtains a training set and constructs a classifier that generalizes the relationship between the input features and the target outcome.
- *Diversity Generator* – It is clear that nothing is gained from an ensemble model if all ensemble members are identical. The diversity generator is responsible for generating the diverse classifiers and decides the type of every base classifier that differs from each other. Diversity can be realized in different ways depending on the accuracy of individual classifiers for improved classification performance. Common diversity creation approaches are: (i) using different training sets, (ii) combining different inducers, and (iii) using different parameters for a single inducer.
- *Combiner* – The task of the combiner is to produce the final decision by combining all classification results of the various base inducers. There are two main methods of combining: weighting methods and meta-learning methods. Weighting methods

give each classifier a weight proportional to its strength and combine their votes based on these weights. The weights can be fixed or dynamically determined when classifying an instance. Common weighting methods are majority voting, performance weighting, Bayesian combination, and voggging. Meta-learning methods learn from new training data created from the predictions of a set of base classifiers. The most well-known meta-learning methods are stacking and grading. While weighting methods are useful when combining classifiers built from a single learning algorithm and they have comparable success, meta-learning is a good choice for cases in which base classifiers consistently classify correctly, or consistently misclassify.

3.1.2.2 Ensemble Strategies

To construct an ensemble model, any of the following strategies can be performed (Tuysuzoglu, Birant, & Pala, 2018):

- *Strategy 1 - Different training sets using random sampling with replacement:* One ensemble strategy is to train different base learners by different subsets of the training set. This can be done by random re-sampling of a dataset (i.e. bagging - Figure 3.1 (a)). When we train multiple base-learners with different training sets, it is possible to reduce variance, and therefore error.
- *Strategy 2 - Different training sets obtained by random instance and feature subset selection:* The combination of bagged decision trees is constructed similar to Strategy 1 using one significant adjustment that random feature subsets are used (i.e. Random Forest - Figure 3.1 (b)). When we have enough trees in the forest, the Random Forest classifier is less likely to overfit the model. It is also useful to reduce the variance of low-bias models, besides handling missing values easily.
- *Strategy 3 - Different training sets using random sampling with replacement over weighted data:* This ensemble strategy can be implemented by weighted re-sampling of the dataset serially by focusing on difficult examples that are not correctly classified in the previous steps (i.e. AdaBoost - Figure 3.1 (c)). Boosting helps to decrease the bias of otherwise stable learners such as linear classifiers or univariate decision trees also known as decision stumps.

- *Strategy 4 - Different algorithms*: The other ensemble strategy (i.e. voting - Figure 3.1 (d)) is to use different learning algorithms to train different base-learners on the same dataset. So the ensemble includes diverse algorithms that each take a completely different approach. The main idea behind this kind of ensemble learning is taking advantage of classification algorithms' diversity to face complex data.

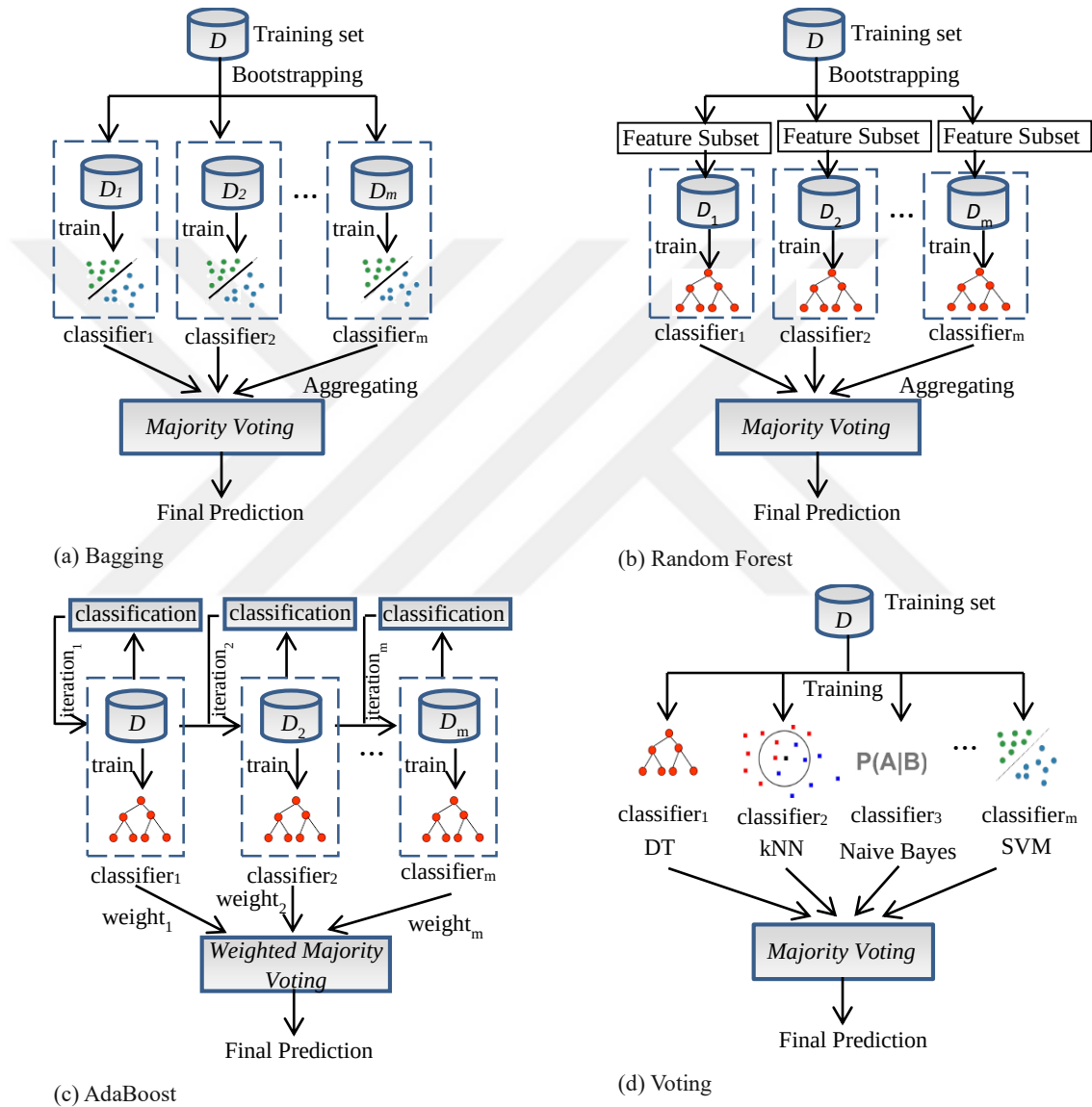


Figure 3.1 Different ensemble strategies

Although ensemble classifiers have a common goal to construct multiple, diverse, and predictive models and finally to combine their outputs, each strategy is carried out in different ways using different training sets, combiner, or inducer. Table 3.1

summarizes the properties of different ensemble strategies, the popular algorithms under each category, and the pros and cons of each ensemble classifier.

Table 3.1 Characteristic of different ensemble classifiers

Algorithm	Training Set	Classifiers	Combiner	Inducer	Ensemble Strategy	Advantage	Weakness
Bagging	random resampling	inducer independent	majority voting	single inducer	1	minimizes variance	a relatively large ensemble size – loss of cooperation with each other
Random Forest	random resampling + feature subset	inducer dependent (decision tree)	majority voting	single inducer	2		
Boosting	weighted resampling	inducer independent	weighted majority voting	single inducer	3	boosts the performance of the weak learners	degrades with noise
AdaBoost	weighted resampling	inducer independent	weighted majority voting	single inducer	3		
Stacking	resampling and k-folding	inducer independent	meta-learning	multi-inducer	1, 4	good performance	storage and time complexity
Grading	resampling and k-folding	inducer independent	meta-learning	multi-inducer	1, 4	predictions are graded	storage and time complexity
Voting	same dataset	inducer independent	majority voting	multi-inducer	4	increase predictive accuracy	how are classifiers selected
Voting	same dataset	inducer independent	majority voting	single inducer	4	simple to understand and implement	limited to a single algorithm performance

After a summative introduction, the detailed explanations of each method are presented in the following subsections.

Assume that $X = \{x_1, x_2, \dots, x_n\}$ is a data set that is comprised of n instances with m distinct features. We can define an ensemble model as $E = \{e_1, e_2, \dots, e_I\}$ where I is the ensemble size and each ensemble member e_i represents the model built by a classification algorithm applied in the i^{th} iteration. The main purpose is the determination of how to construct the classification ensemble so that the diversity among each member is ensured. In this study, well-known classification algorithms

are combined with different ensemble learning methods. Six different ensemble learning methods, one of which is the recently proposed eBagging algorithm, are applied in line with this objective and they are explained in this section.

3.1.2.2.1 Bagging. The data set X is repeatedly resampled with replacement so that different training sets are used in the generation of $EBAG = \{e_1, e_2, \dots, e_I\}$ where each e_i is made up of different training set X_i from $X_{ALL} = \{X_1, X_2, \dots, X_I\}$ for I iterations. All ensemble iterations are applied with the same model parameters. The only difference is the used training sets. The final class labels of instances are determined using a voting mechanism which is applied to the outputs of the ensemble iterations.

Random Forest is one of the variations of the bagging algorithm where different ensemble iterations of the model $E_{RF} = \{e_1, e_2, \dots, e_I\}$ are constructed from decision trees. As in bagging, the bootstrapped replicas of training set X are created to be used in ensemble iterations besides feature subsets can also be differed like in random subspace method. In other words, in each iteration, e_i , different training sets that are made up of different instances and different features are used. Finally, a voting mechanism is applied to combine the outputs of the ensemble members.

The feature-based version of the bagging algorithm is known as *Random Subspace*. The feature set of data set $X = \{x_1, x_2, \dots, x_n\}$, where each x_i is comprised of m dimensions, is updated in each ensemble iteration by randomly selecting t percent of m attributes. In this way, feature set $F = \{f_1, f_2, \dots, f_I\}$ is generated for I iterations where each f_i is used to construct classification model e_i in $E_{RS} = \{e_1, e_2, \dots, e_I\}$. The same model parameters are used for each ensemble member. Majority voting is used in the decision-making process of the final class label prediction.

3.1.2.2.2 *Boosting*. The most commonly applied boosting method is the *AdaBoost*. The procedure starts with a training subset instances (X_i) of which are randomly chosen from data set X , and then, a weak learner classifies the instances in X_i . The next ensemble iteration e_i is constructed with a new training set instances of which are selected with a distribution probability $P_i \in P = \{P_1, P_2, \dots, P_n\}$ in the ensemble model $E_{AdaBoost} = \{e_1, e_2, \dots, e_I\}$ for I iterations. These probabilities are updated in each iteration according to training errors of instances, in other words, hard-to-classify instances have a high chance to be chosen for the new ensemble iteration. According to a weighted majority voting approach, the final class label is assigned to each instance.

When bagging and boosting are compared, boosting leads to a greater reduction in error (Quinlan, 1996) because it concentrates on the challenging elements in the training set during the sample selection. However, boosting suffers from being hard to parallelize because the inputs of one model depend on the outputs of others. In the following parts, a new ensemble method (eBagging) that takes the best characteristics of these two techniques will be presented. While both approaches substantially improve predictive accuracy, eBagging shows a greater benefit.

3.1.2.2.3 *Voting*. Because each classification algorithm produces different generalization error, a strategic combination of them can reduce the total error. By this occasion, in the vote, different classification algorithms are applied in different ensemble iterations so that $E_{VOTE} = \{e_1, e_2, \dots, e_I\}$ is constituted by I classification methods. Decisions of those members are combined using majority voting in a way that the most assigned class label is the final prediction for instance x_i in $X = \{x_1, x_2, \dots, x_n\}$ for n samples.

3.1.2.3 The Proposed Method: Enhanced Bagging (eBagging)

The proposed method is introduced in this part by dividing the procedure into two parts: *Error-based Bootstrapping* and *Enhanced Bagging*.

3.1.2.3.1 Error-based Bootstrapping. Fundamentally, bagging relates to the bootstrap approach where the training sets are randomly selected with replacement from the original instances. By following this procedure, several records may appear more than once as a result of resampling while others may not be present in the training set. The disadvantage with the traditional bootstrap method is that training subsets produced by random selection with replacement are not especially concentrated on misclassified instances. In other words, the hard-to-classify examples may not present in the training sets, so the learning algorithm cannot focus on those data points to minimize the training errors. To overcome this drawback, a novel bootstrapping model, named eBootstrapping, is proposed in this study by ensuring the presence of misclassified instances in training sets to encourage their correct classification.

Consider the dataset $D = \{o_1, o_2, \dots, o_n\} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ with y_i as the class output and x_i as a p -dimensional explanatory variable for the i^{th} object o_i , so $o_i = (x_i, y_i)$ for $(i = 1, \dots, n)$, where n is the number of objects (instances) in the dataset. Assume that we have k different class labels, in this case, $y_i \in Y = \{1, \dots, k\}$.

- *Definition 1.* (Bootstrapping). If we have n data instances denoted by

$$D = \{o_1, o_2, \dots, o_n\} \quad (3.7)$$

Then randomly generated sample with the same size n and with the equal probability ($1/n$ for each observation)

$$D^* = \{o_1^*, o_2^*, \dots, o_n^*\} \quad (3.8)$$

called the bootstrap sample or bootstrap resample and denoted by adding a star to the symbols. The star symbol $*$ represents zero or more instances. This means that D^* is likely to contain repeats. Similarly, just as $\bar{o}$ is the mean of the original dataset, we write $\bar{o}^*$ for the mean of the resamples' data. To create an ensemble E , we get b independent bootstrap samples and denote by

$$E = \{D_1^*, D_2^*, \dots, D_b^*\} \quad (3.9)$$

For a given dataset of n examples, each example has a probability $(1-1/n)^n$ of not be included in the bootstrap sample. If n is large, the probability approaches $1/e = 0.368$, which means 36.8% of the original examples would not be selected, called out-of-bag instances.

$$\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n = \frac{1}{e} \approx 0.368 \quad (3.10)$$

This also means that on average 36.8% of the “difficult” examples (misclassified) would not take part in the training set and for this reason, the underlying learning algorithm may not provide a good compromise between bias and variance on classification. To get over this drawback, a novel bootstrapping method, eBootstrapping, which gives priority to the difficult instances is proposed.

- *Definition 2.* (eBootstrapping). Given a dataset with size n ,

$$D = \{o_1, o_2, \dots, o_n\} \quad (3.11)$$

misclassified instances are identified by a *prior classifier* in the pre-training step and denoted by M

$$M = \{o_1, o_2, \dots, o_m\} \quad (3.12)$$

where m is the number of incorrectly classified instances and it is smaller than instance size n . Classified instances are the rest of the dataset (correctly classified instances) and denoted by C given in (3.13) where $D = M \cup C$ and $M \cap C = \emptyset$.

$$C = \{o_{m+1}, o_{m+2}, \dots, o_n\} \quad (3.13)$$

eBootstrapping is a prediction error-based bootstrapping method which generates a dataset that includes all misclassified instances in M and some correctly classified

instances taken with replacement from C . An *eBootstrap sample* of D is a collection of n instances

$$D^* = \{o_1, o_2, \dots, o_m, o_{m+1}, o_{m+2}, \dots, o_n\} \quad (3.14)$$

where each value o_i is a distinct instance from M for $(1 \leq i \leq m)$, and each value c_j^* is a randomly selected instance from C with replacement and with equal probability $\Pr(c_j^* = o_t) = 1/(n-m)$ for $(m+1 \leq j, t \leq n)$ and $(m+1 \leq t \leq n)$. The star symbol $*$ represents zero or more instances. In particular, repeated values $c_{j1}^* = c_{j2}^* = o_t$ are allowed. Since the sample size of D^* is n , repeated values c_j^* means that some values in C must be left out. To form b independent eBootstrap samples, we repeat this process b times and denote by $E = \{D_1^*, D_2^*, \dots, D_b^*\}$.

As shown in Figure 3.2, an ebootstrap sample includes members of the original dataset with all misclassified members appearing, some correctly classified members appearing zero times, some appearing only once, some appearing twice, and so on. Replacement results in the repetition of correctly classified values in ebootstrap samples. eBootstrap samples have the same sample size as the original sample. As an example, consider a dataset $D = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ with size $n = 10$. The eBootstrap sample $D^* = \{1, 2, 3, 8, 6, 6, 4, 10, 9, 6\}$ has three repeats of the same number (6). The order of the elements in the resample plays no role. The out-of-bag set consists of two elements $\{5, 7\}$.

	misclassified			classified						
Original sample	1	2	3	4	5	6	7	8	9	10
eBootstrap Sample 1	1	2	3	8	6	6	4	10	9	6
eBootstrap Sample 2	1	2	3	5	7	10	4	10	8	9
eBootstrap Sample 3	1	2	3	4	4	7	9	8	6	10
eBootstrap Sample 4	1	2	3	7	10	6	8	4	5	9
eBootstrap Sample 5	1	2	3	9	4	9	6	7	5	9
in bag										

3.1.2.3.2 *Enhanced Bagging (eBagging)*. eBagging improves the traditional bagging technique by replacing the bootstrap method with eBootstrap. The key difference is the creation of training sets by giving higher chance for selection to hard-to-classify instances which are misclassified by the prior learner. As shown in Figure 3.3, the proposed eBagging technique contains four steps (Tuysuzoglu & Birant, 2020):

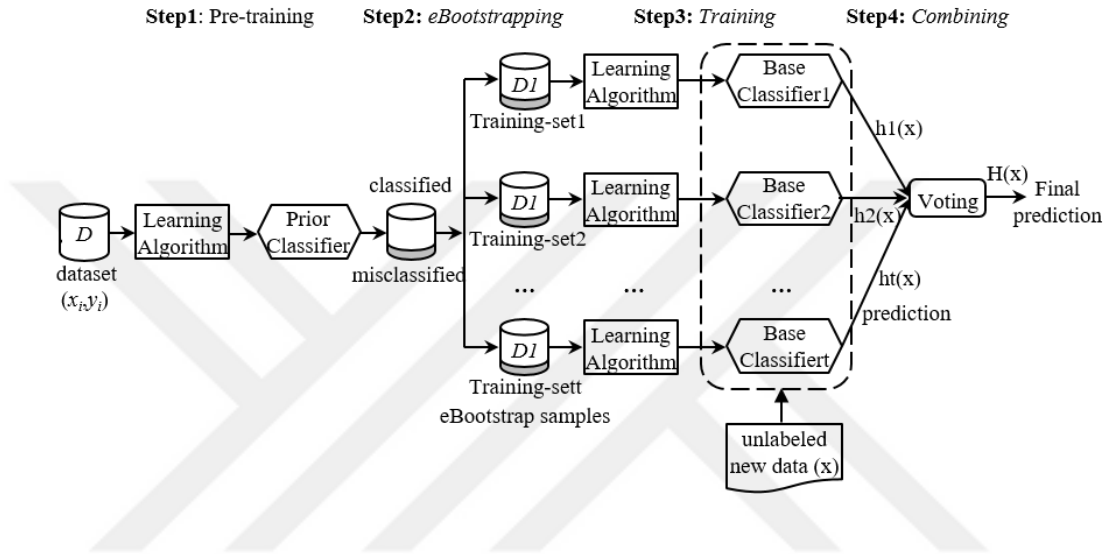


Figure 3.3 The general framework of Enhanced Bagging (eBagging) algorithm

- *Step 1: Pre-training* A prior classifier is applied to the original dataset and the dataset is divided into two parts: one part that contains correctly classified instances and the others (incorrectly classified).
- *Step 2: eBootstrapping* eBootstrapping is used to construct different training sets by directly pushing misclassified instances and resampling with replacement from classified instances. So, the difficult instances are always included in each subset of data. This step both provides diversity and allows the learning algorithm to focus on hard-to-classify examples and so it gives us a reasonable starting point.
- *Step 3: Training* The base classifiers are trained on different subsets of the training patterns. In this study, the learning algorithms used in pre-training and training steps (in other words the learning algorithms of prior classifier and base classifier) are planned to be the same, but different learning algorithms can also be tried in future studies.

- *Step 4: Combining (Aggregating)* The base classifiers perform classification tasks and the final prediction is made by applying majority voting to the outputs of each ensemble subset. If classifiers disagree with each other, then the incorrect errors of the different classifiers can be removed by the voting mechanism.

Boosting technique (i.e., AdaBoost algorithm) also tries to classify the hard-to-classify examples accurately and ignores the ones which are easy-to-classify. However, the proposed method is different from boosting in two-folds. First, eBagging generates training sets from the original dataset in parallel, so it is not an iterative approach as boosting. Second, eBagging doesn't assign weight values to each instance as boosting. Instead of that, eBagging is directly copied all difficult examples into all training sets.

The effectiveness of the proposed method depends on constructing a diverse, yet accurate, collection of classifiers. By eBagging, we obtain a good compromise between a lack of diversity and a premature overfitting situation. Figure 3.4 demonstrates the pseudo-code of the proposed eBagging algorithm. A classifier (or a hypothesis) is a mapping from x to y . Given set D containing n examples, misclassified instances are identified by a prior classifier in the pre-training step and denoted by M . Classified instances are the rest of the dataset and denoted by C . Datasets $D_1...D_t$ are generated by including all misclassified instances in M and some correctly classified data points taken with replacement from C . After training, a new instance is classified by voting.

In Table 3.2, there is a summary of the properties of eBagging, bagging, random forest, and boosting techniques in terms of given criteria.

```

Inputs: Dataset  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ 
       where  $y_i \in Y = \{1, \dots, k\}$ 
        $L$ : Learning algorithm for prior and base classifiers
        $t$ : ensemble size
        $n$ : the number of instances
        $k$ : the number of classes
        $x$ : an unlabeled instance to classify

Process:
// Step1: Pre-training
 $h = L(D)$  // train to build prior classifier
 $C = \emptyset, M = \emptyset$  // generate new datasets
for  $i = 1$  to  $n$ 
    if  $h(x_i) = y_i$ 
         $C.add(x_i, y_i)$  // correctly classified instances
    else
         $M.add(x_i, y_i)$  // misclassified instances
for  $i = 1$  to  $t$ 
    // Step2: eBootstrapping
     $D_i = \emptyset$ 
     $D_i = D_i \cup M$ 
    while  $D_i.length \neq m$ 
         $r = \text{random.next}(1, C.length)$ 
         $D_i.add(C_r)$ 
    // Step3: Training
     $h_i = L(D_i)$  // train to build base classifier
Output:
// Step4: Combining
 $H(x) = \text{voting}(h_1(x), h_2(x), \dots, h_n(x))$  // final hypothesis
        $= \arg \max_{y \in Y} \sum_{i: y = h_i(x)}^t 1$ 

```

Figure 3.4 The pseudo-code of enhanced bagging (eBagging)

Table 3.2 Comparison table of eBagging, Bagging, Random Forest and Boosting techniques

	eBagging	Bagging	Random Forest	Boosting
Training set construction	misclassified samples + random resampling	random resampling	random resampling+random feature subset	weighted resampling
Model construction	parallel	parallel	parallel	iterative (incremental)
Weight assignment to each instance	no	no	no	yes
Classifiers	inducer independent	inducer independent	inducer dependent (decision tree)	inducer independent
Training datasets	giving higher preference to misclassified samples	random	random	giving higher preference to misclassified samples
Goal to achieve	increase the performance of the base learners with a prior learner	minimize variance	minimize variance	increase predictive force
Aim	find a good trade-off between variance and bias that minimizes prediction error	decrease variance	increase diversity	decrease bias
Methods where this is used	error-based random subspace	random instance subspace	random instance subspace and random feature subspace	gradient descent
Function to combine	majority voting	majority voting	majority voting	weighted majority voting
Inducer number	single inducer	single inducer	single inducer	single inducer
Base learner	simple or complex	complex	simple	simple
Effect	loss minimization	risk minimization	risk minimization	margin maximization

3.1.2.4 Advantages of Ensemble Learning in Environmental Informatics

Some of the advantages of applying ensemble learning in environmental informatics are given below:

- Prediction of the expected parameters based on other parameters or under different cases in environmental studies. For example, prediction of rainfall (Kühnlein, Appelhans, Thies, & Nauss, 2014), climate change (Gaál, Moriondo, & Bindi, 2012; Nelson et al., 2014; Muñoz-Mas et al., 2016) species richness/diversity (Knudby, Brenning, & LeDrew, 2010; Kocev & Džeroski, 2013) and atmospheric parameters (Lima et al., 2013)
- Construction of models to reduce the consumption of energy (Fan, Xiao, & Wang, 2014; Araya et al., 2017; Jovanović, Sretenović, & Živković, 2015) and raw materials (Srivastava, 2012) such as wood, grass, metal, steel, plastics, glass, paper, fuel and natural gas
- Analyzing the items in environmental data to describe the current situation more clearly and to plan different activities for different clusters (Bougoudis et al., 2016; Kocev & Džeroski, 2013)
- Classification of environmental audio and environmental noise (Bravo-Moncayo, Naranjo, García, & Mosquera, 2017)
- Processing ecological data for better modeling ecological systems (Knudby et al., 2010, Kocev & Džeroski, 2013) environmental data toward a better quality control such as air quality (Stojić et al., 2016; Lei & Wan, 2012; Singh et al., 2013) and water quality (Granata & de Marinis, 2017; Budka et al., 2010; Rodriguez-Galiano et al., 2014)
- Identifying unexpected patterns and detecting anomalies in environmental data (Srivastava, 2012; Araya et al., 2017) to identify bad values, changes, errors, noises, frauds, and abnormal activities to realize the purpose of giving an alarm
- Determination of the most important factor that affects the environment (Luo & Kathuria, 2013)
- Development of a model to manage resources effectively (Srivastava, 2012; Fan et al., 2014; Jovanović et al., 2015), including environmental resources such as air, water and soil; flow resources such as solar power (Mohammed, Yaqub, & Aung, 2015) and wind energy; and natural resources such as coal, gas, and forests
- Discovering patterns that can be used for better waste management and recycling

- Analyzing the records of financial transactions related to environmental economics for better decision making, i.e. investigating the financial impacts of environmental policies
- Using ensemble methods as a preprocessing step before performing the essential environmental study

3.1.2.5 Challenges of Ensemble Learning in Environmental Informatics

Even ensemble-based environmental data mining is helpful based on the pre-mentioned advantages, some challenges could be overcome when you are aware. Challenges can be grouped under five main titles: selecting ensemble strategy, determining a satisfactory architecture, computational cost, complex nature of environmental data, and finally post-processing.

- *Selecting ensemble strategy*: It is a difficult work to determine the best ensemble strategy in terms of accuracy, scalability, computational cost, usability, compactness, and speed of classification. Environmental researchers should know how to construct an ensemble model and be aware of alternative strategies and advantages/disadvantages of them. To overcome this problem, EDML is mostly addressed to the computer and environmental scientists working together.
- *Determining a satisfactory architecture*: There are two levels of problems in designing ensemble architecture. First, it is necessary to determine the optimal ensemble size. There are three approaches for determining the ensemble size: (i) pre-selection of the ensemble size, (ii) selection of the ensemble size while training, and (iii) post-selection of the ensemble size (pruning). Second, how are learning algorithms and their respective parameters selected to construct the best ensemble? The best values for the input parameters of the algorithms should be determined through a number of tries. These problems are fundamentally different and should be solved separately to improve classification accuracy. Furthermore, it is necessary to update the model when new environmental data is acquired, allowing the up-to-date model to change over time.

- *Computational cost*: Increasing the number of classifiers usually increases computational cost. To overcome this problem, users may predefine a suitable ensemble size limit or classifiers can be trained in parallel.
- *Complex nature of environmental data*: It is necessary to deal with the high dimensionality and complexity of environmental data. To reduce the dimensionality of the feature vector, feature selection techniques can be used such as principal component analysis, information gain, and ReliefF. Another problem is to deal with heterogeneous data by adding problem-specific science algorithms to the solution.
- *Post-processing*: Another critical issue is determining what is the best voting mechanism (majority, weighted, average, etc.) for combining the outputs of base classifiers? Furthermore, the final results should be presented in an appropriate form to help users understand and interpret easily.

3.2 Case Study 1 – Ensemble Methods in Environmental Data Mining

The study presented in this part focuses on several ensemble strategies in addition to the standard single classifiers such as decision tree, naive Bayes, support vector machine, and k-nearest neighbor, popularly used in literature. This is the first study that compares four ensemble strategies for environmental data mining: (i) different training sets formed by random sampling with replacement (bagging), (ii) different training sets obtained by random instance and feature subset selection (Random Forest), (iii) different training sets using random sampling with replacement over weighted data (AdaBoost) and (iv) different algorithms (vote) (Tuysuzoglu, Birant, Kut, & Pala, 2020).

Data mining techniques have been recently utilized in environmental studies for processing environmental data and converting it to useful patterns to obtain valuable knowledge and making the right decisions when dealing with environmental problems. Many of the developed techniques in data mining can often be tailored to fit environmental data.

Recently, ensemble learning has been one of the active research fields in machine learning. Thus, it has been utilized in a very broad range of areas such as marketing, banking, insurance, health, telecommunication, and manufacturing. In contrast to these studies, this work proposes an ensemble learning approach that combines several models to produce a result to solve environmental problems.

In this study, different ensemble learning strategies were compared in terms of classification accuracy, precision, recall, and f-measure. Four ensemble learning strategies were tested on six different real-world environmental datasets. The application was developed by using Weka's open-source data mining library.

3.2.1 Dataset Description

In this experimental study, six different datasets that are available for public use were selected to determine the best ensemble strategy. Basic characteristics of the investigated environmental datasets are given in Table 3.3.

Table 3.3 Environmental datasets and their characteristics

ID	Dataset Name	Year	Attributes	Instances	Type	Link
1	ozone-onehour	2008	74	2536	Air	http://archive.ics.uci.edu/ml/datasets/Ozone+Level+Detection
2	ozone-eighthour	2008	74	2534		
3	leaf	2014	17	340	Ecology	http://archive.ics.uci.edu/ml/datasets/Leaf
4	eucalyptus	1991	20	736	Soil	https://weka.wikispaces.com/Datasets
5	forestType	2015	28	523	Ecology	https://archive.ics.uci.edu/ml/datasets/Forest+type+mapping
6	cloud	1971	8	108	Rainfall	https://github.com/renatopp/arff-datasets/blob/master/statlib/nominal/cloud.arff

3.2.2 Experimental Study and Results

Classification accuracies, precision, recall, and f-measure values for the applied algorithms were obtained using 10-fold cross-validation. A comparison of the classification accuracies of the applied algorithms for each dataset is displayed in

Figure 3.5. Four weak learners (support vector machine (SVM), naive Bayes (NB), decision tree (DT) applied with C4.5 algorithm, k-nearest neighbor (KNN)) and four ensemble learners (bagging, random forest (RF), AdaBoost and vote) were used to construct classification models from environmental data. The base classifiers for the ensemble learners were selected as the one which gave the best classification accuracy among the applied weak learners for the respective dataset.

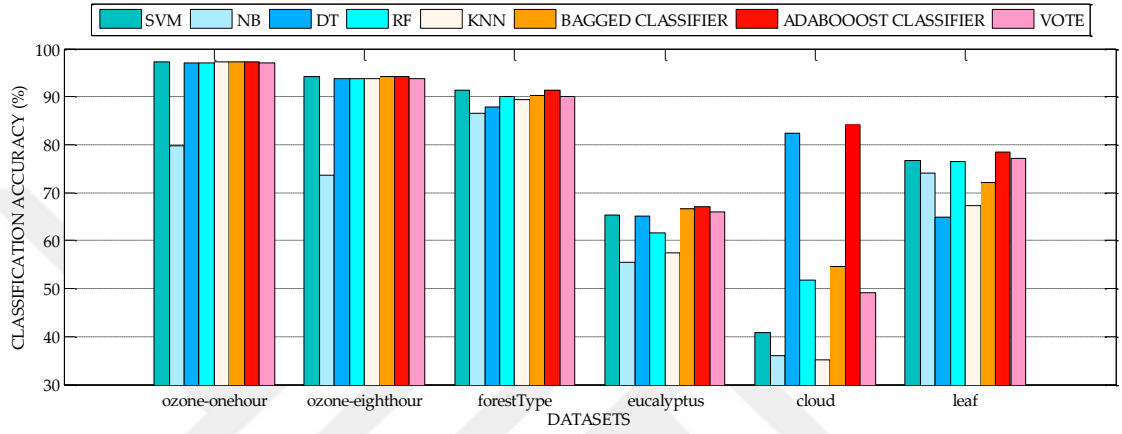


Figure 3.5 Comparison of single and ensemble classifiers in terms of classification accuracies

The experimental results were obtained with optimum parameters (given in Table 3.4) using grid search. The best parameters of SVM were found for the complexity parameter, C and the exponent value, E for polykernel parameters in the interval $[10^k$ for $k \in \{-3, \dots, 3\}]$ and $[1-10]$ respectively. To model DT, confidence factor, C for pruning, and the minimum number of objects, M per leaf were obtained in the intervals of $[0.05 - 0.95]$ and $[1-10]$. The number of neighbors, N for KNN classifier was selected in the range of $[1, 25]$. For the RF classifier, the number of randomly chosen attributes, K and the number of iterations to be performed, I , were found in the interval $[0-15]$ and $[10-100]$ respectively. The number of ensemble classifiers for bagging is 10 for each dataset. Weight threshold for weight pruning, P and the number of iterations to be performed, I were selected in the interval $[10-100]$ for AdaBoost classifier. The Vote was performed using the optimum parameters of SVM, NB, DT, KNN, and RF classifiers.

The objective of this experiment is to remark the success of the ensemble strategies in terms of classification accuracy concerning environmental data. According to the experimental results, it is apparent that the number of correctly classified instances is increased if ensemble strategies are applied. Especially, the AdaBoost classifier provides significant performance gain compared to other models. SVM has superiority over other single-learners hence most of the ensemble models selected it as the base learner.

