# FSOLAP: A FUZZY LOGIC-BASED SPATIAL OLAP FRAMEWORK FOR SPATIAL-TEMPORAL ANALYTICS AND QUERYING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SINAN KESKIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

JANUARY 2023

Approval of the thesis:

**FSOLAP: A FUZZY LOGIC-BASED SPATIAL OLAP FRAMEWORK FOR SPATIAL-TEMPORAL ANALYTICS AND QUERYING**

submitted by **SINAN KESKIN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Mehmet Halit S. Oğuztüzün
Head of Department, **Computer Engineering** ________________

Prof. Dr. Adnan Yazıcı
Supervisor, **Computer Engineering, METU** ________________

**Examining Committee Members:**

Prof. Dr. Mehmet Halit S. Oğuztüzün
Computer Engineering, METU ________________

Prof. Dr. Adnan Yazıcı
Computer Engineering, METU ________________

Prof. Dr. Murat Koyuncu
Computer Engineering, Atılım University ________________

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering, METU ________________

Assoc. Prof. Dr. Tansel Dökeroğlu
Software Engineering, Çankaya University ________________

Date:

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Surname: Sinan Keskin

Signature :

# ABSTRACT

## FSOLAP: A FUZZY LOGIC-BASED SPATIAL OLAP FRAMEWORK FOR SPATIAL-TEMPORAL ANALYTICS AND QUERYING

Keskin, Sinan
Ph.D., Department of Computer Engineering
Supervisor: Prof. Dr. Adnan Yazıcı

January 2023, 137 pages

Nowadays, with the rise in sensor technology, the amount of spatial and temporal data increases day by day. Fast, effective, and accurate analysis and prediction of collected data have become more essential than ever. Spatial Online Analytical Processing (SOLAP) emerged to perform data mining on spatial and temporal data that naturally contains the hierarchical structure used in many complex applications. In addition, uncertainty and fuzziness are inherently essential elements of data in many complex data applications, particularly in spatial-temporal database applications. Also, there is always a need to support flexible queries and analyses on uncertain and fuzzy data, due to the nature of the data in these complex spatiotemporal applications.

In this study, FSOLAP is proposed as a new fuzzy SOLAP-based framework to compose the benefits of fuzzy logic and SOLAP concepts and is extended with inference capability to the framework to support predictive analytics and spatiotemporal predictive querying. Additionally, while FSOLAP primarily includes historical data and associated queries and analyses, we also describe how to handle predictive fuzzy spatiotemporal queries, which typically require an inference mechanism.

The predictive accuracy and resource utilization performance of FSOLAP are compared using real data with some well-known machine learning techniques such as Support Vector Machine, Random Forest, and Fuzzy Random Forest. The extensive experimental results show that the FSOLAP framework for the predictive analysis of various spatiotemporal events using a big meteorological dataset is considerably more accurate and scalable than conventional machine learning techniques.

Keywords: Fuzzy spatiotemporal data mining, spatiotemporal predictive analytics, fuzzy spatiotemporal OLAP, fuzzy association rule mining, fuzzy knowledge base, fuzzy inference system, fuzzy spatiotemporal querying, fuzzy spatiotemporal predictive querying

# ÖZ

## FSOLAP: UZAMSAL-ZAMANSAL ANALİTİK VE SORGULAMA İÇİN BULANIK MANTIK TABANLI UZAMSAL OLAP ÇERÇEVESİ

Keskin, Sinan
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Adnan Yazıcı

Ocak 2023 , 137 sayfa

Günümüzde sensör teknolojisindeki ilerlemeyle birlikte mekansal ve zamansal veri miktarı her geçen gün artmaktadır. Toplanan verilerin hızlı, etkili ve doğru analizi ve bu verilerle tahmin yapmak her zamankinden daha önemli hale geldi. Mekansal Çevrimiçi Analitik İşleme (SOLAP), birçok karmaşık uygulamada kullanılan hiyerarşik yapıyı doğal olarak içermekte olup mekansal ve zamansal veriler üzerinde veri madenciliği yapmak için ortaya çıkmıştır. SOLAP kullanan uygulamalar daha çok kesin veriler üzerinde çalışırlar ancak birçok karmaşık veri uygulamasında, özellikle uzamsal-zamansal veritabanı uygulamalarında, verilerin doğası gereği belirsizlik ve bulanıklık temel unsurlardır. Ayrıca, bu karmaşık uzay-zamansal uygulamalardaki verilerin doğası gereği, belirsiz ve bulanık veriler üzerinde esnek sorgu ve analizlerin desteklenmesine her zaman ihtiyaç vardır.

Bu çalışmada, FSOLAP, bulanık mantık ve SOLAP kavramlarının faydalarını bir araya getirerek yeni bir bulanık SOLAP tabanlı çerçeve olarak önerilmiş olup tahmine dayalı analitiği ve uzay-zamansal tahmine dayalı sorgulamayı desteklemek için çıkarım yeteneği üzerine eklenmiştir. Ek olarak, FSOLAP temel olarak geçmiş verilerle ilişkili sorguları ve analizleri ele alırken, tipik olarak bir çıkarım mekanizması gerektiren tahmine dayalı bulanık uzay-zaman sorgularının nasıl ele alınacağını da açıklıyoruz.

FSOLAP çerçevesinin tahmine dayalı doğruluğu ve kaynak kullanım performansı, Destek Vektör Makinesi, Rastgele Orman ve Bulanık Rastgele Orman gibi bazı iyi bilinen makine öğrenimi teknikleriyle gerçek veriler kullanılarak karşılaştırılmıştır. Kapsamlı deneysel sonuçlar, büyük bir meteorolojik veri seti kullanan çeşitli uzay-zamansal olayların tahmine dayalı analizine yönelik FSOLAP çerçevesinin, geleneksel makine öğrenimi tekniklerinden çok daha doğru ve ölçeklenebilir olduğunu göstermektedir.

Anahtar Kelimeler: Bulanık uzay-zaman veri madenciliği, uzay-zaman tahminsel analitik, bulanık uzay-zamansal OLAP, bulanık birliktelik kuralı madenciliği, bulanık bilgi tabanı, bulanık çıkarım sistemi, bulanık uzay-zamansal sorgulama, bulanık uzay-zaman tahminsel sorgulama

to all my loved ones...

# ACKNOWLEDGMENTS

The meteorological measurement data used in this study was undertaken to contribute to the Turkish State Meteorological Service. The source code of this study is available at https://github.com/skeskin19/solapfuzzyframework.

# TABLE OF CONTENTS

CHAPTERS

# LIST OF TABLES

TABLES

# LIST OF FIGURES

FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ACU | Average CPU Usage |
| AMU | Average Memory Usage |
| AUC | Area Under the Curve |
| CPU | Central Processing Unit |
| COG | Center of Gravity |
| ETL | Extract, Transform, and Load |
| FC | Fuzzy Confidence |
| FCL | Fuzzy Control Logic |
| FCM | Fuzzy C-Means |
| FIS | Fuzzy Inference System |
| FM | Fuzzy Module |
| FKB | Fuzzy Knowledge Base |
| FP | Frequent Pattern |
| FRF | Fuzzy Random Forest |
| FSAS | Fuzzy Storage Assignment System |
| GIDB | Geospatial Information Database |
| IGP | Intelligent Geographical Project |
| JDBC | Java Database Connectivity |
| JSON | JavaScript Object Notation |
| LHS | Left Hand Side |
| MBR | Minimum Bounded Rectangle |
| MCA | Multi-criteria Analysis |
| MDX | Multi-dimensional Expression |
| MBT | Model Building Time |

| | |
|---|---|
| MPT | Model Prediction Time |
| minsup | Minimum Support |
| minconf | Minimum Confidence |
| ML | Machine Learning |
| OLAP | Online Analytical Processing |
| QIn | Query Interface |
| QM | Query Module |
| QPc | Query Processor |
| QPr | Query Parser |
| RF | Random Forest |
| RHS | Right Hand Side |
| RPF | Rule Power Factor |
| SDM | Spatial Data Mining |
| SOLAP | Spatial Online Analytical Processing |
| SQL | Structured Query Language |
| SVM | Support Vector Machines |

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivation and Problem Definition

Automated data collection tools and the development of database technologies have enabled the storage of large amounts of data in databases and other information repositories. These collected data contain spatial and temporal features to a large extent. It is still a fundamental need to make spatial-temporal analyses, querying, and inferences effectively and efficiently on these data. There are basically two types of difficulties when working on the need to do spatial and temporal data analytics and querying. The first is to ensure that spatial and temporal data are appropriately structured. Because the analyses to be made on this type of data should be modeled according to the data structure to produce results with high performance. Another concern is that spatial-temporal analyses and queries naturally contain complexity, making it difficult to deal with conventional techniques. Complex operations such as interrogating a region with an uncertain range on spatial-temporal data inherently include uncertainty and fuzziness. It is necessary to propose an approach that effectively, efficiently, and accurately makes spatial-temporal analytics, provides inferences, and supports spatial-temporal queries by focusing on solving these concerns.

Researchers working in this field mainly make predictions with numerical and statistical models [1, 2, 3], which commonly use precise values as input and output. However, uncertainty and fuzziness are inherent characteristics of most spatiotemporal database applications [4]. Thus, spatiotemporal information and the various relationships and associations involved in such applications often include uncertainty and fuzziness. For example, a region boundary is represented using a fuzzy concept to describe a highly polluted region. In general, spatial and temporal applications contain various types of uncertainty and fuzziness for the following reasons:

- Spatial information such as where the events occur, the topological properties of these places, and their spatial relationships are complex and naturally imprecise or fuzzy and include multiple forms of uncertainty [5].

- Due to their appearance form, many natural phenomena do not occur within concrete boundaries and exhibit fuzzy transitional spread (for example, it is not easy to define fog boundaries with precise edges.) [6, 7, 8].

- With the complex nature of the application domain of spatial databases, it is often tedious and challenging to operate on a set where only precise data is taken into account. For example, it may be necessary to identify "high frost risk" regions or know the number of days a location receives "heavy rainfall." The fuzziness criteria are naturally expressed in linguistic terms instead of treating these analyses and queries with crisp data [9].

In addition, conventional data mining techniques [10] are not adequate for spatiotemporal database applications because these techniques involve complex differential equations and computational algorithms that require high resources. On the other hand, effective and efficient data analysis and prediction are needed to perform with a vast amount of collected spatial and temporal data. On the other hand, effective and efficient data analysis and prediction are needed to perform with a vast amount of collected spatial and temporal data. Spatial Online Analytical Processing (SOLAP) is one widely used geospatial data mining tool for this purpose. It allows the exploration of data cubes to obtain new information effectively and efficiently [11]. A multi-tier computational model needs to be adapted to spatial data mining, as handling spatial relationships involves high computational costs [1]. Morely, SOLAP is a data mining platform that provides spatial and temporal analysis quickly and easily with a multidimensional approach composed of tiers of aggregation.

Spatiotemporal database applications inherently include hierarchical data forms. Spatiotemporal data has geographical information as a spatial part with a hierarchical structure and covers another from outside to inside. This coverage format can be exemplified as country-region-city-town-station from general to specific. Similarly, the time information in the temporal section may have hierarchical levels such as year-month-day-hour or represented as year-quarter-month or year-season-month types depending on the application domain. SOLAP, which has the ability to make fast and effective analyzes and queries on hierarchical data, is also preferred because it allows such data to be easily modeled over metadata. In addition, uncertainty and fuzziness are inherently involved in spatiotemporal applications because they are natural features of spatial-temporal data [5] (i.e., in the definition of a fogy location). Due to these features of spatiotemporal applications, it isn't easy to deal with conventional logic approaches. Since these features are complex, instead of approaching with precise mathematical models, it is necessary to apply the fuzzy logic approach that has emerged for this purpose. In studies [5, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21] carried out with the thought that crisp logic would not be sufficient, and uncertain and fuzzy conditions are handled by developing applications and systems that provide linguistical control with the fuzzy logic approach. In this context, a platform offering efficient and effective analysis and queries can be developed using the concepts of fast and accurate spatiotemporal data processing provided by SOLAP and the easy handling of complex applications provided by fuzzy logic. The objective of the study is to offer a new framework, FSOLAP, to leverage both SOLAP and fuzzy logic to provide fuzzy spatiotemporal data analytics and queries and extend them with inference capability.

Online Analytical Processing (OLAP) is not capable of explaining relationships between the data that resides in the data cube [22], and neither is SOLAP. Association rules, a type of data mining technique for finding associations among data, can discover affinities on SOLAP data. This study focuses on the generation of association rules for the fuzzy inference system. The most recent algorithms [4, 23] presented for fuzzy association rule mining consider that the user provides the fuzzy set. In this study, fuzzy classes and membership values are automatically generated and used to generate fuzzy association rules. In addition, minimum support and minimum confidence values should be satisfied simultaneously by the user when generating the association rules [24]. The constraining issue related to discovering these associations is to choose the threshold of minimum support (minsup) and minimum confidence (minconf) properly because they directly influence the number and the quality of the detected patterns. Therefore, the appropriate minsup and minconf values can be determined using a heuristic approach. Different minsup and minconf values are programmatically chosen to generate fuzzy association rules, and the prediction accuracy is calculated for each generated association rule set.

Furthermore, fuzzy logic with OLAP is widely studied in the literature [4, 25], but there are not many proposals about fuzzy logic with spatial data types of OLAP (i.e., FSOLAP). This study introduces a new framework based on SOLAP and fuzzy logic. The FSOLAP framework includes a database, SOLAP, a fuzzy logic module, an association rule generator, and a fuzzy inference mechanism to perform data analytics and prediction on spatiotemporal database applications.

## 1.2 Proposed Methods and Models

Spatial-temporal database applications naturally contain hierarchical data structures. Spatial data includes hierarchical breakdowns such as country-region-city, while temporal data have hierarchical relationships at levels such as year-month-day. SOLAP, a geospatial data mining tool, emerged to provide effective and efficient analysis and querying of hierarchical data. In addition, conventional data mining techniques are insufficient in spatiotemporal database applications because they often require intensive computations involving complex differential equations and computational algorithms. Spatial and temporal information and several relations in spatial-temporal applications frequently involve uncertainty and fuzziness, which are inherent features of most of these applications. At this point, a combination of fuzzy logic and the SOLAP concepts is proposed in this study to get benefits from them. To accomplish this, we assemble a fuzzy spatial OLAP model and thus support performing fuzzy and OLAP-based operations on fuzzy spatial data. Additionally, the combined proposal is improved to support the inference mechanism by using fuzzy association rule mining with a fuzzy inference system. Finally, a fuzzy spatial-temporal predictive model is built by combining fuzzy logic, SOLAP, fuzzy knowledge base, and fuzzy inference system.

## 1.3 Contributions and Novelties

This study involves in finding patterns in data with spatiotemporal characteristics applying data analytics and making predictions using the knowledge explored. In this context, it is essential to use real data to demonstrate the accuracy of performance of the study. In some of the studies [4, 22], researchers use synthetic or semi-synthetic data to test the performance of their proposed models. Although the proposals are confirmed to some extent from synthetic data, it is impossible to express the actual efficiency and accuracy of the model for real situations. For this reason, it is crucial to represent the accuracy and performance tests of this proposal using real data in the spatiotemporal database. Thus, the accuracy and resource utilization performance of the model proposed in this study is demonstrated using real data that includes spatial and temporal features and different meteorological measurements obtained from the Turkish Meteorological Service.