Table 3.4 Optimum classifier parameters corresponding to each dataset

Dataset	SVM		DT		KNN		RF		ADABOOST	
	C	E	C	M	N	Distance Metric	I	K	I	P
ozone-onehour	10^3	2	0.05	1	17	Euclidean Distance	10	5	10	10
ozone-eighthour	10^3	5	0.55	1	5	Chebyshev Distance	10	5	100	10
leaf	10^0	1	0.05	2	1	Manhattan Distance	60	0	100	10
eucalyptus	10^1	1	0.15	2	9	Manhattan Distance	50	2	80	40
forestType	10^0	1	0.15	3	11	Manhattan Distance	50	11	10	10
cloud	10^0	1	0.05	1	17	Euclidean Distance	100	0	100	40

There are a number of cases resulting in poor classification performance, such as:

- In case of a presence of either noisy or missing data,
- If there is an insufficient number of instances available,
- If there are too much number of classes,
- If a complex relationship is inherent,
- If the feature dependencies are ignored,
- If the feature selection is not well performed,
- If the algorithm parameters are not correctly determined or,
- If the class labels are imbalanced.
- For example, because the number of instances in the “cloud” dataset is very few (due to the insufficient number of instances), inferior results are obtained for most of the applied algorithms as expected. However, even in such cases while some

algorithms fail, some others manage to perform well (e.g. C_{4.5} DT 82%). In this situation, the classifier's performance can also be enhanced by applying ensemble learning methods as in the case of AdaBoost with 84% classification accuracy for the same dataset. AdaBoost is a powerful ensemble learning algorithm because its distribution update step ensures that instances misclassified by the previous classifier are more likely to be included in the training data of the next classifier with the chance of further enhancement.

- Because classification accuracy as a performance metric is not just enough to decide whether a learner is considerably good or not, the precision, recall, and f-measure values were also calculated for each model (Table 3.5). It is also clear from the table values that applying ensemble strategies compared to single-learners makes more sense in terms of classifier performance.

Table 3.5 Precision (PRE), recall (REC) and f-measure (F-MEA) results using 10-fold cross-validation for respective algorithms in each dataset

Dataset	Algorithm	PRE	REC	F-MEA	Dataset	Algorithm	PRE	REC	F-MEA
Ozone(one-hour)	SVM	0.97	0.97	0.95	eucalyptus	SVM	0.65	0.65	0.65
	NB	0.96	0.79	0.86		NB	0.62	0.55	0.55
	C _{4.5} DT	0.94	0.97	0.95		C _{4.5} DT	0.66	0.65	0.64
	RF	0.94	0.97	0.95		RF	0.61	0.61	0.61
	K-NN	0.97	0.97	0.95		K-NN	0.57	0.57	0.56
	K-NN _{Bagged}	0.97	0.97	0.95		SVM _{Bagged}	0.66	0.66	0.66
	K-NN _{AdaBoost}	0.97	0.97	0.95		SVM _{AdaBoost}	0.67	0.67	0.67
	Vote	0.94	0.97	0.95		Vote	0.67	0.65	0.65
Ozone(eight-hour)	SVM	0.93	0.94	0.93	cloud	SVM	0.37	0.40	0.37
	NB	0.92	0.73	0.80		NB	0.49	0.36	0.32
	C _{4.5} DT	0.87	0.93	0.90		C _{4.5} DT	0.82	0.82	0.82
	RF	0.91	0.93	0.91		RF	0.51	0.51	0.51
	K-NN	0.87	0.93	0.90		K-NN	0.33	0.35	0.32
	SVM _{Bagged}	0.92	0.94	0.93		C _{4.5} DT _{Bagged}	0.55	0.54	0.54
	SVM _{AdaBoost}	0.93	0.94	0.93		C _{4.5} DT _{AdaBoost}	0.84	0.84	0.84
	Vote	0.93	0.93	0.91		Vote	0.47	0.49	0.46
ForestTypes	SVM	0.91	0.91	0.91	leaf	SVM	0.78	0.76	0.76
	NB	0.86	0.86	0.86		NB	0.75	0.74	0.74
	C _{4.5} DT	0.88	0.88	0.87		C _{4.5} DT	0.66	0.65	0.64
	RF	0.90	0.90	0.90		RF	0.77	0.76	0.76
	K-NN	0.89	0.89	0.89		K-NN	0.69	0.67	0.67
	SVM _{Bagged}	0.90	0.90	0.90		SVM _{Bagged}	0.72	0.72	0.71
	SVM _{AdaBoost}	0.91	0.91	0.91		SVM _{AdaBoost}	0.79	0.78	0.78
	Vote	0.90	0.90	0.90		Vote	0.77	0.77	0.76

3.3 Case Study 2 – Classification of Air Quality Network based on Meteorological and Pollutant Factors

As a consequence of the generated contamination during the production and consumption activities occurring from various activities of the people, the atmosphere becomes dirty and the living life on earth is affected negatively. Due to the fuels burned for various purposes, smoke from factory and house chimneys, exhaust gases of vehicles air pollution has been increasing in some parts of the world in recent years. Many developed countries have established air quality measurement networks to monitor pollutant concentrations and in case of exceedance from acceptable pollution levels alerts are issued.

Air quality monitoring networks can be composed of different types of stations according to the respective criteria such as the location type (urban, suburban or rural) or the surrounding factors affecting the region (industrial, traffic or background). The determination of which category a station should be placed facilitates the decision making authority of these regions to take similar control mechanisms about air quality. In this way, emission sources are more accurately canalized to the corresponding stations.

In this study, different well-known classification algorithms (SVM, KNN, Naive Bayes, C4.5 and multilayer perceptron), their ensemble models (AdaBoost, bagging, random subspace, vote) and Random Forest in addition to the recently proposed “Enhanced Bagging (eBagging)” algorithm were applied to classify a new station into the correct station category (urban, rural or industrial). The potential contributions of this study can be listed as follows:

- Similar regions in terms of station type can be easily identified so that air quality management strategies for them can be implemented similarly.
- Any time-series data (combined with meteorological factors) of a new station can be analyzed and the process makes it possible to classify it as belonging to one of the groups.

- Common characteristics among air quality monitoring stations in the same class can be discovered.
- Gathering similar information and summarizing the common features as a result of classification help environmental scientists understand the current situation more clearly.
- An optimal air quality monitoring network can be designed by offering adequate information using minimum measurement devices which are assigned according to the station type.
- A novel ensemble learning algorithm is used in environmental informatics for the first time.

In the experimental studies, daily concentrations of five air pollutant data (PM₁₀, SO₂, NO, NO₂, and O₃) in addition to daily meteorological data such as temperature, relative humidity, and air pressure collected from 21 air quality monitoring stations of Turkey in the period of 1 January 2018 and 1 May 2019 were conducted.

3.3.1 Dataset Description

In this study, daily concentrations of air pollutants (NO₂, NO, PM₁₀, O₃, and SO₂) and daily meteorological data (relative humidity, temperature, and air pressure) from 21 air quality monitoring stations in Turkey were considered. The AQMSs are selected from 13 provinces which are from three different regions (Blacksea, Eastern Anatolia, and Marmara) of Turkey. The properties of each station are noted in Table 3.6 by displaying the name of AQMS, the city where the station is located, the county of the city, latitude and longitude information, the type of the network (i.e. rural/urban/industrial), and air pollutants that are regularly measured in the respective station. The concentration values of each pollutant are accessible online at the web page of <http://laboratuvar.cevre.gov.tr>.

Table 3.6 The selected air quality monitoring stations (AQMS) and their features

ID	AQMS Name	City	County	Longitude	Latitude	Type	The Measured Pollutants
1	Artvin	Artvin	Merkez	41.8182	41.1752	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
2	Bayburt	Bayburt	Merkez	40.2255	40.2558	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
3	Canakkale - Biga Icdas	Canakkale	Biga	27.1072	40.4173	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
4	Canakkale - Can - MTHM	Canakkale	Can	27.0498	40.0293	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
5	Edirne - Kesan - MTHM	Edirne	Kesan	26.6352	40.8511	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5}
6	Erzincan	Erzincan	Merkez	39.4950	39.7430	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
7	Erzurum - Pasinler	Erzurum	Pasinler	41.5721	40.0335	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
8	Igdir	Igdir	Merkez	44.0536	39.9261	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
9	Igdir - Aralik	Igdir	Aralik	44.6209	39.7868	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5}
10	Istanbul - Basaksehir - MTHM	Istanbul	Basaksehir	28.7898	41.0954	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
11	Istanbul - Esenyurt - MTHM	Istanbul	Esenyurt	28.6688	41.0192	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
12	Kocaeli - Gebze - MTHM	Kocaeli	Gebze	29.4365	40.8108	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
13	Kocaeli - Korfez - MTHM	Kocaeli	Korfez	29.7888	40.7461	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5} , PM _{2.5} Flow Rate
14	Kocaeli - Yenikoy - MTHM	Kocaeli	Basiskele	29.8844	40.7042	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
15	Rize	Rize	Merkez	40.5328	41.0217	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
16	Rize - Ardesen	Rize	Ardesen	41.0475	41.1273	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5}
17	Trabzon - Akcaabat	Trabzon	Akcaabat	39.5923	41.0143	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
18	Yalova - Armutlu-MTHM	Yalova	Armutlu	28.7845	40.5292	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5}
19	Zonguldak - Eren Enerji Lise	Zonguldak	Catalagzi	31.8801	41.4964	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO, PM _{2.5}
20	Zonguldak - Eren Enerji Santral	Zonguldak	Catalagzi	31.8964	41.5105	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO, PM _{2.5}
21	Zonguldak - Eren Enerji Tepekoy	Zonguldak	Catalagzi	31.9374	41.5269	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO, PM _{2.5}

The National Air Quality Monitoring Network of Turkey is comprised of 313 AQMSs. In this study, all the AQMSs were examined and 21 out of 313 stations were selected since meteorological data and the aforementioned air pollutants are regularly measured in these stations together in the period of 1 January 2018 and 1 May 2019. The pollutant concentrations and meteorological data are mean values of daily (24 hours) measurements. After missing data elimination, 9980 instances are available in the given time interval.

3.3.2 Experimental Study and Results

In this study, a number of classification methods (SVM, KNN, naive Bayes (NB), multilayer perceptron (MLP) and C4.5) are compared to each other using single (no ensemble strategy used) and ensemble learning algorithms (random forest, vote, bagging, AdaBoost, random subspace (RS) and the proposed eBagging method). The application was developed using Weka's open-source data mining library (Eibe, Hall, & Witten, 2016) on Visual Studio. The data set used in the experiments and the experimental results are presented in this section.

Classification accuracy, precision, recall, and F-measure values for the applied algorithms were calculated using ten-fold cross-validation. Table 3.7 displays the results of these performance metrics.

The number of iterations to be performed, parameter I , was selected as 10 for all of the ensemble methods. The other parameters of all classification methods were left as default Weka parameters. The only difference is in the RS; the subspace size parameter which is the number of features selected for each ensemble iteration was set as 0.7. The Vote ensemble strategy is created using SVM, KNN, NB, C4.5, and MLP in different iterations of the ensemble model so that diversity is provided.

Once you have a model it is time to decide whether it is a good enough model to solve your problem. Taking classification accuracy alone into account is typically not enough to make this decision. Precision, recall, and F-measure are other important performance metrics to be taken into consideration and they are also used in this study. Because the data set used here has imbalanced distributed class labels including instances from 11 urban stations, 6 industrial stations, and 4 rural stations, precision and recall values should also be consulted in addition to accuracy. Precision is an important measure to determine when the cost of False Positive is high. It shows how precise/accurate your model is out of those estimated as positive, how many of them are actually positive. On the other side, recall talks about how many of the Actual Positives our model finds through labeling it as Positive (True Positive). It is

appropriate to consider when there is a significant cost associated with False Negative. Finally, F-measure provides a balance between recall and precision.

Table 3.7 The results of performance metrics of the applied algorithms

Algorithm	Accuracy (%)	Precision	Recall	F-Measure
SVM	65.47	0.65	0.66	0.64
KNN	78.97	0.79	0.79	0.79
NB	55.72	0.68	0.56	0.54
C4.5	86.95	0.87	0.87	0.87
MLP	77.52	0.77	0.78	0.77
RANDOM FOREST	90.73	0.91	0.91	0.91
VOTE	84.55	0.85	0.85	0.85
ADABOOST _{SVM}	65.47	0.65	0.66	0.64
ADABOOST _{KNN}	78.97	0.79	0.79	0.79
ADABOOST _{NB}	55.72	0.68	0.56	0.54
ADABOOST _{C4.5}	90.55	0.91	0.91	0.91
ADABOOST _{MLP}	81.51	0.82	0.82	0.82
BAGGING _{SVM}	65.62	0.65	0.66	0.64
BAGGING _{KNN}	79.55	0.80	0.80	0.80
BAGGING _{NB}	55.77	0.68	0.56	0.54
BAGGING _{C4.5}	90.78	0.91	0.91	0.91
BAGGING _{MLP}	82.37	0.82	0.82	0.82
RS _{SVM}	63.78	0.63	0.64	0.57
RS _{KNN}	80.55	0.81	0.81	0.80
RS _{NB}	56.45	0.68	0.57	0.55
RS _{C4.5}	90.38	0.90	0.90	0.90
RS _{MLP}	76.60	0.77	0.77	0.75
eBAGGING _{SVM}	65.49	0.65	0.65	0.64
eBAGGING _{KNN}	87.31	0.87	0.87	0.87
eBAGGING _{NB}	55.76	0.68	0.56	0.54
eBAGGING _{C4.5}	91.29	0.91	0.91	0.91
eBAGGING _{MLP}	78.16	0.78	0.78	0.78

The best classification accuracy, precision, recall, and F-measure values are indicated by bold notation and the best classifier with all, which is eBAGGING_{C4.5}, is shown with the gray background in Table 3.7. It is apparent that single learners cannot achieve well compared to their ensemble versions. When single classifiers are analyzed, the highest classification performance can be observed in the C4.5 algorithm; also, Random Forest which is an ensemble of decision trees gives more accurate results than C4.5, as expected.

In order to examine the performance comparison among ensemble models more in-depth, Figure 3.6 points their accuracies under different base classifiers. In two (C4.5 and KNN) out of five models, eBagging significantly outperforms its counterparts and when Naive Bayes and SVM algorithms are used as a base classifier, it achieved approximately head to head results with Bagging. In just one of the cases (MLP), Bagging performs better than eBagging with the highest score. Furthermore, the purpose of dealing with misclassified instances is also valid for one of the powerful ensemble learning methods, the AdaBoost algorithm. In this study, it has succeeded to compete with AdaBoost, as well.

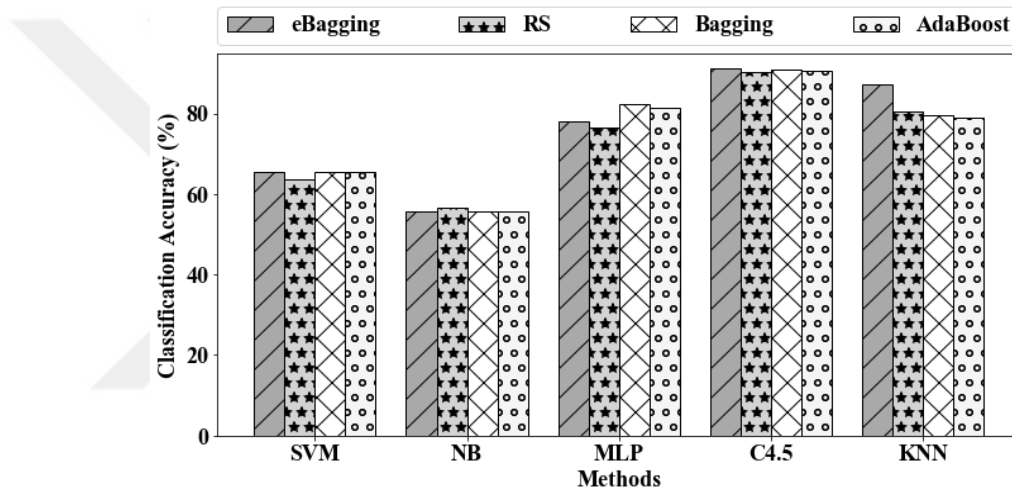


Figure 3.6 Comparisons of the applied ensemble models in terms of accuracy

As a consequence of the experimental work, the proposed algorithm (eBagging) is proven to be an improved version of the standard bagging algorithm by handling hard-to-classify data samples, deeply. The idea is simple, easy to put into practice, and provides major benefits concerning the other applied algorithms. With this study, an optimal air quality monitoring network can be designed by discovering common characteristics among air quality monitoring stations. Moreover, a new ensemble learning algorithm is presented by providing promising results in environmental informatics.

CHAPTER FOUR

REGRESSION STUDIES IN ENVIRONMENTAL INFORMATICS

This chapter proposes a new model, called *Soil Temperature prediction via Self-Training* (STST), which successfully estimates the soil temperature at various soil depths by using machine learning methods. Unlike the previous studies, it aims to raise the sample size with unlabeled data when the amount of pre-labeled data is scarce to form a prediction model. In this study, soil-related data and meteorological data collected for nearly nine months were taken into consideration to estimate the new values. With this study, by detecting how the soil temperature will change in the future, necessary precautions for plant development can be taken earlier and agricultural returns can be obtained beforehand.

4.1 Material and Methods

In this section, the applied methods in experimental studies are explained. A new semi-supervised learning technique called “Soil Temperature prediction via Self-Training (STST)” is introduced. Seven regression algorithms including support vector regression, k-nearest neighbor regression, extra tree regression, decision tree regression, extreme gradient boosting, random forest regression, and multiple linear regression are explained. Self-training and its adaptation to the soil temperature prediction are also expressed fully.

4.1.1 Regression Algorithms

Regression is a supervised learning technique that is used to predict a range of continuous values for future samples of a particular dataset. It determines the relationship between the dependent variable and one or more independent variables.

In this section, the applied regression methods in the experimental study are discussed.

4.1.1.1 Support Vector Regression

Support vector regression (SVR) which is a variant of support vector machines (SVM) is used for the continuous-valued regression problems instead of categorical valued classification problems. The main aim is to minimize the error by maximizing the margin around the separating hyperplane. The general formula can be written as an optimization problem as in (4.1) where w is the normal vector to the bounding planes, C is the penalty associated to the instances which are either misclassified or violate the maximal margin, $\phi(X_i)$ is a function which maps data point X_i into a higher dimensional space, b shows the positions of bounding planes relative to the origin and ξ is a slack variable for soft margins defined for linearly non-separable cases.

$$\begin{aligned} \min & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ \text{subject to } & Y_i(w^T \phi(X_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \end{aligned} \quad (4.1)$$

In non-linear problems, the kernel functions, $K(X_i, X_j) = \phi(X_i)^T \phi(X_j)$, are used to transform the data into a higher dimensional feature space to make it possible to perform the linear separation. Two kernel functions are generally used for these cases as polynomial kernel (*poly kernel*) or Gaussian radial basis function (*RBF kernel*). For two samples X_i and X_j , the poly kernel function and RBF kernel function can be written as in (4.2) and (4.3), respectively.

$$K(X_i, X_j) = (aX_i^T X_j + b)^d \quad (4.2)$$

$$K(X_i, X_j) = \exp\left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right) \quad (4.3)$$

4.1.1.2 K-Nearest Neighbours Regression

It is the adaptation of the k-nearest neighbor algorithm for continuous-valued problems. The difference from the version implemented for classification problems is that the outcome of the k nearest samples to a specific instance are averaged instead of applying majority voting. The optimal size of the neighborhood can be determined by performing cross-validation where the one that gives the minimum error (i.e. mean square error or R^2) is selected.

4.1.1.3 Decision Tree Regression

Decision trees can be applied for both classification and regression problems. The structure is similar to the classification model by constructing a tree with recursive partitioning, however, the impurity measure is changed from entropy measure to another criterion for regression problems. For continuous variables, the mean squared error (MSE) of the children nodes is used.

4.1.1.4 Extra Tree Regression

It is an ensemble of decision trees where cut-points are randomly determined while splitting nodes, on the other hand, the whole samples are used as given at the beginning instead of performing bootstrapping. The final output is assigned by averaging the results of ensemble iterations. Two important parameters are the number of randomly selected features and the minimum sample size for splitting a node.

4.1.1.5 Random Forest Regression

Random forest is also adapted for numerical outputs in regression problems instead of predicting categorical class labels as in the case of the classification task. It is a meta estimator based on decision trees applied to many bootstrapped subsamples of a dataset. First of all, a specified number of decision tree regressors are built. A subset of features is randomly selected to be used as candidates at each split so that the

constructed decision trees do not rely on the same set of features and high correlation among trees can be prevented. The bootstrapped instances also prevent the individual trees from overfitting. The numeric predictions of each estimator are averaged and assigned to the test sample as the final output.

4.1.1.6 Extreme Gradient Boosting

As in the case of extra tree regression and random forest regression methods, extreme gradient boosting (XGBoost) is also one of the ensemble models of decision trees. It is based on gradient boosting in which errors are minimized by the gradient descent algorithm. By adding models on top of each other iteratively, the errors of the previous model are corrected by the next predictor, until the training data is accurately predicted or reproduced by the model. Instead of assigning different weights to the classifiers after every iteration, gradient boosting fits the new model to new residuals of the previous prediction and then minimizes the loss when adding the latest prediction. XGBoost uses this algorithm with an additional custom regularization term in the objective function.

4.1.1.7 Multiple Linear Regression

Multiple Linear Regression (MLR) is a statistical technique which includes various explanatory variables $X_i = \{x_1, x_2, \dots, x_m\}$ for any object $o_i = (X_i, Y_i)$ where there are m features to predict the outcome of a response variable Y_i . It is an extension of ordinary least squares (OLS) (Cameron and Huppert, 1989) which uses only one explanatory variable. The general formula of MLR is as follows:

$$Y_i^* = \alpha_0 + \alpha_1 X_{i1} + \alpha_2 X_{i2} + \alpha_m X_{im} + \epsilon \quad (4.4)$$

In (4.4), Y_i^* is the predicted output for response variable Y_i , α_0 is a constant term for y-intercept, α_i is the slope coefficient for each explanatory variable of X_i and ϵ is the error term (residuals) of the model. The main aim is to predict the values for all

response variables, Y_i for n objects. The explanatory variables should be independent, i.e., there are no major correlations among them.

4.1.2 The Proposed Method: Semi-supervised Soil Temperature Prediction via Self-Training (STST)

Semi-supervised learning is a machine learning approach in which there are a small number of samples whose output is known and a large number of samples with unknown labels to develop a classification/regression model during training (Belkin, Niyogi, & Sindhvani, 2006). One of the semi-supervised learning methods is *self-training* (Zhu & Goldberg, 2009). The rationale behind self-training is to increase the training set size with unlabeled data when there is a very small number of pre-labeled data compared to unlabeled ones so that a more optimized classifier model can be constructed using the updated labeled training set. Considering the problem of soil temperature prediction, we may not always be able to access all past data whose results are known to estimate new values. Because of the high cost of manual labeling, it is hard to obtain sufficient, reliable, and up-to-date labeled data for effective soil temperature prediction. In such cases, by applying the proposed *Soil temperature prediction via the self-training model (STST)*, we first estimate the values for the unlabeled historical samples and then use all past records to estimate the new values for the future records.

The proposed approach (STST) has many advantages that can be summarized as follows:

- The traditional soil temperature prediction is limited to using only labeled data to build a model. The main advantage of the STST approach is that it overcomes this limitation and deals with the design of prediction models in the presence of both labeled and unlabeled data. In addition to labeled data, the STST approach also exploits unlabeled data to help improve soil temperature prediction performance. Thanks to the STST approach, the unlabeled data samples provide additional knowledge that is relevant for prediction, and they can successfully be used to improve the generalization ability of the learning system.

- In many real-world agricultural applications, a huge amount of unlabeled data is available. However, labeling agricultural data is an expensive, difficult, or time-consuming process since it usually requires human efforts and hardware equipment. This is especially true for soil temperature prediction applications that include learning from data collected from the agriculture fields. The proposed STST approach addresses this inherent bottleneck by automatically allowing the model to integrate the available unlabeled data with little or no cost.
- An important advantage of the STST approach is that it can be used with the combination of any supervised base learner such as SVR, KNNReg, and ETReg. The STST approach is entirely unaware of the regression method, in fact, it simply learns from the labeled and pseudo-labeled samples as if they were regular labeled samples.
- Another advantage is that the STST approach can be applied to any soil data without any prior information about the given dataset. It does not make any specific knowledge and specific assumptions for the given data.
- Since the STST approach covers multiple soil depths, it enables enormous agricultural applications, and so it expands the application of the machine learning algorithms in the field of agriculture.

The algorithm in Figure 4.1 displays the pseudo-code of the proposed model which is applied for different soil depths. The aim is to obtain the soil temperature values (Y^*) of new instances in D_{Test} at all depths. In the first part, there are two sets of instances: the labeled dataset (D_L) and the unlabeled dataset (D_U). The hidden patterns affecting the soil temperature can be detected by analyzing the labeled instances. Therefore, using the instances in D_{L_d} , a classifier C_d is trained and a model that facilitates labeling other instances is obtained at the depth d . In this way, the pseudo-label (y_i) of each instance x_i in D_{U_d} is discovered. All the pseudo-labeled instances are then gathered together in D_d . Now, a new labeled training set D_{ALL} is ready by expanding the initially given D_{L_d} with the pseudo-labeled set D_d . The final step is to predict the temperature values (Y_d) of new instances at the depth d by using the new classifier model C_d trained with D_{ALL} . The resulting output is the predicted soil temperature values of all depths as the collection of each Y_d in the set Y^* .

Algorithm Soil temperature prediction via self-training (STST)

Inputs: D_L : the labeled dataset $D_L = \{(x_i, y_i)\}_{i=1}^l$ with l instances D_U : the unlabeled dataset $D_U = \{x_j\}_{j=l+1}^{l+u}$ with u instances D_{Test} : test instances**Outputs:** Y^* : predicted values

for each depth d **do** $C_d = \text{Train}(D_{L_d})$ **foreach** x_i **in** D_{U_d} $y = C_d(x_i)$ $D_d.\text{Add}(x_i, y)$ **end foreach** $D_{ALL} = D_{L_d} \cup D_d$ $C_d = \text{Train}(D_{ALL})$ **foreach** x_i **in** D_{Test} $y = C_d(x_i)$ $Y_d = Y_d \cup y$ **end foreach** $Y^* = Y^* \cup Y_d$ **end foreach**

End Algorithm

Figure 4.1 The pseudo-code of soil temperature prediction via self-training (STST)

Figure 4.2 expresses the main course of this study as a series of processes. The first part is the collection of soil data at different depths, meteorological data, and light intensity by using IoT devices and sensors or by pulling data from web pages. After data collection, if needed, missing data imputation can also be performed, otherwise, the machine learning process including self-training takes part. Future samples are assigned their predicted temperature values as an output. The final part is the reporting and presentation facility to adjust the findings in an interpretable format.

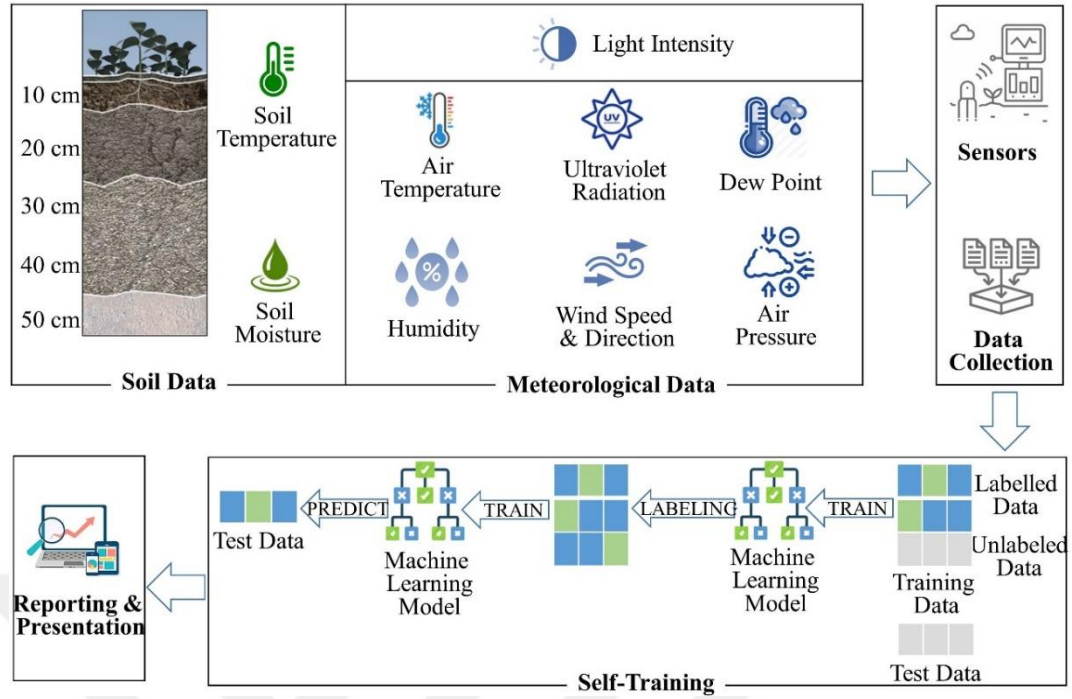


Figure 4.2 The general framework of the proposed “Soil temperature prediction via self-training (STST)” model

4.2 Case Study - Soil Temperature Prediction using Semi-supervised Learning

The purpose of this study is to estimate the soil temperature in Izmir, Turkey by investigating the dynamics of the past soil temperature & soil moisture data at various depths and sunlight lux value which were collected by Arduino and Internet of Things (IoT) sensors and meteorological data (air temperature, humidity, dew point, air pressure, wind speed, wind direction, ultraviolet radiation) taken from the web site (“<https://weather.com>”) between the dates of 01.09.2019 and 22.05.2020.

The main contributions of this study can be listed as below:

- This is the first study that is performed by one of the semi-supervised learning techniques known as “self-training” in the subject of soil temperature prediction.
- It proposes a novel model, called *Soil Temperature prediction via Self-Training (STST)*. The STST facilitates the capability of temperature estimation for new samples in case there are few numbers of labeled samples to discover the hidden patterns.

- This is the first time that the analysis is being reported in detail to determine which regression method provides the most accurate predictions under the self-training framework and the variation in the performances. For this purpose, it compares the self-training versions of machine learning algorithms, including random forest (RF), support vector regression (SVR), k-nearest neighbor regression (KNNReg), extra tree regression (ETReg), decision tree regression (DTReg), and XGBoost.
- This study is also original in that it investigates the performances of the semi-supervised machine learning algorithms on soil temperature prediction with different ratios of labeled data varying from 5% to 85% with an increment of 5.
- It should be highlighted that this study is the first to propose a multi-depth self-training learning framework that considers estimating the soil temperatures at five different soil depths (10, 20, 30, 40, and 50 cm).
- It presents a new application of semi-supervised machine learning to provide a smart way of soil temperature prediction.

4.2.1 Dataset Description

The location of the experimental area is Izmir, Turkey during the dates of 01.09.2019 and 22.05.2020. A part of the data set used in the experiments was collected using IoT devices (Arduino Mega, Arduino Shield) and various sensors (DS18B20 soil temperature (ST) sensor and soil moisture (SM) sensor for measuring hourly data at the soil depths of 10 cm., 20 cm., 30 cm., 40 cm., and 50 cm., and light-dependent resistor (LDR) sensor for hourly light intensity (LI)). The meteorological part of the data set was obtained from the web page of “<https://weather.com>” that provides air temperature (AT), humidity (H), dew point (DP), air pressure (AP), wind speed (WS), wind direction (WD), and ultraviolet index (UV). Hourly values of the aforementioned features were taken into consideration. After data collection, the records with missing values were removed. Finally, 4500 instances were left for data analysis.

A sample fragment of the data set is given in Table 4.1. SM and LI sensors have a limit of 0 to 1023. If the value of SM is close to 1023, it means soil moisture is high,

otherwise low. In the same manner, if the value of LI is high, the light intensity is a lot. For the prediction of ST of each depth, SM values from all depths are taken into consideration. In the sample data, ST at the depth of 50 cm is illustrated.

Table 4.1 A part of the data set used in the experiments

<i>AT</i>	<i>H</i>	<i>DP</i>	<i>AP</i>	<i>UV</i>	<i>WS</i>	<i>WD</i>	<i>LI</i>	<i>SM₅₀</i>	<i>SM₄₀</i>	<i>SM₃₀</i>	<i>SM₂₀</i>	<i>SM₁₀</i>	<i>ST₅₀</i>
(°C)	(%)	(°C)	(mb)		(km/s)								(°C)
28	41	13	1012.9	2/10	14	North	822	272	263	283	342	324	24.69
19	39	4	1020	1/10	13	East	899	347	290	606	968	285	19.50
30	32	11	1011.5	6/10	5	West	849	960	968	1002	1004	933	23.19
11	73	7	1007.1	1/10	2	Northwest	799	44	80	672	694	936	11.75

4.2.2 Experimental Study and Results

In order to indicate the influence of self-training, six regression methods were applied, including random forest (RF), support vector regression (SVR) with the polynomial kernel, k-nearest neighbor regression (KNNReg), extra tree regression (ETReg), decision tree regression (DTReg), and extreme gradient boosting (XGBoost). From here onwards, the abbreviation of the self-training (ST) method followed by the abbreviation of the base classifier technique is used to refer to the related approach. For example, ST-SVR refers to the self-training method with the SVR base classifier. The implementation was done on Python 3.7.6 on Spyder. The parameters of all the applied methods were left as their default values in the “sklearn” library of Python.

Two performance metrics (mean squared error (MSE) and coefficient of determination (R^2)) were measured to evaluate the usability of the proposed methodology and to select the best one in terms of the given criteria. The results show the outputs of ten-fold cross-validation. By considering the ratio of initially given training data as $p\%$, the experimental results were obtained for all varying values of p as 5% to 85% with an increment of 5.

Table 4.2 displays the MSE values of the methods under the self-training framework at the depth of 50 cm. Even though there are very few known cases of soil temperature, in the beginning, the constructed models achieved a remarkable performance by predicting too close to the real values for test cases. The self-training model led to increase the amount of labeled data by pseudo-labeling the past training data. As a result, the constructed regression model had the advantage of discovering more patterns hidden in data due to more sets of instances with known outputs.