The contributions of this thesis can be summarized as follows:

- The main contribution of this study is to propose FSOLAP as a new fuzzy SOLAP-based framework and provide effective and efficient predictive analysis of various spatiotemporal events, including support for various querying capabilities, data visualization, and analysis.
- The FSOLAP framework brings together the strengths of fuzzy and SOLAP concepts for spatiotemporal applications and is extended with inference to offer effective and efficient predictive analysis.
- FSOLAP introduces the fuzzy spatiotemporal predictive query, which includes an inference mechanism, as well as the complex type of fuzzy spatial queries present in the literature.
- We extend the proposed FSOLAP framework using aggregation operators to provide fuzzy summaries for knowledge discovery. We thus offer to generate fuzzy summaries utilizing the result of the executed fuzzy spatial-temporal queries on the FSOLAP framework.
- Determining the number of clusters when performing clustering on fuzzifica-

tion is a tricky problem. In general, empirical approaches have been used to determine the number of clusters in a data set. However, the proper number of clusters determination process is handled automatically.

- Producing an appropriate number of fuzzy association rules, pruning the generated ruleset, weighting the rules in the ruleset, and tuning also becomes an automated process in the FSOLAP framework.
- The predictive accuracy and the utilization performance of the proposed framework are tested, and comparative analysis is made with some of the well-known machine learning (ML) algorithms such as random forest, support vector machines (SVM), and fuzzy random forest using the same real data as a case study of a meteorological application. The utilization performance with these ML models includes the average CPU usage, memory usage, model building time, and model prediction time.

## 1.4 The Outline of the Thesis

The rest of the thesis is organized as follows: Chapter 2 explains the background and previous studies on the subject. Details of the proposed framework and the components of the FSOLAP architecture are demonstrated in Chapter 3. The query module of the framework with the supported query types and fuzzy spatial aggregation is explained in Chapter 4. The form and features of the meteorological data used in the case study of the presented framework are clarified in Chapter 5. Chapter 6 represents the experiment results of the case study. The study results are discussed and compared with the existing works in Chapter 7. Finally, the conclusion and future studies are explained in Chapter 8.

# CHAPTER 2

# BACKGROUND

Data analytics explores data through various algorithms and applications and inferences about the information it contains from these examinations. Data analytics, which emerged from data science, a multidisciplinary field, and has become one of the most critical sciences of today, also has a vast area of study. In data analytics, questions are asked in specific patterns to analyze the information and achieve the desired result. As a result of the different questions asked, data analytics is diversified and divided into various methods. Predictive analytics, a type of data analytics, is where we begin by predicting "potential future outcomes" and turning the results of our descriptive and diagnostic analyzes into actionable concepts for decision making.

A data analyst uses quantitative dataset analysis and gets results, often referred to as predictive modeling in predictive analytics. This operation is a broader approach to characterizing predictions and examining models for their accuracy. Machine learning algorithms, classification models, and regression models are part of the predictive analytics field.

In the following sections, a literature review of previous studies on spatiotemporal data mining and predictive analytics is given, along with brief descriptions of the concepts of OLAP, SOLAP, fuzzy logic and inference mechanisms. In addition, the scope of studies supporting fuzzy queries is also examined.

## 2.1 Data Warehouse

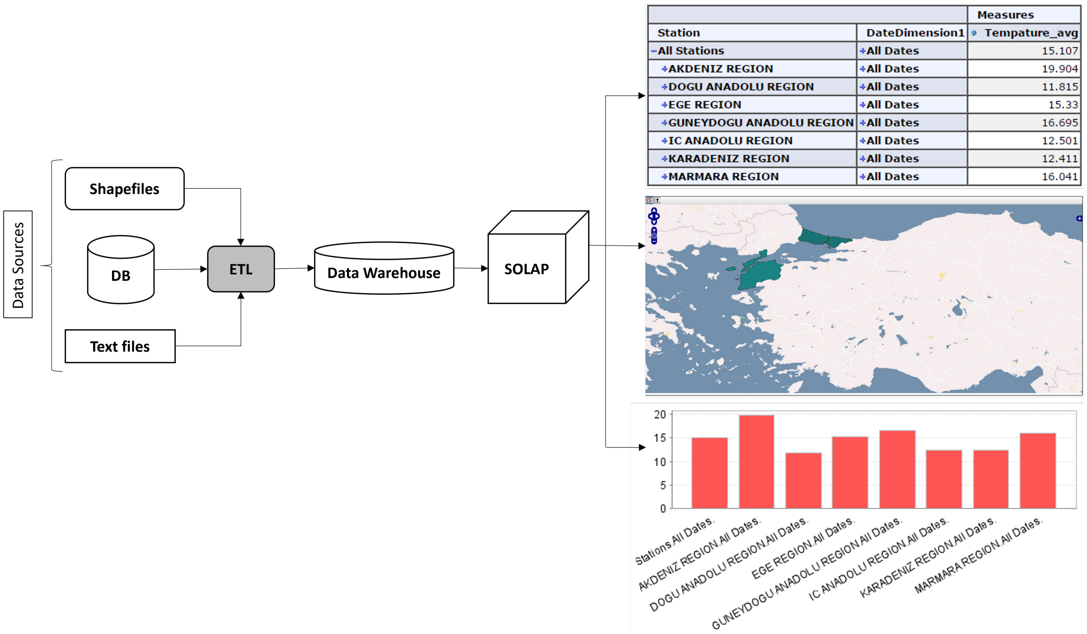
As a result of the critical developments in data collection and storage technologies in recent years, the data stored in businesses have started to reach huge dimensions. In these enterprises, hybrid organizational structures and data may be scattered in different settlements. With the increase in data, which is an essential input to Decision Support Systems, and the need to separate from relational databases, data warehouses have come to the fore. A data warehouse is a large-scale data warehouse that combines the data collected by various units of an enterprise or institution through live systems and the ones that can be evaluated in the future in a system in the background. The data warehouse provides an infrastructure for robust data analysis techniques such as data mining and multi-dimensional analysis and traditional methods where related data can be queried and analyzed.

A data warehouse is a semantically consistent data warehouse that stores the information an organization needs to make strategic decisions and works as a physical representation of the decision support data model. A data warehouse is also seen as an architecture built by integrating data from different types of sources to support structured or unplanned queries, analytical reports, and decision making [10].

## 2.2 Spatial OLAP

With the use of relational databases and the size of the data warehouses that emerged afterward, the need for faster access to data and multidimensional analysis arose. Online Analytical Processing (OLAP) facilitates decision support queries. Spatial OLAP (SOLAP), the specialized type of OLAP for spatial data analysis, naturally includes all the features of OLAP and offers a structure that allows spatial data to be maintained in a hierarchical structure. Thus, the relationship of spatial data with each other can be explored within the scope of geometric properties.

After the data is collected in the data warehouse, this data can be analyzed manually or visually. OLAP technology is used for this, which is a query-based method that supports multi-dimensional data analysis. OLAP is a software technology that provides a variety of insights from the data to analysts, managers, and practitioners by expressing the expert's point of view on the system in a way that the users of the system can understand by accessing the vast majority of possible reviews on information converted from raw data in a fast, consistent, interactive manner. It allows to look at and examine data as a multi-dimensional cube, each dimension corresponding to a field. Thus, grouping by dimension allows exploring the relations between dimensions and presenting the results as graphs or reports. Having a data warehouse does not mean there is no need for OLAP. Data warehouses and OLAP complement each other, as shown in Figure 2.1. A data warehouse is used to store data. On the other hand, OLAP makes sense of this heap of data and makes analysis.


Data Sources
Shapefiles
DB
Text files
ETL
Data Warehouse
SOLAP
20
15
10
5
0
Stations.All Dates.
AKDENIZ REGION.All Dates.
DOGU ANADOLU REGION.All Dates.
EGE REGION.All Dates.
GUNEYDOGU ANADOLU REGION.All Dates.
IC ANADOLU REGION.All Dates.
KARADENIZ REGION.All Dates.
MARMARA REGION.All Dates.

| | | Measures |
|---|---|---|
| Station | DateDimension1 | Tempature_avg |
| −All Stations | +All Dates | 15.107 |
| +AKDENIZ REGION | +All Dates | 19.904 |
| +DOGU ANADOLU REGION | +All Dates | 11.815 |
| +EGE REGION | +All Dates | 15.33 |
| +GUNEYDOGU ANADOLU REGION | +All Dates | 16.695 |
| +IC ANADOLU REGION | +All Dates | 12.501 |
| +KARADENIZ REGION | +All Dates | 12.411 |
| +MARMARA REGION | +All Dates | 16.041 |

Figure 2.1: Using SOLAP with data warehouse

OLAP systems include the ability to look at data in multiple dimensions. The actual performance of an OLAP system can be measured by its power to perform complex calculations. OLAP systems must be capable of performing operations other than just aggregation. OLAP tools support a multi-dimensional conceptual view of data. The multi-dimensional model stores data in events and dimensions instead of rows and columns. Multi-dimensional modeling and examination of data are provided with data cubes. Figure 2.2 represents a sample cube, a logical table used in OLAP databases. It allows dimensions and measurement values to be managed together. The cube creates n-dimensional grids by including many dimensions in it. Each cell of the cube corresponds to only one value. As a result of the intersection of dimensions, that value is reached.

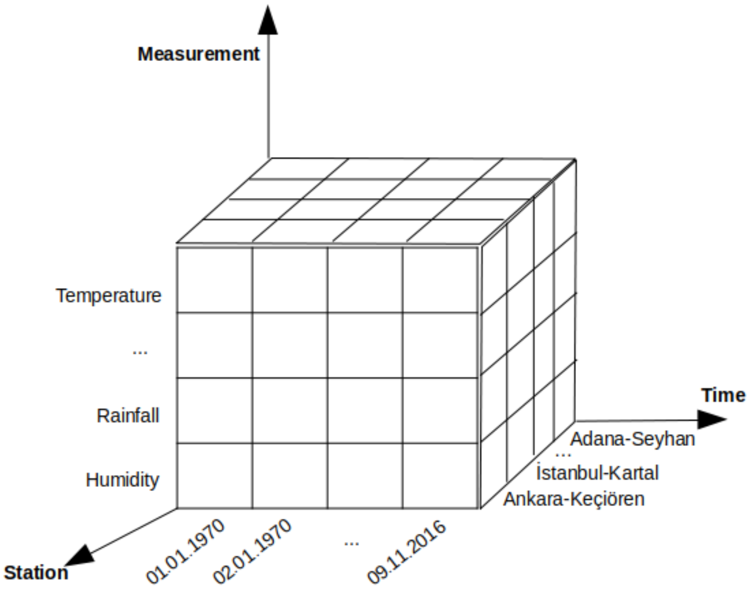


Figure 2.2: A sample cube structure

A multidimensional view of the data represents the conceptual model in repositories. This model forms the technical basis of computation and analysis required for business intelligence. It includes a range of dimensions and numerical measures depending on the dimensions. Dimensions are the business perspective of the database, and metrics are the data of interest. Dimensions in a multidimensional view correspond to fields in a relational database, while measures (i.e., cells) correspond to records. Sales, budget, revenue, etc. Numerical measures constitute the subject of the analysis as they are numerical values that can be used in monitoring the enterprise.

In the multidimensional model, dimensions correspond to coordinates, and measurements are uniquely determined by dimensions coordinating to values (i.e., cells). It can be thought of as a multidimensional space corresponding to a set of multidimensional numerical measurements, as shown in the figure.

OLAP is generally used to make numerical analyses and generate historical data reports. Although historical analyses and inferences are frequently performed on data in spatial OLAP, it is also a desirable to query this data based on verbal language. At this point, supporting queries that surround fuzzy concepts is an essential feature for any spatial database. When the previous studies on this subject are examined, Winter [26] discusses the relationships of discrete regions to topological relationships. Another study [27] is about examining the spatial relationships using inquiries. Cobb and Petry [28] explore spatial relationships from the point of view of fuzziness. As part of the study, the binary topological and directional relationships between 2D objects are modeled. Yang et. al. [29] concern querying binary spatial relations. They introduce binary data structures that enable fuzzy queries of spatial relationships in 2D objects. In addition, Laurent [30] explains the properties of unary operators in the context of queries on fuzzy multidimensional databases. In summary, all these proposals explore the development of basic operators in multidimensional spatial databases.

It is essential to query in the spatial OLAP structure. In particular, making queries in linguistic form provides convenience to the users. We can use linguistic terms for querying in verbal languages with the fuzzy concept. In studies [31, 32] dealing with fuzzy relationships on spatial data, researchers examine the directional and topological relationships in fuzzy concepts. In other studies [33, 34], the binary model is defined, and fuzzy queries are provided on this model. Another study [30] explains using unary operators in fuzzy queries on the multidimensional model. In this study, the management of query types in different combinations is explained besides the handling of unary operators on the fuzzy cube.

Data mining and fuzzy data mining are performed on OLAP [12, 13, 14, 15], and these studies are focussed on providing information from a more general perspective. In these studies, the advantages of OLAP and fuzzy concepts are combined. Also, a multidimensional data model is built on the data warehouse, and knowledge discovery is performed through imprecise data. These studies typically focus on finding knowledge about fuzzy spatial data, but more complex queries (e.g., select cold regions) are not considered.

## 2.3 Data Mining

The essence of data mining is finding and extracting unusual, significant, precise, previously unknown, and valuable information or patterns from vast amounts of data.

Models used in data mining can be examined under two main categories: descriptive and predictive. In descriptive models, patterns in existing data that can be used to aid decision-making are defined. Predictive models aim to develop a model based on the data with known results and estimate the result values for data sets whose results are unknown using this established model.

Performing data mining tasks on data warehouses has been frequently studied [35, 2, 22, 10]. Some of them [36, 22] related to mining patterns and generating association rules on data cubes. For example, Agrawal et. al. [36] express that OLAP is closely entangled with association rules and offers the objective with association rules for discovering patterns in data. To discover knowledge from data cubes, association rule mining is used together with OLAP. Further, spatial data mining in a spatial data cube as well as a spatial database can be performed. To this end, Han [1] has proposed a prototype spatial OLAP and data mining system to explore spatial information visually called GeoMiner. Another study focuses on a framework using data cubes in association rule generation. In this study, Messaoud et. al. [22] use aggregate measures to determine support and confidence values by guiding data mining analysis with a meta-rule based contextual approach. Although these studies deal with data cube and association rules, they do not examine the dimensions of the data cube in terms of spatial and temporal hierarchies and they do not consider uncertainty and fuzziness.

In another study, Kaya and Alhajj [4] proposed a multi-agent learning approach based on the use of the data mining process for modular cooperative learning systems, which is an extension of mining patterns with association rules which integrates the modular approach with fuzziness and OLAP concepts. They merged fuzzy logic and OLAP-based mining for processing the knowledge informed by agents, then applied it to a few simple problems.