Table 4.2 The comparison of the applied methods under self-training framework in terms of MSE values

%	Methods					
	<i>ST-RF</i>	<i>ST-SVR</i>	<i>ST-KNNReg</i>	<i>ST-ETReg</i>	<i>ST-XGBoost</i>	<i>ST-DTReg</i>
5	2.277	18.594	6.374	5.150	2.189	4.336
10	1.721	16.753	4.084	3.804	1.474	3.177
15	1.252	15.506	3.133	3.075	0.866	3.093
20	1.134	14.509	2.858	2.866	0.814	2.493
25	1.045	14.077	2.559	2.356	0.660	2.142
30	0.915	13.663	2.299	2.197	0.637	1.768
35	0.842	13.222	2.171	1.937	0.653	1.530
40	0.790	12.842	2.023	2.100	0.551	1.486
45	0.740	12.535	1.899	1.595	0.494	1.334
50	0.678	12.234	1.799	1.629	0.486	1.296
55	0.618	12.034	1.734	1.521	0.486	1.143
60	0.570	11.768	1.658	1.665	0.462	1.156
65	0.550	11.504	1.601	1.400	0.416	1.139
70	0.537	11.301	1.552	1.413	0.414	1.015
75	0.537	11.087	1.512	1.445	0.426	0.993
80	0.450	10.896	1.479	1.262	0.391	0.996
85	0.453	10.729	1.443	1.165	0.365	0.957

Furthermore, the most noticeable thing is the reduction in error values as the percentage (p) value increases. It is because as the number of training instances real labels of which are known rises, the pseudo-labeled training data is less required to predict the outcome of new cases. More accurate estimations can be obtained as a result.

The best results for all the ratio values were achieved when ST-XGBoost was performed. XGBoost is one of the ensemble learning methods which boosts high

performance compared to single learners so it was expected that its results were good. ST-RF is another ensemble learning method that followed ST-XGBoost in terms of low error values. The highest errors were obtained when ST-SVR was the learner. It can be inferred that ST-SVR required more labeled training data in order to find the optimal hyperplane to make more accurate predictions. It could not manage to perform well when the value of p is small. The performance of ST-SVR may be improved by changing the kernel type or updating the parameters instead of using default values.

The comparisons of the methods according to R^2 values are indicated in Figure 4.3 at the depth of 50 cm. R^2 expresses the proportion of the variance for a dependent variable that is explained by the model's inputs. In this study, it is the relationship between soil temperature and other independent variables such as soil moisture, air temperature, ultraviolet radiation, etc. According to the results, the general impression is that there is a steady increase as the value of p increases. The same condition as in MSE is valid for R^2 that ST-SVR has the lowest coefficient of determination (0.331 to 0.614) while the best results are found in ST-XGBoost (0.921 to 0.987).

To show the trend in the results as the soil depth changes, Table 4.3 demonstrates the MSE values of each applied method by taking their average for all p values from 5 to 85. Especially for ensemble learning methods, there is a general pattern that the error first increases for the soil depth of 10 to 30 and then a decreasing trend follows as the depth increases from 40 to 50 cm. Their common characteristic is that they can better estimate the deepest soil temperature. On the contrary, there is a decreasing trend in most of the cases for single learners from shallow to deeper parts. ST-SVR and ST-DTReg manage to predict the best at the depth of 50 cm. while ST-KNNReg performs well at the depth of 40 cm. The main inference for the proposed model is that the more accuracy is generally obtained the deeper the soil.

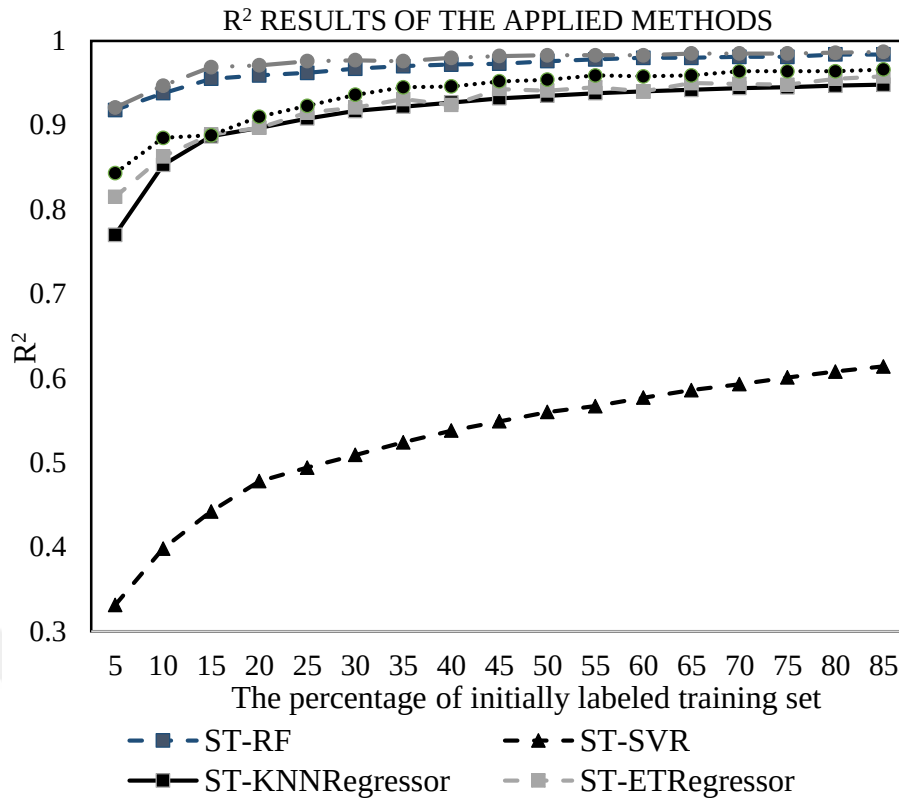


Figure 4.3 The comparison of the applied methods under self-training framework in terms of R2 values

Table 4.3 The comparison of the mean values of the applied methods in terms of MSE values under self-training framework at different soil depths

<i>Methods</i>	<i>Depth</i>				
	<i>50</i>	<i>40</i>	<i>30</i>	<i>20</i>	<i>10</i>
ST-RF	0.889	0.953	0.986	0.968	0.951
ST-ETReg	2.152	2.432	2.591	2.424	2.341
ST-SVR	13.133	14.001	14.430	14.327	14.397
ST-XGBoost	0.693	0.834	0.894	0.846	0.756
ST-KNNReg	2.363	2.310	2.345	2.351	2.655
ST-DTReg	1.768	1.957	2.033	1.981	2.012

This study contributes to the agricultural field in a way that plant growth can be handled more efficiently by taking the forecasted soil temperature values into account. An automated irrigation system or cooling/heating system can be set up according to the variation in the temperature of the forecasted time intervals.

CHAPTER FIVE

CLUSTERING STUDIES IN ENVIRONMENTAL INFORMATICS

Air pollution, which is the result of the urbanization brought by modern life, has a dramatic impact on the global scale as well as local and regional. Since air pollution has important effects on human health and other living things, the issue of air quality is of great importance all over the world. Accordingly, many studies based on classification, clustering, and association rule mining applications considering air pollution have also been proposed in the field of data mining and machine learning to extract hidden knowledge from environmental parameters. One approach is to model a region in a way that cities having similar characteristics are determined and placed into the same clusters. Instead of using traditional clustering algorithms, a novel algorithm, named *Majority Voting based Multi-Task Clustering* (MV-MTC), is proposed and utilized to consider multiple air pollutants jointly. In the experimental studies, five well-known clustering algorithms, including k-means, expectation-maximization, canopy, farthest first, and hierarchical clustering methods are also implemented.

5.1 Material and Methods

In this section applied methodologies are presented. The overall goal of the used techniques is to create clusters with a consistent set of similar behavioral points by ensuring the maximum similarities in intra-cluster objects while keeping inter-cluster differences high. The clustering algorithms or techniques used in this study are k-means, expectation-maximization, canopy, farthest first, and hierarchical clustering in addition to the proposed technique Multi-task clustering.

5.1.1 Clustering Algorithms

Clustering is unsupervised learning where a set of objects are put into different groups by considering the intra-cluster similarity and inter-cluster difference. There is no pre-defined label as in the case of the classification task. Dividing the data into

clusters can be on the basis of centroids (i.e. k-means), distributions (i.e. expectation-maximization), densities (i.e. DBSCAN), or connectivity type (i.e. hierarchical clustering), etc.

In this section, the applied clustering methods in the environmental case study including k-means, expectation-maximization, canopy, farthest first, and hierarchical clustering are given.

5.1.1.1 K-Means

Consider a dataset $D = \{o_1, o_2, \dots, o_n\}$ where each o_i represents an object as a p -dimensional explanatory variable and n is the number of objects (instances) in the dataset. Assume that the problem domain is to be divided into k clusters combination of which is represented as a vector $C_{KM} = \{C_1, C_2, \dots, C_k\}$ and the centroids of k clusters are denoted by $\mu = \{m_1, m_2, \dots, m_k\}$.

The first step is to assign k points as cluster centers at random. The distance between each data point o_i and each cluster centroids m_j , where $i = \{1, \dots, n\}$ and $j = \{1, \dots, k\}$, are calculated using one of the distance metrics such as *Euclidean*, *Manhattan*, *Chebyshev* or *Minkowski* distance etc. as $\text{argmin}_j \text{dist}(o_i, m_j)$ to find the nearest cluster for the respective instance to be assigned. New cluster centroids are calculated by $m_j = (1/n_j) \sum_{o_i \in m_j} o_i$, where n_j denotes the number of objects in cluster j , C_j . This process iteratively continues until no data point changes cluster membership. According to the method used for the initialization of the process, different techniques instead of random initialization can be used such as *k-means++*, *farthest first*, or *canopy*.

5.1.1.2 Expectation Maximization

It extends the k-means paradigm differently. While the k-means algorithm assigns each data point to a cluster, each object in the expectation-maximization (EM) model is assigned to each cluster according to a weight representing the probability of

membership. In other words, there is no definite limit between clusters, and new centers are calculated in terms of weighted measures (Han, Pei, & Kamber, 2011). EM clusters data points using a finite mixture density model, i.e. normal distribution, of k probability distributions, where each distribution represents a cluster.

As in the k-means clustering, the process starts with selecting cluster centroids randomly. The procedure continues with two steps to refine the parameters (i.e. clusters) iteratively based on statistical modeling: Expectation (E) step and Maximization (M) step (Dempster, Laird, & Rubin, 1977). In step E, a function to determine the probability of cluster membership of an instance is generated using the present estimate for the attributes using (5.1) where $p(o_i|C_j)$ follows the normal distribution and $i = \{1, \dots, n\}, j = \{1, \dots, k\}$.

$$P(o_i \in C_j) = p(C_j|o_i) = \left(p(C_j)p(o_i|C_j) \right) / p(o_i) \quad (5.1)$$

M step is applied as in (5.2) for re-estimating the model parameters by discovering the attributes which maximize the expected log-likelihood found in E. The iterative process continues until obtaining the optimal value.

$$m_j = (1/n) \sum_{i=1}^n \frac{o_i P(o_i \in C_j)}{\sum_t P(o_i \in C_t)} \quad (5.2)$$

5.1.1.3 Canopy

The general application of canopy is on the preprocessing step of other clustering algorithms such as k-means or hierarchical clustering to speed up the process in case of large data sets (McCallum, Nigam, & Ungar, 2000). The procedure uses two distance metrics $T1 > T2$ to be used for later processing and a list of data points to cluster. The initial canopy center is determined randomly from one of the data points and then distances of all other instances to this canopy center are approximated. The instances whose distance values fall within the threshold of $T1$ is placed into a canopy while the data points whose distance value fall within the threshold of $T2$ are removed

from the list. These removed ones are excluded from being selected as a new canopy center or creating new canopies. The process iteratively continues until the list is empty.

5.1.1.4 *Farthest First*

It is one of the variants of k-means clustering where each cluster centroid is selected in turn at the point furthest from the existing cluster centers. This point must lie within the data area. This significantly boosts the speed of clustering in general due to the need for less reassignment and modification (Sharma, Bajpai, & Litoriya, 2012).

5.1.1.5 *Hierarchical Clustering*

Hierarchical clustering is used to group data objects into a tree of clusters either by bottom-up (agglomerative) or top-down (divisive) fashion (Liu, Lu, Nie, Su, & Yang, 2016). In the agglomerative version, each instance of data set is put into its own cluster initially and all of these atomic clusters are merged continuously until a single cluster is formed to hold all data points inside or if there is a termination condition. The divisive version is just the opposite of agglomerative clustering because it begins the process with a single cluster where all data points are placed and the later steps are the subdivision of the cluster into smaller distinct ones until a termination criterion is satisfied such as a predetermined number of clusters is obtained.

According to the distance calculation method between different clusters, there are many link types used in hierarchical clustering such as *Single* (a.k.a. the minimum link which is the closest distance between any items of two different clusters), *Complete* (a.k.a. the maximum link which is the largest distance between any items of two different clusters), *Average* (a.k.a. average distance between the elements of two clusters), *Mean* (the mean distance of merged cluster) and *Centroid* (the distance from one centroid to another).

5.1.2 Proposed Method: Majority Voting Based Multi-Task Clustering (MV-MTC)

Multi-task learning (MTL) is a learning technique in which useful information contained in several relevant tasks is leveraged to help improve the overall performance of all tasks. All of these tasks or at least a subset of them are assumed to be related to each other. In many of the applications, it is found that learning these tasks jointly leads to much performance improvement compared with learning them individually. In most of the fields comprising of computer vision, bioinformatics, health informatics, speech and natural language processing, web applications, and ubiquitous computing, MTL is used to enhance the overall performance of the applications involved. Learning paradigms including supervised learning (e.g., classification or regression problems) (Xue, Liao, Carin, & Krishnapuram, 2007; Liu, Qiu, & Huang, 2016; Liu, Gao, He, Deng, Duh, & Wang, 2019), unsupervised learning (Zhang, 2015; Zhang et al., 2018; Zhang, Zhang, Liu, & Liu, 2017; Siahpirani, Ay, & Roy, 2016; Zhang, Zhang, Liu, & Liu, 2016; Yan, Ricci, Liu, & Sebe, 2015; Zhang, Zhang, Liu, 2015), semi-supervised learning (Liu, Liao, & Carin, 2007; Qi, Tasthan, Carbonell, Klein-Seetharaman, & Weston, 2010; Lu, Li, & Mou, 2015; Zhang & Yeung, 2009), active learning (Reichart, Tomanek, Hahn, & Rappoport, 2008; Zhang, 2010; Acharya, Mooney, & Ghosh, 2017; Fang & Tao, 2015), reinforcement learning (Wilson, Fern, Ray, & Tadepalli, 2007; Parisotto, Ba, & Salakhutdinov, 2015; Li, Liao, & Carin, 2009; Lazaric & Ghavamzadeh, 2010), multi-view learning (Zhang et al., 2016; Zhao, Xie, Xu & Sun, 2017; He & Lawrence, 2011; Gao et al., 2017), and graphical models (Honorio & Samaras, 2010; Oyen & Lane, 2012; Yan, Ricci, Subramanian, Lanz, & Sebe, 2013) are generally combined with MTL.

Multi-task classification and *multi-task clustering* are two well-known types of multi-task learning recently presented in the literature. In the study of Wang, Yan, Lu, Zhang, & Li (2018), Multi-task classification was used in the prediction of air pollution particles by implementing a deep multi-task learning framework. On the other hand, multi-task clustering has not been studied until now for air quality management, neither in environmental science.

There is an issue to be addressed: "what to share" while learning multiple tasks. The form of sharing type determines which knowledge sharing among all the tasks could occur. Usually, there are three forms of sharing including feature, instance, and parameter. *Feature-based MTL* aims to learn common features among different tasks. *Instance-based MTL* identifies useful data objects in a task for other tasks and then shares knowledge via the identified instances. *Parameter-based MTL* uses model parameters in a task to help learn model parameters in other tasks (Zhang & Yang, 2017). The proposed method in this study (MV-MTC) is among the popularly applied unsupervised learning schemes of instance-based MTL applications.

MTC has been applied in many different areas including bioinformatics, text mining, web mining, image mining, daily activity recognition, and so on (Zhang et al., 2018; Zhang et al., 2017; Siahpirani et al., 2016; Zhang et al., 2016; Yan et al., 2015; Zhang et al., 2015; Zhang, Zhang, Liu, & Luo, 2008). The resulting clustering template of MTC has generally outperformed the single clustering algorithms' outputs. Table 5.1 presents a brief list of studies in which different MTC algorithms were proposed and applied in various subject areas. It is experimentally proven that MTC algorithms provide a remarkably significant performance when compared to single-task learners.

A *task* is generally referred to the construction of a model using a specific dataset for a single target or a sub-goal. In this sense, "*multiple tasks*" could mean the modeling of multiple output targets simultaneously by using task-related datasets and by considering task relations. Depending on the definition of "multiple tasks", we can define multi-task clustering as follows: *multi-task clustering* (MTC) is a process of generating global clusters that are shared by the multiple related tasks. MTC is desired to merge information among tasks to improve the clustering performance of individual tasks. The most important aspect in MTC is to discover the shared information among tasks. In this study, a novel algorithm, named Majority Voting based Multi-task Clustering (MV-MTC), is proposed to provide this aspect (Tuysuzoglu, Birant, & Pala, 2019).

Consider the unlabeled dataset $D = \{o_1, o_2, \dots, o_n\}$ where each o_i represents an object as a p -dimensional explanatory variable and n is the number of objects (instances) in the dataset. Assume that the problem domain consists of r different tasks $T = \{t_1, t_2, \dots, t_r\}$ each of which is represented as t_i .

In the first step of the algorithm, the instance set allotted to each task should be properly clustered using one of the traditional clustering algorithms. For r different tasks, let's denote the resulting clustering assignments as $C = \{C_{t_1}, C_{t_2}, \dots, C_{t_r}\}$ where $C_{t_i} = \{c_1, c_2, \dots, c_k\}$ for the predetermined number of clusters as k and each c_i consists of different o_i 's from the dataset D . In order to take the joint decision from all C_{t_i} 's, a common factor should be determined because the same cluster names do not have to represent the same clustering structure among the task groups. We need to determine common cluster labels meaning the same information through all tasks.

In this context, after clustering instances of each task by one of the single clustering algorithms, all clusters are labeled from the common label set $L = \{L_1, L_2, \dots, L_k\}$ as in Table 5.2 in terms of the mean weights of intra-cluster objects and k cluster labels for k clusters are produced according to the cluster weights. To illustrate if we have 3 clusters, the heaviest one, the medium-weighted one and the lightest one can be labeled as 'L3', 'L2' and 'L1', respectively. The same procedure is applied to r tasks. As shown in the following example, all instances in the dataset are labeled with a suitable cluster label L_i for each task t_i . As the final stage, like in the majority voting approach, the most common cluster label among all tasks for a given instance o_i is selected as the final cluster assignment. Therefore, the novel MTC algorithm is called Majority Voting based Multi-task Clustering (MV-MTC).

Table 5.1 The summary table of the data mining studies taking “Multi-Task Clustering” as the main subject

Ref.	Subject Area	Aim	Algorithms/Methods	Performance Metrics
Zhang et al., 2018	Bioinformatics	Generate a model that utilizes multiple single-cell populations from biological replicates or different samples to address the cross-population clustering problem of scRNA-seq data	scVDMC , KM, Pooled KM, SNN-Cliq, CellTree, Seurat, SC3	ARI, Cluster Error (measured on the best one-to-one matching between the detected clusters and the true clusters)
Zhang et al., 2017	Text Mining and Image Mining	Propose a general multi-task clustering algorithm by transferring knowledge of instances through reweighting the distance between samples in different tasks by learning a shared subspace and selecting the nearest neighbors for each sample from the other tasks	MTCTKI , KM, KKM, Ncut-GK, Ncut-SNN, LSSMTC, DMTFC, SMT-NMF, MTCTKI0	Clustering Accuracy, NMI
Siahpirani et al., 2016	Bioinformatics	Identify common and context-specific aspects of genome architecture	Arboretum-Hi-C , KM, HC, SC	DBI, SI, D, Number of enriched clusters, log <i>P</i> value of ANOVA
Zhang et al., 2016	Web Page Mining and Image Mining	Develop a co-clustering based multi-task multi-view clustering framework which integrates within-view-task clustering, multi-view relationship learning and multi-task relationship learning	KM, KKM, NSC, BiCo, SNMTF, CoRe, CoTr, LSSMTC, DMTFC, BMTMVC , SMTMVC	Clustering Accuracy, NMI
Liu et al., 2016	Bioinformatics	Automated HEp-2Cells Classification	CMTL and the other 28 methods presented in the HEp-2Cells Classification contest held at the 2012 International Conference on Pattern Recognition	Accuracy
Yan et al., 2015	Activity Recognition	Daily living analysis from visual data gathered from wearable cameras	EMD-MTC with linear and RBF kernel denoted as CEMD-MTC and KEMD-MTC respectively; KM, KKM, CNMF and SemiNMF, SemiEMD-MTC, KSemiEMD-MTC and the LSMTC method	Clustering Accuracy, NMI
Zhang et al., 2015	Document Clustering	Identifying and avoiding negative effects of the boosting process of MBC and also dealing with nonlinear separable data in the clustering of documents	SMBC and S-MKC : KM and KKM; MBC	Clustering Accuracy, NMI

NMF: nonnegative matrix factorization, Ncut-GK: normalized cut with Gaussian kernel similarity, BiCo: bipartite graph co-clustering, DMTFC: convex discriminative multi-task feature clustering, MTCTKI0: MTCTKI without using shared subspace, NSC: normalized spectral clustering, SNMTF: semi-nonnegative matrix tri-factorization, CoRe: co-regularized multi-view spectral clustering, CoTr: co-trained multi-view spectral clustering, SC3: single-cell consensus clustering, SNN-Cliq: shared nearest neighbor cliq, LSSMTC: learning the shared subspace for multi-task clustering, DBI: Davies-Bouldin index, SI: Silhouette index, D: delta contact count, SMT-NMF: symmetric multi-task non-negative matrix factorization, ARI: adjusted rand index, HC: hierarchical clustering, SC: spectral clustering, EMD-MTC: earth mover's distance regularized multi-task clustering, NMI: normalized mutual information, KM: k-means, S-MKC: smart multi-task kernel clustering, KKM: kernel k-means, CMTL: clustered multi-task learning, MBC: multitask Bregman clustering, SMBC: smart multi-task bregman clustering, BMTMVC: bipartite graph based multi-task multi-view clustering, Ncut-SNN: normalized cut with shared nearest neighbor similarity, MTCTKI: multi-task clustering by transferring knowledge of instances, scVDMC: variance-driven multitask clustering of singlecell RNA-seq data, SMTMVC: semi-nonnegative matrix tri-factorization based multi-task multi-view clustering, Arboretum-Hi-C: multi-task spectral clustering algorithm for comparative analysis of Hi-C data

Table 5.2 Assignments of cluster labels under different tasks

Instance	Cluster Label Assignments for t_1	Cluster Label Assignments for t_2	...	Cluster Label Assignments for t_{r-1}	Cluster Label Assignments for t_r
o_1	L_1	L_2	...	L_{k-1}	L_k
o_2	L_2	L_k	...	L_{k-1}	L_3
...	...	...	...	...	...
o_{n-1}	L_k	L_2	...	L_1	L_4
o_n	L_1	L_{k-2}	...	L_{k-1}	L_k

This study proposes two novel concepts of single-task clusters and multi-task clusters. In the first phase, the proposed algorithm discovers local clusters (*single-task clusters*) from each task data separately, and in the second phase, these local clusters are combined to produce the global result (*multi-task clusters*).

- *Definition 1. (Single-Task Clusters)*

Single-task clusters are groups of instances discovered from the data partition D_t of a particular task t , i.e. $D = \bigcup_{t=1}^r D_t$, and denoted by $C_{t_i} = \{c_1, c_2, \dots, c_k\}$, where k is the number of clusters.

- *Definition 2. (Multi-Task Clusters)*

Given r tasks $T = \{t_i\}_{i=1}^r$ where all the tasks are related but not identical, *multi-task clusters*, which are denoted by $C = \{C_{t_1}, C_{t_2}, \dots, C_{t_r}\}$, are groups of instances that mostly appear in the same level of the clusters of the tasks.

Based on these definitions, it is possible to say that there are two elementary factors for multi-task clustering. The first factor is the definition of the task. Many real-world problems consist of a number of related subtasks. For instance; PM₁₀, SO₂, NO₂, NO, and O₃ air pollutants can be considered as the tasks of air quality monitoring problem. The second factor is the definition of the ensemble method to combine multiple tasks. In this study, a majority voting mechanism, which selects the cluster that is the one with the most votes was used.

To figure out the rationale behind the algorithm, the example scenario in Table 5.3 and Table 5.4 explain the process step by step. In the first stage, the dataset D which is full of instances with only one feature is given. There are three tasks (t_1 , t_2 , and t_3) and the aim is to group the dataset into three clusters by taking the joint decision from

each task. The attribute value of instances can change according to different tasks. The next step is applied for clustering instances by one of the clustering algorithms simultaneously for each task. Instances are properly assigned to one of three clusters (C_1 , C_2 , or C_3). On the other hand, we need to determine a common decision point on the cluster groups of different tasks to get the final cluster assignments. Therefore, three labels (L_1 , L_2 , and L_3) are used to generalize the clusters and mean the same groupings under different tasks according to average intra-cluster weights. In the final part, after instances are labeled with the new label set for every task (C_{t_1} , C_{t_2} and C_{t_3}), a majority voting scheme is applied to obtain final cluster labels for the MV-MTC algorithm.

Table 5.3 An example showing MV-MTC algorithm step by step

Ins. ID	Task t_1	Task t_2	Task t_3
1	3	23	43
2	10	15	30
3	21	40	54
4	9	32	89
5	18	14	72
6	12	27	28
7	6	24	26
8	4	41	22
9	17	33	79
10	19	28	58
11	10	15	47
12	3	18	73

(1) Dataset D and its instance values in terms of three tasks

Cluster ID	Instance Assignments		
	Task t_1	Task t_2	Task t_3
C_1	{2, 4, 6, 11}	{3, 4, 8, 9}	{1, 3, 10, 11}
C_2	{3, 5, 9, 10}	{2, 5, 11, 12}	{4, 5, 9, 12}
C_3	{1, 7, 8, 12}	{1, 6, 7, 10}	{2, 6, 7, 8}

(2) Instance assignments to clusters under different tasks

Cluster ID	Common Label Assignments					
	Task t_1	Avg. Weight	Task t_2	Avg. Weight	Task t_3	Avg. Weight
C_1	L2	10.25	L3	36.50	L2	50.50
C_2	L3	18.75	L1	15.50	L3	78.25
C_3	L1	4.00	L2	25.50	L1	26.50

(3) Determination of the common cluster labels according to average intra-cluster weights

Table 5.4 Assignments of final cluster labels under different tasks and the output of MV-MTC algorithm

Ins. ID	C_{t_1}	C_{t_2}	C_{t_3}	$MV-MTC$	Ins. ID	C_{t_1}	C_{t_2}	C_{t_3}	$MV-MTC$
1	L1	L2	L2	L2	7	L1	L2	L1	L1
2	L2	L1	L1	L1	8	L1	L3	L1	L1
3	L3	L3	L2	L3	9	L3	L3	L3	L3
4	L2	L3	L3	L3	10	L3	L2	L2	L2
5	L3	L1	L3	L3	11	L2	L1	L2	L2
6	L2	L2	L1	L2	12	L1	L1	L3	L1

(4) Assignments of final cluster labels to the instances in terms of different tasks and the output of MV-MTC from majority voting rule

Figure 5.1 displays the general framework of the multi-task clustering algorithm where t_i 's showing every single task of the task space and D as the unlabeled data. The main purpose is to ensure that the instances in the clusters that are created before the MV-MTC result remain in the same set in the final step. The number of instances remaining in the same cluster is maximized according to (5.3) where $C_{ij}(o_r)$ means that the instance o_r is the member of cluster c_j of task t_i and $MV - MTC_j$ indicates the resulting cluster c_j of the MV-MTC algorithm. The pseudo-code of the proposed algorithm is given in Figure 5.2.

$$\max \left\{ \sum_{i=1}^r \sum_{j=1}^k \sum_{r=1}^n 1 : C_{ij}(o_r) \in MV - MTC_j, o_r \in C_{ij} \right\} \quad (5.3)$$

As shown in Figure 5.2, the methodology is made up of four steps. In the first step, single-task clusters are generated by considering each individual task. Step 2 is performed to calculate intra-cluster weights under different tasks. In Step 3, cluster labels are assigned to clusters according to their weight values, assigning L_1 to the cluster which has the lowest mean value and then increasing the label values until giving L_k to the highest one. The last step is the place where a joint decision from different tasks is taken by applying a majority voting mechanism. As a result, all data points are placed into the most suitable clusters and final cluster labels are assigned from the joint decision.

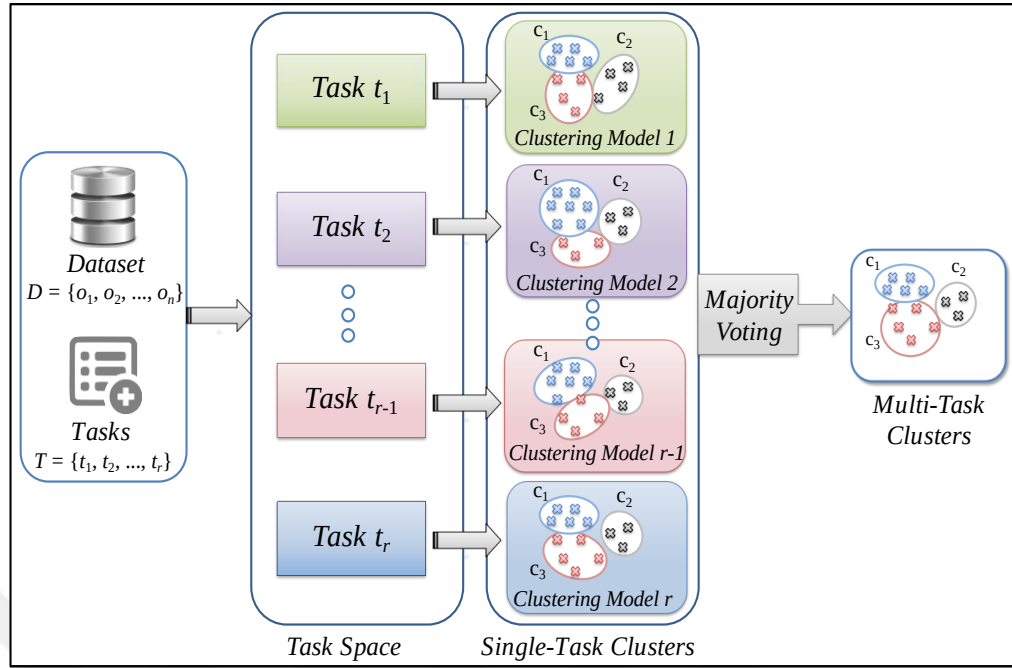


Figure 5.1 The general MV-MTC clustering scheme

Algorithm Majority Voting based Multi-task Clustering (MV-MTC)

Inputs: Dataset $D = \{o_1, o_2, \dots, o_n\}$

Task space $T = \{t_1, t_2, \dots, t_r\}$

CA: a clustering algorithm

Cluster label set $L = \{L_1, L_2, \dots, L_k\}$

n : the number of instances

k : the number of clusters

r : the number of tasks

Process:

// Step1: Clustering according to task t_i

1. **for** $i = 1$ to r

2. $C_{t_i} = CA(D, t_i)$

3. $C.add(C_{t_i})$

Output:

$C = \{C_{t_1}, C_{t_2}, \dots, C_{t_r}\}$ // cluster assignments in terms of different tasks

$C_{t_i} = \{c_1, c_2, \dots, c_k\}$ // k different clusters under the task t_i

// Step2: Determine average intra-cluster weights

4. **for each** C_{t_i} in C

5. **for** $i = 1$ to k

6. **for each** o in c_k

7. $sum = sum + o$ // value of the instance

8. $m_i = sum / |c_k|$

Output:

$\mu_i = \{m_1, m_2, \dots, m_k\}$ // k different average intra-clusters weights under the task t_i

// Step3: Label each cluster c_i in C_{t_i} for all tasks according to μ_i values

9. **for each** c_i in C_{t_i}

10. **for** $i = 1$ to k

11. $L_{c_i} = L_{index(m_i)}$ // an appropriate cluster label from L in terms of the index of m_i

// Step4: Obtaining joint decision

12. **for** $i = 1$ to n

13. **for** $j = 1$ to r

14. $L(o_i) = \arg \max_{L_i \in L} \sum_{L_{t_j}(o_i)} 1$

// final cluster assignment of o_i where each L_{t_j} is the cluster label of o_i in task t_j and $j \in \{1, 2, \dots, r\}$

Figure 5.2 The algorithm of the MV-MTC clustering

Different types of MTC algorithms have also been proposed. For instance, multi-task multi-view clustering is presented to handle the learning problem of multiple related tasks with one or more common views (Zhang et al., 2016). Each view is associated with one task or multiple related tasks, the inter-task knowledge is transferred to one another, and multi-task and multi-view relationships are exploited to improve clustering performance. In (Zhang et al., 2016), it is applied for webpage and image mining operations under the clustering framework.

5.2 Case Study - Majority Voting Based Multi-Task Clustering of Air Quality Monitoring Network in Turkey

Air pollution is now recognized as an important problem all over the world. It can be referred to as a mixture of multiple pollutants that vary in size and composition. Air pollutants (a.k.a. "criteria pollutants") are commonly grouped as particulate matters such as PM_{10} and $PM_{2.5}$, and ground-level pollutants such as ozone (O_3), carbon monoxide (CO), sulfur dioxide (SO_2), nitrogen oxides (NO), and nitrogen dioxide (NO_2).

It is a known fact that air pollution has negative impacts on human health, range of vision, materials, plants, and animal health. Air pollutants trigger or worsen chronic diseases such as asthma, pneumonia, heart attack, bronchitis, and other respiratory problems. Since particulate matters are very small and light, they tend to stay in the air longer than heavier particles. This increases the likelihood of humans and animals inhaling these particles through respiration. Due to their small size, these particles can easily pass through the nose and throat and penetrate the lungs, and some may even enter the circulatory system. So briefly, the high concentrations of air pollutants can harm human health, adversely influence the environment, and also cause property damage (Rich, 2017; Xing, Xu, Shi, & Lian, 2016; Mannucci & Franchini, 2017).

Due to the seriousness of the issue, air pollution control policies require systematic monitoring and evaluation of air quality. The causes of air pollution should be investigated and necessary precautions should be taken under the findings. Therefore, it is very important to develop an appropriate tool to understand the air quality in an area. For this purpose, effective methods are continuously developed with new studies.

In this context, this study is aimed at examining the air quality monitoring stations in Turkey according to their similarities in terms of five air pollutants PM_{10} , SO_2 , NO_2 , NO, and O_3 , and making appropriate inferences based on the analysis of the levels of air pollutants measured at these stations in the time interval of 1 November 2017 and 1 November 2018. In this way, city areas with similar air pollution behavior can be

identified so that decision-making authority can canalize emission sources to be located in the regions in need. In order to perform the experiments, a novel algorithm, named *majority-based multi-task clustering* (MV-MTC), is proposed, instead of applying traditional clustering algorithms, to benefit from the common decision coming from different pollutant sources. The novelty of this study is the implementation of multi-task clustering (MTC) in the field of environmental science and air pollution in Turkey is examined with this method for the first time.

The proposed MV-MTC algorithm has many advantages over existing multi-task clustering methods. First, some methods mainly have a complicated theoretical foundation, which leads to implementation difficulties. For instance, graph-based methods and matrix factorization for nonnegative data are commonly applied as in (Zhang et al., 2016) by implementing a semi-nonnegative matrix tri-factorization method to co-cluster the data in each view of each task. Likewise, the algorithm introduced in (Liu et al., 2016) has several sophisticated steps: feature extraction, clustering-based regularization, convex relaxation, and optimization. Spectral clustering that uses the eigenvectors of the Laplacian of a graph for clustering is another way to implement multi-task clustering (Siahpirani et al., 2016). In addition to graph-based methods, as in (Zhang et al., 2017), multi-task clustering can be performed by reweighting the distance between data points in different tasks by learning a shared subspace. In this way, clustering operation for each task is generated by selecting the nearest neighbors for each sample from the other tasks in the learned shared subspace.

Second, some proposed MTC methods (i.e. scVDMC (Zhang et al., 2018) and Arboretum-Hi-C (Siahpirani et al., 2016)) were designed as field-specific methods and have a valid use only for bioinformatics data to analyze the genome architecture or to simultaneously capture the differentially expressed genes. These methods are not suitable for the analysis of geographical data (or for the identification of air pollution levels of a region).

Third, the proposed algorithm is particularly advantageous since it does not need any a-prior information about the data. However, Yan et al. (2015) proposed a novel

algorithm, named Convex Multi-task Clustering (CMTC), which requires some prior knowledge about the data relationship.

Fourth, some multi-task clustering algorithms (i.e. Zhang et al., 2015) require additional parameters and the results change significantly with different parameter values. It makes it difficult to use the algorithm since the user should determine the optimal parameter for each problem. The proposed algorithm does not require any additional parameter tuning.

Fifth, the execution time of some multi-task clustering algorithms (i.e. Zhang et al., 2008) increases exponentially when the input data increases. However, the proposed algorithm (MV-MTC) requires computation time that grows linearly with the number of instances, clusters, and tasks.

Sixth, the proposed method can effectively avoid the imbalance of cluster distribution by merging multiple models according to majority voting. Besides, the MV-MTC framework can effectively reduce clustering errors by selecting the best clustering algorithm for the problem under consideration.

The goal is to propose an easily implemented, generally applicable, fast, prior knowledge, and parameter independent multi-task clustering method. Unlike existing methods, the proposed algorithm in this study is a new kind of multi-task clustering method which is much easier to understand and implement by taking the jointly obtained common decision from different tasks using cluster labels. It was developed as a new method that can appeal to every area rather than being specific to an area (Zhang et al., 2018; Siahpirani et al., 2016).

5.2.1 Dataset Description

There are seven geographical regions, namely Eastern Anatolia, Central Anatolia, Southeastern Anatolia, Blacksea, Mediterranean, Aegean, and Marmara, in Turkey and numerous air quality monitoring stations (AQMS) at each region. This study is

conducted on 49 AQMSs from 32 provinces which are from different regions. The features of each station are listed in Table 5.5 by showing the name of AQMS, in which city it is located, the corresponding county of the city, longitude and latitude information, network type (urban/rural/industrial), and which air pollutants are regularly measured in there.

The National Air Quality Monitoring Network of Turkey reached 330 Air Quality Monitoring Stations. The air quality of all provinces in this country is monitored. To facilitate public access to information on air quality, the monitoring results are published online at the website of <http://laboratuvar.cevre.gov.tr>. In all of the air pollution measurement stations, SO₂ and PM₁₀ parameters are measured, also, NO, NO₂, NO_x, CO, and O₃ are measured automatically in many of them. In this study, all of the AQMSs were investigated and 49 out of 330 stations were selected because the aforementioned air pollutants (PM₁₀, SO₂, NO₂, NO, and O₃) are regularly measured in these stations together.

Since the data becomes roughly periodic after one year, only one year's (November 2017 through November 2018) data is used in the experiments. The pollutant concentrations are mean values of daily (24 hours) measurements.

Table 5.5 The selected air quality monitoring stations (AQMS) and their features

ID	AQMS NAME	CITY	COUNTY	LONGITUDE	LATITUD	TYPE	THE MEASURED
1	Adana - Catalan	Adana	Yüreğir	35.2619	37.1864	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
2	Adana - Dogankent	Adana	Yüreğir	35.3491	36.8545	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
3	Adana - Meteoroloji	Adana	Karaisali	35.3440	37.0041	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
4	Adana - Valilik	Adana	Seyhan	35.3124	36.9991	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
5	Agri	Agri	Merkez	43.0396	39.7213	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
6	Agri - Dogubeyazit	Agri	Dogubeyazit	44.0835	39.5476	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
7	Agri - Patnos	Agri	Patnos	42.8530	39.2365	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
8	Ankara - Kecioren	Ankara	Kecioren	32.8628	39.9672	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO, PM _{2.5}
9	Ankara - Sıhhiye	Ankara	Cankaya	32.8594	39.9272	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO, PM _{2.5}
10	Ardahan	Ardahan	Merkez	42.7055	41.0000	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
11	Artvin	Artvin	Merkez	41.8182	41.1752	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
12	Bartın	Bartın	Merkez	32.3564	41.6248	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO, PM _{2.5}
13	Bayburt	Bayburt	Merkez	40.2255	40.2558	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
14	Canakkale - Biga İcdas	Canakkale	Biga	27.1072	40.4173	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
15	Canakkale - Can - MTHM	Canakkale	Can	27.0498	40.0293	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
16	Edirne - Kesan - MTHM	Edirne	Kesan	26.6352	40.8511	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5}
17	Erzincan	Erzincan	Merkez	39.4950	39.7430	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
18	Erzurum	Erzurum	Yakutiye	41.2728	39.8982	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
19	Erzurum - Palandoken	Erzurum	Palandoken	41.2752	39.8676	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
20	Erzurum - Pasinler	Erzurum	Pasinler	41.5721	40.0335	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
21	Giresun - Gemilercekegi	Giresun	Merkez	38.3985	40.9144	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO, PM _{2.5} , PM ₁₀ Flow Rate, PM _{2.5} Flow Rate
22	Gumushane	Gumushane	Merkez	39.4808	40.4608	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
23	Hatay - Iskenderun	Hatay	Iskenderun	36.2239	36.7141	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
24	Iğdir	Iğdir	Merkez	44.0536	39.9261	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
25	Iğdir - Aralık	Iğdir	Aralık	44.6209	39.7868	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5}
26	Istanbul - Basakşehir - MTHM	Istanbul	Basakşehir	28.7898	41.0954	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
27	Istanbul - Esenyurt - MTHM	Istanbul	Esenyurt	28.6688	41.0192	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
28	Karabük - Kardemir 1	Karabük	Merkez	32.6274	41.1920	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
29	Kars - İstasyon Mah.	Kars	Merkez	43.1044	40.6050	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO

Table 5.5 continues

ID	AQMS NAME	CITY	COUNTY	LONGITUDE	LATITUD	TYPE	THE MEASURED
30	Kirklareli - Limankoy-MTHM	Kirklareli	Limankoy	28.0559	41.8852	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
31	Kirsehir	Kirsehir	Merkez	34.1686	39.1381	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO, PM _{2.5}
32	Kocaeli - Gebze - MTHM	Kocaeli	Gebze	29.4365	40.8108	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
33	Kocaeli - Korfez - MTHM	Kocaeli	Korfez	29.7888	40.7461	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5} , PM _{2.5} Flow
34	Kocaeli - Yenikoy - MTHM	Kocaeli	Basiskele	29.8844	40.7042	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
35	Manisa - Soma	Manisa	Soma	27.6129	39.1814	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
36	Ordu - Unye	Ordu	Unye	37.2802	41.1214	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM ₁₀ Flow Rate
37	Rize	Rize	Merkez	40.5328	41.0217	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
38	Rize - Ardesen	Rize	Ardesen	41.0475	41.1273	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5}
39	Samsun - Atakum	Samsun	Atakum	36.2965	41.3253	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5} , PM ₁₀ Flow Rate, PM _{2.5} Flow Rate
40	Seyyar - 1 (06 THL 77) - Malatya Arapgir	Malatya	Arapgir	38.4878	39.0457	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃
41	Seyyar - 2 (06 THL 79) - Sincan OSB	Ankara	Mamak	33.0364	39.9008	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃
42	Seyyar - 4 (06 DV 9975) - Isparta	Isparta	Sarkikaraagac	31.3549	38.0442	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃
43	Tekirdag - Corlu - MTHM	Tekirdag	Corlu	27.8154	41.1806	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃
44	Trabzon - Akcaabat	Trabzon	Akcaabat	39.5923	41.0143	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO
45	Trabzon - Uzungol	Trabzon	Uzungol	40.2980	40.6173	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃
46	Trabzon - Valilik	Trabzon	Merkez	39.7123	41.0059	Urban	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x
47	Yalova - Armutlu-MTHM	Yalova	Armutlu	28.7845	40.5292	Rural	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , PM _{2.5}
48	Zonguldak - Eren Enerji Lise	Zonguldak	Catalagzi	31.8801	41.4964	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO, PM _{2.5}
49	Zonguldak - Eren Enerji Tepekoy	Zonguldak	Catalagzi	31.9374	41.5269	Industrial	PM ₁₀ , SO ₂ , NO ₂ , NO, O ₃ , NO _x , CO, PM _{2.5}

5.2.2 Experimental Study and Results

In this study, the proposed MTC method, MV-MTC, is compared with traditional clustering algorithms K-Means (KM), Expectation-Maximization (EM), Hierarchical Clustering (HIER), Canopy and Farthest First (FFIRST). Each task was clustered by the selected algorithm and then their decision from consensus was obtained in the MV-MTC framework. Performance evaluation is done via the sum of squared error (SSE) calculation. Before constructing the model, data were normalized and missing data imputation was performed using the mean values.

The number of clusters, k , was selected as 10% of the number of instances in the dataset therefore it was 5. Distance metric was chosen as Euclidean distance. To take the joint decision from each single clustering algorithm, each cluster was labeled according to the weights calculated as the average value of the instances of intra-cluster. According to this scheme, 5 cluster labels were determined as 'L1', 'L2', 'L3', 'L4' and 'L5'. Table 5.6 displays the average normalized weight of each cluster in terms of different air pollutants and their corresponding cluster labels. As a result of the joint decision of different tasks, where the evaluation of either of PM₁₀, SO₂, NO₂, NO, and O₃ pollutants were assumed as a new task, final cluster assignments were done.

To evaluate the performance of the applied methodology, values of the sum of squared error are calculated. SSE is the sum of the squared differences between each observation and its group's mean. According to (5.4), o_i represents an instance of dataset D , C_j represents the j^{th} cluster, m_j is the centroid value of the specified cluster j where o_i is assigned and k is the number of clusters. The total SSE of a method is the sum of all separate SSE calculations coming from distinct clusters.

$$\text{SSE} = \sum_{j=1}^k \sum_{o_i \in C_j} (o_i - m_j)^2 \quad (5.4)$$

Table 5.6 The mean weight values of each cluster group and their intra-cluster labels obtained by KM++

	Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5	
	Weight	Label	Weight	Label	Weight	Label	Weight	Label	Weight	Label
PM ₁₀	0.2002	L2	0.3142	L3	0.1309	L1	0.3982	L4	0.5977	L5
SO ₂	0.0713	L1	0.2328	L3	0.1497	L2	0.5799	L5	0.4996	L4
NO ₂	0.0902	L1	0.2177	L2	0.3322	L3	0.4396	L4	0.6593	L5
NO	0.0726	L1	0.1382	L2	0.5186	L5	0.1781	L3	0.1783	L4
O ₃	0.4208	L3	0.3237	L2	0.7482	L5	0.4873	L4	0.2082	L1

Final assignments are obtained both by MV-MTC and single clustering algorithms. Clustering algorithms are applied on every single task i.e. the model is formed just by taking one pollutant into consideration. The SSE results of different pollutants under different algorithms are shown as $C_{\text{pollutantName}}$ where "pollutantName" is one of the pollutants (PM₁₀, SO₂, NO₂, NO, or O₃) in Table 5.7. C_{ALL} is the average SSE value coming from all pollutants. KM is applied with two different versions in terms of the

initialization method used. KM with the random initialization is denoted as KM and KM initialized with K-Means++ is displayed as KM++. Hierarchical clustering is implemented with different link types among clusters. HIER_SING, HIER_COMP, HIER_AVG, HIER_MEAN, and HIER_CENTRO represent the hierarchical clustering types with single link, complete link, average link, mean link, and centroid link, respectively. The bold notations in Table 5.7 show the best results in the respective rows.

It can be concluded that the proposed MV-MTC method outperforms all single clustering algorithms that similar AQMSs are assigned to the same cluster group more accurately when multi-task clustering is applied. Besides, the most promising output of MV-MTC is obtained by KM++. In case of single clustering algorithms, EM performs the best among the other applied techniques.

Final cluster assignments after performing MV-MTC with KM++ are shown in Figure 5.3. It points out the geographical locations of AQMSs in the map of Turkey with different colored markers where each color represents a cluster.

Table 5.7 The results of the performance evaluation in terms of SSE values

	KM	KM++	EM	Canopy	FFirst	HIER_SING	HIER_COMP	HIER_AVG	HIER_MEAN	HIER_CENTRO
C _{PM10}	396.40	406.53	392.42	476.39	415.03	532.51	411.91	516.16	456.02	526.03
C _{SO2}	260.31	194.96	226.03	415.04	214.27	216.13	204.56	214.27	214.27	214.27
C _{NO2}	367.95	315.79	319.25	406.52	356.39	437.90	317.32	437.90	373.54	437.90
C _{NO}	335.17	351.29	294.79	426.31	305.73	319.82	303.40	303.40	319.82	319.82
C _{O3}	393.83	394.42	371.31	478.08	418.25	645.29	402.39	410.27	514.38	446.52
C _{ALL}	350.73	332.60	320.76	440.47	341.93	430.33	327.92	376.40	375.61	388.91
MV-MTC	113.80	108.12	116.16	122.84	134.41	161.96	124.92	161.96	144.16	161.96



Figure 5.3 Results from the implementation of multi-task clustering obtained by KM++

In the MV-MTC approach, a clustering algorithm is performed for t tasks, and a merging operation is done in the final step (“majority voting”). In this study, the best results were obtained using the KMeans++ algorithm. The time complexity of K-Means++ is $O(n*k + n*k*I)$, where n is the number of instances, k is the number of clusters, and I is the number of iterations needed for convergence (Wu et al., 2017). After the single-task clustering step, the merging process takes the runtime cost of $O(t*n)$, where t is the number of tasks. Considering this, the total time complexity of the MV-MTC algorithm is $O(n*k + n*k*I + t*n)$. This time complexity indicates that the proposed MV-MTC algorithm requires computation time that grows linearly with the number of instances, clusters, and tasks. So, the execution time of the algorithm will be still reasonable even if we process a large volume of data.

Table 5.8 shows the execution time (in seconds) to perform the MV-MTC algorithm in terms of different clustering methods. Single task clustering results are also shown as $C_{PollutantName}$, and C_{ALL} represents the sum of the running time of all single task clustering results. Experiments were performed on a desktop computer with Intel Core i7-6700 3.40 GHz processor and 8 GB memory. In each experiment, the algorithms were executed 10 times, and then the average values were reported. The empirical results show that the running time of the proposed K-means++ algorithm under the MV-MTC framework is better than EM and hierarchical clustering algorithms.

Besides, The MV-MTC algorithm has comparable speed with the traditional clustering algorithms when we compare C_{ALL} and MV-MTC results on the datasets.

Table 5.8 Comparisons of different clustering methods in terms of execution time (in seconds)

	KM	KM++	EM	Canopy	FFirst	HIER_SING	HIER_COMP	HIER_AVG	HIER_MEAN	HIER_CENTRO
C_{PM10}	0.04	0.05	0.32	0.03	0.03	0.11	0.11	0.11	0.11	0.16
C_{SO2}	0.04	0.03	0.36	0.04	0.03	0.11	0.11	0.11	0.11	0.17
C_{NO2}	0.03	0.04	0.29	0.03	0.03	0.11	0.11	0.11	0.11	0.16
C_{NO}	0.04	0.04	0.26	0.03	0.03	0.11	0.11	0.11	0.11	0.17
C_{O3}	0.05	0.04	0.30	0.03	0.03	0.11	0.11	0.11	0.12	0.16
C_{ALL}	0.20	0.20	1.53	0.16	0.15	0.55	0.55	0.55	0.56	0.82
MV-MTC	0.31	0.60	1.55	0.33	0.29	0.73	0.73	0.73	0.76	1.05

MV-MTC algorithm was compared with one of the recently proposed MTC methods, MTCMRL, in terms of time complexity (Zhang et al., 2008). Multi-task clustering was combined with the model relation learning (MTCMRL) method to automatically learn the model parameter relatedness between each pair of tasks by providing a solution to a non-convex optimization problem. Even though the algorithm has proposed a better clustering performance compared to other multi-task clustering methods, it still does not offer the expected performance in terms of time complexity, which is $O(n^2*m)$, where m is the number of features and n is the number of instances per task, so it increases exponentially when n is increased to larger volumes. On the other hand, MV-MTC is still reasonable to be preferred because of its linearly changing time complexity.

5.2.3 Discussions

With this study, it is aimed to identify similar regions in terms of air quality. It enables flexible decision-making at the cluster level. So decision-makers on the control of air quality can take actions similarly for the members of the same cluster. Since many air quality monitoring station data is summarized in several clusters, it provides richer but compacted information for control and modeling. It finds structure in air quality data and is therefore exploratory in nature. Representing the whole

environmental data by a few clusters may offer the great advantage of simplification in analyzing the data. Identification of the monitoring station groups can be used to understand why these stations in the same cluster are similar. Clustering monitoring stations minimize the overload of information. Grouping similar information and summarizing common characteristics help environmental scientists understand the current situation more clearly. In addition, it is also possible to classify a new station by assigning it to the cluster with the closest center. The potential contributions of this study to the prediction of air quality can be listed as follows:

- The multi-task clustering can be also used to label all the observed elements before air quality prediction, by calculating the distance between each centroid and each element in the data, and then selecting the cluster label (or level) with minimum distance.
- Multi-task clustering can also be used as a preprocessing step to improve the speed and performance of the classification algorithm that is used to predict the air quality index.
- In the application to predict air quality index, temporal data clustering results can give information about air quality variations, so that a set of forecasting systems, which are dedicated to reflecting temporal changes, can be formed.
- The identification of the air pollution levels of the different regions by clustering can be useful to design an air quality monitoring network structure. Such networks must consider the monitoring location, sampling frequencies, and the pollutants concern. For instance, clustering results lead to design of an optimal network, i.e. a network providing maximum data with minimum measurement devices. The spatial relationship analysis is used to compare the information given by the potential sites that may form the network.
- On forecasting the level of air pollution, it is possible to find the closest cluster of a new instance to be predicted, and then use the values in this cluster for prediction.
- Multi-task clustering can also be useful for detecting extreme air pollution events and can help predict future exceedances. In this sense, an air pollutant value of a region may be considered an outlier if it exceeds the minimum and maximum value of the cluster it belongs to.

CHAPTER SIX

ASSOCIATION RULE MINING STUDIES IN ENVIRONMENTAL INFORMATICS

Through the agency of IoT (Internet of Things) sensors in air quality monitoring stations, different pollutant concentrations and meteorological parameters can be regularly measured. In case of the availability of an unusual condition such as some amounts of exceedance from normal pollutant levels, a smart technological system can give a warning and the appropriate air quality management strategy is automatically performed so that intelligent cities are developed. In this direction, the objective of this study is to discover important relationships and significant patterns among air pollution features. Determination of the frequently observed association rules triggers the background smart environment system in case of the detection of a critical situation. In the experimental studies, traditional Association Rule Mining (ARM) and Weighted Association Rule Mining (WARM) methods are compared on the real-world dataset that was collected from 21 monitoring stations in Turkey.

6.1 Material and Methods

In this section, traditional association rule mining, weighted association rule mining, and the proposed methodology throughout this study will be introduced in detail as well as the performance measures such as support, confidence, or lift metrics used in the evaluation of the models.

6.1.1 Association Rule Mining

Identification of interesting patterns to obtain useful knowledge is the basic purpose of data mining. By analyzing the data, the hidden structure is inferred from the problem domain so that predictive models can be generated by classification or clustering tasks to take critical decisions for future problems. Besides, another main task of data mining called association rule mining (ARM) allows us to discover important relationships among data samples by using association rules.

ARM models consider several transactions, $T = \{T_1, T_2, \dots, T_n\}$, each of which contains a set of items from the itemset, I , as shown in $I = \{i_1, i_2, \dots, i_m\}$, where n is the number of transactions and m is the number of items. The question is: “What does the value of one feature tell us about the value of another feature?” in this scenario. The basic purpose is the determination of the significant relationships among these items i.e. finding a set of items that frequently appear together by considering whole transactions. ARM methods are broadly utilized in many different application areas such as market basket analysis, discovering patterns in biological databases, fraud detection in credit card transactions, etc. Some of the well-known methods introduced under ARM are Apriori, FP-Growth, and Eclat.

To illustrate, in the following example in Table 6.1, there are seven items in the itemset $I = \{A, B, C, D, E, F, G\}$ and five transactions T_i 's which include items from I . The aim is to extract all association rules using these recorded transactions in the past and select the most important ones among them. An association rule is the combination of two concepts which are antecedent and consequent, both of which are a list of items. As an example, $\{X\} \rightarrow \{Y\}$ is an association rule where both X and Y are itemsets and X is known as *antecedent* and Y is known as *consequent*. This rule implies the transactions which contain itemset X that also contain itemset Y .

Various metrics are considered in the evaluation process to understand the strength of associations among each item. *Support* is one of these metrics which tells us the fraction of the total number of transactions where a specific itemset occurs. It is calculated as in (6.1):

$$\text{Support}(\{X\} \rightarrow \{Y\}) = \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Total number of transactions}} \quad (6.1)$$

In the given scenario in Table 6.1, $\text{Support}(\{A\} \rightarrow \{B\}) = 2 / 5 = 0.4$. In the same way, support the values of all other association rules can be calculated. Association rules with higher support values than the given support value by the user are called *frequent* and they are worth considering for further analysis because they provide enough information on the relationships between their items.

Table 6.1 An example itemset and transactions

Original Itemset (I)	A, B, C, D, E
Transaction T_1	A, C, E
Transaction T_2	A, B, D, E
Transaction T_3	B, E
Transaction T_4	A, B, D, E
Transaction T_5	E

Another important evaluation metric is known as *confidence*. Confidence can be defined as the likeness of occurrence of consequent on the cart given that the cart already has the items in the antecedents. By using the formula in (6.2), $\text{Confidence}(\{A\} \rightarrow \{B\}) = 2 / 3 = 0.667$ according to the scenario in Table 6.1. Association rules which satisfy both minimum support and minimum confidence thresholds are called *strong*.

$$\text{Confidence}(\{X\} \rightarrow \{Y\}) = \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Transactions containing } X} \quad (6.2)$$

The general problem of association rule mining can be reduced into a two-step process: *finding all frequent itemsets* using minimum support threshold and *detection of strong association rules from these frequent itemsets* using minimum confidence threshold (Han et. al., 2011).

The other three essential measures are *lift*, *leverage*, and *conviction* given in (6.3), (6.4), and (6.5), respectively. *Lift* shows the interestingness of a rule in other words how many times more often antecedent and consequent occur together than expected if they were statistically independent. When calculating the *conviction*, the probability that the antecedent appears without the consequent is calculated; if the antecedent and consequent are dependent, the probability that the antecedent is seen independently of the consequent is calculated. We can think of the *leverage* as the difference between the probability of the set which includes both antecedent and the consequent and the sum of their individual probabilities.

$$\text{Lift}(\{X\} \rightarrow \{Y\}) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X) * \text{Support}(Y)} \quad (6.3)$$

$$\text{Leverage}(\{X\} \rightarrow \{Y\}) = \text{Support}(X \cup Y) - \text{Support}(X) * \text{Support}(Y) \quad (6.4)$$

$$\text{Conviction}(\{X\} \rightarrow \{Y\}) = \frac{1 - \text{Support}(Y)}{1 - \text{Confidence}(\{X\} \rightarrow \{Y\})} \quad (6.5)$$

6.1.1.1 Weighted Association Rule Mining

One drawback of the traditional ARM algorithms is that each item is assumed to have the same significance even though some of them have greater importance compared to others within a transaction or the whole item space. Therefore, the setting of the traditional ARM models should be organized in a weighted manner according to importance levels. In this respect, Tao, Murtagh, & Farid (2003) developed an improved version of ARM, which is called Weighted Association Rule Mining (WARM). In this study, because different air pollutants/meteorological factors have different effects on air pollution, the rules are generated using WARM by taking account of item weights.

Association rule mining (ARM) methods have been applied in many studies to find high-quality relationships among the attributes of a problem domain. Weighted association rule mining (WARM) which is the sub-branch of ARM has also been used in many research areas. Some of these studies will be explained in this section.

Firstly, Tao et. al. (2003) proposed the weighted scheme of association rule mining in discovering significant relationships among the items of transactions. In traditional association rule mining applications, all items were given the same weights even though their importance within the transactions or the item spaces was different. In their presented algorithm, each item should have its weight according to its role in the given scenario and also different item sets or transactions may have specific itemset/transaction weights. Besides, the downward closure property, which is a rule of ARM that if an itemset is large, then all of its subsets are also large, is invalidated by this method. Weighted downward closure property is taken into account by using weighted support in the process of finding significant item sets.

To construct a personalization recommendation system based on user focus, Zhang, Chen, Wang, & Wang (2004) developed an algorithm of text information retrieval which considers a user's entire query requests at a period. The algorithm was implemented using weighted association rule mining with FP-Growth (frequent pattern growth). Their proposed method increased the efficiency in the subject of the time requirement.

Wang & Zhang (2003) combined fuzzy set theory with weighted association rule mining by assigning weights to the relative attributes having a different significance. Kaya & Alhajj (2004) expanded the research by mixing fuzziness, weighted association rules, and genetic algorithms. Fuzzy quantitative association rules were determined by employing a clustering method based on a genetic algorithm, which dynamically adjusts and optimizes membership functions.

Li (2005) implemented another work of weighted association rule mining using fuzzy set theory to obtain user-guided association rules to analyze data of application loans including 650 instances of applicants for appliance loans. Their method resulted in flexible, natural, and understandable rules in the final stage.

As in the work of Yun & Leggett (2005) used weighted association rule mining with the FP-Growth algorithm in which a minimum weight restraint and a weight range were specified and items were supplied with different weights.

Sun & Bai (2008) presented a new method that is different from other classical WARM models that it did not require the pre-assigned item weights. A support metric was defined for the itemsets in the database which only contains binary attributes to find interesting patterns. The weight of each transaction was evaluated by the HITS (Hyperlink-Induced Topic Search) algorithm. Based on the computed weights, the new support was defined to give the significance of itemsets. Their model outperformed the link-based methods.

Dhanalakshmi & Porkodi (2018) applied their proposed WARM model to analyze the Tobacco smoke effect on the placenta dataset. After the calculation of weighted support, they separately implemented one existing (based on FP-Growth) and two novel ARM techniques to obtain association rules. In the final step, association rules were filtered depending on the weighted support and significant gene ontology terms. Their proposed WARM model provided more optimal results compared to the existing one.

Another application area was the analysis of the performance of a cricket team using a WARM method (Saraswat, Dev, & Singh, 2018). The resulting output was aimed at being used in the planning of game-winning. The important relationships among attributes such as the effect of toss, venue (home/away condition), batting first or second and the game result (win/loss) were detected using weighted support and weighted confidence.

Qiu et al. (2019) introduced their study to automatically identify the building operational strategies (on/off, sequencing, and coordinated control). The experimental results were validated by a power metering system data. They applied two data mining methods: decision tree (CART) to determine on/off and sequencing control strategies and WARM to specify coordinated control strategy. Their proposed framework worked accurately.

Table 6.2 presents some recent studies which use WARM models. The bold notation in the “Methods” column indicates the algorithm which outperforms the others. It is apparent that WARM methods provide good performance in different areas of expertise such as bioinformatics (Lakshmi & Vadivu, 2019; Indhumathy, Nabhan, & Arumugam, 2018), recommendation systems (Cagliero, Garza, Pasini, & Baralis, 2018; Bhavithra & Saradha, 2018), and activity recognition (Hui, Chen, & Zhan, 2019; Wen, Zhong, & Wang, 2015).

Table 6.2 Recent studies taking WARM methods as the main methodology

Ref.	Aim	Data	Methods
Lakshmi & Vadivu, 2019	Disease comorbidity prediction	Protein-protein interaction data, Pathway data and Gene Ontology annotation data	PCID and the proposed model: WARBCP
Hui et al., 2019	Classification of video scenes by discovering the relationship between action semantic and scene	Daily human activities data	The proposed model: WTARM , SVM, ARM, and TARM
Rahimi, Noferesti, & Shamsfard, 2019	Identification of sentiment shifters in reviews	A medical drug corpus and a corpus including reviews of products, restaurants and movies	SentiWordNet, Domain Specific Lexicon, ML-unigram, Proposed-DM-Dependency, Proposed-DM-SRL, Proposed-ML and Proposed-ML + Shifter
Cagliero et al., 2018	Recommendation of additional reviewers for papers with critical assignments	Conference and journal datasets	WAR
Bhavithra & Saradha, 2018	Personalized web page recommendation	Web query log data	K-NN, CF, CBR and the proposed model: CBR with WARM
Indhumathy et al., 2018	Prediction of potentially new HCV-Human protein interactions	HCV-human protein interactions data (HCVpro)	The proposed model: WARL
He, Zhu, He, Gu, & Cui, 2017	Root cause identification to find the hidden cause-related relationships	Big data of product infant failure	ARM and WARM
Wang, Li, Huang, & Su, 2016	Discovering correlated alarm sequences	Quantitative alarm sequences data	Fuzzy weighted alarm association rules mining
Wen et al., 2015	Finding the frequent patterns of human activities	The sensor readings data from smart environments	Naïve Bayesian, Decision Tree, HMM and the proposed weighted frequent patterns mining model
Cao, Ma, & Cao, 2014	Software reliability qualitative evaluation	Databases about unpredictable cases in the software development process and software reliability	Delphi Hierarchy Process Based on Cloud Model and WARM

PCID: predicting disease comorbidity based on integration of multi-scale data, WARBCP: weighted association rule-based disease comorbidity prediction, WTARM: weighted temporal apriori rule mining, TARM: temporal apriori rule mining, DM: data mining, SRL: semantic role labeling, CF: collaborative filtering, WARL: weighted association rule mining, WAR: weighted association rules, CBR: case-based reasoning, HMM: hidden Markov model

Different from traditional association rule mining methods, in the weighted association rule mining framework, all items are assigned predefined weights that represent the significance of items within a transaction. The general framework is implemented as below:

1. Assign all item weights as $W = \{ w_1, w_2, \dots, w_m \}$, where m is the number of items
2. Normalize all item weights as $W' = \{ w'_1, w'_2, \dots, w'_m \}$
3. Calculate transaction weights as $W_t = \{ w_{T_1}, w_{T_2}, \dots, w_{T_n} \}$ by using normalized item weights, W'
4. Normalize transaction weights as $W'_t = \{ w_{T_1}', w_{T_2}', \dots, w_{T_n}' \}$, where n is the number of transactions in the dataset
5. Calculate weighted support values of each itemset
6. Eliminate redundant itemsets which are under the predetermined support value and put the remaining ones into frequent itemsets
7. Discover the strong association rules which are above the predefined confidence threshold

According to the given scenario in Table 6.1, the first thing is to assign corresponding weights to the items of I . Then, all weights are normalized for later processing. Table 6.3 (a) displays the initially given weights by the user and their normalized versions. After the calculation of the normalized item weights, the next step is to find the weights of each transaction T_i as w_{T_i} according to (6.6), where $i = \{1, 2, \dots, n\}$ and n is the number of transactions. The relative importance of each transaction is determined in this way. As in the calculation of item weights, transaction weights are also normalized as shown in Table 6.3 (b).

$$\text{Weight}(T_i) = w_{T_i} = \frac{\sum_{j=1}^{|T_i|} w_j}{|T_i|} \quad (6.6)$$

The next, perhaps the most essential, step is to find all frequent itemsets which occur together very often. In the beginning, the user should introduce the weighted support threshold to control the selection of itemsets. Here, the weighted support is given as 0.3. All itemsets from Level 1 to Level 5 are evaluated that the equal ones or

above 0.3 are marked as *frequent* and they are shown as bold and gray shaded in Table 6.3 (c).