Prediction of the weather or natural phenomenon is also an extensive field of study.

In this context, some recent studies have been conducted to predict meteorological events such as drought [37, 38, 39], operational streamflow [40], rainfall [41], and reservoir operation under climate change [42]. Mohamadi et al. [37] basically made a drought modeling study to efficiently manage water resources using the adaptive neuro-fuzzy interface system (ANFIS). They used ANFIS, multilayer perceptron (MLP), radial basis function neural network (RBFNN), and support vector machine (SVM) models to forecast meteorological droughts in Iran. These models predicted the three-month standardized precipitation index (SPI). The results of their study indicated that hybrid soft computing models and wavelet coherence are appropriate tools for predicting hydrological variables. In addition, Taormina and Chau [40] dealt with the application of the lower upper bound estimation (LUBE) method for the construction of artificial neural networks (ANN) based prediction intervals (PIs) of streamflow discharges at three confidence levels. They tested the suitability of the LUBE approach in producing PIs at different confidence levels (CL) for the 6 h ahead streamflow discharges of the Susquehanna and Nehalem Rivers, US. They also proposed the Multi-Objective Fully Informed Particle Swarm (MOFIPS) optimization algorithm to return valid PIs for both rivers and the three CL considered in their study. They concluded that MOFIPS-based LUBE represented a viable option for the straightforward design of more reliable interval-based streamflow forecasting models. Wu and Chau [41] attempted to seek a relatively optimal data-driven model for rainfall forecasting from three aspects: model inputs, modeling methods, and data-preprocessing techniques. They suggested using a modular artificial neural network(MANN) coupled with data-preprocessing techniques for improving four rainfall predictions from India and China consisting of two monthly and two daily series. To reasonably evaluate MANN's performance, three models, LR, K-NN, and ANN, were used for comparison. Their experiments showed that MANN distinguishes itself from the other monthly and daily rainfall forecasting models. While all models reasonably forecast two monthly rainfall series, only MANN can simulate each daily rainfall series without a noticeable lag effect. Data mining offers a way to statistically analyze data and extract or derive such rules that can be used for predictions. Shamshirband et al. [38] studied to predict an index of the hydrological drought of standardized streamflow index (SSI) for the Navrood drainage basin using SPI and standardized precipitation evapotranspiration index (SPEI) through the implementation of data-driven models of support vector regression (SVR), gene expression programming (GEP), and M5 model trees (MT). In their proposal, three drought indices, i.e., SPI, SSI, and SPEI, were modeled using SVR, GEP, and MT. They selected these simple data-driven models for easy predictive modeling applications and high decision-making capabilities. Their results showed that SPI delivered higher accuracy, and the MT model better predicted SSI. In another study, Zhao et al. [39] proposed the gravity recovery and climate experiment (GRACE) based modulated water deficit (GRACE-MWD) process for drought monitoring under the modulated annual cycle (MAC) reference frame in southwest China. They achieved a higher agreement ratio using GRACE-MWD with the standardized precipitation evapotranspiration index at a time of 3 months (SPEI03). They observed that GRACE-MWD hits the drought situation twice following SPEI03 for the three severe droughts over the past ten years in the examined study area. Also, Ehteram et al. [42] investigated reservoir operation under climate change for a base period (1981–2000) and a future period (2011–2030). In their study, they considered reservoir operation for irrigation demand-supply and investigated reservoir operation under climate change based on different climate change models. The study of climate change models showed that the temperature of Dez basin in Iran would increase from 2011–2030, and precipitation would decrease for this period. Based on all evolutionary algorithms, they noted that the volume of water to be released for the future period was less than for the base period. Their results also showed that the bat algorithm with high reliability and low vulnerability index performed better than other evolutionary algorithms. The research [2] makes predictions about future variations in sea salinity and temperature associations in surrounding waters by exploring the historic salinity temperature. Their study offers to use inter-dimensional association rule mining with fuzzy inference to find salinity-temperature patterns with spatial-temporal affinities. Another proposal [3] presents a method for predicting daily rainfall data. They use precipitation, wind speed, visibility, temperature, and dew point as atmospheric parameters for predictive analysis. They apply an apriori algorithm to explore the unknown association between different atmospheric parameters to predict rainfall. Their proposal uses conventional techniques with precise data, but a fuzzy approach can be used in this domain since meteorological data inherently contains fuzziness.

### 2.3.1 Data Mining Techniques

The first thing to do in the data mining process is determine the information needed and define the problem. After this phase, the data sources should be determined, and the data should be examined. The data is prepared, and then the model is built. This step, which can be considered the most critical step of the process, is the way to reach the necessary information. The next step is to evaluate the model and interpret the results.

It is possible to examine the main methods used extensively in descriptive and predictive models, such as Classification and Regression, Clustering, Association Rules, memory-based methods, artificial neural networks, and decision trees. Classification and regression models are predictive models, clustering, association rules, and sequential pattern models are descriptive models.

#### 2.3.1.1 Clustering

The primary purpose is to find natural groups (clusters) within multidimensional data. Objects can be included in the same set if they are similar to each other (to the same extent) and not similar to objects in different clusters. Clustering deals with how domain information can be combined with clustering mechanisms. In some applications, the clustering model can act as a preprocessor of the classification model. With the increase in the amount of data collected in databases, cluster analysis has recently become an active topic in data mining research. The choice of a clustering algorithm depends on the data type and purpose.

#### 2.3.1.2 Association Rules

The aim is to reveal the rules of events that are likely to occur together. Association rules find association relationships between datasets. Discovering interesting association relationships from large volumes of transaction records makes decision-making processes more efficient. While large databases have association rules, frequently repeated items are found first. Each element is repeated at least as often as the predetermined minimum number of supports. Strict association rules are then created from frequently repeated items. These rules must meet the minimum support and minimum confidence values.

A prediction approach [10] utilizes association rules in machine learning algorithms with fuzzy contributions. In this context, they integrated fuzzy association rules into machine learning techniques to introduce a classification framework. The framework has multi-objective optimization to determine the fuzzy sets, the fuzzy class association rules, and a classifier on the forged feature vectors to forecast the class of unnoticed objects. Another study [43] focuses on the construction of the application of various data mining techniques to cluster, classify, associate, or predict the pattern of spatiotemporal data. They explain data mining techniques for meteorological applications such as classification, clustering, and association rules. Although machine learning algorithms seem adequate when prediction alone is considered, they are not sufficient for complex systems including inquiry, visualization, and hierarchical analysis on spatiotemporal data.

## 2.4 Fuzzy Spatial Data Mining

Spatial Data Mining (SDM) applies data mining techniques to spatial data. SDM also discovers hidden features and exciting relationships in spatial databases. The main task of SDM is to distinguish and extract some obscure, unknown, and interesting spatial/temporal pattern distributions. Current methods for geospatial data analysis include spatial clustering algorithms, deviation analysis methods, spatial classification, association analysis, and visual data mining methods.

In classical logic, classifications are precise. An element is either a member of a set or not and cannot be a partial membership. In short, classical sets have logic 0 and 1. On the other hand, unlike classical logic, fuzzy logic can operate in uncertain and approximate situations by imitating the human sense. In fuzzy logic, an element can be a member of more than one set. Within the scope of fuzzy spatial data analysis, instead of examining spatial data with precise borders, it is ensured that the areas where the events occur are explained with a fuzzy approach, taking into account the transitive structure. This approach converges more closely to the human-like examination of events in daily life.

Some researchers [16, 17] focused on fuzzy spatial data mining, not including SOLAP or multidimensional expression (MDX) query. They provided an extension of the standard Structured Query Language (SQL) for performing spatial and temporal data analytics. In their study, techniques invented in fuzzy and spatial data mining are merged and extended to handle the uncertainty of characteristic spatial data. However, the query performance of the proposals was not shown. In another proposal [18], fuzzy logic and spatial databases are combined to provide OLAP queries and decision support processes. They discussed the fuzzy spatial data warehouse design for a methodological application, while they did not focus on the effectiveness and efficiency of the system and queries.

### 2.5 Fuzzy Spatial Querying

One of the critical research areas in data analytics is efficient and effective data querying. The flexibility of the query is also essential. The users should know the fields and the data structure for querying. They should also have sufficient knowledge of the query language of the corresponding system. These are some basic requirements for querying the database. It should be able to define the query correctly and logically to get good efficiency from querying. Flexible systems need to be developed to adapt and adequately meet these requirements. These systems minimize the logical errors in the query, increase the efficiency and effectiveness of the query, and offer easy use to the user. The MDX query structure is executed by collecting the data in the data warehouse and making inquiries with the support of the SOLAP server. While making inferences with data analytics on fuzzy spatial data, it is vital to provide relevant queries for this data.

In studies [16, 17], the Structured Query Language (SQL) structure is extended to query spatial and temporal data. In these studies, spatial and fuzzy data mining techniques and their advantages are brought together, and the uncertainties in spatial data are discussed. Researchers need to emphasize performance metrics such as resource usage and accuracy revealed by the queries. In another study [18], a fuzzy spatial data model is built to provide making decision support and query on the data warehouse. This study uses OLAP and fuzzy logic concepts but does not contain the methodology's effectiveness.

Among the studies [19, 20] conducted for fuzzy spatial query, studies that made specific types of queries are also presented. These studies show the management of nearest-neighbor and range types queries. In these queries, researchers emphasize fuzzy spatial objects with uncertain boundaries. Queries contain basic fuzzy spatial concepts but do not support complex and flexible queries. Support for complex spatial query types is still required.

Studies [21, 44] explain the use of special structures that support the efficient and effective querying of fuzzy spatial data in spatiotemporal databases. In these studies, especially index structures desired to access the data with the least cost have been focused. In these studies, novel indexes such as R*-tree [45] and X-tree [46] were used for efficient and effective queries, but there were no queries showing the benefits of spatial OLAP.

# CHAPTER 3

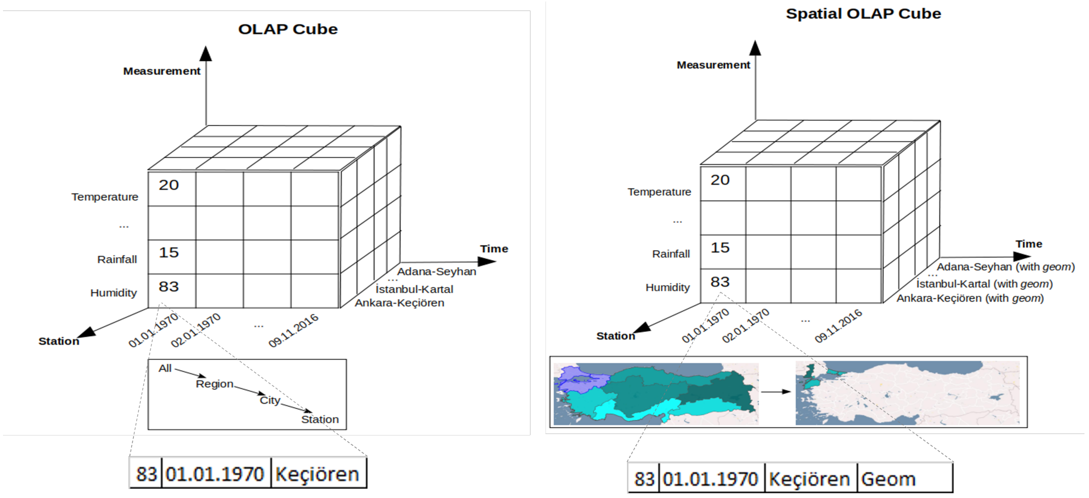
# FSOLAP FRAMEWORK

In this section, we first explain building the model to handle fuzzy spatial-temporal data. While making the necessary model for efficient and effective analysis and queries on fuzzy spatiotemporal data is essential, it is also vital to construct an integrated framework that includes many components that provide easy use for the analytical, querying, and inference functions of this model.

## 3.1 Fuzzy Spatial OLAP Model

Before explaining the proposed fuzzy spatial OLAP cube model, it is helpful to lecture about the fuzzy OLAP and spatial OLAP cube model in the literature. Figure 3.1(a) shows the basic OLAP cube structure consisting of the dimensions, hierarchies, and precise data in the cell. The OLAP cube cell contains a precise value, measurement date, and spatial information of the relevant data type. Figure 3.1(b) shows a spatial OLAP cube structure. The spatial OLAP cube cell contains geometric spatial information in addition to the OLAP cube cell. This info may be a polygonal data structure that allows spatial operations. In addition to the basic model, this structure also includes functions that provide spatial data to be handled hierarchically in spatial relationships.

Figure 3.2 shows the fuzzy OLAP cube's dimensions, hierarchies, membership class, and value. An example record in the cube cell contains the fuzzy membership class, fuzzy membership value, measurement date, and station information of the related feature.



(a) (b)

Figure 3.1: (a) OLAP, (b) Spatial OLAP models in literature

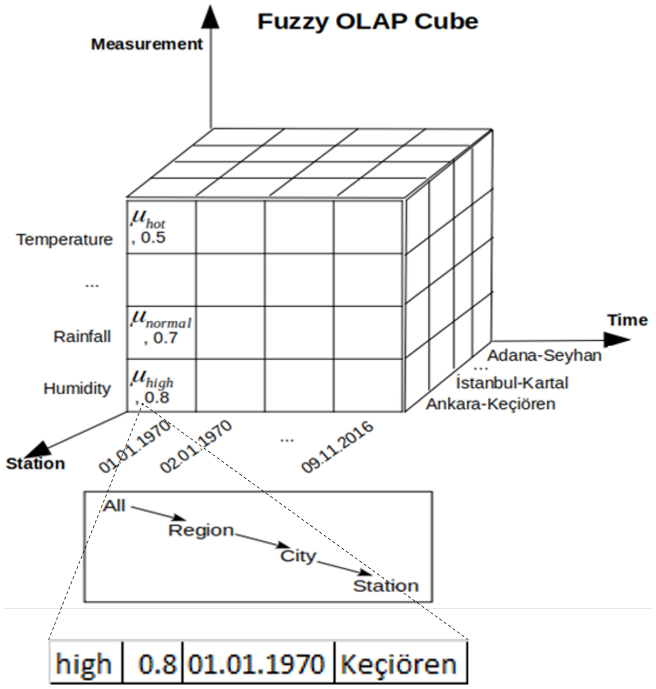
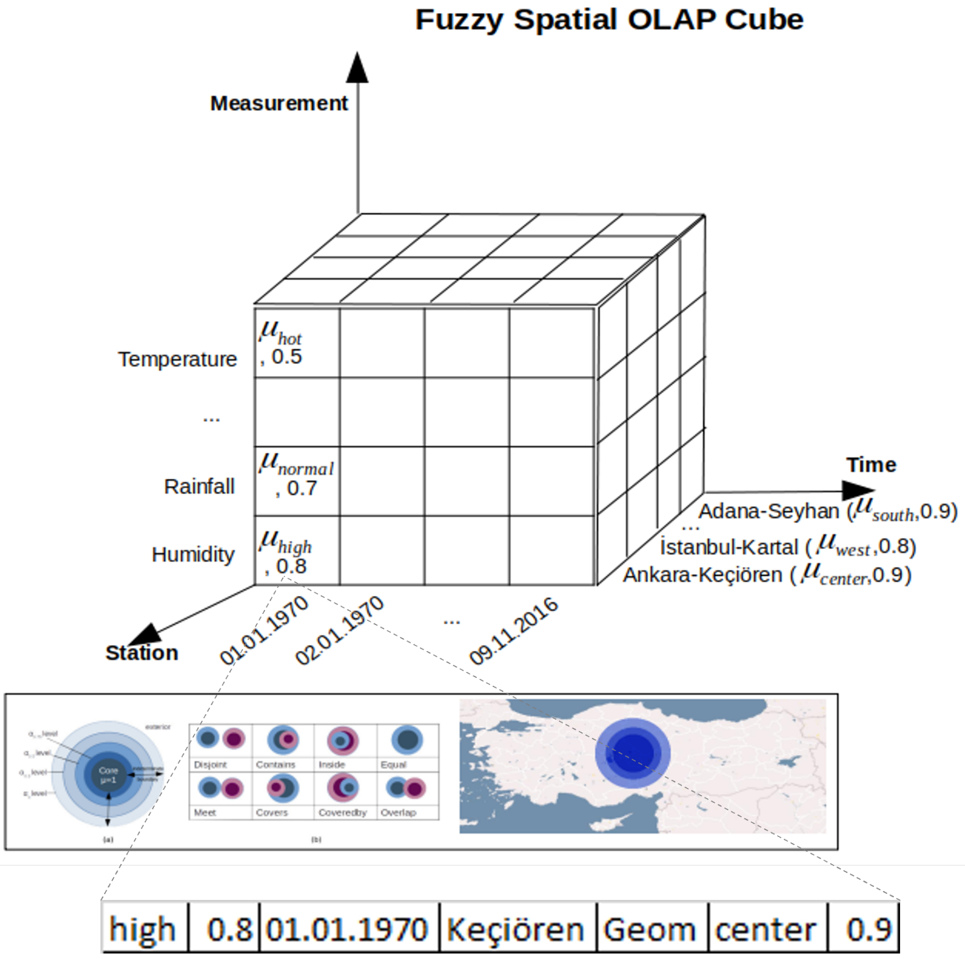