Table 6.3 An example scenario using warm to find frequent itemsets

Items	Weight (W)	Normalized Weight (W')
A	1	0.04
B	5	0.20
C	4	0.16
D	5	0.20
E	10	0.40
Total	25	1.00

(a)

Transactions	Transaction Weight (W_T)	Normalized Transaction Weight (W_T')
T_1 A, C, E	0.200	0.172
T_2 A, B, D, E	0.210	0.181
T_3 B, E	0.300	0.258
T_4 A, B, C, D, E	0.200	0.172
T_5 B, C, E	0.253	0.217
Total	1.163	1.00

(b)

Level 1	A	B	C	D	E					
Weighted Support	0.525	0.828	0.561	0.353	1.0					
Count	3	4	3	2	5					
Level 2	A, B	A, C	A, D	A, E	B, C	B, D	B, E	C, D	C, E	D, E
Weighted Support	0.353	0.344	0.353	0.525	0.389	0.353	0.828	0.172	0.561	0.353
Count	2	2	2	3	2	2	4	1	3	2
Level 3	A, B, C	A, B, D	A, B, E	A, C, D	A, C, E	A, D, E	B, C, D	B, C, E	B, D, E	C, D, E
Weighted Support	0.172	0.353	0.353	0.172	0.344	0.353	0.172	0.389	0.353	0.172
Count	1	2	2	1	2	2	1	2	2	1
Level 4	A, B, C, D	A, B, C, E	A, B, D, E	A, C, D, E	B, C, D, E					
Weighted Support	0.172	0.172	0.353	0.172	0.172					
Count	1	1	2	1	1					
Level 5	A, B, C, D, E									
Weighted Support	0.172									
Count	1									

(c)

In ARM methods, if an itemset is found as frequent, then its subsets are also assumed as frequent. This assumption is called *the downward closure property*. Here, an important difference from traditional ARM is attracted attention that downward closure property does not have to hold anymore. For instance, if the itemset $\{A, B, C\}$ is classified as frequent, then $\{A, B\}$ may not be included in frequent itemsets because the probability of the selection of $\{A, B, C\}$ as a frequent itemset can arise from the high weight of item C.

After finding all frequent itemsets, strong association rules are determined using the initially given weighted confidence threshold. According to the scenario in Table 6.3 (c), for example, $\{A, E\}$ is a frequent itemset. Two association rules can be produced from $\{A, E\}$ as $A \rightarrow E$ and $E \rightarrow A$. Weighted confidence values of these two rules do not have to be the same according to (6.4), because they express a different sequence of itemsets. While Weighted Confidence ($A \rightarrow E$) = Weighted Support ($\{A, E\}$) / Weighted Support ($\{A\}$) = 1, on the other hand, Weighted Confidence ($E \rightarrow A$) = Weighted Support ($\{A, E\}$) / Weighted Support ($\{E\}$) = 0.525. In the scenario of Table 6.3, the user-defined weighted confidence threshold is 0.6. According to that, while $A \rightarrow E$ is a strong association rule, $E \rightarrow A$ is not.

$$\text{Weighted Confidence}(\{X\} \rightarrow \{Y\}) = \frac{\text{Weighted Support}(X, Y)}{\text{Weighted Support}(X)} \quad (6.7)$$

Using (6.7), all association rules are evaluated and the ones providing the required weighted confidence value are chosen as strong association rules in the final step.

6.1.2 The Proposed Method: Weighted Association Rule Mining (WARM)

Various WARM models with different properties have been proposed. Wang, Yang, & Yu (2000) introduced an algorithm, called WAR (Weighted Association Rules), which consists of two main parts. In the first part, the frequent itemsets are generated through traditional ARM algorithms by ignoring the weights; then post-processing is applied to the frequent itemsets during rule-generation by introducing

the intervals of item weight. WUARM (Weighted Utility ARM) developed by Khan, Muyeba, & Coenen (2008) considered both the significance of items and their utility as the frequency of occurrences in transactions. Some of the WARM models were implemented using frequent pattern growth algorithms such as WIP (Weighted Interesting Pattern mining) and WFIM (Weighted Frequent Itemset Mining) (Yun & Leggett, 2005, 2006). While WFIM was applied by creating an ascending weight ordered prefix tree, WIP used a new concept of a weighted hyperclique pattern which considered weight confidence to discover frequent patterns with similar levels of weights and the h-confidence to determine affinity patterns with strong support. The proposed model used in this chapter was implemented using the Apriori approach because pattern growth algorithms take huge memory while storing in a tree structure, which may not fit in memory when data size is very large.

Some studies have focused on fuzzy set theory combined with WARM to mine weighted boolean and quantitative data (Wang & Zhang, 2003; Yue, Tsang, Yeung, & Shi, 2000; Lu & Qian, 1999; Gyenesei, 2000; Muyeba, Khan, & Coenen, 2008). While FWARM (Fuzzy Weighted Association Rule Mining) proposed by Muyeba et al. (2008) resulted in a valid downward closure property, others could not guarantee the validation. However, fuzzy-based algorithms have several disadvantages such as requiring knowledge about the problem solved and difficulties in setting the rules and choosing membership functions. The presented algorithm differs from many others reported in the literature in terms of easy implementation and application.

In this study, an intelligent environmental system is introduced by using a WARM methodology. Figure 6.1 displays the general framework of the proposed system. The concentrations of air pollutants and some meteorological parameters from three different types of monitoring stations (urban, rural, and industrial) are daily recorded in data storage/servers and Cloud. Each log data is treated as a transaction in the WARM scenario. The data management team is in the charge of analyzing and processing all recorded data to obtain useful information related to air quality factors. For this purpose, the framework is implemented using the WARM methodology by the data management team. Differently from the example in Table 6.3, there is more

than one value of each item showing the degree of the amount as $item_i = \{ item_{i_1}, item_{i_2}, \dots, item_{i_k} \}$. For example, the concentration of each pollutant has six different degrees such as “Good, Moderate, Sensitive, Unhealthy, Bad, Dangerous”. Instead of only giving weights to items themselves, the proposed system assigns weights to item values as in Table 6.4 to correctly apply the methodology. The remaining parts of the original WARM method are applied in the same way by calculating transaction weights using the weights of item values. After that, frequent itemsets and strong association rules are found as in the traditional WARM method.

Table 6.4 Weight assignment to item values

PM ₁₀		SO ₂		NO ₂		O ₃	
Good	w ₁	Good	w ₇	Good	w ₁₃	Good	w ₁₉
Moderate	w ₂	Moderate	w ₈	Moderate	w ₁₄	Moderate	w ₂₀
Sensitive	w ₃	Sensitive	w ₉	Sensitive	w ₁₅	Sensitive	w ₂₁
Unhealthy	w ₄	Unhealthy	w ₁₀	Unhealthy	w ₁₆	Unhealthy	w ₂₂
Bad	w ₅	Bad	w ₁₁	Bad	w ₁₇	Bad	w ₂₃
Dangerous	w ₆	Dangerous	w ₁₂	Dangerous	w ₁₈	Dangerous	w ₂₄

(a)

TEMP		HUM		PRES		STATION	
Low	w ₂₅	Low	w ₂₈	Low	w ₃₁	Urban	w ₃₄
Moderate	w ₂₆	Moderate	w ₂₉	Moderate	w ₃₂	Industrial	w ₃₅
High	w ₂₇	High	w ₃₀	High	w ₃₃	Rural	w ₃₆

(b)

In the following example, according to the given scenario, three strong rules were identified. For example, in Rule 1, when the concentration of SO₂ is moderate and the concentration of NO₂ is high besides there is a very high temperature, then it is likely to observe the station type as urban with 40% of support and 75% of confidence. Other rules are also interpreted in the same way.

- Rule 1:

SO₂=moderate NO₂=sensitive Temp=low → Station=urban (40% support, 75% confidence)

- **Rule 2:**
 $PM_{10}=\text{good } O_3=\text{moderate } SO_2=\text{good } NO_2=\text{good} \rightarrow \text{Station}=\text{rural}$ (25% support, 85% confidence)
- **Rule 3:**
 $PM_{10}=\text{moderate } SO_2=\text{sensitive } Pres=\text{high} \rightarrow \text{Station}=\text{industrial}$ (30% support, 80% confidence)

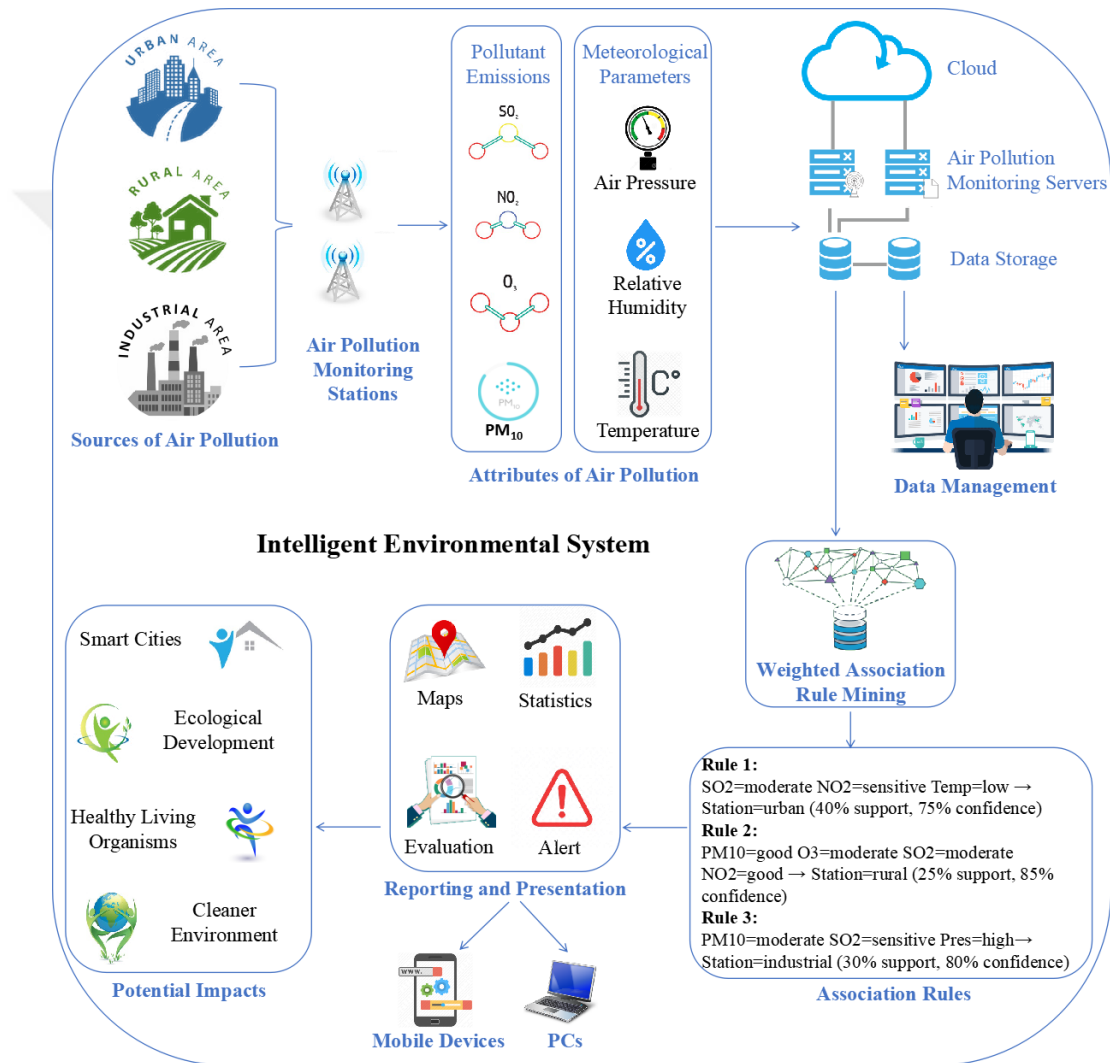


Figure 6.1 The general framework with potential impacts of the proposed intelligent environmental system

After the exploration of all significant association rules, the phase of reporting and presentation takes place to express the experimental results of the applied methodology

in a user-friendly manner. The system allows accessing the results using both mobile devices and personal computers. According to the outcomes, the system puts the air quality management team on alert in case of the detection of an unexpected situation to take the required precautions.

The general advantages of the proposed system are as follows:

- Smart cities are designed to monitor changing air conditions and provide appropriate warning systems.
- Ecological development is sustained and reinforced.
- Living things in the fresh air environment continue to live in a healthier and prosperous environment.
- A cleaner environment is obtained by inspecting the air quality in a regular and controlled manner.

6.2 Case Study - Air Pollution Monitoring in Smart Cities using Weighted Association Rule Mining

With the developing technology in recent years, various types of electronic data are collected via sensors, and then assets and resources are managed more efficiently thanks to intelligent systems that analyze the collected data. Environmental science is one of the areas that use intelligent systems to produce a solution for environmental problems and to protect the environment against factors such as pollution, climate change, and environmental degradation. In this context, observing air pollution data and analyzing and processing them through intelligent systems are among the useful studies of this field.

In this study, it is aimed to design an intelligent environment system that provides air quality control in cities by processing air pollution data collected from IoT sensors in monitoring stations. For this purpose, the association rule mining (ARM) method, which is one of the data mining tasks, is applied. According to the obtained rules from different combinations of the specified features of the dataset, normal conditions shall

be determined in terms of air quality and an intelligent system shall be established to detect the critical situations.

The novelty of this study is the implementation of WARM in the field of environmental science for the first time. This research is also original in that it assesses air pollution parameters in Turkey with this method. In the experimental studies, traditional ARM and WARM methods were compared on the real-world dataset that was collected from 21 monitoring stations in Turkey between the dates of 1 January 2018 and 1 May 2019.

The potential contributions can be listed as follows:

- Important relationships among different pollutants (PM_{10} , SO_2 , NO_2 , and O_3), meteorological parameters (temperature, relative humidity, and air pressure), and monitoring stations (urban, rural, and industrial) are identified.
- It is apparent that as in the subject of air pollution, the other environmental criteria such as water pollution parameters or soil contamination factors can also be evaluated, likewise.
- The applicability of WARM in an environmental problem is shown and its advantages compared to traditional ARM methods are proven.
- Smart cities with a system that helps manage air quality control can be developed.

6.2.1 Dataset Description

The dataset is the same as the one explained in Chapter 3. All attributes are categorized such that daily concentrations of pollutants are labeled as “Good, Moderate, Sensitive, Unhealthy, Bad, Dangerous” in terms of the level of air pollution and meteorological factors are also labeled in the same manner. Table 6.5 shows the sample dataset of this study.

The proposed model was applied in such a way that a number of transactions with the itemsets, which are full of air pollution data, are conducted to infer association

rules and observe useful relations. The itemset of the mentioned scenario is given in Figure 6.2.

Table 6.5 Sample dataset

Season	PM ₁₀	SO ₂	NO ₂	O ₃	TEMP	HUM	PRES	CLASS
Autumn	Moderate	Good	Good	Good	Moderate	Moderate	Medium	Urban
Summer	Good	Good	Good	Good	High	Moderate	High	Urban
Summer	Moderate	Good	Good	Good	Low	Low	High	Urban
Winter	Good	Sensitive	Unhealthy	Good	High	High	High	Industrial
Spring	Good	Good	Good	Good	High	Moderate	High	Rural

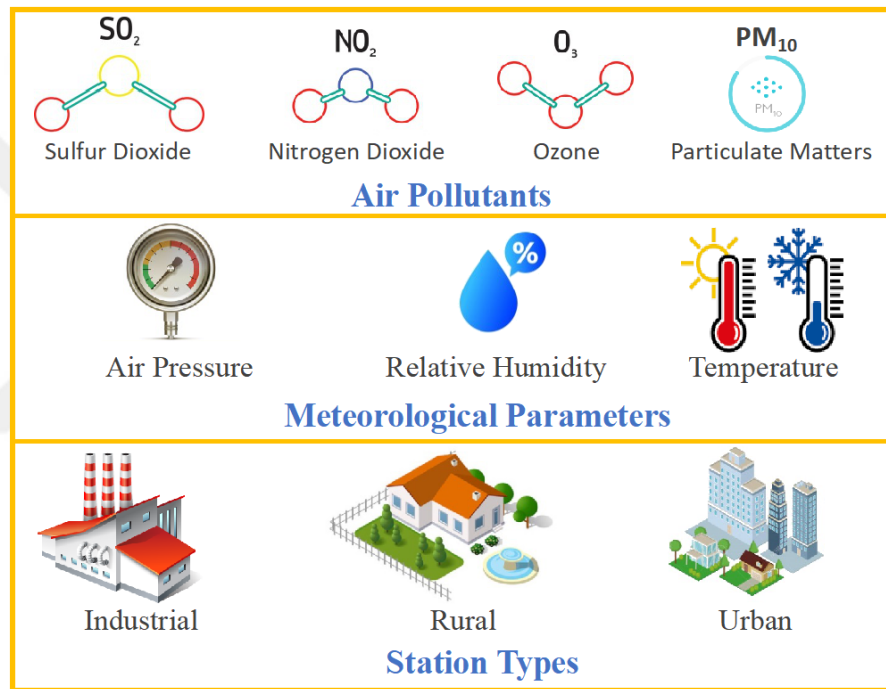


Figure 6.2 Itemset used in the transactions of air pollution dataset

6.2.2 Experimental Study and Results

Two main methodologies were implemented in the experimental studies: traditional ARM model based on Apriori and WARM. The predefined confidence threshold was 0.6 and the support threshold was ranged from 0.2 to 0.9 to show the change of the number of k-itemsets for both of the applied models. The WARM model was implemented in Visual Studio 2019 and the results of the Apriori model were obtained using the WEKA application (Holmes, Donkin, & Witten, 1994).

The first thing before applying ARM models was to discretize all numeric values into their respective intervals. Therefore, the daily concentration values of all air pollutant factors were categorized into the corresponding index categories “Good, Moderate, Sensitive, Unhealthy, Bad, Dangerous” using National Air Quality Index Breakpoints defined at the website of <http://laboratuvar.cevre.gov.tr> as shown in Table 6.6.

Meteorological parameters were also categorized as in Table 6.7 (b) according to the equal-sized intervals between their minimum and maximum values displayed in Table 6.7 (a) where Temp, Hum, and Pres are the abbreviations of Temperature, Relative Humidity, and Air Pressure, respectively.

The season when the measurement was performed were also added separately according to the month of the date. The amount of some air pollutant factors rises in some seasons. For example, the concentration of air pollutants increases in a parallel way as more fuel is burned due to the need for heating in winter. With this study, the relationship between season and pollutant concentrations can also be observed.

Table 6.6 National air quality index breakpoints for the air pollutants (SO₂, NO₂, O₃ and PM₁₀)

Index	SO ₂ [µg/m ³]	NO ₂ [µg/m ³]	O ₃ [µg/m ³]	PM ₁₀ [µg/m ³]
Good	0-100	0-100	0-120	0-50
Moderate	101-250	101-200	121-160	51-100
Sensitive	251-500	201-500	161-180	101-260
Unhealthy	501-850	501-1000	181-240	261-400
Bad	851-1100	1001-2000	241-700	401-520
Dangerous	>1101	>2001	>701	>521

Table 6.7 Breakpoints determined for the meteorological parameters

minTemp [°C]	maxTemp [°C]	minHum [%]	maxHum [%]	minPres [mbar]	maxPres [mbar]
-31.44	34.23	0	100.95	713.6	1051.38

(a)

Index	Temperature Breakpoints	Relative Humidity Breakpoints	Air Pressure Breakpoints
Low	< -10	< 35	< 827
Moderate	-10 - 10	35 - 70	827 - 940
High	> 10	> 70	> 940

(b)

Several example strong association rules found using WARM with support threshold = 0.2 was found by displaying their confidence, lift, leverage and conviction values as below:

Examples of rules with 2 itemsets:

- *Rule1:*
SEASON=Spring → SO2=Good (Confidence: 0.907, Lift: 0.9704, Leverage: - 0.0087, Conviction: 0.703)
- *Rule2:*
PM10=Good → TEMP=High (Confidence: 0.6252, Lift: 1.0514, Leverage: 0.0205, Conviction: 1.0815)
- *Rule3:*
STATION=Urban → SO2=Good (Confidence: 0.9135, Lift: 0.9774, Leverage: - 0.0111, Conviction: 0.7563)
- *Rule4:*
NO2=Good → PM10=Good (Confidence: 0.6716, Lift: 1.0009, Leverage: 0.0006, Conviction: 1.0018)

Examples of rules with 3 itemsets:

- *Rule1:*
O3=Good, HUM=Moderate → STATION=Urban (Confidence: 0.6183, Lift: 1.1732, Leverage: 0.03, Conviction: 1.2392)
- *Rule2:*
SEASON=Winter, TEMP=Moderate → O3=Good (Confidence: 0.9739, Lift: 1.0003, Leverage: 0.0001, Conviction: 1.0125)
- *Rule3:*
PM10=Good, HUM=Moderate → SO2=Good (Confidence: 0.9737, Lift: 1.0418, Leverage: 0.0086, Conviction: 2.4839)
- *Rule4:*
TEMP=High, STATION=Urban → SO2=Good (Confidence: 0.9573, Lift: 1.0243, Leverage: 0.007, Conviction: 1.5312)

Examples of rules with 4 itemsets:

- *Rule1:*
SO2=Good, PRES=High, STATION=Industrial → O3=Good (Confidence: 0.9746, Lift: 1.001, Leverage: 0.0003, Conviction: 1.0397)
- *Rule2:*
O3=Good, TEMP=High, HUM=Moderate → SO2=Good (Confidence: 0.9754, Lift: 1.0437, Leverage: 0.0095, Conviction: 2.6587)
- *Rule3:*
SEASON=Spring, SO2=Good, PRES=High → O3=Good (Confidence: 0.9879, Lift: 1.0147, Leverage: 0.003, Conviction: 2.1851)
- *Rule4:*
STATION=Urban → O3=Good, PRES=High, NO2=Good (Confidence: 0.6376, Lift: 0.9274, Leverage: -0.0263, Conviction: 0.8623)

Examples of rules with 5 itemsets:

- *Rule1:*
NO2=Good, O3=Good, PRES=High, STATION=Industrial → SO2=Good (Confidence: 0.9818, Lift: 1.0505, Leverage: 0.0126, Conviction: 3.5937)
- *Rule2:*
SEASON=Winter, SO2=Good, NO2=Good, TEMP=Moderate → O3=Good (Confidence: 0.9831, Lift: 1.0098, Leverage: 0.002, Conviction: 1.5677)
- *Rule3:*
SO2=Good, O3=Good, TEMP=High, HUM=Moderate → NO2=Good (Confidence: 0.9699, Lift: 1.0338, Leverage: 0.0072, Conviction: 2.0534)
- *Rule4:*
SO2=Good, NO2=Good, O3=Good, STATION=Urban → PM10=Good (Confidence: 0.6346, Lift: 0.9457, Leverage: -0.0164, Conviction: 0.9003)

Examples of rules with 6 itemsets:

- *Rule1:*
PM10=Good, SO2=Good, NO2=Good, HUM=High, PRES=High → TEMP=High (Confidence: 0.6514, Lift: 1.0955, Leverage: 0.0202, Conviction: 1.1629)

- *Rule2:*
SO2=Good, NO2=Good, HUM=High, PRES=High, STATION=Urban → O3=Good (Confidence: 0.988, Lift: 1.0149, Leverage: 0.0031, Conviction: 2.2127)
- *Rule3:*
PM10=Good, SO2=Good, TEMP=High, HUM=High, PRES=High → NO2=Good (Confidence: 0.956, Lift: 1.0191, Leverage: 0.0043, Conviction: 1.4067)
- *Rule4:*
PM10=Good, SO2=Good, NO2=Good, O3=Good, TEMP=High → HUM=High (Confidence: 0.6085, Lift: 1.0185, Leverage: 0.0043, Conviction: 1.0282)

Examples of rules with 7 itemsets:

- *Rule1:*
PM10=Good, SO2=Good, O3=Good, TEMP=High, HUM=High, PRES=High → NO2=Good (Confidence: 0.9558, Lift: 1.0188, Leverage: 0.0042, Conviction: 1.3997)
- *Rule2:*
PM10=Good, NO2=Good, HUM=High, PRES=High → TEMP=High, SO2=Good, O3=Good (Confidence: 0.6242, Lift: 1.0989, Leverage: 0.0206, Conviction: 1.1496)
- *Rule3:*
PM10=Good, SO2=Good, NO2=Good, O3=Good, TEMP=High, HUM=High → PRES=High (Confidence: 0.9662, Lift: 1.2984, Leverage: 0.0526, Conviction: 7.5779)

As shown in Table 6.8, the number of k-itemsets was analyzed with varying minimum support thresholds from 0.2 to 0.9 in increments of 0.1. According to the results, the maximum and the minimum number of itemsets were found as 108 and 1 for $k=3$ and $k=7$, respectively, for support threshold = 0.2. There were no 8-itemsets generated from the attributes given in Table 6.8. Figure 6.3 shows the variations of the number of k-itemsets, for the support value of 0.2. It can be observed that the number of frequent 2-itemsets and 5-itemsets are close to each other. The WARM algorithm

generally constructs a form quite similar to the bell curve. However, the skewness and kurtosis of the curve can change according to the minimum support threshold.

Table 6.8 The variations of the k-itemsets according to different support thresholds

Support	Size of Itemset						
	1	2	3	4	5	6	7
0.2	14	61	108	96	45	10	1
0.3	12	34	50	38	16	3	0
0.4	8	23	30	16	3	0	0
0.5	8	18	16	5	0	0	0
0.6	5	9	7	2	0	0	0
0.7	4	5	1	0	0	0	0
0.8	3	3	1	0	0	0	0
0.9	3	2	0	0	0	0	0

As shown in Figure 6.4, Figure 6.5, and Figure 6.6, when the minimum support value decreases, the number of frequent rules increases almost exponentially. When the minimum support value is too high, the ARM and WARM algorithms cannot discover enough information from data. On the other hand, a large amount of frequent rules is generated when the algorithms are executed with small support values. For this reason, a reasonable threshold can be set to mine valuable and sufficient frequent rules.

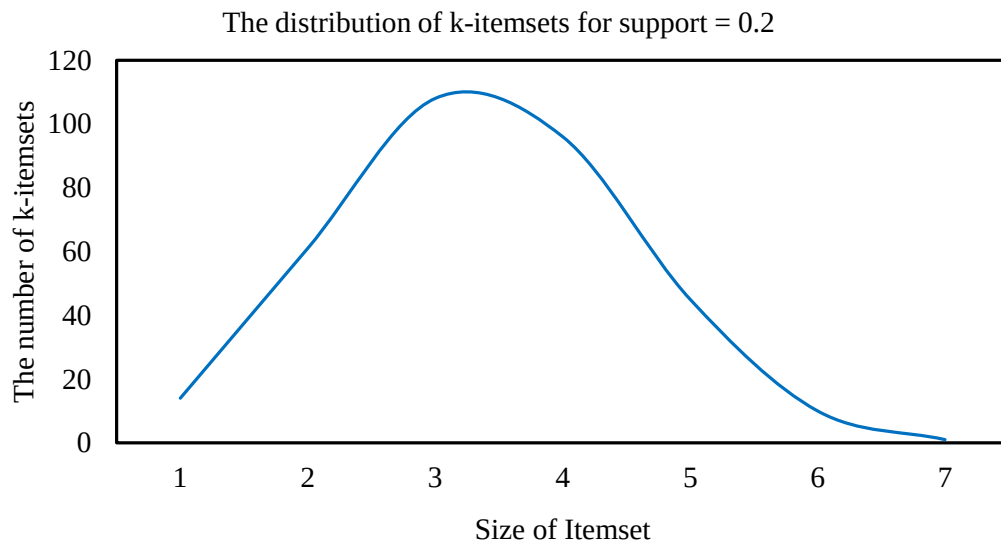


Figure 6.3 The variations in the number of k-itemsets found by WARM model

Figure 6.4 displays the comparison results of the applied methods, Apriori, and the proposed WARM model at the confidence threshold of 0.6. For different support thresholds, the number of strong association rules was determined. In this configuration, the difference between the value of item weights is one. For example, assume that the weights of the attribute values of PM₁₀ “Good”, “Moderate”, “Sensitive”, “Unhealthy”, “Bad”, and “Dangerous” are determined as {1, 2, 3, 4, 5, 6}, respectively, with only one difference among different item values. According to the results, WARM can detect more association rules than Apriori. It means that WARM explored other important rules that the Apriori method could not detect. For example, when the minimum support is 0.45, WARM discovers 324 frequent rules, while standard ARM finds 280 rules for the same dataset. From this point of view, WARM and ARM algorithms have comparable effects on the number of rules. The effect of adding item weights into the generation of frequent itemsets contributes much to find the undiscovered association rules. If the weights related to air pollutants and meteorological factors are considered, then some missing, probably important, rules can be detected.

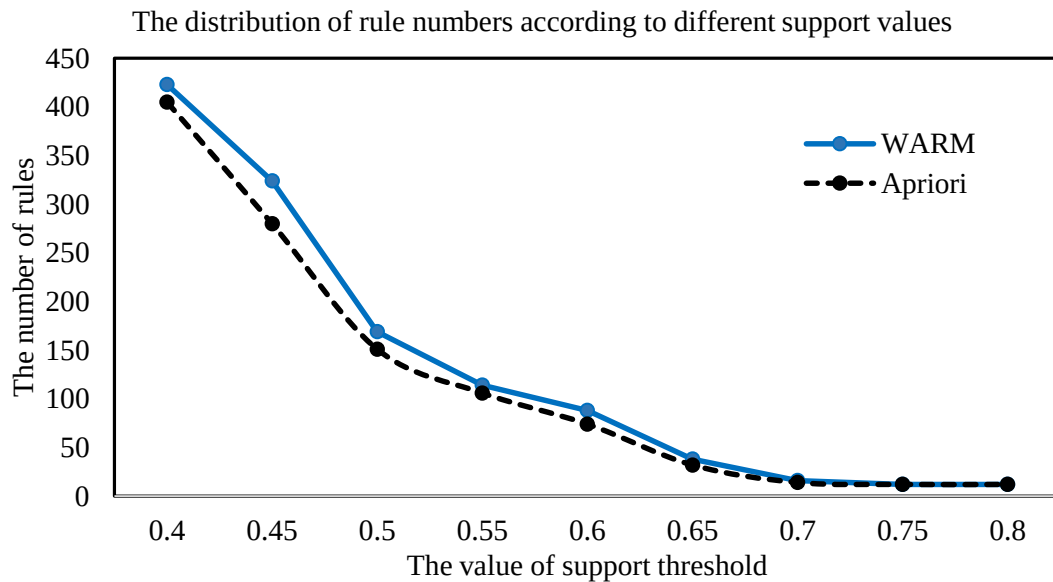


Figure 6.4 The comparisons of WARM and Apriori according to the number of rules in terms of different support thresholds at 0.6 confidence

In order to interpret the impact of different confidence levels in the generation of association rules, experiments were also done at the confidence threshold of 0.9. Figure 6.5 displays the comparison results which are obtained with the same item weights as in Figure 6.4. As expected, increasing the confidence value from 0.6 to 0.9 decreased the total number of found rules. For instance, 180 rules were now detected by WARM at the support value of 0.4 while the model generated 423 rules when the confidence value was 0.6. The same condition was valid for Apriori. By the way, WARM performed even better than Apriori at 0.9 in finding more association rules when compared with the results of 0.6. Because the proportional difference between them increased. To illustrate, the ratios of the difference of the number of found association rules among two models at the support value of 0.55 are 7.02% and 13.85% in confidence thresholds of 0.6 and 0.9, respectively. It is apparent that the detection of the important rules was performed by the improvement with approximately 7%.

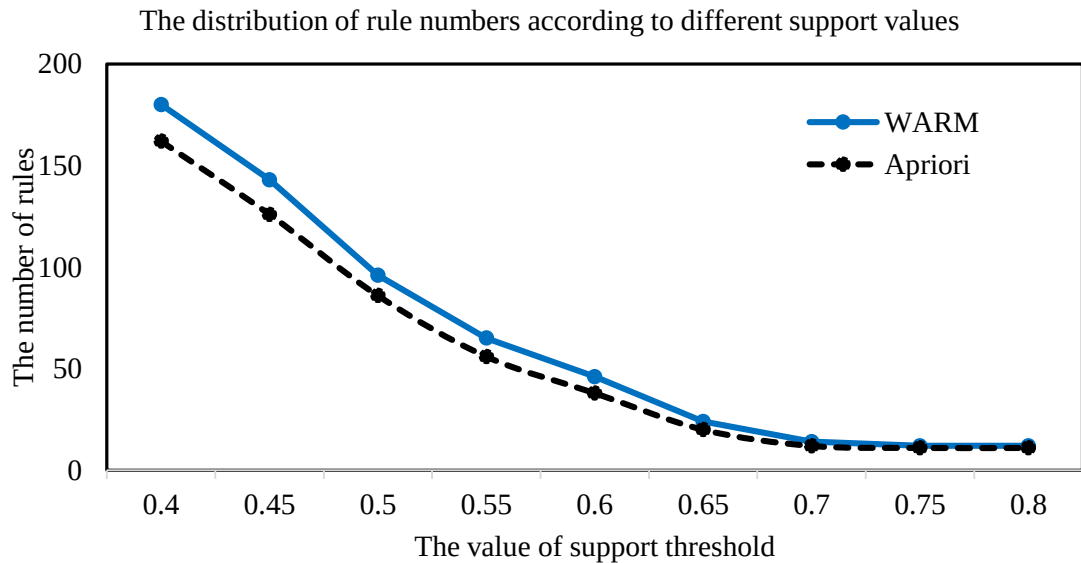


Figure 6.5 The comparisons of WARM and Apriori according to the number of rules in terms of different support thresholds at 0.9 confidence

The other important issue is the effect of different weight assignments. To show the difference, experiments were repeated with item weights which were doubled, respectively. For example, assume the weights of the attribute values of PM₁₀ “Good”, “Moderate”, “Sensitive”, “Unhealthy”, “Bad”, and “Dangerous” are now determined

as $\{1, 2, 4, 8, 16, 32\}$ according to this scenario. Figure 6.6 demonstrates the results according to this case at the confidence value of 0.9. There is an increase in the number of association rules at the support values of 0.45, 0.55, and 0.65. The change of item weights positively affected the output. Therefore, the determination of weights properly is very important for the WARM method.