Figure 3.2: Fuzzy OLAP model

Fuzzy OLAP enables the extraction of relevant information in a more natural language and supports the reliability of the information, giving results to the queries. Fuzzy OLAP can be built with a multidimensional database to deal with real-world data and perform OLAP-based mining. In other words, since fuzzy set theory handles numerical values more naturally, we can combine OLAP mining and fuzzy data mining to increase the intelligibility of the discovered information. It also helps deduce more generalizable rules as numerical data is manipulated with words.

The fuzzy spatial data cube model, which is built as a composition of OLAP cube structures in the literature, is shown in Figure 3.3. This model supports relational operations of spatial data and provides fuzzy functions by containing fuzzy membership classes and values in its cells.



| | | | | | | |
|---|---|---|---|---|---|---|
| high | 0.8 | 01.01.1970 | Keçiören | Geom | center | 0.9 |

Figure 3.3: Fuzzy Spatial OLAP model

We explain the basic steps in building the model through the following definitions. A fuzzy spatial cube consists of cells that contain fuzzy measurement data, spatial information of the measure, and the temporal attribute of the data when measured. This information in the cell provides the association with spatial and temporal hierarchies, constructing separate hierarchies for each measurement type.

Suppose that $R$ is a reference set that includes tuples of records $t$. A record is a tuple of $t(m, d, s)$ which consists of the $m$ measurement value, $d$ date of measure, and $s$ station of measure. A fuzzification process $fz$ on $R$ generates fuzzy membership functions, classes, and values for each tuples $t_i$. After fuzzification we build the fuzzy SOLAP cube which consists of cells with fuzzified tuples $t_f(m_f(v_m, d_m), d, s_f(v_s, d_s))$ where $fz(m) \rightarrow m_f(v_m, d_m)$, $fz(s) \rightarrow s_f(v_s, d_s)$ and $v_m \in$ fuzzy measurement set (for example cold), $v_s \in$ fuzzy spatial set (for example north). $v_m$ and $v_s$ are defined by a fuzzy set. $d_m$, $d_s \in$ [0, 1] are the confidence degree associated with these fuzzy values. Domain $D \in f : S \rightarrow \{D_1, D_2, ..., D_n\}$ finite set $f$ and dimension $D_s$, $D_m$ can be defined on a domain as $D_s$, $D_m \in D$. Tuple $t_f(m_f(v_m, d_m), d, s_f(v_s, d_s))$ represents a cell element $d_m \in D_m$ and $d_s \in D_s$. $D_{m1} \times D_{m2} \times ... \times D_{mk} \rightarrow D_m \times$ [0, 1] and $D_{s1} \times D_{s2} \times ... \times D_{sk} \rightarrow D_s$ X [0, 1] are the dimension of measure and spatial respectively. The definition of fuzzy spatial cube is $D_m \times D_s \rightarrow D_c$ X [0, 1] where $D_m \times D_s$ are dimensions and $D_c$ is measure in the cell. The degree between 0 to 1 indicates how much each cell belongs to the relevant domain and is denoted by $\mu$.

### 3.2 Proposed Architecture

In this study, a new framework called FSOLAP is proposed to provide fuzzy spatiotemporal data analytics and prediction [47]. The FSOLAP framework provides for fuzzy spatial-temporal data analytics and supports flexible and complex querying. The framework includes a multi-layered system architecture that consists of four layers. The layers are data sources, structured data, logic, and presentation layers (from the bottom to the top). The system architecture of FSOLAP is represented in Figure 3.4. The multi-layered architecture in this figure has been designed based on dealing with a complex problem with smaller pieces. It starts from a raw dataset and processes it by making it structural, with multiple functionalities such as SOLAP server, fuzzy module, query module and other components. It provides an environment that includes components and supports a modular approach by transferring data between layers. It is a structure that allows us to develop our layered architecture framework according to a certain standard and hierarchy, increases the control of the communication of the components within the layers, makes our framework tidier, and makes error management easier. In software, we can easily manage operations such as accessing data, performing operations on it and displaying these operations to the user with a layered architecture. With the support of the layered architecture, we disassemble this structure and ensure that these processes can be better managed.

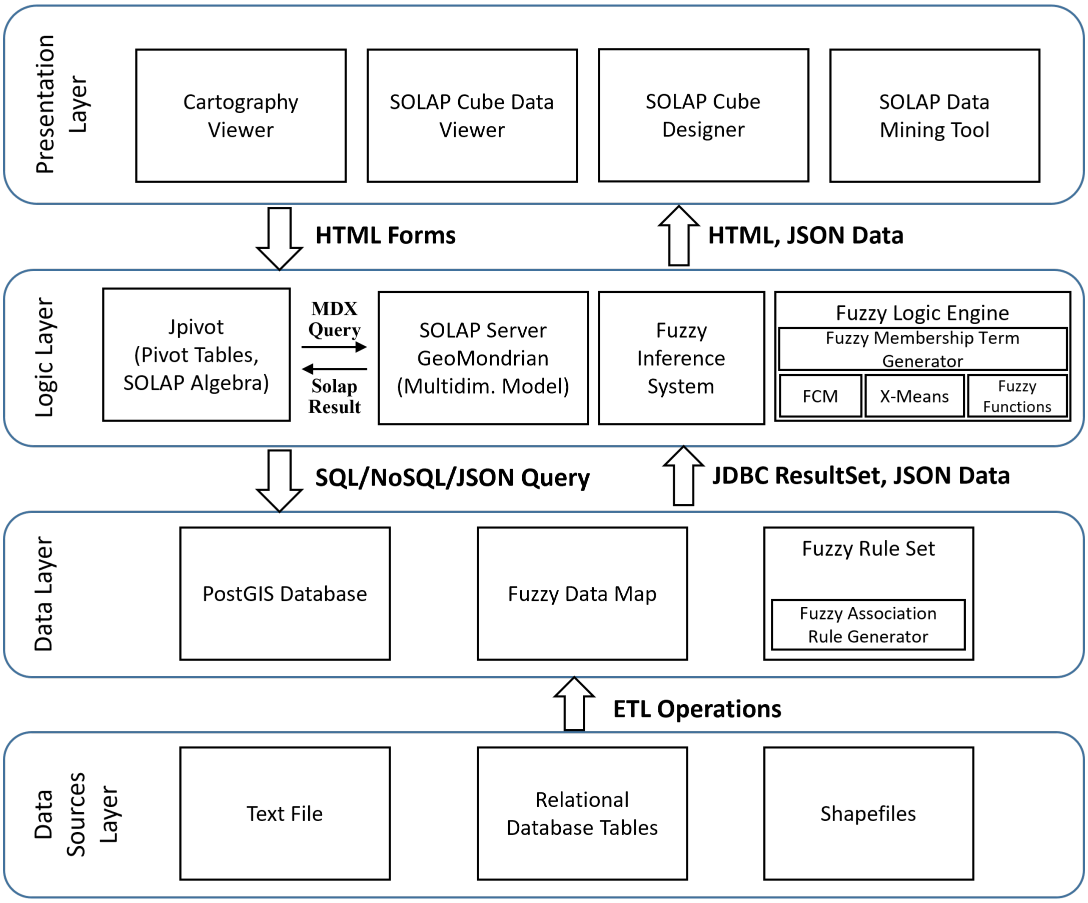


Figure 3.4: Layers of the FSOLAP framework

The data sources layer is at the bottom of the frame and feeds the system with plain data. The structured data layer is at the second level of the architecture and contains structured data. Then, the logical layer is built one level up that handles fuzzy operations, query operations, and prediction processes. The presentation layer, which supplies user interfaces and visualization components, is at the top of the framework.

As the first layer below the framework, the data comes in the form of text files containing raw data, unstructured data collected from the website, semi-structured database tables, and shape files for spatial data. This data is moved to a higher layer by structuring it with Extract Transform and Load (ETL) operations. During this process, raw or semi-structured data is read and wrong or missing data is determined and cleaned up in the preprocessing process, and validation is performed to assure the information is consistent.

The next layer is the data layer, where the clean and structured data is stored. Unlike a lower layer, the connection between the relational database tables is defined in this layer. The relationship between the tables may be associated with the snowflake schema or star schema structure. We chose snowflake because multidimensional data is better managed with the snowflake schema. Snowflaking is better when the dimension table is relatively big in size as it reduces the space. Also, the snowflake schema is easier to maintain and change as data is less duplicative. It supports complex queries, although there are more foreign keys. More joins cause a longer query execution time. It is also suitable for data warehouse core to simplify complex relationships. In this study we use the snowflake schema model to connect tables. In addition, spatial data in shape files are held in the [48] database in polygon format. Also, this layer contains the fuzzy data generated by the fuzzification process. As another generated information, the fuzzy rule set produced as a result of fuzzy association rule generation is also comprised in this layer. Users can execute SQL queries through this layer and call JavaScript Object Notation (JSON) requests for retrieving data. While SQL query results are presented as Java Database Connectivity (JDBC) result sets or SQL tuples, JSON formatted responses are returned in a standard response structure.

In the middle of the framework, the logic layer, which contains the SOLAP server, Fuzzy Inference System (FIS) and Fuzzy Logic Engine components, manages fuzzy logic operations and fuzzy inference mechanisms along with the data mining tools. It also provides querying with a multidimensional expression (MDX) on the data cube in the SOLAP server. The Fuzzy Logic Engine supports precise-to-fuzzy transformations and fuzzy-to-crisp conversion as part of defuzzification, and fuzzy class identification processes are handled by performing membership computations before these. In the details of these processes, fuzzy clustering algorithms are performed. The inference capability of the framework comes with FIS. This layer manages the logical operations between the presentation and data layers and performs spatial data mining tasks.

In this study, we aim to build a framework that makes spatial data analysis, querying, and inference by combining the strengths of SOLAP in the analysis of multidimensional data and the advantages of fuzzy logic in handling complex problems. For this purpose, we first provide the modeling of fuzzy spatial data on OLAP. In this modeling, the number of clusters for each attribute of the multidimensional data is determined, and fuzzification is performed. The association rule set is generated through the fuzzy dataset. The rules in the rule set are pruned and weighted. FIS is built with the obtained association rules, fuzzy membership classes, and membership functions.

The framework supports fuzzy spatial, temporal, and spatiotemporal types of complex queries. In order to do this, fuzzy regions and topological relations, which are spatial features, need to be addressed. Figure 3.5(a) represents a fuzzy region as an illustration and shows the core, the indeterminate boundary, exterior, and $\alpha - cut$ levels according to the membership grades. Figure 3.5(b) shows the fuzzy topological relationships [49] between fuzzy regions.

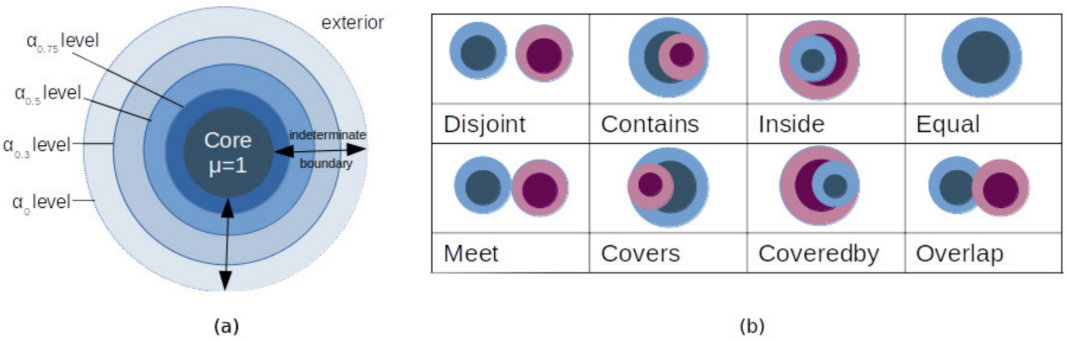


Figure 3.5: (a) $\alpha - cut$ levels of a fuzzy region. (b) The topological relations of the two fuzzy regions.

The detailed explanation of the fuzzy topological relations are given in [47, 50, 51, 52]. According to these definitions, $R1$ and $R2$ are two fuzzy regions, and $\tau(R1, R2)$ is the function that represents the topological relation between them. To consider $\alpha - cut$ level, $R1_{\alpha i}$ and $R2_{\alpha j}$ are the $\alpha - cut$ level regions and $\tau(R1_{\alpha i}, R2_{\alpha j})$ is the topological relation between these regions. The generalized form of the function can be specified as

$$\tau(R1, R2) = \sum_{i=1}^{n}\sum_{j=1}^{m} m(R1_{\alpha i})m(R2_{\alpha j})\tau(R1_{\alpha i}, R2_{\alpha j}) \tag{3.1}$$

The derivation of this function for the overlap relation can be given as an example:

$$\tau(R1, R2) = \sum_{i=1}^{n}\sum_{j=1}^{m} m(R1_{\alpha i})m(R2_{\alpha j})\tau_{overlap}(R1_{\alpha i}, R2_{\alpha j}) \tag{3.2}$$

The presentation layer at the top of the framework includes the components that users use interactively, as shown in Figure 3.4. This layer includes visual components, SOLAP cube designer, SOLAP cube viewer, and query interfaces. The components of this layer make it as easy to use as possible by allowing interaction between the framework and the user. The SOLAP cube designer provides a new cube metadata definition, and similarly the cube viewer provides a visual display of the constructed cube.

The implementation part of the study includes a fuzzy logic-based OLAP spatial framework (FSOLAP) which mainly consists of PostGIS, SOLAP, fuzzy logic module, association rule generation and fuzzy inference system. The steps of the prediction workflow using these components are shown in Figure 3.6. In the first step, a training data set is retrieved from data warehouse tables, and then the proper number of clusters is obtained with X-Means clustering [53] on this data. The data is fuzzified by running Fuzzy C-Means (FCM) [54] with the number of clusters found, and membership classes and functions are obtained. And a fuzzy SOLAP cube is assembled with fuzzified data. A fuzzy association rule set is generated with Frequent Pattern Growth (FP-Growth) [55] on the fuzzified dataset. By pruning and weighting this rule set, the rule set is assembled better for inference. FIS is built by using membership classes, membership functions, and association rules. During the prediction process of FIS, inferences are made by executing FIS with testing data.

The data selection process in the framework is accomplished by executing the MDX query on the SOLAP cube and fetching a sub-cube. The definition of a sub-cube be-

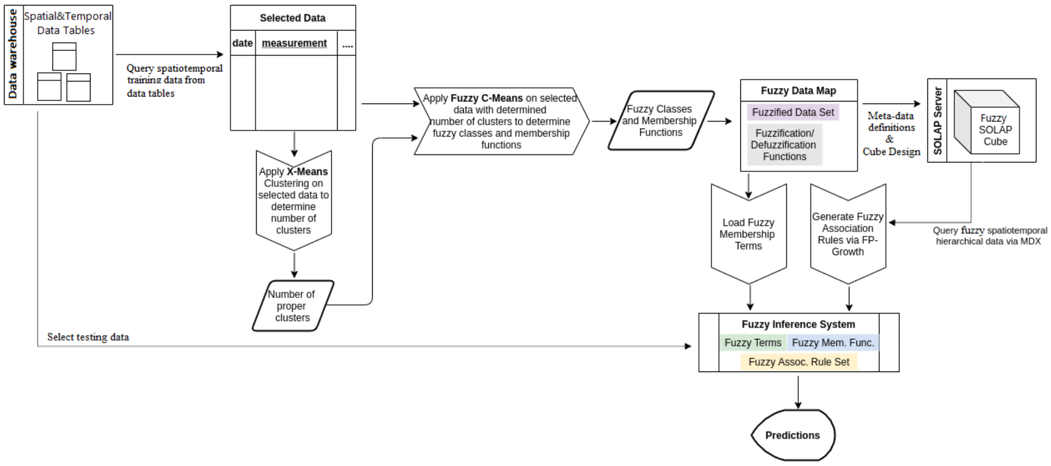

Data warehouse
Spatial&Temporal Data Tables
Query spatiotemporal training data from data tables
Selected Data
date
measurement
....
Apply X-Means Clustering on selected data to determine number of clusters
Number of proper clusters
Apply Fuzzy C-Means on selected data with determined number of clusters to determine fuzzy classes and membership functions
Fuzzy Classes and Membership Functions
Fuzzy Data Map
Fuzzified Data Set
Fuzzification/ Defuzzification Functions
Meta-data definitions & Cube Design
SOLAP Server
Fuzzy SOLAP Cube
Load Fuzzy Membership Terms
Generate Fuzzy Association Rules via FP-Growth
Query fuzzy spatiotemporal hierarchical data via MDX
Select testing data
Fuzzy Inference System
Fuzzy Terms
Fuzzy Mem. Func.
Fuzzy Assoc. Rule Set
Predictions

Figure 3.6: Execution steps of data analytics and prediction with FSOLAP

longing to the SOLAP cube is as follows: Let $B_s \subseteq B$ be a non-empty set of $n$ dimensions $\{B_1, B_2, \ldots, B_n\}$ from the data cube $C(n \leq d)$. The n-tuple $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ defines a sub-cube on $C$ according to $B_s\, iff\ \forall_i \in \{1, \ldots, n\}, \sigma_i \neq \emptyset$ and there exists a unique $j$ such that $\sigma_i \subseteq A_{ij}$ and can be visualized as shown in Figure 3.7.

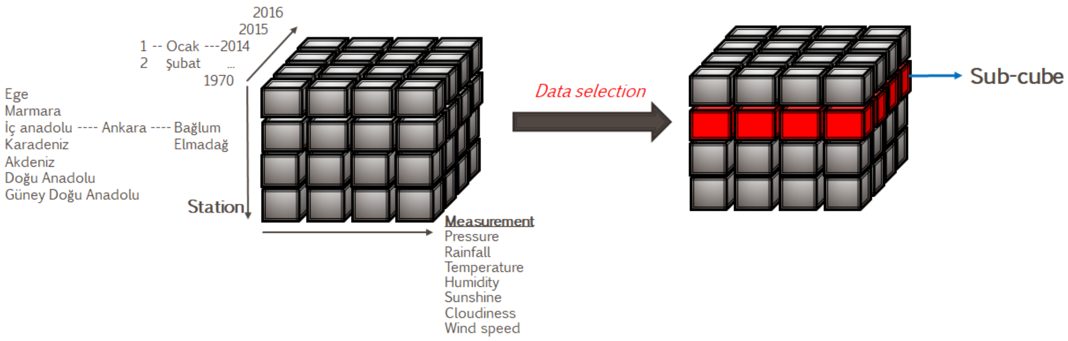


Figure 3.7: Sub-cube from SOLAP data selection.

The appropriate number of clusters is determined for each measurement data by X-Means [53] clustering. And then, Fuzzy C-Means (FCM) [54] is performed for fuzzy clustering and labeling. Fuzzy association rule sets and membership functions are acquired on fuzzy data. Next, the fuzzy control logic (FCL) is created for the Fuzzy Inference System (FIS). The FIS predicts by applying a fuzzy inference process that applies the membership functions to the fuzzy rules to emanate the fuzzy output. The prediction execution steps are given in Algorithm 1.

**Algorithm 1** Algorithm of fuzzy SOLAP based prediction

```
Input: Set of data $D$ of size N, set of attributes $A$ of size F, testing data ratio R
Output: Set of prediction $P$ of size N*R
    Initialization : Set the data $D$ in an accessible format
    $FD \leftarrow \{\}$            //all fuzzified data
    $FD_p \leftarrow \{\}$          //fuzzified data for an attribute
    $FT \leftarrow \{\}$            //all fuzzy terms for FIS
    $FT_p \leftarrow \{\}$          //fuzzy terms for an attribute
    $FAR \leftarrow \{\}$           //fuzzy association rules
    $Train_D \leftarrow \{\}$       //training data
    $Test_D \leftarrow \{\}$        //testing data
    $Train_D \leftarrow D$ of size $N * (1 - R)$
    for attribute $atr$ in $Train_D$ do
        $atrVals \leftarrow$ Select attribute $atr$ values from $Train_D$
        $k \leftarrow$ Execute X-means with $atrVals$ to get proper number of cluster
        $FD_p \leftarrow$ Execute FCM with $k$ and $atrVals$ then generate fuzzy classes and memberships
        $FD \leftarrow FD$ union $FD_p$
        for fuzzy class $f$ in $FD_p$ do
            $FT_p \leftarrow$ generate fuzzy terms for $f$ in form of triangular, trapezoidal or Gauss
            $FT \leftarrow FT$ union $FT_p$
        end for
    end for
    Create fuzzy SOLAP with $FD$
    Select subset $S$ from fuzzy SOLAP
    $FAR_k \leftarrow$ Execute FP-Growth with $S$
    $FAR_{k-n} \leftarrow$ Execute pruning on $FAR_k$
    $FAR \leftarrow$ Determine weight of rules with Rule Power Factor method on $FAR_{k-n}$
    $FIS \leftarrow$ Create fuzzy inference system with $FT$ and $FAR$
    $Test_D \leftarrow D$ of size $N * R$
    $P \leftarrow$ Execute FIS rules with $Test_D$ for prediction
    return $P$
```

The algorithm starts by accepting as parameters the D dataset containing N amount of data, the F feature set including several features in each record, and the percentage of data it will use for testing. Firstly, the training dataset considering the ratio of testing percentage is fetched by making a query containing the attribute to be estimated from the tables. Then, the loop is run for each attribute item. X-Means clustering is applied for the current feature in the loop, and the appropriate number of clusters of the relevant attribute is acquired. The fuzzification is performed using the calculated number of attribute clusters in Fuzzy C-Means clustering. So, membership classes and membership functions are determined for the related attribute. The fuzzy spatial cube is constructed by making SOLAP cube metadata definitions with fuzzified data. The fuzzy association rule set is generated by performing FP-Growth on the fuzzified training dataset. Unnecessary or repetitive rules are removed from the association rule set by pruning in a way that does not affect the prediction accuracy. The rules in the pruned rule set are weighted with the support of the rule power factor. The final rule set, the fuzzy membership classes, and the fuzzy membership functions are used in FIS. Also, testing data not in the training data set is fetched from the database tables. Prediction results are acquired by running the testing dataset with the FIS. The accuracy of the prediction results is calculated in performance evaluation.

The architectural environment of the FSOLAP framework is explained in the following subsections. In this part, meteorological measurements are given as an example to clarify our descriptions.

### 3.2.1 PostGIS Environment

Structured data after ETL operations are stored in PostGIS, which is an extension of the PostgreSQL database and provides spatial data operations.

The raw data in the text files are processed and inserted into the relevant database table. There are only stations with latitude, longitude, and altitude values in the raw data, but the region and city of the stations are not available. The region and city information are gathered, converted into polygons, and inserted into the relevant tables in the ETL phase. Fuzzy geometries are not implemented in PostGIS. They are calculated, stored, and retrieved from the fuzzy module. The fuzzy data map in the fuzzy module works as an in-memory database, and it provides fuzzify/defuzzfy methods for fuzzy operations of the data contained here. Fuzzy geometric attributes are defuzzified during query execution and operated with the help of the spatial aggregate functions supplied by the spatial extension of PostGIS. This operation provides spatial queries on hierarchical data.

### 3.2.2 SOLAP Environment

In order to support spatial and temporal queries on structured data, designing and building a SOLAP cube is necessary. While creating the meta-data of the SOLAP cube in the SOLAP cube designer, three types of primary elements are constructed. The first of these is the temporal hierarchy, which indicates the measurement date of the data, the second is the spatial hierarchy, which shows the measurement position of the data. And the last is the measurement results, which include the measurement values of the data. While the temporal hierarchy has day-month-year breakdowns, the spatial scale includes station-city-region levels. There are data belonging to ten different measurement results in the remaining dimensions. These are average temperature, sunshine hours, wind direction, average pressure, average humidity, vapor pressure, total precipitation (manual), cloudiness, average wind speed, and total precipitation (omgi). The visual representation of the cube design is shown in Figure 3.8. This figure shows spatial and temporal hierarchies, basic cube dimensions, and examples of operations on the cube.

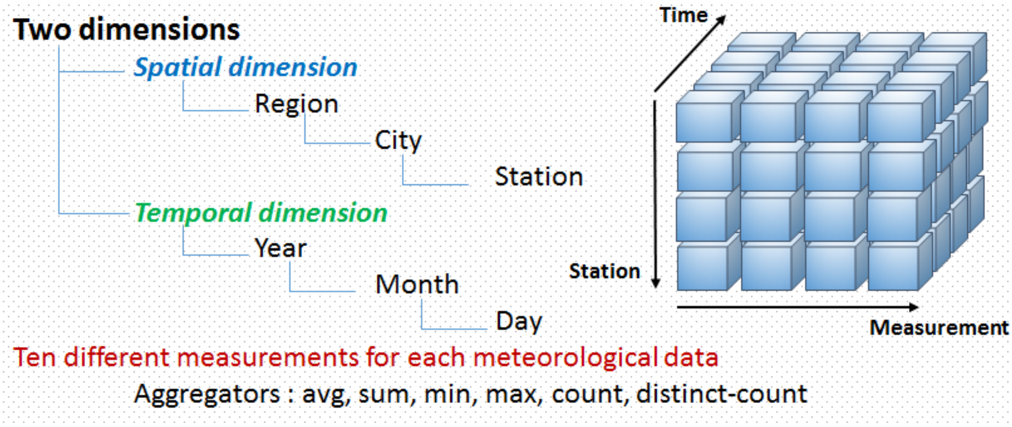

Two dimensions
Spatial dimension
Region
City
Station
Temporal dimension
Year
Month
Day
Time
Station
Measurement
Ten different measurements for each meteorological data
Aggregators : avg, sum, min, max, count, distinct-count

Figure 3.8: Dimensions and hierarchies of the SOLAP cube design

The Workbench [56] is used as the SOLAP cube meta-data design tool and is preferred for ease of integration with the SOLAP server. After the meta-data is designed, a connection with the PostGIS database is provided for the SOLAP server to manage the data using this metadata. Thus, MDX queries are executed on GeoMondrian, which we use as a SOLAP server. The drill up/down operations are performed on the data by displaying temporal and spatial hierarchical breakdowns in the MDX query results. In addition, Spatialytics [57], which is a visualization tool which is used to display the spatial data in the query results on the map. The methodology from processing and structuring the raw data to querying and showing the results is represented in Figure 3.9.

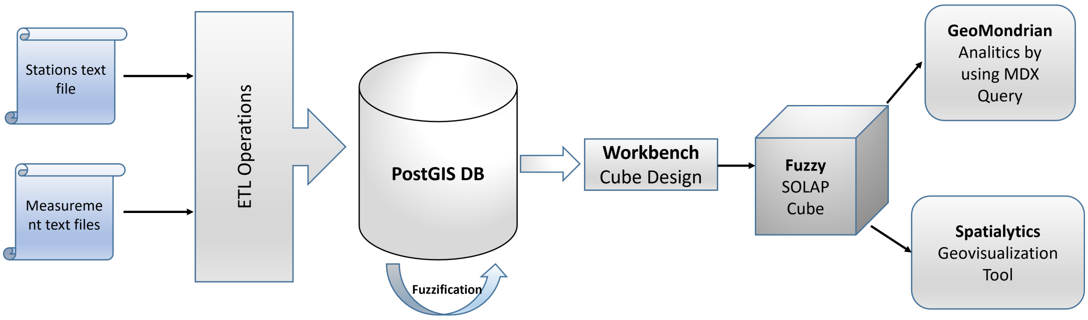


Figure 3.9: SOLAP data processing flow

Firstly, the files containing the station and measurement data are read, and the ETL process is performed. Then, the formatted data is inserted into the database to be structured. Fuzzification proceeds on the structured data for fuzzy transformation. Meta-data definitions are made using Workbench on the fuzzy data. A fuzzy spatial cube is designed on GeoMondrian with fuzzified data.