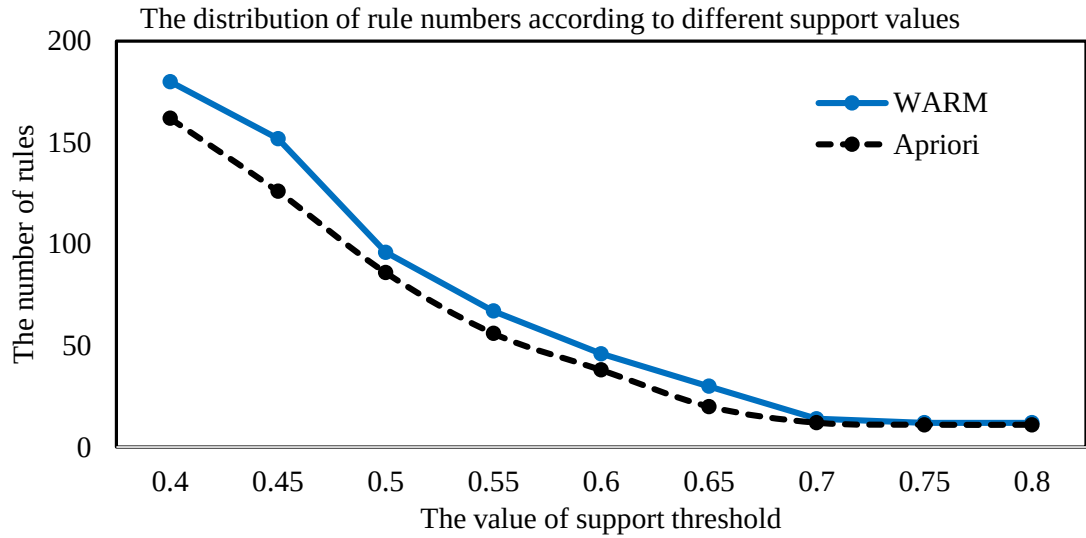


Figure 6.6 The comparisons of WARM and Apriori according to the number of rules in terms of different support thresholds at 0.9 confidence with doubled item weights

CHAPTER SEVEN

TIME SERIES STUDIES IN ENVIRONMENTAL INFORMATICS

Soil temperature has a great effect on plant development, yield increase, chemical and biological activities in the soil. Estimation of soil temperature is of great importance for this purpose. This chapter proposes a novel multi-view multi-depth learning framework for soil temperature prediction. Under the proposed framework, soil temperature prediction in 16 stations of the United States between the years of 2011 and 2019 at various soil depths is performed using multivariate time series and machine learning methods. This is the first study that two different views to represent antecedent soil data (i.e. soil moisture and soil temperature) and past meteorological data (air temperature, precipitation, relative humidity, solar radiation, wind speed, and vapor pressure) were designed to capture different features. It is the first time that multi-view multi-depth learning has provided a very powerful opportunity to estimate soil temperature at different soil depths when both time series and machine learning methods are used as base learners.

7.1 Material and Methods

In this study, multi-view learning was applied to estimate soil temperature at different depths using three machine learning methods (SVR, MLR, and MLPReg) and two multivariate time series methods (ARIMAX and SARIMAX). In this part, time series methods and the proposed multi-view multi-depth learning technique are discussed.

7.1.1 Time Series Algorithms

Time series analysis is a statistical technique that comprises methods to extract meaningful and significant statistics or other characteristics inherent in time series data. As in the data mining and machine learning tasks, time series forecasting can be employed to predict future values of test cases by making inferences based on the previously observed values. Multivariate time series analysis is the analysis of

relationships between multiple variables while univariate time series analysis is a quantitative analysis of only one variable.

The experimental data of this chapter consist of more than one variable to forecast future values. Therefore, in this section, two popular multivariate time series methods (ARIMAX and SARIMAX) that were used in the experiments are explained.

7.1.1.1 Autoregressive Integrated Moving Average with Explanatory Variable (ARIMAX)

Autoregressive Integrated Moving Average with Explanatory Variable (ARIMAX) (Williams, 2001) is an extension of Autoregressive Integrated Moving Average (ARIMA) (Contreras, Espinola, Nogales, & Conejo, 2003) model through the inclusion of exogenous variables X_i . ARIMA models are the combination of two stochastic processes which are Auto-Regressive (AR) and Moving Average (MA). AR reflects the effect of the past values on the current observation while MA assumes that the value of the dependent variable depends on the residual errors of previous time intervals. The term, Integrated (I) is the use of differencing of raw observations to make the time series stationary. The general model of ARIMA is in (7.1) where c is the intercept of the model, ϕ and θ are unknown parameters, ϵ_t is white noise ($\sim N(0, \sigma^2)$) and Y'_i is the differenced series (it may have been differenced more than once).

$$Y'_i = c + \underbrace{\phi_1 Y'_{i-1} + \phi_2 Y'_{i-2} + \dots + \phi_p Y'_{i-p}}_{\text{AR term}} + \underbrace{\epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}}_{\text{MA term}} \quad (7.1)$$

ARIMAX is designed for multivariate datasets and it is represented by three parameters (p, d, q) where p is the number of AR lags, d is the degree of differencing (the number of times the dependent variable Y_i has to be differenced to produce the stationary Y'_i) and q is the number of MA lags. The standard ARIMAX would be written as in (7.2) where β is an unknown coefficient.

$$Y'_i = c + \beta X_i + \phi_1 Y'_{i-1} + \phi_2 Y'_{i-2} + \dots + \phi_p Y'_{i-p} + \epsilon_i + \theta_1 \epsilon_{i-1} + \theta_2 \epsilon_{i-2} + \dots + \theta_q \epsilon_{i-q} \quad (7.2)$$

The difference from ARIMA is the inclusion of additional explanatory variables of $X_i = \{x_1, x_2, \dots, x_m\}$ to show the impact of all features on the dependent variable. This model does not consider seasonal effects.

7.1.1.2 Seasonal Autoregressive Integrated Moving Average with Explanatory Variable (SARIMAX)

Another version of the ARIMAX model with the seasonal effect is known as Seasonal Autoregressive Integrated Moving Average with Explanatory Variable (SARIMAX) (Nourani, Kisi, & Komasi, 2011). A new parameter list is added to the aforementioned (p, d, q) values of the ARIMAX model as seasonal terms (P, D, Q, s) where P is the seasonal AR order, D is the seasonal difference order, Q is the seasonal MA order and s is the number of periods in a season (i.e. 1 for yearly data, 4 for quarterly data, 12 for monthly data, 52 for weekly data and 365 for daily data). The new model is specified as SARIMAX $(p, d, q) \times (P, D, Q, s)$.

An example of SARIMAX (2, 1, 0) $\times$ (1, 1, 0, 4) can be expressed as in (7.3):

$$Y'_i = c + \beta X_i + \phi_1 Y'_{i-1} + \phi_2 Y'_{i-2} + \tilde{\phi}_1 Y'_{i-4} + \phi_1 \tilde{\phi}_1 Y'_{i-5} + \phi_2 \tilde{\phi}_1 Y'_{i-6} + \epsilon_i \quad (7.3)$$

In (7.3), there are no non-seasonal and seasonal MA lags, the number of periods is quarterly and $\tilde{\phi}$ term expresses the seasonal AR lags. Another example of SARIMAX (0, 0, 1) $\times$ (0, 0, 1, 12) can be written as follows:

$$Y'_i = c + \beta X_i + \theta_1 \epsilon_{i-1} + \tilde{\theta}_1 \epsilon_{i-12} + \theta_1 \tilde{\theta}_1 \epsilon_{i-13} + \epsilon_i \quad (7.4)$$

In (7.4), there are no non-seasonal and seasonal AR lags, the number of periods is monthly and $\tilde{\theta}$ term expresses the seasonal MA lags.

7.1.2 The Proposed Method: Multi-View Multi-Depth Soil Temperature Prediction (MV-MD-STP)

Let's consider a base learner for a multi-view learning framework as L which is selected from one of the machine learning or time series methods. Firstly, the whole dataset is separated into several views, z , $V = \{V_1, V_2, \dots, V_z\}$ where each V_i consists of several explanatory variables from the whole feature set X and the features in different views are independent of each other. The procedure starts with training prediction models for all views using the selected method L . Using the trained models $L = \{L_1, L_2, \dots, L_z\}$, all test objects in $O = \{o_1, o_2, \dots, o_k\}$ are forecasted in each view and prediction outputs are obtained as $Y^* = \{Y_1^*, Y_2^*, \dots, Y_z^*\}$ where each $Y_i^* = \{Y_{i1}, Y_{i2}, \dots, Y_{ik}\}$ for k test instances. The thing to do is now the aggregation of outputs obtained in each view by taking the average of the forecasted values in Y^* for every instance. The final prediction for all test instances is $Y = \{Y_{avg_1}^*, Y_{avg_2}^*, \dots, Y_{avg_k}^*\}$ as a consequence. The problem can be expressed as in (7.5) where f is the general formula of the selected learner, L .

$$Y_{avg_k}^* = \frac{1}{z} \sum_{i=1}^z f_L(V_i) \quad (7.5)$$

The proposed approach employs two main principles: the complementary principle and the consensus principle. Thanks to the *complementarity principle*, it improves the learning performance by comprehensively using information from multiple views since each view may contain some information non-exist in other views. Thanks to the *consensus principle*, it maximizes the agreement on multiple views since minimizing the disagreement among the views reduces their individual errors. Consequently, both complementary and consensus principles play important roles in the proposed method.

The proposed method is based on the motivation of multi-view learning by combining two feature sets (soil data and meteorological data) into a single predictor to forecast soil temperature values at different depths (2, 4, 8, 20, and 40 inches). The general framework of the proposed system is shown in Figure 7.1. In this study, two views are formed as V_M and V_S to express the effect of meteorological data (M) and

soil-related data (S), respectively. Multivariate time series methods (ARIMAX and SARIMAX) and machine learning methods (SVR, MLR, and MLPReg) are used as base learners to train V_M and V_S . Using the procedure in (7.10), final predictions for the soil temperature at different depths are obtained.

The pseudo-code is shown in Figure 7.2. The algorithm has two main phases: the training phase and the testing phase. In the training phase, the algorithm applies the learning method for each view V_i separately and this process is repeated for each depth d . In the testing phase, the class label Y_i^* is assigned to instance X_i by averaging the outputs of all views at the depth of d . This process is repeated k times (for each instance in the test set). The time complexities of the training and testing parts of the algorithm are given by $O(z.d.T(n))$ and $O(k.d.z)$, respectively, where z is the number of views, d is the number of depths, T represents the time required for the execution of a learning method on n training instances, and k is the number of instances in the test set.

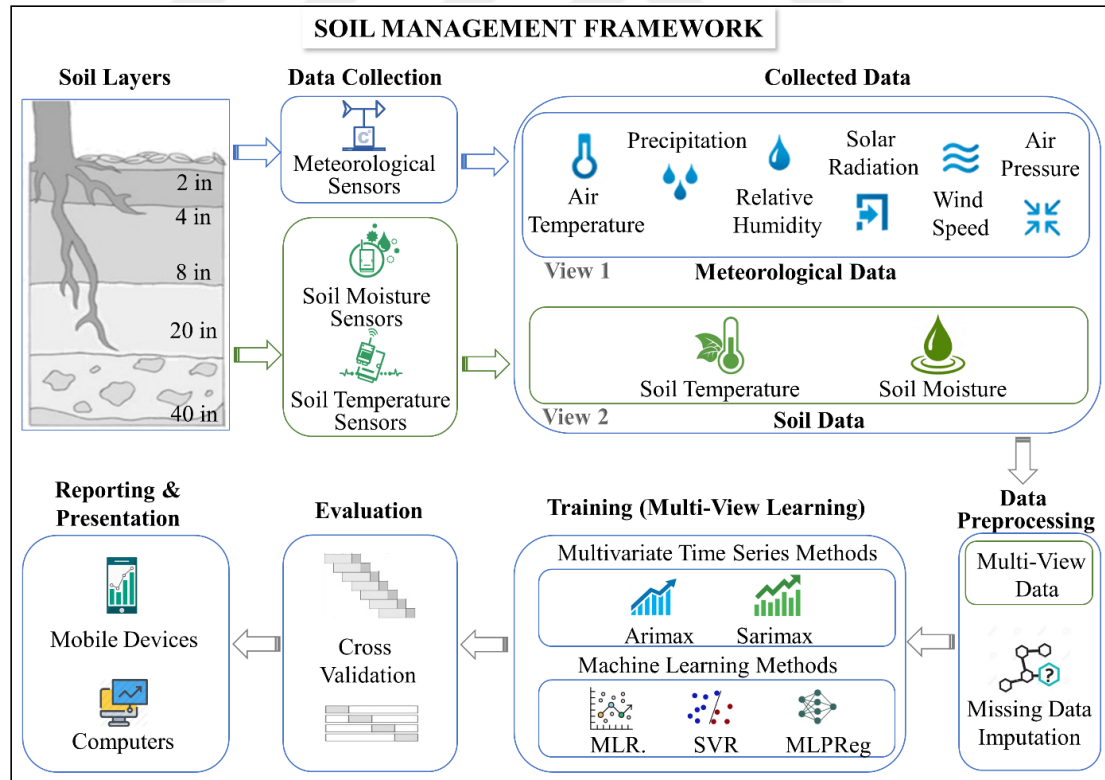


Figure 7.1 The proposed multi-view multi-depth soil temperature prediction framework

Algorithm Multi-View Multi-Depth Learning (MV-MD-STP)

Inputs:

D : Multi-view multi-depth training set with z views d depths
 O : Test set with k instances

Output:

Y^* : Predicted values for the test set

for each depth d **in** D **do**

for each view V **in** D **do**

$L_{d,V} = f_L(D, d, V)$

end for each

end for each

for $i = 1$ **to** k **do**

for $j = 1$ **to** d **do**

$\text{sum} = 0$

for $m = 1$ **to** z **do**

$\text{sum} = \text{sum} + L_{j,m}(X_i \in O)$

end for

$Y_{avg_j}^* = \text{sum} / z$

$Y_i^* = Y_i^* \cup Y_{avg_j}^*$

end for

$Y^* = Y^* \cup Y_i^*$

end for

End Algorithm

Figure 7.2 The pseudo-code of the proposed method

7.2 Case Study - Comparison of Machine Learning and Time Series Methods for Multi-View Multi-Depth Soil Temperature Prediction

In order to take precautions and follow the appropriate policy for critical situations, regular measurement of soil temperature is required. In that sense, many studies present methods to predict the future values of soil temperature in a specific region by considering the recorded past data. In this study, daily values of soil data (past values of soil moisture and soil temperature at different depths) and meteorological data (air temperature, precipitation, relative humidity, solar radiation, wind speed, and vapor pressure) in 16 stations from three states (Utah, Alabama and New Mexico) of United States between the years of 2011 and 2019 are considered to estimate the soil temperatures at five soil depths (2 inches, 4 inches, 8 inches, 20 inches, and 40 inches).

To forecast soil temperature values, both machine learning and multivariate time series approaches are applied under a multi-view learning framework. Machine learning methods include Support Vector Regression (SVR), Multiple Linear

Regression (MLR), and Multilayer Perceptron Regression (MLPReg) while time series methods are Autoregressive Integrated Moving Average with Explanatory Variable (ARIMAX) and Seasonal Autoregressive Integrated Moving Average with Explanatory Variable (SARIMAX) methods. Multi-view learning has been proposed in a number of studies from various subject areas such as urban water quality prediction (Liu, Zheng, Liang, Liu, & Rosenblum, 2016), air quality estimation (Lv, Li, Chen, & Chen, 2019), EEG seizure detection (Yuan, Xun, Jia, & Zhang, 2018), software comprehension (Saeidi, Hage, Khadka, & Jansen, 2019), city event extraction from Twitter data streams (Farajidavar, Kolozali, & Barnaghi, 2017), identification of risky traffic locations in intelligent transportation systems (Zhang, Lu, Zhang, Shang, & Wang, 2018) and many others until now on. Multi-view learning provides a novelty to this study, which is applied for the first time in the estimation of soil temperature, by considering soil data and meteorological data as two different views.

The major contributions of this study can be listed as below:

- This study proposes a novel multi-view multi-depth learning framework for soil temperature prediction.
- This is the first study that multivariate time series and machine learning approaches are compared to each other so that which one is the most suitable for this kind of study is presented.
- Multi-view learning is implemented for the first time in the prediction of soil temperature and its applicability has been shown.

The main benefits of the proposed multi-view multi-depth framework can be summarized as follows:

- Using the direct relations of meteorological and soil data, the future soil temperature value of a region can be estimated.
- A smart soil management system can be designed using the forecasted values to apply irrigation in case of need.
- The yield rate in agriculture can be increased by using the estimated soil temperature values.
- Resource consumption in agriculture can be remarkably decreased.

7.2.1 Dataset Description

All the data including both meteorological and soil parameters were gathered from the website (<https://www.wcc.nrcs.usda.gov>) of the National Water and Climate Center (NWCC) in National Resources Conservation Service (NRCS) under the United States Department of Agriculture. 16 stations from three states (Utah, Alabama, and New Mexico) of the United States were considered in the interval of 01.01.2011 and 10.31.2019. Figure 7.3 shows the study area and the aforementioned stations which are as follows: Cave Valley, Hals Canyon, Harms Way, Manderfield, Panguitch, Sand Hollow, Split Mountain, Spooky, Tule Valley, Vermillion, Grouse Creek from Utah, Dee River Ranch, Isbell Farms, Morris Farms, and Sudduth Farms from Alabama and Sevilleta from New Mexico.

Meteorological and soil data were collected hourly. Meteorological data include min/max values of air temperature, precipitation increment, relative humidity, average of solar radiation, avg/max values of wind speed, and partial/saturated vapor pressure. Soil data are the values of soil moisture percent and soil temperature observed at 2, 4, 8, 20, and 40 inches of soil depths.

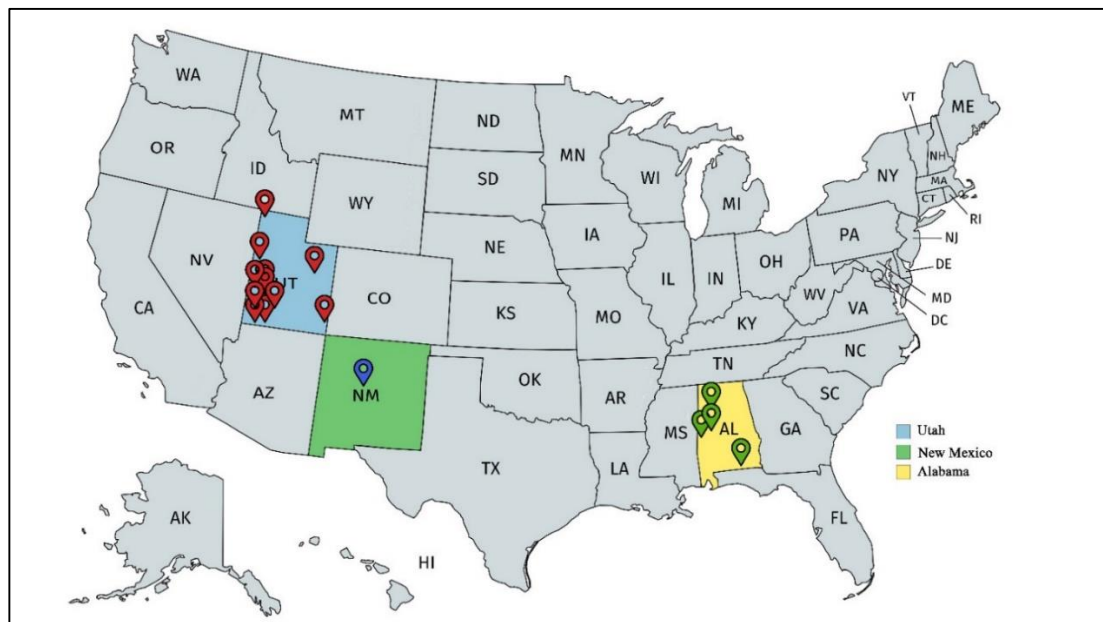


Figure 7.3 The study area for soil temperature prediction

Data preprocessing is an important process before data analysis to provide more accurate and reliable results. Therefore, missing data imputation was performed in the beginning. The records were extracted if all of the values in the respective rows are empty. After that, the empty indices in specific columns were filled according to the average values of the same index in other years. Then, the conversion of hourly data to daily data using the mean values of 24 hours was performed since daily trends are desired to be observed in this study.

7.2.2 Experimental Study and Results

All experiments were performed in Python. Multivariate time series and machine learning strategies were applied. Multivariate time series methods are ARIMAX and SARIMAX as Machine Learning part includes SVR, MLPReg, and MLR. “scikit-learn” and “statsmodels” libraries were used to implement the mentioned methods. All methods were applied as a base learner under a multi-view learning framework to handle and combine two views (meteorological view and soil view). For tuning hyperparameters and producing robust measurements of model performance, all of the experiments were applied using ten-fold cross-validation.

In the first part, the data was analyzed to discover the remarkable correlations among the features. Three correlograms were created to represent the relationships of attributes in three states (Alabama, New Mexico, and Utah) of the U.S. These matrices were created by taking the average of correlation values of all stations under a state. To compute the outputs, Pearson Correlation that measures the linear dependence between the two features was used. Figure 7.4 displays the findings by showing both negative and positive correlations (distinguished by the color of circles) between parameters with their magnitudes (distinguished by the size of each circle). In the matrices, positive and negative correlations are represented by blue and red colors, respectively. The intensity of the color and the size of the circle are proportional to the correlation coefficients. According to the results, there is almost no correlation between soil temperature (ST) and precipitation increment (PI) for all states while air temperature (AT), and vapor pressure saturated (VPS) have high positive correlations

with ST. In addition, vapor pressure partial (VPP) and solar radiation (SR) have moderate positive correlations. Relative humidity (RH) is negatively correlated with ST in New Mexico and Utah as there is a little correlation in Alabama. Wind speed (WS) is negatively correlated with ST in Alabama and it shows very little correlation in other states. Soil moisture (SM) is usually negatively correlated with ST.

In order to get the most accurate prediction results, a grid search was applied for all models and for all soil depths to obtain the best parameters. The range of (p, d, q) parameters of ARIMAX and seasonal terms (P, D, Q) parameters of SARIMAX were set between 0 and 4, and the number of seasons in a period, s parameter of SARIMAX was determined as 12 for monthly time series analysis. SVR was implemented using *RBF kernel* and other parameters (C, ϵ, γ) were set from the intervals of $[0.1, 1, 100, 100]$, $[0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10]$, $[0.0001, 0.001, 0.005, 0.1, 1, 3, 5]$, respectively. For MLPReg, hidden layer sizes were selected from $[10, 20, 30, 40, 50, 100, 200, 300, 400, 500]$, activation function was either *tanh* or *relu*, the solver for weight optimization was either *adam* or *lbfgs* and *alpha* (L2 penalty term) parameter was set from the interval of 10^{-1} to 10^{-5} . Any other parameters were set as their default values.

After model creation and the application on the dataset, it is important to assess the performance of them according to a different set of criteria. For this purpose, four performance metrics were calculated in this study as mean square error (MSE) (defined as in (7.6)), mean absolute error (MAE) (defined as in (7.7)), root mean square error (RMSE) (defined as in (7.8)) and coefficient of determination (R^2) (defined as in (7.9)). The evaluation results of 16 stations for all soil depths (2, 4, 8, 20, and 40 inches) are listed in Table 7.1, Table 7.2, Table 7.3, Table 7.4 and Table 7.5. Besides, the average MSE, MAE, RMSE and R^2 values of the applied methods at different depths are shown in Figure 7.5 (a and b) and Figure 7.6 (c and d), respectively.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - Y_i^*)^2 \quad (7.6)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - Y_i^*| \quad (7.7)$$

$$\text{RMSE} = \sqrt{\frac{\sum_1^n (Y_i - Y_i^*)^2}{n}} \quad (7.8)$$

$$R^2 = 1 - \frac{\sum_1^n (Y_i - Y_i^*)^2}{\sum_1^n (Y_i - Y_{\text{avg}})^2} \quad (7.9)$$

According to the experimental results, it is obvious that SVR is the best model under a multi-view learning framework at all of the soil depths to forecast soil temperature. MLPReg comes second in terms of performance metrics. Even though MLR performed better than time series methods at a depth of 40, it stayed behind others when soil depth gets lower. ARIMAX and SARIMAX have mostly achieved head to head results. For instance, in terms of RMSE, the average performances of the SVR, MLPReg, MLR, ARIMAX, and SARIMAX methods were observed as 2.52, 2.61, 3.12, 3.35, and 3.36 respectively at the depth of 40 inches as shown in Figure 7.6 (c).

It is also seen that machine learning methods generally made better predictions at all depths than time series methods. This means that machine learning has a better capability of generalization under the multi-view multi-depth framework. For example, when considering the MAE metric, the SVR method provides improvements of about 2.43 to 5.66 in Figure 7.5 (b) at the various soil depths compared to the time series method.

Another important thing is that soil temperature is estimated more accurately when soil depth is decreased from 40 to 2 inches. It is valid for all of the applied methods that multi-view learning has successfully performed soil temperature estimation by aggregating both meteorological and soil dependent parameters effectively. For example, in Figure 7.5 (a), the MSE values obtained from the ARIMAX method are 12.03, 10.88, 8.95, 6.33, and 5.60 for the soil depths 40, 20, 8, 4, and 2 inches respectively. Also, when Figure 7.6 (d) is interpreted, it is seen that R^2 values get lower as soil depth increases, so estimation performance deteriorates in a parallel way.

The best performance of SVR at high soil depths (40 inches and 20 inches) was observed at the Panguitch station of Utah, which includes a semi-arid climate where

the summers are warm, dry, and mostly clear and the winters are freezing, snowy, and partly cloudy. At soil depths of 8 and 4 inches, SVR obtained the most accurate predictions at the Seville station of New Mexico, which is located in the Chihuahuan desert. The region mostly has a semi-arid climate and the mountains and high plains are covered by the desert. As the region receiving low rainfall has quite hot days and cool nights in summer, winters are cold. At the lowest soil depth (2 inches), the SVR results of Tule Valley station of Utah were the best among others. Tule Valley is also a semi-arid region. There is a large playa that collects all precipitation from the drainage basin in the center of the valley. On the other hand, the lowest accuracies of SVR were generally obtained (at the depths of 4, 8, and 20 inches) in the Dee River Ranch and Isbell Farms stations of Alabama, where the climate is warm and humid subtropical resulting from its mid-latitude location and proximity to the Gulf of Mexico. We can deduce that it was easier to accurately model soil temperature prediction of semi-arid regions compared to modeling for humid regions by using the proposed multi-view multi-depth method.

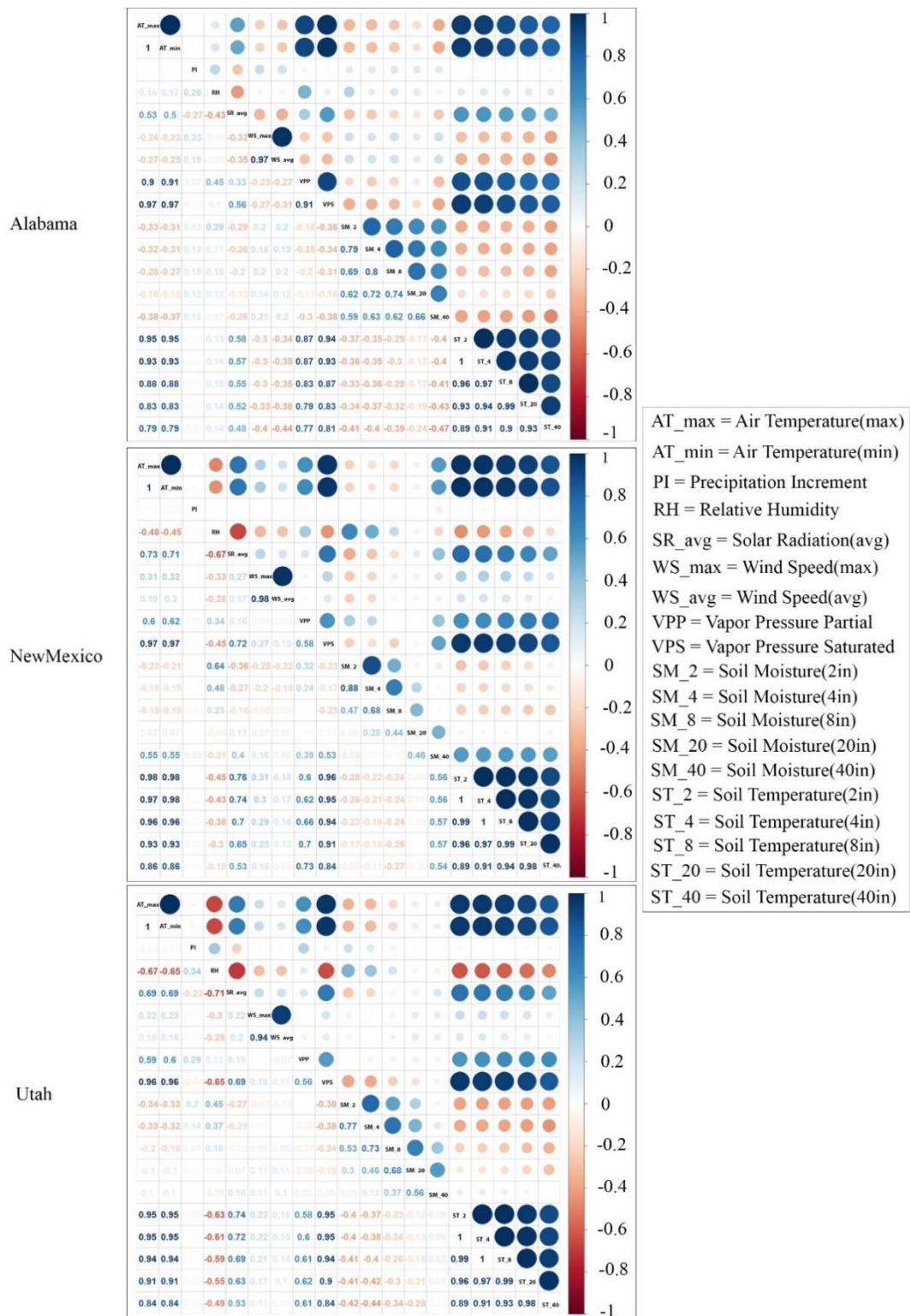


Figure 7.4 Correlations among the used parameters for three states of the U.S.