### 3.2.3 Fuzzy Environment

When generating fuzzy association rules using the SOLAP cube, it is necessary to fuzzify the meteorological data. In this context, nine different types of meteorological data are individually fuzzified. Thus, each measurement data item requires a fuzzy environment composed of fuzzy classes and membership functions. In this case, the number of clusters is determined for average pressure, cloudiness, vapor pressure, sunshine hour, the average wind speed, wind direction, average humidity, average temperature, total rainfall measurement by applying the fuzzification process. If the same number of clusters is used for each measurement, the accuracy of the predictions is reduced. For example, four different clusters, such as cold, normal, hot, and boiling, seem to be suitable for temperature measurement data. However, mostly cloudy, partly sunny, partly cloudy, sunny, overcast, broken, and fair for cloudiness measurement would be appropriate clusters. To determine the most appropriate number of clusters, the experiences of meteorologists are important. This study applies an approach to determine the proper number of clusters by considering each distribution of measurements in the dataset. In this context, the X-Means algorithm is used to determine the appropriate number of clusters for each feature in the data set. X-Means is an adaptation of the K-Means algorithm. We can also use Elbow or Silhouette methods for this operation, but X-Means was preferred in our study in terms of software ease of use. Also, cluster labeling is applied by considering the appropriate number of clusters, which is the output of X-Means algorithms. For this purpose, the meteorological literature terminology is investigated, and meaningful labels are used in the cluster labeling process domains. Then, the cluster labels are evaluated as membership class names during the fuzzification phase. The number of clusters as output of X-Means clustering and defined cluster labels are represented in Table 3.1. The cluster name list column in the table is the definitions used in labeling the membership classes made after the clustering process.

For each measurement of data, the number of cluster output of the X-Means clustering and the precise measurement value are given as input to Fuzzy C-Means clustering to calculate fuzzy membership values. After determining the appropriate number of clusters with the X-Means algorithm for each attribute in the dataset, fuzzy membership classes and functions are acquired with FCM. This process is done automatically in the framework.

Table 3.1: Determined clusters after FCM execution

| Measurement | Clusters | Cluster Name List |
|---|---|---|
| Actual Pressure | 6 | very-low, low, normal, high, very-high, extreme |
| Cloudiness | 8 | mostly-cloudy, partly-sunny, partly-cloudy, mostly-Sunny, overcast, broken, sunny, fair |
| Rainfall | 8 | nearly-dry, very-low, low, normal, high, very-high, over-much, flood |
| Humidity | 4 | nearly-dry, low, normal, high |
| Sunshine hour | 4 | nearly-dark, low, normal, high |
| Temperature | 4 | cold, normal, hot, boiling |
| Vapor Pressure | 8 | very-low, low, below-normal, normal, above-normal, high, very-high, extreme |
| Wind Speed | 8 | very-low, low, below-normal, normal, above-normal, high, very-high, extreme |

The algorithm of X-Means clustering is shown in the Algorithm 2 which includes primarily two operations such as improve parameters and improve structure and repeats until fulfillment. This clustering algorithm is an adaption of K-means that purifies cluster appointments by continually trying subdivisions and maintaining the most suitable resulting splits until some criterion is reached [53].

The objective function of the K-means algorithm is as follows:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} || \, x_i^j - c_j \, ||^2 \tag{3.3}$$

where $|| \, x_i^j - c_j \, ||^2$ is a selected distance measure between a data point $x_i^j$ and the cluster center $c_j$, is an demonstrator of the distance of the $n$ data points from their relevant cluster centers.
The predefined number of clusters is given as a parameter to Fuzzy C-Means (FCM) [54] clustering algorithm during the fuzzification of each attribute. Choosing a particular clustering algorithm depends solely on the type of data to be clustered and the

**Algorithm 2** X-means Clustering Algorithm

```
Input: initial data set: $d_1$,...,$d_n$
Output: $C \leftarrow$ number of clusters
Improve-Params $\leftarrow$ drive traditional K-means in conjunction
Improve-Structure $\leftarrow$ detect whether new centroids and determine the point should be arise
if $C > C_{max}$ then
    stop and report the most acceptable scoring instance found during the quest
else if $C <= C_{max}$ then
    Go to 1
end if
return $C$
```

purpose of the clustering applications. A hard clustering algorithm like the K-Means algorithm is suitable for exclusive clustering tasks; conversely, a fuzzy clustering algorithm like FCM is ideal for overlapping clustering tasks. In some situations, we cannot directly consider that data belongs to only one cluster. It may be possible that some properties of data contribute to more than one cluster. As in the case of document clustering, a particular document may be categorized into two different categories. For those purposes, we generally prefer membership value-based clustering like FCM. The objective function of the FCM is defined as follows:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} \mid\mid x_i - c_j \mid\mid^2, 1 \leq m < \infty \tag{3.4}$$

where $m$ is any real number greater than 1, $u_{ij}$ is the degree of membership of $x_i$ in the cluster j, $x_i$ is the $ith$ of d-dimensional measured data, $c_j$ is the d-dimension center of the cluster, and $\mid\mid * \mid\mid$ is any norm expressing the similarity between any measured data and the center [54]. Fuzzy partitioning is taken out via an iterative optimization of the objective function represented above, with the update of membership $u_{ij}$ and the cluster centers $u_j$ by:

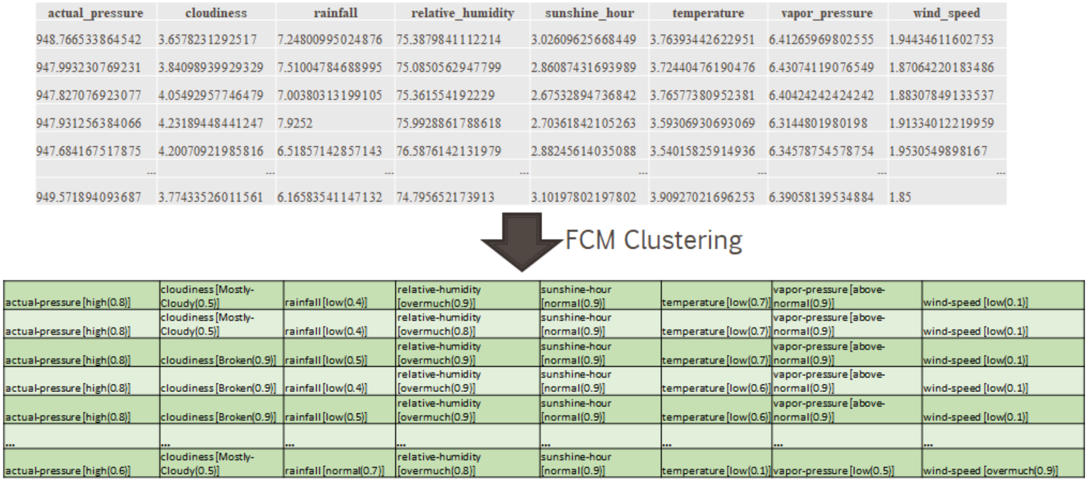
$$u_{ij} = \left( \frac{1}{\sum_{k=1}^{C} u_{ij} (\frac{||x_i - c_j||}{||x_i - c_k||})^{\frac{2}{m-1}}} \right) \tag{3.5}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m}.x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{3.6}$$

This repetition continues until $max_{ij} = |u_{ij}^{(k+1)} - u_{ij}^{(k)}| < \delta$, where $\delta$ is an ending criterion between 0 and 1, whereas the number of $k$ is the iteration steps. This procedure converges to a local minimum or a tackle point of $J_m$. The algorithm steps are given as follows:

1. Initialize $U = [u_{ij}]$ matrix, $U^{(0)}$
2. At k-step: calculate the center vectors $C^{(k)} = [c_j]$ with $U^{(k)}$
$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m}.x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{3.7}$$
3. Update $U^{(k)}$, $U^{(k+1)}$
$$u_{ij} = \left( \frac{1}{\sum_{k=1}^{C} u_{ij} \left( \frac{||x_i - c_j||}{||x_i - c_k||} \right)^{\frac{2}{m-1}}} \right) \tag{3.8}$$
4. If $||\ U^{(k+1)} - U^{(k)}\ || < \delta$ then STOP, otherwise return to step 2.

For example, the fuzzification inputs and outputs for meteorological measurements is shown in Figure 3.10.



| actual_pressure | cloudiness | rainfall | relative_humidity | sunshine_hour | temperature | vapor_pressure | wind_speed |
|---|---|---|---|---|---|---|---|
| 948.766533864542 | 3.6578231292517 | 7.24800995024876 | 75.3879841112214 | 3.02609625668449 | 3.76393442622951 | 6.41265969802555 | 1.94434611602753 |
| 947.993230769231 | 3.84098939929329 | 7.51004784688995 | 75.0850562947799 | 2.86087431693989 | 3.72440476190476 | 6.43074119076549 | 1.87064220183486 |
| 947.827076923077 | 4.05492957746479 | 7.00380313199105 | 75.361554192229 | 2.67532894736842 | 3.76577380952381 | 6.40424242424242 | 1.88307849133537 |
| 947.931256384066 | 4.23189448441247 | 7.9252 | 75.9928861788618 | 2.70361842105263 | 3.59306930693069 | 6.3144801980198 | 1.91334012219959 |
| 947.684167517875 | 4.20070921985816 | 6.51857142857143 | 76.5876142131979 | 2.88245614035088 | 3.54015825914936 | 6.34578754578754 | 1.9530549898167 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 949.571894093687 | 3.77433526011561 | 6.16583541147132 | 74.795652173913 | 3.10197802197802 | 3.90927021696253 | 6.39058139534884 | 1.85 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| actual-pressure [high(0.8)] | cloudiness [Mostly-Cloudy(0.5)] | rainfall [low(0.4)] | relative-humidity [overmuch(0.9)] | sunshine-hour [normal(0.9)] | temperature [low(0.7)] | vapor-pressure [above-normal(0.9)] | wind-speed [low(0.1)] |
| actual-pressure [high(0.8)] | cloudiness [Mostly-Cloudy(0.5)] | rainfall [low(0.4)] | relative-humidity [overmuch(0.8)] | sunshine-hour [normal(0.9)] | temperature [low(0.7)] | vapor-pressure [above-normal(0.9)] | wind-speed [low(0.1)] |
| actual-pressure [high(0.8)] | cloudiness [Broken(0.9)] | rainfall [low(0.5)] | relative-humidity [overmuch(0.9)] | sunshine-hour [normal(0.9)] | temperature [low(0.7)] | vapor-pressure [above-normal(0.9)] | wind-speed [low(0.1)] |
| actual-pressure [high(0.8)] | cloudiness [Broken(0.9)] | rainfall [low(0.4)] | relative-humidity [overmuch(0.9)] | sunshine-hour [normal(0.9)] | temperature [low(0.6)] | vapor-pressure [above-normal(0.9)] | wind-speed [low(0.1)] |
| actual-pressure [high(0.8)] | cloudiness [Broken(0.9)] | rainfall [low(0.5)] | relative-humidity [overmuch(0.9)] | sunshine-hour [normal(0.9)] | temperature [low(0.6)] | vapor-pressure [above-normal(0.9)] | wind-speed [low(0.1)] |
| ... | ... | ... | ... | ... | ... | ... | ... |
| actual-pressure [high(0.6)] | cloudiness [Mostly-Cloudy(0.5)] | rainfall [normal(0.7)] | relative-humidity [overmuch(0.8)] | sunshine-hour [normal(0.9)] | temperature [low(0.1)] | vapor-pressure [low(0.5)] | wind-speed [overmuch(0.9)] |

Figure 3.10: Fuzzification of meteorological measurements

After the fuzzification step, the fuzzy values of each measurement data are stored in the text file. Then, the fuzzy measurement data stored in separate files are merged by unification and transferred into a single structure. Here, the measurement date and station information are taken into account.

Within the scope of the fuzzy environment, membership functions are prepared and stored in triangular and trapezoidal forms. There are different forms of membership functions such as Triangular, Trapezoidal, Piecewise linear, Gaussian, and Singleton. The most straightforward membership functions are formed using straight lines. These straight-line membership functions have the advantage of simplicity. These are the triangular membership function and trapezoidal membership function. When we use simple, intuitively clear methods, we utilize both the formulas and our intuition. While creating membership functions, the triangular form is used as an isosceles triangular structure to be the peak value. The cluster value is determined as the midpoint of the upper base of the trapezoid to generate a trapezoidal form. And the isosceles lateral edges are concluded by considering the data frequency values. Figure 3.11 shows the fuzzy membership functions constructed for relative humidity, temperature, sunshine hour, and actual pressure. These membership functions are used in FIS for inferencing and explained in the following subsections.

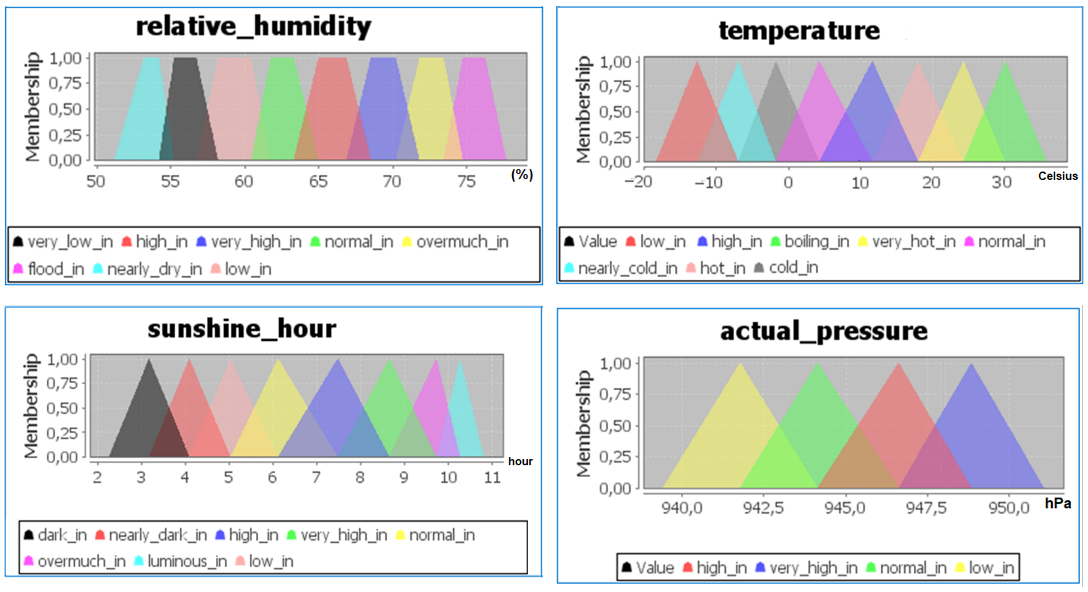
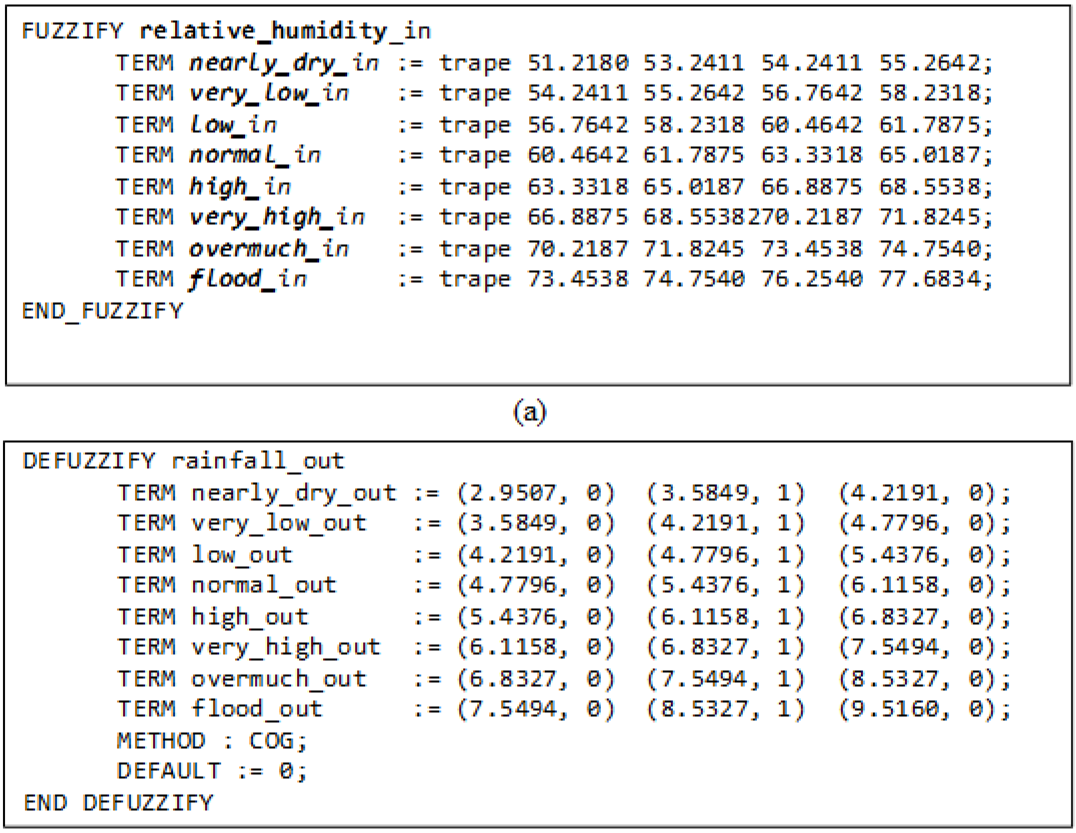


Figure 3.11: Fuzzy membership function of all measurements

As shown in the figure, relative humidity has eight clusters, and the trapezoidal membership function, and rainfall also have eight clusters and triangular membership functions. A detail of the membership functions for each measurement is given in the following figures. Figure 3.12(a) shows trapezoidal membership functions of relative humidity. Here, we provide the trapezoidal formation by referencing each term's points after the $trape$ keyword. Similarly triangular membership functions of rainfall are also shown in Figure 3.12(b). A triangular function is defined by defining three points of the triangle for each fuzzy membership class.



```
FUZZIFY relative_humidity_in
      TERM nearly_dry_in := trape 51.2180 53.2411 54.2411 55.2642;
      TERM very_low_in   := trape 54.2411 55.2642 56.7642 58.2318;
      TERM low_in        := trape 56.7642 58.2318 60.4642 61.7875;
      TERM normal_in     := trape 60.4642 61.7875 63.3318 65.0187;
      TERM high_in       := trape 63.3318 65.0187 66.8875 68.5538;
      TERM very_high_in  := trape 66.8875 68.5538270.2187 71.8245;
      TERM overmuch_in   := trape 70.2187 71.8245 73.4538 74.7540;
      TERM flood_in      := trape 73.4538 74.7540 76.2540 77.6834;
END_FUZZIFY
```

(a)

```
DEFUZZIFY rainfall_out
      TERM nearly_dry_out := (2.9507, 0)  (3.5849, 1)  (4.2191, 0);
      TERM very_low_out   := (3.5849, 0)  (4.2191, 1)  (4.7796, 0);
      TERM low_out        := (4.2191, 0)  (4.7796, 1)  (5.4376, 0);
      TERM normal_out     := (4.7796, 0)  (5.4376, 1)  (6.1158, 0);
      TERM high_out       := (5.4376, 0)  (6.1158, 1)  (6.8327, 0);
      TERM very_high_out  := (6.1158, 0)  (6.8327, 1)  (7.5494, 0);
      TERM overmuch_out   := (6.8327, 0)  (7.5494, 1)  (8.5327, 0);
      TERM flood_out      := (7.5494, 0)  (8.5327, 1)  (9.5160, 0);
      METHOD : COG;
      DEFAULT := 0;
END DEFUZZIFY
```

(b)

Figure 3.12: Membership function of relative humidity (a) and rainfall (b)

### 3.2.4 Fuzzy Association Rule Generation

In our study, a fuzzy association rule set is generated to make predictions with fuzzified data. Association rules are composed using frequent pattern generation algorithms. These algorithms are generally divided into two main groups, including candidate generation or not. The Frequent Pattern Growth (FP-Growth) [55] algorithm finds frequent itemsets without candidate generation. On the other hand, the Apriori algorithm [58] has a candidate generation phase. This study performs both approaches to notice which one brings better performance.

All non-empty sets included in a frequent itemset must also be frequent in the apriori algorithm. This situation is a fundamental property called apriori property. In addition, the non-empty itemset that provides minimum support can be used in association rule production. The Apriori algorithm keeps the candidate itemset it finds in each step. Without the database transactions, the next pass creates the candidate itemset using the essential itemset in the previous step.

Apriori and apriori-like algorithms have two crucial problems. One of them is they generate a vast amount of candidate sets, and the other one, they are continuously scanning the database and specifying a broad set of candidates by using pattern matching [24]. Han et al. proposed the FP-Growth method to overcome the weaknesses of the Apriori algorithm. Their approach mines frequent itemsets without candidate generation. A highly compressed data structure, called Frequent Pattern Tree (FP-Tree) [55], is constructed to condense the original transaction database. It separates the compressed database into a set of dependent databases. Each database mines individually and every item is associated with one frequent item. The FP-Growth algorithm annihilates both disadvantages of the Apriori algorithm. FP-Growth algorithm functions are much better than all its ancestors due to the following reasons:

1. First of all, FP-Tree describes a compact exposition of the original database because it is built by using only frequent items. Other irrelevant information is eliminated.

2. Secondly, it increases efficiency in that the database scan twice only.

3. Lastly, FP-Tree embraced the divide and conquer approach, reducing the conditional FP–Tree size.

In FP-Tree, the user cannot change the support and confidence threshold value during the process. This situation is the disadvantage of FP-Tree, so it is not proper for interactive and incremental mining. However, databases are dynamic. When new transactions occur in the database, this insertion may repeat the whole association rule mining process [24]. Association is about exploring rules related to the items which occur together in a transaction (i.e., buying something in a market transaction).

Association rule mining has some formal definitions, which can be explained as follows: Let $T = \{t_1, t_2, \cdots, t_n\}$ is a transaction which contains items. These items have a set of $k$ binary attributes. Let $D = \{d_1, d_2, \cdots, d_m\}$ is a database which includes a set of transactions. $\forall d \in D$ has a unique transaction ID and $t_s$ where $t_s \subseteq T$. $A \Rightarrow B$ can be defined as a rule in an implication form, where $B \subseteq T$. This rule definition can only be between a set and a single item, $A \Rightarrow i_j$ for $i_j \in T$. In composition of a rule two different sets of items, $A$ and $B$, are used as itemsets, where

- $A$ is antecedent or also called left-hand-side (LHS)
- $B$ is consequent or also called right-hand-side (RHS)

In this study, both Apriori and FP-Growth are used to generate a fuzzy association rule set. FP-Growth is better in execution time; however, Apriori exists better memory usage. Table 3.2 shows the number of fuzzy association rules produced due to FP-Growth, executed with additional support and confidence values. The maximum accuracy rate of predicted measurement and the execution time for rule generation are also given in the table. As shown in the sixth line of the table, when minimum support is 0.05, and minimum confidence is 0.5, 2028 association rules are produced in 809 milliseconds. After the pruning process, when the system discarded the extra or duplicative ones, 1894 rules remained. An accuracy of 90.63 percent is obtained when the remaining rules are weighted and used in the prediction process with sample testing data. When a higher confidence value as 0.6 is selected in the bottom line, fewer rules are generated, and the prediction accuracy falls. For this reason, the minimum support 0.05 and minimum confidence 0.5 values are determined as appropriate values in the association rule generation method.

A sample association rule set generated by using the FP-Growth algorithm is given as follows. The form of association rules $[A, B] \rightarrow C$ in the examples is shown in the upper section of each Table 3.3 row and $if..then..$ form is shown in the lower section. Support, confidence, lift, and leverage values for the rules are also given in the upper part.

Table 3.2: Generated number of the rule according to the support and confidence

| min support | min confidence | number of rule generated | number of rule after pruning | max accuracy | rule generation time (ms) |
|---|---|---|---|---|---|
| 0.03 | 0.1 | 14084 | 9871 | 90.63 | 4061 |
| 0.05 | 0.1 | 3245 | 2942 | 90.63 | 1175 |
| 0.05 | 0.2 | 3193 | 2863 | 90.63 | 985 |
| 0.05 | 0.3 | 3002 | 2709 | 90.63 | 919 |
| 0.05 | 0.4 | 2599 | 2328 | 90.63 | 892 |
| **0.05** | **0.5** | **2028** | **1894** | **90.63** | **809** |
| 0.05 | 0.6 | 1538 | 1373 | 89.04 | 742 |

Table 3.3: Sample of generated fuzzy association rules

| Fuzzy Association Rules |
|---|
| actual-pressure-*[high(0.9)]*, sunshine-hour-*[overmuch(0.9)]* ⇒ wind-speed-*[high(0.3)]* (support = 0.19, confidence = 1.0, lift = 1.64, leverage = 0.07)<br>IF actual_pressure IS high AND sunshine_hour IS overmuch THEN wind_speed IS high |
| cloudiness-*[Mostly-Cloudy(0.5)]*, sunshine-hour-*[normal(0.9)]*, vapor-pressure-*[above-normal(0.9)]*, wind-speed-*[low(0.1)]* ⇒ actual-pressure-*[high(0.8)]* (support = 0.12, confidence = 1.0, lift = 1.69, leverage = 0.05)<br>IF cloudiness IS mostly_cloudy AND sunshine_hour IS normal AND vapor_pressure IS above_normal THEN actual_pressure IS high |
| cloudiness-*[Mostly-Cloudy(0.7)]*, relative-humidity-*[high(0.7)]*, sunshine-hour-*[high(0.5)]* ⇒ rainfall-*[very-low(0.8)]* (support = 0.03, confidence = 0.79, lift = 4.23, leverage = 0.02)<br>IF cloudiness IS mostly_cloudy AND relative_humidity IS high AND sunshine_hour IS high THEN rainfall IS very_low |

### 3.2.5 Fuzzy Association Rule Pruning Based on Confidence

As a result of association rule mining, there are many generated rules, as shown in the number of rule-generated columns in Table 3.2. The number of association rules in the FIS directly affects the performance at runtime. For this reason, it is essential to discard unnecessary or repetitive ones among the rules produced to get better execution time. There are two main approaches to decrease the number of the rules in the rule base, which are given as follows [59]:

- In a subjective method, some tools allow the user to specify which rules are potentially interesting and which are not, such as templates [60] and constraints [61, 62].

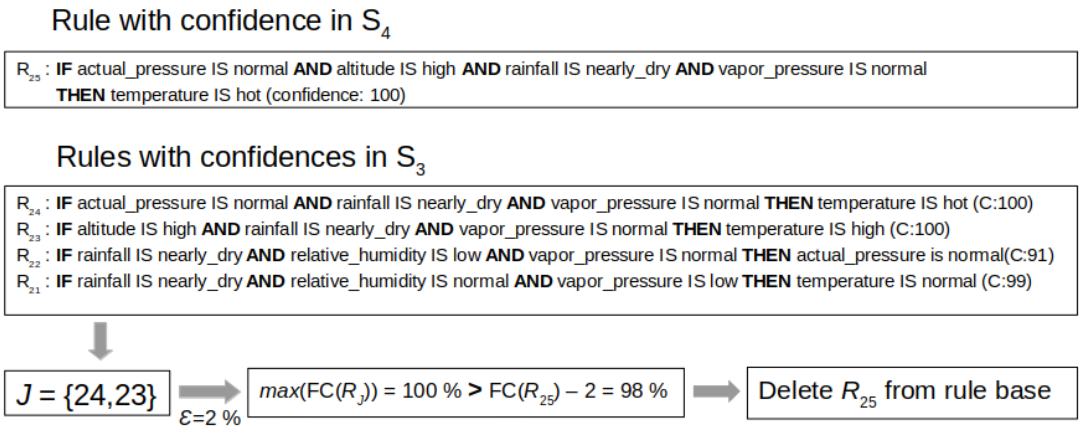
- In an objective method, user-independent quality standards are realized according to association rules. While interest depends on the user to a large extent, objective measures are needed to decrease the similarity inherent in a set of rules. Objective approaches can also be divided into two groups. One of them is pruning by determining the lift, support, or confidence values for each rule independently of the other rule. Another approach checks if it has an exact value of confidence and support to identify the rule with the most common condition and the most noticeable result. This approach potentially eliminates duplicate rules [63].

The number of all rules is $K$, but the rules can vary (a rule length is the number of antecedent fuzzy sets), and fuzzy confidence values belong to all rules. The monotonic property serves the benefit of the application of the fuzzy confidence measure. Namely, if given a rule of length $k$ with a maximal confidence of a fuzzy association rule ($FC$) value in the rule base and a rule of size $(k + 1)$ contain added input variable, then the $FC$ value of the rule improves, or it does not change. This study uses a rule-based pruning algorithm that withdraws the unnecessarily complicated rules as illustrated in Algorithm 3.

**Algorithm 3** Algorithm of Rule Pruning

```
Input: $R = \{R_1, ..., R_K\}$ is a set of fuzzy association rules
    $K \leftarrow$ size of rule set ($R_K$), $k$ = 1,...,$K$
    $T$ is an empty set
Output: $PR$: pruned fuzzy association rule base with reduced number of rule
for k = $K$,...,2 do
    for all $F \epsilon R_k$ do
        for all $F' \epsilon R_{k-1}$ do
            if size($F' \cap F$) = $k$ then
                $T \leftarrow T \cup$ index of $F'$
            end if
        end for
        if max($FC(F_T)) > FC(F)$) - $\varepsilon$ then
            remove rule $F$ from the rule base $PR$
        end if
    end for
end for
return $PR$
```

According to the algorithm, the most comprehensive rule which contains the most terms in its antecedent is selected. Then, shorter rules with fewer antecedents are chosen and compared with the extensive rule, taking the $FC$ values into account. In the comparison process, the $\varepsilon$ value as a correction factor is subtracted from the $FC$ value of the rule to be pruned, and the result is expected to be lower than the $FC$ value of the shorter rule. Determining the correction factor value is a complex process, and we chose this value as 2 percent in our study. As a result, pruning provides to design of a rule base with shorter rules. Although the number of rules has been reduced, the classification accuracy of the inference processes does not change. The system works better by not using unnecessary time and resources by operating comprehensive rules. We show the number of generated rules before applying the pruning method and the number of remaining rules after the pruning method in Table 3.2. Figure 3.13 represents a sample execution of the pruning method.