Table 7.1 Comparisons of the applied methods in different stations at the depth 40 inch

Data	Depth -40																			
	SVR				MLR				MLPReg				ARIMAX				SARIMAX			
	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²
Cave Valley	1.95	5.90	2.43	0.96	2.20	7.58	2.75	0.95	2.02	6.71	2.59	0.95	2.30	8.38	2.86	0.92	2.30	8.38	2.86	0.92
Dee River Ranch	2.07	6.65	2.58	0.91	2.32	8.24	2.87	0.89	2.02	6.83	2.61	0.91	2.77	11.60	3.38	0.74	2.77	11.60	3.38	0.74
Grause Creek	1.82	6.29	2.50	0.94	2.29	8.15	2.85	0.92	2.03	7.10	2.66	0.93	2.43	9.07	3.00	0.91	2.43	9.07	3.00	0.91
Hals Canyon	1.92	6.79	2.60	0.95	2.62	10.84	3.29	0.92	2.15	7.78	2.78	0.94	3.03	14.48	3.74	0.88	3.03	14.48	3.74	0.88
Harms Way	1.77	5.77	2.40	0.95	2.21	7.85	2.80	0.93	2.21	8.52	2.83	0.92	2.45	9.84	3.08	0.90	2.45	9.84	3.08	0.90
Isbell Farms	2.04	6.84	2.61	0.94	2.88	12.80	3.57	0.88	1.42	3.30	1.82	0.99	3.70	24.70	4.62	0.69	3.79	25.37	4.73	0.68
Manderfield	1.78	6.14	2.47	0.94	2.26	7.80	2.82	0.92	2.00	6.82	2.60	0.93	2.39	9.21	3.00	0.90	2.39	9.21	3.00	0.90
Morris Farms	2.15	7.46	2.73	0.92	2.68	11.64	3.40	0.87	2.40	9.40	3.06	0.90	3.30	17.60	4.03	0.72	3.30	17.60	4.03	0.72
Panguitch	1.44	3.68	1.92	0.92	1.83	5.24	2.29	0.89	1.52	3.93	1.98	0.92	2.04	6.46	2.52	0.86	2.04	6.46	2.52	0.86
Sand Hallow	1.96	6.07	2.46	0.96	2.22	8.14	2.85	0.95	2.12	7.26	2.69	0.96	2.41	9.65	3.08	0.93	2.41	9.65	3.08	0.93
Sevilleta	1.82	5.77	2.40	0.96	2.27	8.56	2.93	0.94	1.88	6.19	2.49	0.96	2.54	10.94	3.26	0.92	2.54	10.94	3.26	0.92
Split Mountain	2.06	8.01	2.83	0.95	2.82	12.26	3.50	0.93	2.17	8.21	2.86	0.95	3.02	13.77	3.67	0.91	3.02	13.77	3.67	0.91
Spooky	1.81	5.43	2.33	0.96	2.19	8.14	2.85	0.95	1.93	6.21	2.49	0.96	2.33	8.92	2.96	0.93	2.33	8.92	2.96	0.93
Sudduth Farms	1.97	5.71	2.39	0.94	2.28	7.87	2.81	0.92	1.91	5.80	2.41	0.94	2.36	8.71	2.94	0.89	2.36	8.71	2.94	0.89
Tule Valley	2.33	9.97	3.16	0.97	3.12	15.20	3.90	0.93	2.66	11.64	3.41	0.95	3.45	18.30	4.25	0.91	3.45	18.30	4.25	0.91
Vermillion	1.92	6.05	2.46	0.94	2.39	22.91	4.42	0.77	1.92	6.32	2.51	0.94	2.47	10.79	3.22	0.89	2.47	10.79	3.22	0.89

Table 7.2 Comparisons of the applied methods in different stations at the depth 20 inch

Data	Depth -8																			
	SVR				MLR				MLPReg				ARIMAX				SARIMAX			
	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²
Cave Valley	1.71	4.91	2.22	0.98	2.20	7.58	2.75	0.95	2.07	7.09	2.66	0.97	2.15	7.44	2.71	0.95	2.15	7.44	2.71	0.95
Dee River Ranch	2.02	6.38	2.52	0.94	2.40	9.06	3.01	0.92	2.05	7.07	2.65	0.94	2.73	11.76	3.36	0.83	2.71	11.62	3.34	0.83
Grause Creek	1.79	5.95	2.44	0.96	2.24	7.98	2.82	0.95	2.00	6.97	2.63	0.95	2.14	8.40	2.89	0.94	2.32	8.40	2.89	0.94
Hals Canyon	1.93	6.89	2.62	0.97	2.62	11.13	3.33	0.95	2.14	7.98	2.82	0.96	2.64	11.46	3.36	0.94	2.64	11.46	3.36	0.94
Harms Way	1.68	5.22	2.28	0.97	2.15	7.40	2.72	0.96	2.25	9.56	2.91	0.95	2.24	8.15	2.82	0.95	2.24	8.15	2.82	0.95
Isbell Farms	2.40	15.12	3.87	0.92	3.13	18.33	4.27	0.90	2.98	19.07	4.35	0.90	3.31	30.96	4.43	0.85	3.39	29.33	4.41	0.86
Manderfield	1.87	5.84	2.42	0.96	2.28	8.12	2.85	0.95	1.97	6.73	2.59	0.96	2.37	8.96	2.96	0.93	2.37	8.96	2.96	0.93
Morris Farms	1.81	5.70	2.38	0.96	2.31	8.42	2.90	0.94	2.09	7.22	2.68	0.95	2.34	8.84	2.91	0.86	2.31	8.66	2.88	0.87
Panguitch	1.47	3.75	1.93	0.96	1.86	5.35	2.31	0.95	1.64	4.44	2.11	0.96	1.93	5.77	2.40	0.94	1.93	5.77	2.40	0.94
Sand Hallow	1.59	4.25	2.06	0.98	1.93	5.98	2.44	0.98	1.81	5.29	2.30	0.98	1.99	6.34	2.49	0.97	1.99	6.35	2.49	0.97
Sevilleta	1.63	4.70	2.17	0.98	1.97	6.82	2.61	0.97	1.74	5.36	2.31	0.98	2.15	19.51	3.54	0.92	2.14	19.49	3.54	0.92
Split Mountain	1.93	6.55	2.56	0.97	2.54	10.23	3.20	0.96	2.02	7.14	2.67	0.97	2.59	10.71	3.25	0.96	2.59	10.71	3.25	0.96
Spooky	1.86	5.71	2.39	0.97	2.29	8.42	2.90	0.96	2.02	6.70	2.59	0.97	2.42	9.30	3.03	0.95	2.42	9.30	3.03	0.95
Sudduth Farms	1.60	4.39	2.09	0.97	1.96	6.13	2.48	0.95	1.64	4.56	2.13	0.96	2.01	6.55	2.54	0.94	2.01	6.55	2.54	0.94
Tule Valley	1.85	6.35	2.52	0.98	2.45	10.09	3.18	0.97	2.04	7.27	2.69	0.98	2.57	11.22	3.31	0.96	2.57	11.22	3.31	0.96
Vermillion	1.56	5.19	2.27	0.97	2.25	45.84	5.52	0.71	1.79	6.13	2.47	0.96	2.10	8.68	2.88	0.94	2.23	8.42	2.83	0.94

Table 7.3 Comparisons of the applied methods in different stations at the depth 8 inch

Data	Depth -8																			
	SVR				MLR				MLPReg				ARIMAX				SARIMAX			
	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²
Cave Valley	1.52	3.93	1.98	0.99	1.89	5.89	2.43	0.98	1.71	4.88	2.21	0.98	1.86	5.75	2.38	0.97	1.86	5.75	2.38	0.97
Dee River Ranch	1.76	4.92	2.22	0.97	1.99	6.23	2.49	0.96	1.85	5.71	2.38	0.97	2.18	7.48	2.68	0.92	2.18	7.47	2.68	0.92
Grause Creek	1.57	4.48	2.12	0.98	2.02	6.47	2.54	0.97	1.79	5.39	2.32	0.97	2.11	7.00	2.64	0.96	2.11	7.00	2.64	0.96
Hals Canyon	1.58	4.86	2.20	0.99	2.07	7.34	2.71	0.98	1.68	5.21	2.28	0.98	2.12	7.48	2.73	0.98	2.12	7.48	2.73	0.98
Harms Way	1.52	4.38	2.09	0.98	1.90	5.96	2.44	0.98	1.75	5.24	2.29	0.98	2.07	7.23	2.65	0.97	2.07	7.23	2.65	0.97
Isbell Farms	2.25	14.67	3.80	0.93	2.82	16.59	4.06	0.93	3.31	21.98	4.57	0.9	3.02	25.94	4.00	0.89	2.87	22.86	3.82	0.91
Manderfield	1.57	4.47	2.11	0.98	1.97	6.25	2.50	0.97	1.74	5.17	2.27	0.98	2.04	6.81	2.58	0.96	2.04	6.81	2.58	0.96
Morris Farms	1.62	4.58	2.14	0.97	1.99	6.47	2.54	0.96	1.89	5.85	2.42	0.96	1.96	6.29	2.46	0.91	1.96	6.18	2.45	0.92
Panguitch	1.41	3.42	1.85	0.98	1.71	4.63	2.15	0.97	1.48	3.69	1.92	0.98	1.84	5.29	2.30	0.96	1.94	5.29	2.30	0.96
Sand Hallow	1.39	3.31	1.82	0.99	1.54	3.99	2.00	0.99	1.50	3.84	1.96	0.99	1.58	4.22	2.04	0.99	1.58	4.22	2.04	0.99
Sevilleta	1.35	3.18	1.78	0.99	1.64	4.51	2.12	0.98	1.51	3.78	1.94	0.99	1.78	24.48	3.37	0.92	1.78	24.47	3.37	0.92
Split Mountain	1.71	5.04	2.24	0.99	2.15	7.53	2.74	0.98	1.73	5.06	2.25	0.99	2.24	8.09	2.82	0.98	2.24	8.09	2.82	0.98
Spooky	1.51	4.16	2.04	0.99	1.82	5.55	2.35	0.98	1.66	4.83	2.19	0.98	1.87	5.74	2.38	0.98	1.87	5.74	2.38	0.98
Sudduth Farms	1.42	3.40	1.84	0.98	1.67	4.50	2.12	0.97	1.48	3.65	1.91	0.98	1.62	4.37	2.07	0.97	1.62	4.37	2.07	0.97
Tule Valley	1.55	4.55	2.13	0.99	2.14	7.75	2.78	0.98	1.71	5.13	2.26	0.99	2.17	7.86	2.78	0.98	2.17	7.86	2.78	0.98
Vermillion	1.39	3.67	1.91	0.98	2.10	62.61	5.92	0.71	1.59	4.48	2.11	0.98	1.88	9.17	2.81	0.95	2.00	6.84	2.59	0.97

Table 7.4 Comparisons of the applied methods in different stations at the depth 4 inch

Data	Depth -4																			
	SVR				MLR				MLPReg				ARIMAX				SARIMAX			
	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²
Cave Valley	1.37	3.44	1.85	0.99	1.78	5.36	2.31	0.98	1.58	4.25	2.06	0.99	1.82	5.56	2.34	0.98	1.82	5.56	2.34	0.98
Dee River Ranch	1.78	5.07	2.25	0.97	1.82	5.20	2.28	0.97	1.77	4.94	2.22	0.97	2.03	6.59	2.53	0.94	2.02	6.58	2.53	0.94
Grause Creek	1.50	3.90	1.97	0.98	1.84	5.56	2.36	0.98	1.68	4.63	2.15	0.98	1.89	5.85	2.41	0.97	1.89	5.85	2.41	0.97
Hals Canyon	1.29	2.94	1.71	0.99	1.66	4.76	2.18	0.99	1.37	1.39	1.78	0.99	1.75	5.25	2.28	0.99	1.75	5.25	2.28	0.99
Harms Way	1.34	3.28	1.81	0.99	1.72	4.76	2.18	0.98	1.52	4.03	2.01	0.99	1.82	5.52	2.31	0.98	1.82	5.52	2.31	0.98
Isbell Farms	1.66	4.70	2.17	0.98	2.05	6.80	2.61	0.97	1.79	5.17	2.27	0.98	2.27	8.87	2.85	0.95	2.26	9.01	2.85	0.95
Manderfield	1.66	5.03	2.24	0.98	2.04	6.73	2.59	0.97	1.90	6.18	2.48	0.97	2.17	7.82	2.75	0.96	2.17	7.83	2.76	0.96
Morris Farms	1.42	3.47	1.86	0.98	1.64	4.51	2.12	0.98	1.53	3.93	1.98	0.98	1.71	4.71	2.12	0.94	1.71	4.72	2.12	0.94
Panguitch	1.33	2.98	1.73	0.98	1.57	3.94	1.98	0.98	1.39	3.16	1.78	0.98	1.69	4.56	2.13	0.97	1.69	4.56	2.13	0.97
Sand Hallow	1.22	2.58	1.61	0.99	1.34	3.05	1.74	0.99	1.43	3.34	1.82	0.99	1.36	3.17	1.75	0.99	1.36	3.17	1.75	0.99
Sevilleta	1.18	2.34	1.53	0.99	1.38	3.22	1.79	0.99	1.30	2.83	1.68	0.99	1.49	7.63	2.31	0.98	1.49	7.63	2.31	0.98
Split Mountain	1.52	4.11	2.02	0.99	2.01	6.56	2.56	0.98	1.68	4.60	2.14	0.99	2.16	7.59	2.73	0.98	2.15	7.57	2.72	0.98
Spooky	1.41	3.38	1.84	0.99	1.69	4.56	2.13	0.99	1.55	3.88	1.97	0.99	1.85	5.46	2.32	0.98	1.85	5.46	2.32	0.98
Sudduth Farms	1.36	3.08	1.75	0.98	1.58	3.99	2.00	0.97	1.43	3.35	1.83	0.98	1.58	4.25	2.04	0.97	1.58	4.25	2.04	0.97
Tule Valley	1.30	3.08	1.75	0.99	1.81	5.66	2.38	0.99	1.45	3.62	1.90	0.99	1.87	6.03	2.43	0.99	1.87	6.03	2.43	0.99
Vermillon	1.40	3.47	1.86	0.99	2.08	85.07	6.54	0.70	1.60	4.29	2.07	0.99	1.94	12.48	3.04	0.94	2.06	9.62	2.93	0.96

Table 7.5 Comparisons of the applied methods in different stations at the depth 2 inch

Data	Depth -2																			
	SVR				MLR				MLPReg				ARIMAX				SARIMAX			
	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE	R ²
Cave Valley	1.54	4.69	2.16	0.99	2.01	7.29	2.70	0.98	1.69	5.31	2.30	0.98	2.08	7.84	2.68	0.97	2.08	7.82	2.68	0.97
Dee River Ranch	1.44	3.16	1.78	0.98	1.53	3.66	1.91	0.98	1.52	3.62	1.90	0.98	1.81	5.26	2.20	0.95	1.81	5.26	2.20	0.95
Grause Creek	1.36	3.06	1.75	0.99	1.67	4.53	2.13	0.99	1.47	3.54	1.88	0.99	1.73	4.85	2.19	0.98	1.73	4.86	2.19	0.98
Hals Canyon	1.48	4.34	2.08	0.99	1.89	6.48	2.54	0.98	1.60	4.79	2.18	0.99	1.96	6.74	2.52	0.98	1.95	6.69	2.51	0.98
Harms Way	1.27	2.78	1.67	0.99	1.58	4.00	2.00	0.99	1.42	3.30	1.82	0.99	1.68	4.64	2.13	0.98	1.68	4.64	2.13	0.98
Isbell Farms	1.49	3.84	1.96	0.99	1.69	4.69	2.16	0.98	1.60	4.24	2.06	0.98	1.95	5.97	2.41	0.97	1.95	5.97	2.41	0.97
Manderfield	1.48	3.80	1.95	0.99	1.80	5.31	2.3	0.98	1.64	4.41	2.10	0.98	2.05	7.01	2.59	0.97	2.05	7.01	2.59	0.97
Morris Farms	1.32	3.06	1.75	0.99	1.55	4.19	2.04	0.98	1.57	4.03	2.00	0.98	1.68	4.40	2.06	0.96	1.68	4.40	2.06	0.96
Panguitch	1.30	2.89	1.70	0.99	1.60	4.04	2.01	0.98	1.37	3.10	1.76	0.98	1.69	4.74	2.16	0.97	1.69	4.74	2.16	0.97
Sand Hallow	1.13	2.30	1.52	0.99	1.27	2.80	1.67	0.99	1.31	2.91	1.70	0.99	1.32	3.17	1.70	0.99	1.32	3.17	1.70	0.99
Sevilleta	1.11	2.03	1.42	0.99	1.29	2.66	1.63	0.99	1.21	2.38	1.54	0.99	1.34	4.41	1.91	0.99	1.32	4.34	1.90	0.99
Split Mountain	1.42	3.58	1.88	0.99	1.85	5.57	2.36	0.99	1.53	3.91	1.98	0.99	2.04	6.70	2.56	0.99	2.04	6.70	2.56	0.99
Spooky	1.42	3.64	1.91	0.99	1.63	4.57	2.14	0.99	1.52	4.13	2.03	0.99	1.71	4.96	2.17	0.99	1.73	5.01	2.19	0.99
Sudduth Farms	1.27	2.62	1.62	0.99	1.43	3.24	1.80	0.98	1.30	2.78	1.66	0.98	1.44	3.44	1.84	0.98	1.46	3.49	1.85	0.98
Tule Valley	1.03	1.86	1.36	0.99	1.50	3.95	1.98	0.99	1.14	2.18	1.47	0.99	1.51	4.03	1.96	0.99	1.51	4.03	1.96	0.99
Vermillon	1.39	3.14	1.77	0.99	2.08	86.08	6.60	0.76	1.58	4.06	2.01	0.99	2.08	11.38	3.05	0.96	2.08	11.38	3.05	0.96

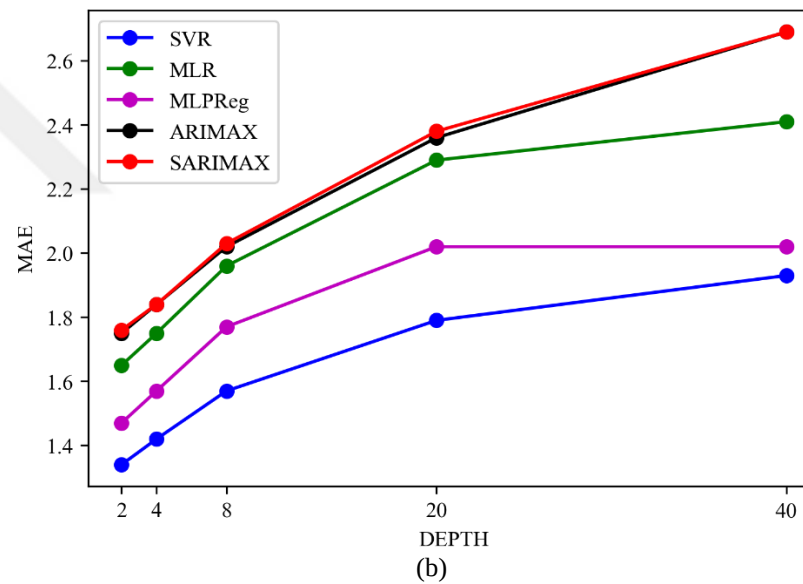
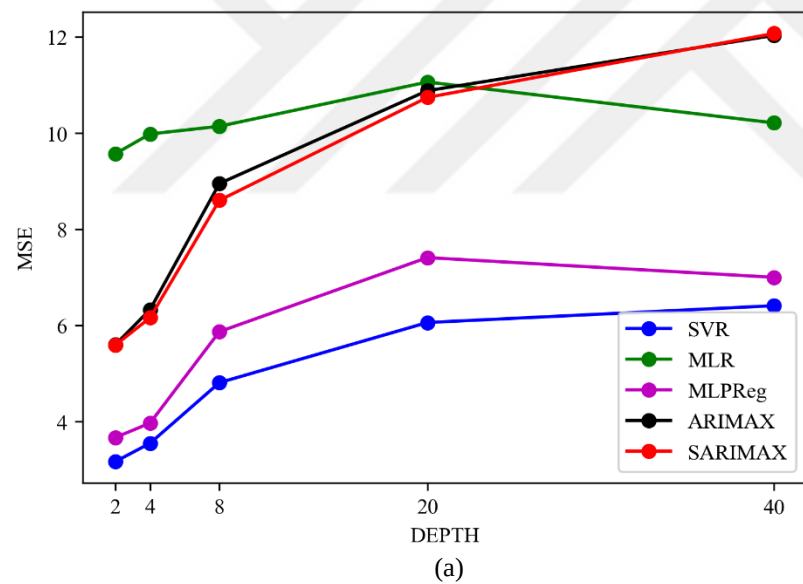


Figure 7.5 Comparisons of the average MSE values (a) and average MAE values (b) of the applied methods according to stations at different depths

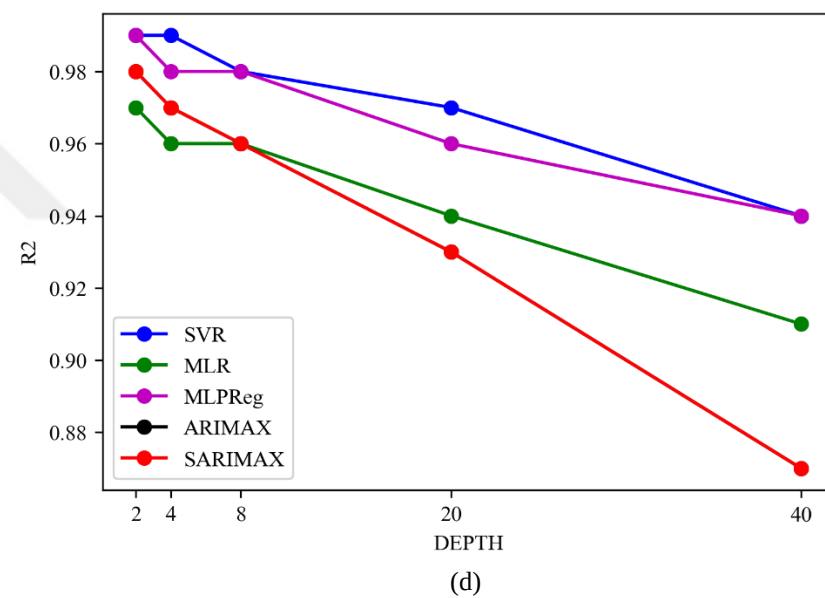
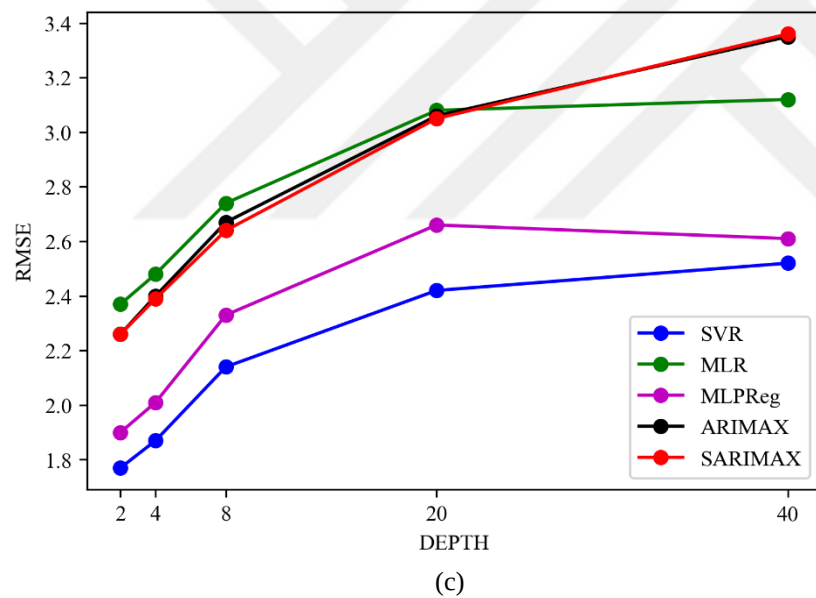


Figure 7.6 Comparisons of the average RMSE values (c) and average R2 values (d) of the applied methods according to stations at different depths

7.2.3 Discussions

In the agricultural environment, the sustainability of farming activities and crop production drastically depends on the quality of the soil and the factors affecting the soil characteristics. Soil temperature is one of the major determinants that its variation directly affects agricultural activities. Necessary precautions can be taken for crop production and the consumption of natural resources can be reduced in agriculture by predicting soil temperature in the early period.

Although a variety of approaches have been implemented and optimized, their consideration for estimating soil temperature was either using univariate time series techniques or applying machine learning methods without dealing with multi-view properties. On the other hand, the soil is affected differently by soil-related parameters and meteorological factors. Therefore, they should be handled differently by dividing the problem domain into two specific views. Furthermore, estimating based on only one factor (the past values of soil temperature) is inefficient because soil temperature highly depends on other factors such as air temperature, vapor pressure, solar radiation, soil moisture, etc., therefore, these parameters should also be taken into account in the model construction.

In this study, the multi-view multi-depth soil temperature prediction method was employed by considering multivariate parameters and splitting the features of the problem into two views as soil-related view and meteorological view. The experimental results showed that values very close to the real soil temperature were obtained at five soil depths in terms of the performance metrics such as MSE, MAE, RMSE, and R^2 with the applied improvements.

As the main returns of this study, the following subjects can be listed:

- The proposed framework, *multi-view multi-depth learning*, has proven to be very efficient and useful when multiple views and depths coexist.

- Two knowledge analysis techniques (supervised machine learning and time series forecasting) were compared to each other under the proposed framework and the machine learning side was superior.
- The multi-view learning framework, which was used for the first time in the estimation of soil temperature, has proven to be effective in this area as well.



CHAPTER EIGHT

CONCLUSION AND FUTURE WORK

8.1 Conclusion

In this thesis, several environmental data were analyzed and assessed using machine learning and data mining techniques, novel methods were developed for relevant case studies, and useful solutions were offered. Six environmental case studies were taken into consideration under five data mining and machine learning tasks (classification, regression, clustering, association rule mining, and time series). The proposed methods used in the case studies are as follows:

- eBagging: Enhanced Bagging
- STST: Soil Temperature Prediction via Self-training
- MV-MTC: Majority Voting Based Multi-Task Clustering
- WARM: Weighted Association Rule Mining
- MV-MD-STP: Multi-View Multi-Depth Soil Temperature Prediction

In the first study, a new ensemble learning strategy named *Enhanced Bagging* (*eBagging*) was proposed to classify air quality monitoring stations in Turkey as belonging to one of the categories of industrial, rural, or urban. The remarkable part of *eBagging* is giving hard-to-classify instances that are misclassified by the prior learner a higher chance for selection. The experimental results showed that it achieved superior performance when compared to single learners and other ensemble learning methods (bagging, AdaBoost, random subspace, and voting).

In another case study, soil temperature prediction of Izmir, Turkey was handled using a new self-training method, named *STST* in a regression problem. The presented method succeeded results with the lowest error when compared to other methods by constructing a prediction model through increasing the sample size of the training set with unlabeled data when pre-labeled data are scarce.

As a clustering case study, Turkey's air quality monitoring network was taken into consideration to determine the optimal clustering application for monitoring stations based on air pollutant factors. For this purpose, a new method called *MV-MTC* was introduced by applying majority voting to take the common decision coming from different pollutant sources. The best results were obtained by the proposed *MV-MTC* method in the experimental studies.

In another study, the relationship among different air pollution factors and meteorological parameters was examined using a new version of the *WARM* method by assigning weights to item values according to their effects on air pollution. The results were compared with the traditional *ARM* method under different support and confidence values. As a result, the most important association rules were obtained and it was shown that *WARM* discovered the rules that could not be explored by *ARM* with the Apriori method.

In the next study, soil temperature prediction was also managed by applying a new methodology named *Multi-view Multi-depth Learning* through representing the problem space in two different views (soil-related view and meteorological view). The common decision from different views was assigned as the final temperature value at each soil depth by taking their average. Machine learning and time-series methods were implemented under this proposed framework. Very close results to the real values of soil temperature were obtained as the output of the experimental studies.

In brief, the proposed methods in this thesis were tested in real-world environmental datasets and they were proven to be the best by obtaining the most accurate solutions with the least error.

8.2 Future Work

In the future, the following issues can be addressed:

- Multi-strategy ensemble learning that combines several ensemble strategies can be considered, instead of a single ensemble strategy as the proposed method (eBagging) in Chapter 3.
- Text mining, web mining, and process mining have been used in many engineering fields. However, there is very limited usage of them in environmental engineering. Future research can focus on these subjects.
- Some ontologies can be developed for the environmental domain. It is believed that the future environmental data mining studies will be supported by the ontologies to extract semantic relationships, to improve accuracy, and to develop better decision support systems.
- The conducted studies have focused on air and soil data. The validity of the proposed methods can also be tested on the other environmental data (water, energy, ecology, etc.) in future studies by adapting the dynamics inside the models for the relevant parameters of the problem domain.
- The proposed model (STST) in Chapter 4 may be customized and updated to estimate the soil moisture, which is another important parameter in plant production.
- Seasonal changes in air quality data can also be observed instead of using yearly data.
- The severity of air quality may be clustered based on the impact on the health issue or the potential damage to the environment.
- A new study can be conducted to investigate the main causes of pollution by utilizing data such as fuel, exhaust, and industrial waste.
- PM_{2.5} is one of the most dangerous particulate matters. However, in Turkey, there is a missing data problem considering the measurements of PM_{2.5} particular matter. Likewise, the same case is also valid for CO pollutants so it is not dealt with in this study. If the study is extended to be applied to other countries, new air pollutants can also be handled.
- The proposed model (multi-view multi-depth soil temperature prediction) in Chapter 7 can be slightly modified by the inclusion of weights to increase or decrease the effect of different views on the prediction results.

REFERENCES

- Acharya, A., Mooney, R. J., & Ghosh, J. (2017). Active multitask learning using supervised and shared latent topics. In A. Pal, & S. K. Pal, (Eds.). *Pattern recognition and big data* (75-112). Singapore: World Scientific Publishing.
- Alatrasta-Salas, H., Azé, J., Bringay, S., Cernesson, F., Selmaoui-Folcher, N., & Teisseire, M. (2015). A knowledge discovery process for spatiotemporal data: Application to river water quality monitoring. *Ecological Informatics*, 26, 127-139.
- Alizamir, M., Kisi, O., Ahmed, A. N., Mert, C., Fai, C. M., Kim, S., et al. (2020). Advanced machine learning model for better prediction accuracy of soil temperature at different depths. *Plos One*, 15 (4), e0231055.
- Alsouda, Y., Pllana, S., & Kurti, A. (2019). IoT-based urban noise identification using machine learning: Performance of SVM, KNN, Bagging, and Random Forest. *International Conference on Omni-Layer Intelligent Systems (COINS)*, 62-67.
- Alvarez-Guerra, E., Molina, A., Viguri, J. R., & Alvarez-Guerra, M. (2011). A SOM-based methodology for classifying air quality monitoring stations. *Environmental Progress & Sustainable Energy*, 30 (3), 424-438.
- Al-Hajj, R., Assi, A., & Fouad, M. M. (2019). Stacking-based ensemble of support vector regressors for one-day ahead solar irradiance prediction. *International Conference on Renewable Energy Research and Applications (ICRERA)*, 428-433.
- Anton, C. A., Matei, O., & Avram, A. (2019). Collaborative data mining in agriculture for prediction of soil moisture and temperature. *Computer Science On-line Conference (CSOC)*, 141-151.
- Arabameri, A., Yamani, M., Pradhan, B., Melesse, A., Shirani, K., & Bui, D. T. (2019). Novel ensembles of COPRAS multi-criteria decision-making with logistic

regression, boosted regression tree, and random forest for spatial prediction of gully erosion susceptibility. *Science of The Total Environment*, 688, 903-916.

Araújo, J. P. C., Palha, C. A., Martins, F. F., Silva, H. M., & Oliveira, J. R. (2019). Estimation of energy consumption on the tire-pavement interaction for asphalt mixtures with different surface properties using data mining techniques. *Transportation Research Part D: Transport and Environment*, 67, 421-432.

Araya, D. B., Grolinger, K., ElYamany, H. F., Capretz, M. A., & Bitsuamlak, G. (2017). An ensemble learning framework for anomaly detection in building energy consumption. *Energy and Buildings*, 144, 191-206.

Atacak, I., Arici, N., & Guner, D. (2017). Modelling and evaluating air quality with fuzzy logic algorithm-Ankara-Cebeci sample. *International Journal of Intelligent Systems and Applications in Engineering*, 5, 263-268.

Barkett, M. O., & Akün, E. (2018). Heavy metal contents of contaminated soils and ecological risk assessment in abandoned copper mine harbor in Yedidalga, Northern Cyprus. *Environmental Earth Sciences*, 77 (10), 378.

Barrero, M. A., Orza, J. A. G., Cabello, M., & Cantón, L. (2015). Categorisation of air quality monitoring stations by evaluation of PM10 variability. *Science of The Total Environment*, 524, 225-236.

Belkin, M., Niyogi, P., & Sindhvani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7 (Nov), 2399-2434.

Bhavithra, J., & Saradha, A. (2019). Personalized web page recommendation using case-based clustering and weighted association rule mining. *Cluster Computing*, 22 (3), 6991-7002.

- Bilgili, M. (2010). Prediction of soil temperature using regression and artificial neural network models. *Meteorology and Atmospheric Physics*, 110 (1-2), 59-70.
- Biswal, M., & Dash, P. K. (2013). Detection and characterization of multiple power quality disturbances with a fast S-transform and decision tree based classifier. *Digital Signal Processing*, 23 (4), 1071-1083.
- Bonakdari, H., Moeeni, H., Ebtehaj, I., Zeynoddin, M., Mahoammadian, A., Gharabaghi, B., et al. (2019). New insights into soil temperature time series modeling: linear or nonlinear?. *Theoretical and Applied Climatology*, 135 (3-4), 1157-1177.
- Bougoudis, I., Demertzis, K., & Iliadis, L. (2016). HISYCOL a hybrid computational intelligence system for combined machine learning: the case of air pollution modeling in Athens. *Neural Computing and Applications*, 27 (5), 1191-1206.
- Bravo-Moncayo, L., Naranjo, J. L., García, I. P., & Mosquera, R. (2017). Neural based contingent valuation of road traffic noise. *Transportation Research Part D: Transport and Environment*, 50, 26-39.
- Budka, M., Gabrys, B., & Ravagnan, E. (2010). Robust predictive modelling of water pollution using biomarker data. *Water Research*, 44 (10), 3294-3308.
- Cagcag, O., Yolcu, U., Egrioglu, E., & Aladag, C. H. (2013). A novel seasonal fuzzy time series method to the forecasting of air pollution data in Ankara. *American Journal of Intelligent Systems*, 3, 13-19.
- Cagliero, L., Garza, P., Pasini, A., & Baralis, E. M. (2018). Additional reviewer assignment by means of weighted association rules. *IEEE Transactions on Emerging Topics in Computing*, 1-12.

- Cao, B. Y., Ma, S. Q., & Cao, H. H. (Eds.). (2014). *Ecosystem assessment and fuzzy systems management*. Switzerland: Springer.
- Chati, Y. S., & Balakrishnan, H. (2016). Statistical modeling of aircraft engine fuel flow rate. *30th Congress of the International Council of the Aeronautical Science*, 1-10.
- Chen, J., Tang, P., Rakstad, T., Patrick, M., & Zhou, X. (2020). Augmenting a deep-learning algorithm with canal inspection knowledge for reliable water leak detection from multispectral satellite images. *Advanced Engineering Informatics*, 46, 101161.
- Chen, L., Cai, Y., Ding, Y., Lv, M., Yuan, C., & Chen, G. (2016). Spatially fine-grained urban air quality estimation using ensemble semi-supervised learning and pruning. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 1076-1087.
- Chen, L. J., Ho, Y. H., Hsieh, H. H., Huang, S. T., Lee, H. C., & Mahajan, S. (2017). ADF: An anomaly detection framework for large-scale PM2. 5 sensing systems. *IEEE Internet of Things Journal*, 5 (2), 559-570.
- Chou, J. S., Ho, C. C., & Hoang, H. S. (2018). Determining quality of water in reservoir using machine learning. *Ecological Informatics*, 44, 57-75.
- Chou, J. S., Yang, K. H., & Lin, J. Y. (2016). Peak shear strength of discrete fiber-reinforced soils computed by machine learning and metaensemble methods. *Journal of Computing in Civil Engineering*, 30 (6), 04016036.
- Chougule, A., Jha, V. K., & Mukhopadhyay, D. (2019). Crop suitability and fertilizers recommendation using data mining techniques. In C. Panigrahi, A. Pujari, S. Misra, B. Pati, K. C. Li, (Eds.). *Progress in advanced computing and intelligent engineering* (205-213). Singapore: Springer.

- Contreras, J., Espinola, R., Nogales, F. J., & Conejo, A. J. (2003). ARIMA models to predict next-day electricity prices. *IEEE Transactions on Power Systems*, 18 (3), 1014-1020.
- Coutinho, F. H., Thompson, C. C., Cabral, A. S., Paranhos, R., Dutilh, B. E., & Thompson, F. L. (2019). Modelling the influence of environmental parameters over marine planktonic microbial communities using artificial neural networks. *Science of the Total Environment*, 677, 205-214.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39 (1), 1-22.
- Deng, W., & Wang, G. (2017). A novel water quality data analysis framework based on time-series data mining. *Journal of Environmental Management*, 196, 365-375.
- Dhanalakshmi, P., & Porkodi, R. (2018). Extraction of association rules from tobacco smoke effect on the placenta microarray dataset using gene ontology based optimized association rule mining. *International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 1-9.
- Di, Z., Chang, M., Guo, P., Li, Y., & Chang, Y. (2019). Using real-time data and unsupervised machine learning techniques to study large-scale spatio-temporal characteristics of wastewater discharges and their influence on surface water quality in the Yangtze River basin. *Water*, 11 (6), 1268.
- Dincer, N. G., & Akkuş, Ö. (2018). A new fuzzy time series model based on robust clustering for forecasting of air pollution. *Ecological Informatics*, 43, 157-164.
- Dos Reis, A. A., Carvalho, M. C., De Mello, J. M., Gomide, L. R., Ferraz Filho, A. C., & Junior, F. W. A. (2018). Spatial prediction of basal area and volume in

- Eucalyptus stands using Landsat TM data: an assessment of prediction methods. *New Zealand Journal of Forestry Science*, 48 (1), 1.
- Dwevedi, A., Kumar, P., Kumar, P., Kumar, Y., Sharma, Y. K., & Kayastha, A. M. (2017). Soil sensors: Detailed insight into research updates, significance, and future prospects. In A. M. Grumezescu, (Ed.). *New pesticides and soil sensors* (561-594). San Diego: Academic Press.
- Eibe, F., Hall, M. A., & Witten, I. H. (2016). *The WEKA workbench. Online appendix for data mining: practical machine learning tools and techniques* (4th ed.). San Francisco: Morgan Kaufmann.
- Fan, C., Xiao, F., & Wang, S. (2014). Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques. *Applied Energy*, 127, 1-10.
- Fang, M., & Tao, D. (2015). Active multi-task learning via bandits. *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 505-513.
- Farajidavar, N., Kolozali, S., & Barnaghi, P. (2017). *A deep multi-view learning framework for city event extraction from twitter data streams*. Retrieved March 21, 2020, from <https://arxiv.org/abs/1705.09975>.
- Fernandes, J. A., Irigoien, X., Goikoetxea, N., Lozano, J. A., Inza, I., Pérez, A., et al. (2010). Fish recruitment prediction, using robust supervised classification methods. *Ecological Modelling*, 221 (2), 338-352.
- Flemming, J., Stern, R., & Yamartino, R. J. (2005). A new air quality regime classification scheme for O₃, NO₂, SO₂ and PM₁₀ observations sites. *Atmospheric Environment*, 39 (33), 6121-6129.

- Franceschi, F., Cobo, M., & Figueredo, M. (2018). Discovering relationships and forecasting PM₁₀ and PM_{2.5} concentrations in Bogotá, Colombia, using artificial neural networks, principal component analysis, and k-means clustering. *Atmospheric Pollution Research*, 9 (5), 912-922.
- Gaál, M., Moriondo, M., & Bindi, M. (2012). Modelling the impact of climate change on the Hungarian wine regions using random forest. *Applied Ecology and Environmental Research*, 10 (2), 121-140.
- Gao, Z., Li, S. H., Zhang, G. T., Zhu, Y. J., Wang, C., & Zhang, H. (2017). Evaluation of regularized multi-task learning algorithms for single/multi-view human action recognition. *Multimedia Tools and Applications*, 76 (19), 20125-20148.
- Gómez-Losada, Á. (2017). Clustering air monitoring stations according to background and ambient pollution using hidden Markov models and multidimensional scaling. In F. Palumbo, A. Montanari, & M. Vichi, (Eds.). *Data science: Innovative developments in data analysis and clustering* (123-132). Switzerland: Springer.
- Granata, F., & de Marinis, G. (2017). Machine learning methods for wastewater hydraulics. *Flow Measurement and Instrumentation*, 57, 1-9.
- Guo, C., Chen, Y., Liu, H., Lu, Y., Qu, X., Yuan, H., et al. (2019) Modelling fish communities in relation to water quality in the impounded lakes of China's South-to-North Water Diversion Project. *Ecological Modelling*, 397, 25-35.
- Güler, N., & İşçi, Ö. G. (2016). The regional prediction model of PM₁₀ concentrations for Turkey. *Atmospheric Research*, 180, 64-77.
- Gyenesi, A. (2000). Mining weighted association rules for fuzzy quantitative items. *European Conference on Principles of Data Mining and Knowledge Discovery*, 416-423.

- Halmy, M. W. A., & Gessler, P. E. (2015). The application of ensemble techniques for land-cover classification in arid lands. *International Journal of Remote Sensing*, 36 (22), 5613-5636.
- Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques* (3rd ed.). USA: Morgan Kaufmann.
- Harrou, F., Dairi, A., Sun, Y., & Senouci, M. (2018). Reliable detection of abnormal ozone measurements using an air quality sensors network. *IEEE International Conference on Environmental Engineering (EE)*, 1-5.
- He, D., Le, B. T., Xiao, D., Mao, Y., Shan, F., & Ha, T. T. L. (2019). Coal mine area monitoring method by machine learning and multispectral remote sensing images. *Infrared Physics & Technology*, 103, 103070.
- He, J., & Lawrence, R. (2011). A graph based framework for multi-task multi-view learning. *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 25-32.
- He, Y., Zhu, C., He, Z., Gu, C., & Cui, J. (2017). Big data oriented root cause identification approach based on Axiomatic domain mapping and weighted association rule mining for product infant failure. *Computers & Industrial Engineering*, 109, 253-265.
- Hillel, D. (2005). Thermal properties and processes. In D. Hillel, J. H. Hatfield, D. S. Powlson, C. Rosenzweig, K. M. Scow, M. J. Singer, et al. (Eds.). *Encyclopedia of Soils in the Environment*, (156-163). Amsterdam: Academic Press.
- Ho, J. Y., Afan, H. A., El-Shafie, A. H., Koting, S. B., Mohd, N. S., Jaafar, W. Z. B., et al. (2019). Towards a time and cost effective approach to water quality index class prediction. *Journal of Hydrology*, 575, 148-165.

- Holmes, G., Donkin, A., & Witten, I. H. (1994). Weka: A machine learning workbench. *Proceedings of ANZIIS'94-Australian New Zealand Intelligent Information Systems Conference*, 357-361.
- Honorio, J., & Samaras, D. (2010). Multi-task learning of Gaussian graphical models. *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 447-454.
- Hui, K., Chen, C., & Zhan, Y. (2019). A novel approach to classify video scenes via mining weighted association rules of actions in temporal sequence. *Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing*, 145-152.
- Ignaccolo, R., Ghigo, S., & Giovenali, E. (2008). Analysis of air quality monitoring networks by functional clustering. *Environmetrics*, 19 (7), 672-686.
- Indhumathy, M., Nabhan, A. R., & Arumugam, S. (2018). A weighted association rule mining method for predicting HCV-human protein interactions. *Current Bioinformatics*, 13 (1), 73-84.
- Isikli, E., Ustundag, A., & Cevikcan, E. (2015). The effects of environmental risk factors on city life cycle: A link analysis. *Human and Ecological Risk Assessment: An International Journal*, 21 (5), 1379-1394.
- Ji, W., Liu, Y., & Zhen, J. Q. (2020). Prediction of soil humidity based on random weight Particle Swarm Optimized Extreme Learning Machine. *Journal of Physics: Conference Series*, 1486, 042043.
- Ji, X., Shang, X., Dahlgren, R. A., & Zhang, M. (2017). Prediction of dissolved oxygen concentration in hypoxic river systems using support vector machine: a case study of Wen-Rui Tang River, China. *Environmental Science and Pollution Research*, 24 (19), 16062-16076.

- Jia, X., Fu, T., Hu, B., Shi, Z., Zhou, L., & Zhu, Y. (2020). Identification of the potential risk areas for soil heavy metal pollution based on the source-sink theory. *Journal of Hazardous Materials*, 393, 122424.
- Jia, X., Hu, B., Marchant, B. P., Zhou, L., Shi, Z., & Zhu, Y. (2019). A methodological framework for identifying potential sources of soil heavy metal pollution based on machine learning: A case study in the Yangtze Delta, China. *Environmental Pollution*, 250, 601-609.
- Joly, M., & Peuch, V. H. (2012). Objective classification of air quality monitoring sites over Europe. *Atmospheric Environment*, 47, 111-123.
- Jovanović, R. Ž., Sretenović, A. A., & Živković, B. D. (2015). Ensemble of various neural networks for prediction of heating energy consumption. *Energy and Buildings*, 94, 189-199.
- Karimipour, F., & Kanani-Sadat, Y. (2016). Mapping the vulnerability of asthmatic allergy prevalence based on environmental characteristics through fuzzy spatial association rule mining. *Journal of Environmental Informatics*, 28 (1), 1-10.
- Kaya, K., & Ögüdücü, Ş. G. (2018). A binary classification model for PM₁₀ levels. *Proceedings of the 3rd International Conference on Computer Science and Engineering (UBMK)*, 361-366.
- Kaya, M., & Alhajj, R. (2004). Genetic algorithms based optimization of membership functions for fuzzy weighted association rules mining. *Proceedings of the Ninth International Symposium on Computers and Communications*, 110-115.
- Khan, M. S., Muyeba, M., & Coenen, F. (2008). A weighted utility framework for mining association rules. *Second UKSIM European Symposium on Computer Modeling and Simulation*, 87-92.

- Knudby, A., Brenning, A., & LeDrew, E. (2010). New approaches to modelling fish–habitat relationships. *Ecological Modelling*, 221 (3), 503-511.
- Kocev, D., & Džeroski, S. (2013). Habitat modeling with single-and multi-target trees and ensembles. *Ecological Informatics*, 18, 79-92.
- Kurt, A., & Oktay, A. B. (2010). Forecasting air pollutant indicator levels with geographic models 3 days in advance using neural networks. *Expert Systems with Applications*, 37 (12), 7986-7992.
- Kühnlein, M., Appelhans, T., Thies, B., & Nauss, T. (2014). Improving the accuracy of rainfall rates from optical satellite sensors with machine learning—A random forests-based approach applied to MSG SEVIRI. *Remote Sensing of Environment*, 141, 129-143.
- Kwon, Y., Lee, H., & Lee, H. (2018). Implication of the cluster analysis using greenhouse gas emissions of Asian countries to climate change mitigation. *Mitigation and Adaptation Strategies for Global Change*, 23 (8), 1225-1249.
- Lakshmi, K. S., & Vadivu, G. (2019). A novel approach for disease comorbidity prediction using weighted association rule mining. *Journal of Ambient Intelligence and Humanized Computing*, 1-8.
- Lazaric, A., & Ghavamzadeh, M. (2010). Bayesian multi-task reinforcement learning. *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 599-606.
- Lei, K. S., & Wan, F. (2012). Applying ensemble learning techniques to ANFIS for air pollution index prediction in Macau. *International Symposium on Neural Networks (ISNN)*, 509-516.

- Li, H., Liao, X., & Carin, L. (2009). Multi-task reinforcement learning in partially observable stochastic environments. *Journal of Machine Learning Research*, 10 (5), 1131-1186.
- Li, R., Cui, L., Meng, Y., Zhao, Y., & Fu, H. (2019). Satellite-based prediction of daily SO₂ exposure across China using a high-quality random forest-spatiotemporal Kriging (RF-STK) model for health risk assessment. *Atmospheric Environment*, 208, 10-19.
- Li, Y. (2005). A user-guided association rules mining method and its application. *The Fifth International Conference on Computer and Information Technology (CIT)*, 150-156.
- Li, T., Li, Y., An, D., Han, Y., Xu, S., Lu, Z., et al. (2019). Mining of the association rules between industrialization level and air quality to inform high-quality development in China. *Journal of Environmental Management*, 246, 564-574.
- Lima, A. R., Cannon, A. J., & Hsieh, W. W. (2013). Nonlinear regression in environmental sciences by support vector machines combined with evolutionary strategy. *Computers & Geosciences*, 50, 136-144.
- Liu, A., Lu, Y., Nie, W., Su, Y., & Yang, Z. (2016). HEp-2 cells classification via clustered multi-task learning. *Neurocomputing*, 195, 195-201.
- Liu, Q., Liao, X., & Carin, L. (2007). Semi-Supervised multitask learning. *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS)*, 937-944.
- Liu, Y., Zheng, Y., Liang, Y., Liu, S., & Rosenblum, D. S. (2016). Urban water quality prediction based on multi-task multi-view learning. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 2576–2582.

- Lu, J., & Qian, Z. (1999). Mining boolean and general fuzzy weighted association rules database. *Mathematica Applicata*, 12, 38-43.
- Lu, W. Z., He, H. D., & Dong, L. Y. (2011). Performance assessment of air quality monitoring networks using principal component analysis and cluster analysis. *Building and Environment*, 46 (3), 577-583.
- Lu, X., Li, X., & Mou, L. (2014). Semi-supervised multitask learning for scene recognition. *IEEE Transactions on Cybernetics*, 45 (9), 1967-1976.
- Luo, Q., & Kathuria, A. (2013). Modelling the response of wheat grain yield to climate change: a sensitivity analysis. *Theoretical and Applied Climatology*, 111 (1-2), 173-182.
- Lv, M., Li, Y., Chen, L., & Chen, T. (2019). Air quality estimation by exploiting terrain features and multi-view transfer semi-supervised regression. *Information Sciences*, 483, 82-95.
- Mad Saad, S., Andrew, A. M., Md Shakaff, A. Y., Mat Dzahir, M. A., Hussein, M., Mohamad, M., et al. (2017). Pollutant recognition based on supervised machine learning for indoor air quality monitoring systems. *Applied Sciences*, 7 (8), 823.
- Maji, K. J., Dikshit, A. K., & Deshpande, A. (2016). Classification of air quality monitoring stations using fuzzy similarity measures: a case study. In L. A. Zadeh, A. M. Abbasov, R. R. Yager, S. N. Shahbazova, & M. Reformat, (Eds.). *Recent developments and new direction in soft-computing foundations and applications* (489-501). Switzerland: Springer.
- Mannucci, P. M., & Franchini, M. (2017). Health effects of ambient air pollution in developing countries. *International Journal of Environmental Research and Public Health*, 14 (9), 1048.

- McCallum, A., Nigam, K., & Ungar, L. H. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM:KDD)*, 169-178.
- Mehdizadeh, S., Fathian, F., Safari, M. J. S., & Khosravi, A. (2020). Developing novel hybrid models for estimation of daily soil temperature at various depths. *Soil and Tillage Research*, 197, 104513.
- Miskell, G., Pattinson, W., Weissert, L., & Williams, D. (2019). Forecasting short-term peak concentrations from a network of air quality instruments measuring PM_{2.5} using boosted gradient machine models. *Journal of Environmental Management*, 242, 56-64.
- Mohammed, A. A., Yaqub, W., & Aung, Z. (2015). Probabilistic forecasting of solar power: An ensemble learning approach. In R. Neves-Silva, L. C. Jain, R. J. Howlett, (Eds.). *Intelligent decision technologies*, (449-458). Switzerland: Springer.
- Moosa, A., Shabir, H., Ali, H., Darwade, R., & Gite, B. (2018). Predicting solar radiation using machine learning techniques. *2nd International Conference on Intelligent Computing and Control Systems (ICICCS)*, 1693-1699.
- Muñoz-Mas, R., Lopez-Nicolas, A., Martinez-Capel, F., & Pulido-Velazquez, M. (2016). Shifts in the suitable habitat available for brown trout (*Salmo trutta* L.) under short-term climate change scenarios. *Science of the Total Environment*, 544, 686-700.
- Muyeba, M., Khan, M. S., & Coenen, F. (2008). Fuzzy weighted association rule mining with weighted support and confidence framework. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 49-61.

- Nelson, T. A., Coops, N. C., Wulder, M. A., Perez, L., Fitterer, J., Powers, R., et al. (2014). Predicting climate change impacts to the Canadian boreal forest. *Diversity*, 6 (1), 133-157.
- Nguyen, H. T., & Kim, K. H. (2006). Comparison of spatiotemporal distribution patterns of NO₂ between four different types of air quality monitoring stations. *Chemosphere*, 65 (2), 201-212.
- Nourani, V., Kisi, Ö., & Komasi, M. (2011). Two hybrid artificial intelligence approaches for modeling rainfall–runoff process. *Journal of Hydrology*, 402 (1-2), 41-59.
- Onal, A. E., Bayramlar, O. F., Ezirmik, E., Gulle, B. T., Canatar, F., Calik, D., et al. (2017). Evaluation of air quality in the city of Istanbul during the years 2013 and 2015. *Journal of Environmental Science and Engineering*, 6, 465-470.
- Othman, Z. A., Ismail, N., Bakar, A. A., Latif, M. T., & Abdullah, S. M. S. (2018). Rules discovery of high ozone in Klang areas using data mining approach. *International Journal on Advanced Science, Engineering and Information Technology*, 8 (6), 2683-2689.
- Oyen, D., & Lane, T. (2012). Leveraging domain knowledge in multitask Bayesian network structure learning. *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, 1091-1097.
- Pais, M. P., Henriques, S., Batista, M. I., Costa, M. J., & Cabral, H. (2013). Seeking functional homogeneity: a framework for definition and classification of fish assemblage types to support assessment tools on temperate reefs. *Ecological Indicators*, 34, 231-245.

- Parisotto, E., Ba, J. L., & Salakhutdinov, R. (2015). *Actor-mimic: Deep multitask and transfer reinforcement learning*. Retrieved January 8, 2020, from <https://arxiv.org/abs/1511.06342>.
- Peerzada, A. M., & Chauhan, B. S. (2018). Thermal weed control: History, mechanisms, and impacts. In K. Jabran, & B. S. Chauhan, (Eds.). *Non-chemical weed control* (9-31). London: Academic Press.
- Pekel, E. (2020). Estimation of soil moisture using decision tree regression. *Theoretical and Applied Climatology*, 139 (3), 1111-1119.
- Penghui, L., Ewees, A. A., Beyaztas, B. H., Qi, C., Salih, S. Q., Al-Ansari, N., et al. (2020). Metaheuristic optimization algorithms hybridized with artificial intelligence model for soil temperature prediction: novel model. *IEEE Access*, 8, 51884-51904.
- Pucer, J. F., Pirš, G., & Štrumbelj, E. (2018). A Bayesian approach to forecasting daily air-pollutant levels. *Knowledge and Information Systems*, 57 (3), 635-654.
- Qi, Y., Tastan, O., Carbonell, J. G., Klein-Seetharaman, J., & Weston, J. (2010). Semi-supervised multi-task learning for predicting interactions between HIV-1 and human proteins. *Bioinformatics*, 26 (18), i645-i652.
- Qin, S., Liu, F., Wang, C., Song, Y., & Qu, J. (2015). Spatial-temporal analysis and projection of extreme particulate matter (PM₁₀ and PM_{2.5}) levels using association rules: A case study of the Jing-Jin-Ji region, China. *Atmospheric Environment*, 120, 339-350.
- Quinlan, J. (1996). Bagging, boosting, and C4.5. *Proceedings of the 13th National Conference on Artificial Intelligence*, 725-730.

- Qiu, S., Feng, F., Li, Z., Yang, G., Xu, P., & Li, Z. (2019). Data mining based framework to identify rule based operation strategies for buildings with power metering system. *Building Simulation*, 12, 195–205.
- Rahimi, Z., Noferesti, S., & Shamsfard, M. (2019). Applying data mining and machine learning techniques for sentiment shifter identification. *Language Resources and Evaluation*, 53 (2), 279-302.
- Ran, P., Wang, Y., Liu, P., & Li, Z. (2016). A fuzzy association rule mining based approach to identifying SO₂ emission concentration patterns of coal-fired boilers. *Chemical Engineering Transactions*, 52, 631-636.
- Ratolojanahary, R., Ngouna, R. H., Medjaher, K., Junca-Bourié, J., Dauriac, F., & Sebilo, M. (2019). Model selection to improve multiple imputation for handling high rate missingness in a water quality dataset. *Expert Systems with Applications*, 131, 299-307.
- Reder, M., Yürüşen, N. Y., & Melero, J. J. (2018). Data-driven learning framework for associating weather conditions and wind turbine failures. *Reliability Engineering & System Safety*, 169, 554-569.
- Reichart, R., Tomanek, K., Hahn, U., & Rappoport, A. (2008). Multi-Task active learning for linguistic annotations. *Proceedings of Association for Computational Linguistics: Human Language Technology Conference (ACL:HLT)*, 861-869.
- Reid, C. E., Considine, E. M., Watson, G. L., Telesca, D., Pfister, G. G., & Jerrett, M. (2019). Associations between respiratory health and ozone and fine particulate matter during a wildfire event. *Environment International*, 129, 291-298.
- Ren, C., Liang, Y. J., Lu, X. J., & Yan, H. B. (2019). Research on the soil moisture sliding estimation method using the LS-SVM based on multi-satellite fusion. *International Journal of Remote Sensing*, 40 (5-6), 2104-2119.

- Rich, D. Q. (2017). Accountability studies of air pollution and health effects: lessons learned and recommendations for future natural experiment opportunities. *Environment International*, 100, 62-78.
- Rodriguez-Galiano, V., Mendes, M. P., Garcia-Soldado, M. J., Chica-Olmo, M., & Ribeiro, L. (2014). Predictive modeling of groundwater nitrate pollution using Random Forest and multisource variables related to intrinsic and specific vulnerability: A case study in an agricultural setting (Southern Spain). *Science of the Total Environment*, 476, 189-206.
- Sadat, Y. K., Nikaein, T., & Karimipour, F. (2015). Fuzzy spatial association rule mining to analyze the effect of environmental variables on the risk of allergic asthma prevalence. *Geodesy and Cartography*, 41 (2), 101-112.
- Saeidi, A., Hage, J., Khadka, R., & Jansen, S. (2019). Applications of multi-view learning approaches for software comprehension. *The Art, Science, and Engineering of Programming*, 3 (3), 14.
- Salman, A., Jalal, A., Shafait, F., Mian, A., Shortis, M., Seager, J., et al. (2016). Fish species classification in unconstrained underwater environments based on deep learning. *Limnology and Oceanography: Methods*, 14 (9), 570-585.
- Sanikhani, H., Deo, R. C., Yaseen, Z. M., Eray, O., & Kisi, O. (2018). Non-tuned data intelligent model for soil temperature estimation: A new approach. *Geoderma*, 330, 52-64.
- Saraswat, D., Dev, V., & Singh, P. (2018). Analyzing the performance of the Indian Cricket Team using Weighted Association Rule Mining. *International Conference on Computing, Power and Communication Technologies (GUCON)*, 161-164.
- Sha, Z., Tan, X., & Bai, Y. (2014). Localized spatial association: a case study for understanding vegetation successions in a typical grassland ecosystem.

International Conference on Geo-Informatics in Resource Management and Sustainable Ecosystem, 33-45.

Sharma, N., Bajpai, A., & Litoriya, M. R. (2012). Comparison the various clustering algorithms of Weka tools. *Facilities*, 4 (7), 78-80.

Siahpirani, A. F., Ay, F., & Roy, S. (2016). A multi-task graph-clustering approach for chromosome conformation capture data sets identifies conserved modules of chromosomal interactions. *Genome Biology*, 17 (1), 114.

Silva, M. D. A., Trevisan, D. Q., Prata, D. N., Marques, E. E., Lisboa, M., & Prata, M. (2013). Exploring an ichthyoplankton database from a freshwater reservoir in legal amazon. *International Conference on Advanced Data Mining and Applications (ADMA)*, 384-395.

Singh, K. P., Gupta, S., & Rai, P. (2013). Identifying pollution sources and predicting urban air quality using ensemble learning methods. *Atmospheric Environment*, 80, 426-437.

Soares, J., Makar, P. A., Aklilu, Y., & Akingunola, A. (2018). The use of hierarchical clustering for the design of optimized monitoring networks. *Atmospheric Chemistry and Physics*, 18 (9), 6543-6566.

Soares, J., Makar, P., Aklilu, Y. A., & Akingunola, A. (2018). Hierarchical clustering for optimizing air quality monitoring networks. *International Technical Meeting on Air Pollution Modelling and its Application*, 299-303.

Sparke, C. (2018) Environmental noise classification through machine learning. *Proceedings of Acoustics 2018: Hear to Listen (AAS)*, 1-9.

Srivastava, A. N. (2012). Greener aviation with virtual sensors: a case study. *Data Mining and Knowledge Discovery*, 24 (2), 443-471.

- Stojić, A., Stojić, S. S., Reljin, I., Čabarkapa, M., Šoštarić, A., Perišić, M., et al. (2016). Comprehensive analysis of PM₁₀ in Belgrade urban area on the basis of long-term measurements. *Environmental Science and Pollution Research*, 23 (11), 10722-10732.
- Sun, K., & Bai, F. (2008). Mining weighted association rules without preassigned weights. *IEEE Transactions on Knowledge and Data Engineering*, 20 (4), 489-495.
- Şahin, Ü. A., Ucan, O. N., Bayat, C., & Tolluoglu, O. (2011). A new approach to prediction of SO₂ and PM₁₀ concentrations in Istanbul, Turkey: Cellular Neural Network (CNN). *Environmental Forensics*, 12 (3), 253-269.
- Tanweer, S., Mobin, A., & Alam, A. (2016). Environmental noise classification using LDA, QDA and ANN methods. *Indian Journal of Science and Technology*, 9 (33), 1-8.
- Tao, F., Murtagh, F., & Farid, M. (2003). Weighted association rule mining using weighted support and significance framework. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 661-666.
- Taşpınar, F., & Bozkurt, Z. (2014). Application of artificial neural networks and regression models in the prediction of daily maximum PM₁₀ concentration in Düzce, Turkey. *Fresenius Environmental Bulletin*, 23 (10), 2450-2459.
- Tiwari, S., Babbar, R., & Kaur, G. (2018). Performance evaluation of two ANFIS models for predicting water quality Index of River Satluj (India). *Advances in Civil Engineering*, 2018, 1-10.
- Tooke, T. R., Coops, N. C., & Webster, J. (2014). Predicting building ages from LiDAR data with random forests for building energy modeling. *Energy and Buildings*, 68, 603-610.

- Toti, G., Vilalta, R., Lindner, P., Lefer, B., Macias, C., & Price, D. (2016). Analysis of correlation between pediatric asthma exacerbation and exposure to pollutant mixtures with association rule mining. *Artificial Intelligence in Medicine*, 74, 44-52.
- Tsai, W. P., Huang, S. P., Cheng, S. T., Shao, K. T., & Chang, F. J. (2017). A data-mining framework for exploring the multi-relation between fish species and water quality through self-organizing map. *Science of the Total Environment*, 579, 474-483.
- Tuysuzoglu, G., & Birant, D. (2020). Air pollution monitoring in intelligent cities using weighted association rule mining. In *Developing and monitoring smart environments for intelligent cities* (171-197). Pennsylvania: IGI Global.
- Tuysuzoglu, G., & Birant, D. (2020). Enhanced bagging (eBagging): A novel approach for ensemble learning. *International Arab Journal of Information Technology*, 17 (4), 515-528.
- Tuysuzoglu, G., Birant, D., & Pala, A. (2018). Ensemble methods in environmental data mining. In C. Thomas, (Ed.). *Data mining* (1-16). London: IntechOpen.
- Tuysuzoglu, G., Birant, D., & Pala, A. (2019). Majority voting based multi-task clustering of air quality monitoring network in Turkey. *Applied Sciences*, 9 (8), 1610.
- Tuysuzoglu, G., Birant, D., Kut, A., & Pala A. (2020). Classification of air quality network based on meteorological and pollutant factors. *International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 1-6.

- Wang, B., Yan, Z., Lu, J., Zhang, G., & Li, T. (2018). Deep multi-task learning for air quality prediction. *International Conference on Neural Information Processing*, 93-103.
- Wang, B. Y., & Zhang, S. M. (2003). A mining algorithm for fuzzy weighted association rules. *Proceedings of the International Conference on Machine Learning and Cybernetics*, 2495-2499.
- Wang, C., Zhao, L., Sun, W., Xue, J., & Xie, Y. (2018). Identifying redundant monitoring stations in an air quality monitoring network. *Atmospheric Environment*, 190, 256-268.
- Wang, F., Li, K., Duić, N., Mi, Z., Hodge, B. M., Shafie-khah, M., et al. (2018). Association rule mining based quantitative analysis approach of household characteristics impacts on residential electricity consumption patterns. *Energy Conversion and Management*, 171, 839-854.
- Wang, J., Li, H., Huang, J., & Su, C. (2016). Association rules mining based analysis of consequential alarm sequences in chemical processes. *Journal of Loss Prevention in the Process Industries*, 41, 178-185.
- Wang, W., Yang, J., & Yu, P. S. (2000). Efficient mining of weighted association rules (WAR). *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 270-274.
- Waring, R. H., & Schlesinger, W. H. (1985). *Forest ecosystems: analysis at multiples scales* (3rd ed.). United States: Elsevier.
- Wen, J., Zhong, M., & Wang, Z. (2015). Activity recognition with weighted frequent patterns mining in smart environments. *Expert Systems with Applications*, 42 (17-18), 6423-6432.

- Williams, B. M. (2001). Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. *Transportation Research Record*, 1776 (1), 194-200.
- Wilson, A., Fern, A., Ray, S., & Tadepalli, P. (2007). Multi-task reinforcement learning: a hierarchical Bayesian approach. *Proceedings of the 24th international conference on Machine learning (ICML)*, 1015-1022.
- Wu, H., Li, H., Jiang, M., Chen, C., Lv, Q., & Wu, C. (2017). Identify high-quality protein structural models by enhanced k-means. *BioMed Research International*, 2017, 7294519.
- Wu, X., Wang, Q., Liu, Y., & Li, Y. (2017). Data-driven regional analysis of urban atmosphere pollution based on density clustering. *IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 412-417.
- Xing, Y. F., Xu, Y. H., Shi, M. H., & Lian, Y. X. (2016). The impact of PM_{2.5} on the human respiratory system. *Journal of Thoracic Disease*, 8 (1), E69.
- Xu, J., Wang, A., Schmidt, N., Adams, M., & Hatzopoulou, M. (2020). A gradient boost approach for predicting near-road ultrafine particle concentrations using detailed traffic characterization. *Environmental Pollution*, 265, 114777.
- Xue, Y., Liao, X., Carin, L., & Krishnapuram, B. (2007). Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8 (Jan), 35-63.
- Yan, Y., Ricci, E., Subramanian, R., Lanz, O., & Sebe, N. (2013). No matter where you are: flexible graph-guided multi-task learning for multi-view head pose classification under target motion. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1177-1184.

- Yan, Y., Ricci, E., Liu, G., & Sebe, N. (2015). Egocentric daily activity recognition via multitask clustering. *IEEE Transactions on Image Processing*, 24 (10), 2984-2995.
- Yang, M., Nelson, F. E., Shiklomanov, N. I., Guo, D., Wan, G. (2010). Permafrost degradation and its environmental effects on the Tibetan Plateau: A review of recent research. *Earth-Science Reviews*, 103 (1-2), 31-44.
- Yang, G., Huang, J., & Li, X. (2018). Mining sequential patterns of PM2.5 pollution in three zones in China. *Journal of Cleaner Production*, 170, 388-398.
- Yang, H., Zhou, J., Feng, J., Zhai, S., Chen, W., Liu, J., et al. (2019). Ditch-buried straw return: A novel tillage practice combined with tillage rotation and deep ploughing in rice-wheat rotation systems. *Advances in Agronomy*, 154, 257-290.
- Yang, J., Wang, J., Zheng, Y., Lei, M., Yang, J., Wan, X., et al. (2018). Method for identifying outliers of soil heavy metal data. *Environmental Science and Pollution Research*, 25 (13), 12868-12875.
- Yilmaz, T., Özbek, A., Yilmaz, A., & Büyükalaca, O. (2009). Influence of upper layer properties on the ground temperature distribution. *Isi Bilimi ve Teknigi Dergisi/Journal of Thermal Science & Technology*, 29 (2), 43-51.
- Yu, Z., Bedig, A., Montalto, F., & Quigley, M. (2018). Automated detection of unusual soil moisture probe response patterns with association rule learning. *Environmental Modelling & Software*, 105, 257-269.
- Yuan, Y., Xun, G., Jia, K., & Zhang, A. (2018). A multi-view deep learning framework for EEG seizure detection. *IEEE Journal of Biomedical and Health Informatics*, 23 (1), 83-94.

- Yue, J. S., Tsang, E., Yeung, D., & Shi, D. (2000). Mining fuzzy association rules with weighted items. *IEEE International Conference On Systems, Man and Cybernetics. 'Cybernetics Evolving to Systems, Humans, Organizations, And Their Complex Interactions'*, 1906-1911.
- Yun, U., & Leggett, J. J. (2005). WFIM: Weighted frequent itemset mining with a weight range and a minimum weight. *Proceedings of the 2005 SIAM International Conference on Data Mining*, 636-640.
- Yun, U., & Leggett, J. J. (2006). WIP: Mining weighted interesting patterns with a strong weight and/or support affinity. *Proceedings of the 2006 SIAM International Conference on Data Mining*, 624-628.
- Zeng, Y., & Xiang, K. (2017). Adaptive sampling for urban air quality through participatory sensing. *Sensors*, 17 (11), 2531.
- Zeynoddin, M., Ebtehaj, I., & Bonakdari, H. (2020). Development of a linear based stochastic model for daily soil temperature prediction: One step forward to sustainable agriculture. *Computers and Electronics in Agriculture*, 176, 105636.
- Zhang, H., Lee, C. A. A., Li, Z., Garbe, J. R., Eide, C. R., Petegrosso, R., et al. (2018). A multitask clustering approach for single-cell RNA-seq analysis in Recessive Dystrophic Epidermolysis Bullosa. *PLoS Computational Biology*, 14 (4), e1006053.
- Zhang, L., Yang, G., & Li, X. (2020). Mining sequential patterns of PM_{2.5} pollution between 338 cities in China. *Journal of Environmental Management*, 262, 110341.
- Zhang, Y. (2010). Multi-task active learning with output constraints. *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, 667-672.

- Zhang, Y., Lu, Y., Zhang, D., Shang, L., & Wang, D. (2018). Risksens: A multi-view learning approach to identifying risky traffic locations in intelligent transportation systems using social and remote sensing. *IEEE International Conference on Big Data*, 1544-1553.
- Zhang, Y., & Yang, Q. (2017). *A survey on multi-task learning*. Retrieved February 5, 2020, from <https://arxiv.org/abs/1707.08114>.
- Zhang, Y., & Yeung, D. Y. (2009). Semi-supervised multi-task regression. *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 617-631.
- Zhang, X. L. (2014). Convex discriminative multitask clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37 (1), 28-40.
- Zhang, X., Zhang, X., & Liu, H. (2015). Smart multitask Bregman clustering and multitask kernel clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10 (1), 1-29.
- Zhang, X., Zhang, X., Liu, H., & Liu, X. (2017). Multi-task clustering through instances transfer. *Neurocomputing*, 251, 145-155.
- Zhang, X., Zhang, X., Liu, H., & Liu, X. (2016). Multi-task multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 28 (12), 3324-3338.
- Zhang, X., Zhang, X., Liu, H., & Luo, J. (2018). Multi-task clustering with model relation learning. *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 3132-3140.
- Zhang, Z., Chen, E., Wang, J., & Wang, X. (2004). Enabling personalization recommendation with Weightedfp for text information retrieval based on user-

focus. *Proceedings of the International Conference on Information Technology: Coding and Computing*, 272-276.

Zhang, Z., Ma, C., Xu, J., Huang, J., & Li, L. (2015). A novel combinational forecasting model of dust storms based on rare classes classification algorithm. In *Communications in computer and information science* (vol. 482) (520-537). Berlin: Springer.

Zhao, J., Guo, H., Han, M., Tang, H., & Li, X. (2019). Gaussian process regression for prediction of sulfate content in lakes of China. *Journal of Engineering and Technological Sciences*, 51 (2), 198-215.

Zhao, J., Xie, X., Xu, X., & Sun, S. (2017). Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38, 43-54.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3 (1), 1-130.