Rule with confidence in $S_4$

$R_{25}$ : **IF** actual_pressure IS normal **AND** altitude IS high **AND** rainfall IS nearly_dry **AND** vapor_pressure IS normal **THEN** temperature IS hot (confidence: 100)

Rules with confidences in $S_3$

$R_{24}$ : **IF** actual_pressure IS normal **AND** rainfall IS nearly_dry **AND** vapor_pressure IS normal **THEN** temperature IS hot (C:100)
$R_{23}$ : **IF** altitude IS high **AND** rainfall IS nearly_dry **AND** vapor_pressure IS normal **THEN** temperature IS high (C:100)
$R_{22}$ : **IF** rainfall IS nearly_dry **AND** relative_humidity IS low **AND** vapor_pressure IS normal **THEN** actual_pressure is normal(C:91)
$R_{21}$ : **IF** rainfall IS nearly_dry **AND** relative_humidity IS normal **AND** vapor_pressure IS low **THEN** temperature IS normal (C:99)

$J = \{24,23\}$
$\mathcal{E}$=2 %
$max(FC(R_J))$ = 100 % **>** FC($R_{25}$) – 2 = 98 %
Delete $R_{25}$ from rule base

Figure 3.13: A sample execution of the pruning method

### 3.2.6 Weighting The Fuzzy Association Rules

The design of a fuzzy inference system (FIS) can be decomposed from data into two major phases. The first is rule generation, leading to a basic design with a given space partitioning and corresponding rules. And the second is rule-based optimization which aims to select the most valuable rules and optimize rule conclusions. The number of rules is pruned in the rule base in the prior subsection. Here, determining vital rules for decision-making and avoiding misleading rules and decisions are explored. Considerable studies [64] have presented several interest measures for rule mining to fulfill practical decision-making needs. Commonly, types of classifications are given as objective and subjective. There is no user attention in an objective measure because it is generally built on probability, correlation, and statistics theories. On the other hand, a subjective measure considers both the data and the user.

The significance of the rule can be obtained by direct or indirect interaction with the user through the data mining process. Each rule in the rule base of FSOLAP is determined using traditional *IF state THEN decision [WITH significance]* clauses. The optional *WITH significance* statement provides weighting factors for each rule as to interest measure. This study uses interest measures with the name Rule Power Factor (RPF) [64] to assign weight to each fuzzy association rule and mine the fuzzy association rule between them. The formulation of RPF is defined in the following equation. Weights represent the relative importance of each rule. RPF focuses on the significance of the association between an antecedent and consequent rules. And it produces better results when support, confidence, lift, chi-square, and other measures fail [64].

$$rpf(A \rightarrow B) = support(A \cup B) * confidence(A \cup B) \tag{3.9}$$

The RPF produces better results even when the lift (a well-known and accepted measure of interest) fails. The following example illustrates how the RPF works:

- Case 1: If item X appeared in 30 transactions and Y in 60 out of 100 transactions and items, X and Y both together seem in 20 transactions.

  $Lift(X \rightarrow Y) = \frac{support(X \cup Y)}{support(X) * support(Y)}$

  $Lift = 0.2/(0.3 * 0.6) = 0.2/0.18 = 1.11$

  In the 100 transaction database, when items X and Y appear 20 times, the lift expresses positive for the rule.

- Case 2: If item X appeared in 40 transactions and Y in 70 out of 100 transactions and items, X and Y seemed together in 30 transactions.

  $Lift = 0.3/(0.4 * 0.7) = 0.3/0.28 = 1.07$

  Surprisingly, in the same 100 transaction database, when items X and Y appear 30 instead 20 times (case 2), lift says rule 1 is essential.

Now let's see RPF for both cases

$RPF = confidence(X \rightarrow Y) * support(X),$

- Case 1: $RPF = 0.66 * 0.2 = 0.13,$

- Case 2: $RPF = 0.75 * 0.3 = 0.22$

As a result, RPF correctly predicted that Case 2 is more important than Case 1.

### 3.2.7 Fuzzy Inference System

Fuzzy inference systems (FIS) are also known as fuzzy-rule-based systems, fuzzy models, fuzzy associative memories (FAM), or fuzzy controllers when used as controllers. Basically, a fuzzy inference system is composed of five functional blocks shown in Figure 3.14. The FIS contains the following components:

- a rule base including several fuzzy association rules
- a database which defines the membership functions of the fuzzy sets used in the fuzzy rules
- a decision-making unit that executes the inference operations on the rules
- a fuzzification/defuzzification interface which converts the crisp inputs into degrees of the match with linguistic values or via versa

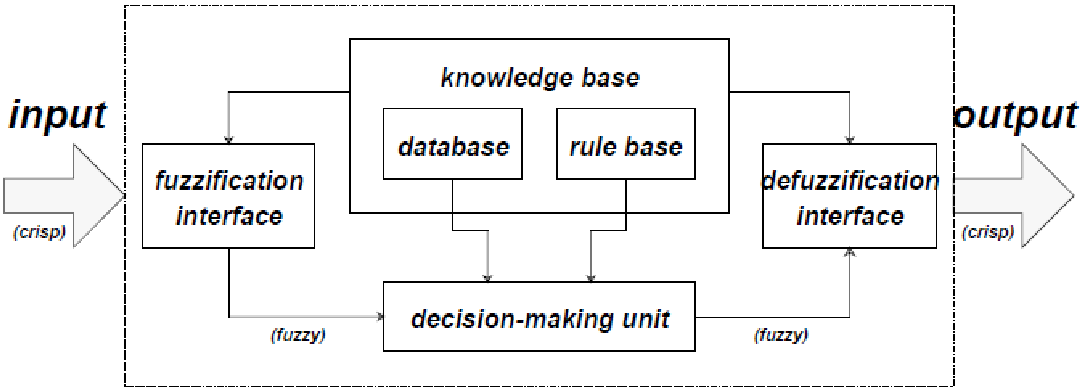


Figure 3.14: Fuzzy Inference System.

This system works as follows: $A' = F(x_0)$ where $x_0$ is a precise value defined in the input universe $\cup$, $A_0$ is a fuzzy set defined in the same universe and $F$ is a fuzzifier operator. The FIS is based on the application of the Generalized Modus Ponens, an extension of the classical Modus Ponens, proposed by Zadeh in which:

$$\frac{\text{(If X is A then Y is B)} \cap \text{(X is A')}}{\text{(Y is B')}} \tag{3.10}$$

in this equation, linguistic variables are $X$ and $Y$, the fuzzy sets are $A$ and $B$, and implication output is $B'$, which is also a fuzzy set. During the inference, the degree of matching is evaluated for each rule using a conjunctive operator; after that, a fuzzy implication operator is performed to deduce the implication. The size of the rules included in the FKB is equal to the number of rules constructed by the FIS.

So far, fuzzy association rules and membership functions are generated and stored in the fuzzy environment of FSOLAP. The framework also has a fuzzification interface for precise to fuzzy operations and vice versa a defuzzification interface for fuzzy to crisp transformation. The main component of FIS is jFuzzyLogic [23], an open-source fuzzy logic library.

FIS of FSOLAP uses a special file structure called Fuzzy Control Logic (FCL). The FCL file contains the membership functions for the fuzzy input fuzzification step. It also has the membership functions which are used in the defuzzification phase for the fuzzy output. Additionally, the FCL file contains the fuzzy association rules used for the prediction. When FIS is executed, precise values are provided as input; after the inputs are fuzzified with the fuzzification, the rule (s) containing the input on the left-hand side is selected. If more than one rule is chosen for execution, the highest weight value has become essential in determining the result. Once the chosen rules have been executed, the fuzzy value on the right-hand side is output.

There are at least one functional blocks that the FCL has. These blocks contain variables, fuzzy operations, and rule definitions represented as follows:

- two types of variables; one of them is input variable defined as $var\ input$, and the other is output variable specified as $var\ output$.
- two types of fuzzy operations; the $fuzzify$ function for precise to fuzzy transformation, other is the $defuzzify$ function vice versa.
- the $ruleblock$ section is the place of defined fuzzy rules.

Defining a variable is a simple operation that needs to write the variable name, type, and default value if necessary. Definition of a membership function in $fuzzify$ or $defuzzify$ starts with the $TERM$ statement and continues as a function definition for each lingual term. Functions are defined as piecewise linear functions using a series of points $(x_0, y_0)(x_1, y_1) \cdots (x_n, y_n)$, thus, a trapezoidal form of a membership function can be represented as $TERMcold := (10, 0)(15, 1)(20, 1)(25, 0)$.

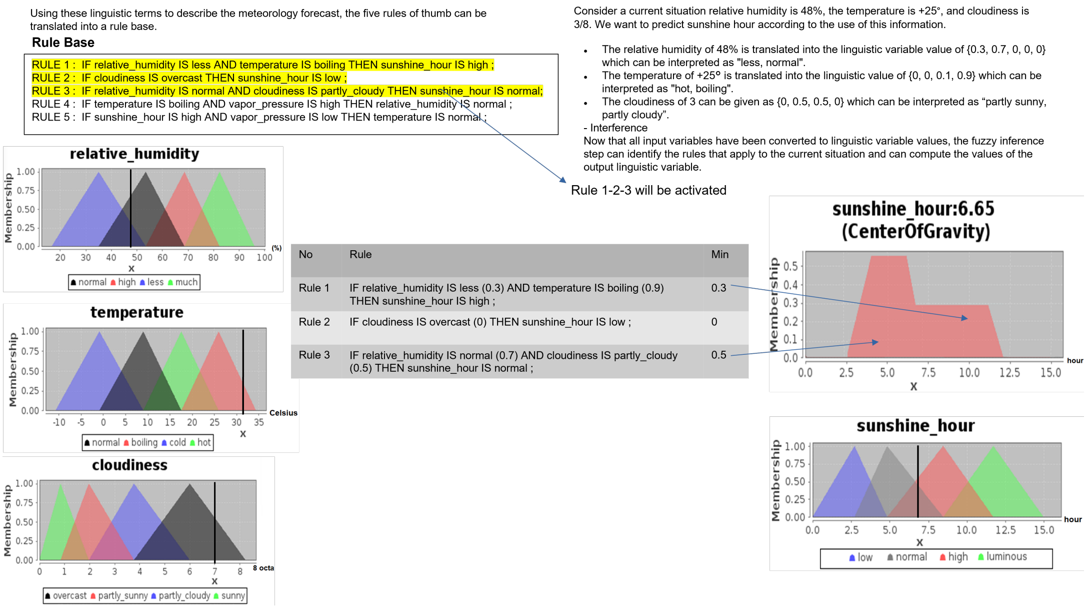
A $ruleblock$ contains fuzzy association rules, and FIS may have one or more of them. Since rules are naturally run simultaneously, they are not executed in any particular order. Each rule is defined using the classic *IF state THEN decision [WITH significance]* clauses. The weighting factor of a rule can be specified $WITHsignificance$ statement optionally. $IFclause$ part of the rule has test condition with the format of *"variable IS [NOT] linguistic term"*. Membership of a variable to a linguistic term is tested using the membership function in the relevant FUZZIFY block. $NOT$ operand can be used optionally to negate the membership function such as $mf(a) = 1 - mf(a)$. Various states can be joined using $OR$ and $AND$ operators [23].

#### 3.2.7.1 FCL example

Suppose we have a meteorology rule base with five rules and linguistic terms that refer to temperature, relative humidity, cloudiness, and vapor pressure values, as shown in Figure 3.15. The five fuzzy association rules defined in the rule base can be used to make meteorological predictions. For example, it is desired to predict how long the sunshine duration will be when the cloudiness is 3/8, the relative humidity is 48%, and the temperature is +25°.

- The cloudiness of 3 can be given as 0, 0.5, 0.5, 0, which can be interpreted as "partly sunny" and "partly cloudy".
- The membership function generates 0.3, 0.7, 0, 0, 0 values for 48% relative humidity, which can be translated as "less" and "normal" membership classes.
- The temperature of +25° has 0, 0, 0.1, 0.9 membership degrees after fuzzy transformation, and these values can be translated into the linguistic term as "hot" and "boiling".

Fuzzy inferences can be made after linguistic transformations of the inputs are made with the help of membership functions. First of all, it is necessary to determine which fuzzy association rules should be fired. Afterward, the determined rules will be executed with the input variables, and the predictions will be made. Figure 3.15 demonstrates how to select the appropriate rules and use the input variables, as well as the evaluation of the results.



Using these linguistic terms to describe the meteorology forecast, the five rules of thumb can be translated into a rule base.

**Rule Base**

RULE 1 : IF relative_humidity IS less AND temperature IS boiling THEN sunshine_hour IS high ;
RULE 2 : IF cloudiness IS overcast THEN sunshine_hour IS low ;
RULE 3 : IF relative_humidity IS normal AND cloudiness IS partly_cloudy THEN sunshine_hour IS normal;
RULE 4 : IF temperature IS boiling AND vapor_pressure IS high THEN relative_humidity IS normal ;
RULE 5 : IF sunshine_hour IS high AND vapor_pressure IS low THEN temperature IS normal ;

Consider a current situation relative humidity is 48%, the temperature is +25°, and cloudiness is 3/8. We want to predict sunshine hour according to the use of this information.

- The relative humidity of 48% is translated into the linguistic variable value of {0.3, 0.7, 0, 0, 0} which can be interpreted as "less, normal".
- The temperature of +25° is translated into the linguistic value of {0, 0, 0.1, 0.9} which can be interpreted as "hot, boiling".
- The cloudiness of 3 can be given as {0, 0.5, 0.5, 0} which can be interpreted as “partly sunny, partly cloudy”.

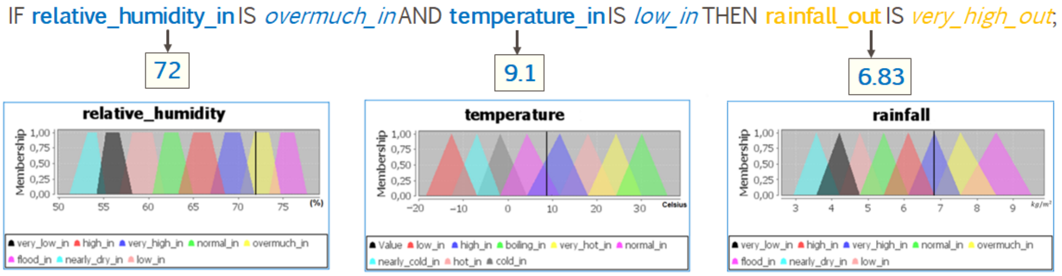
- Interference
Now that all input variables have been converted to linguistic variable values, the fuzzy inference step can identify the rules that apply to the current situation and can compute the values of the output linguistic variable.

Rule 1-2-3 will be activated

| No | Rule | Min |
|---|---|---|
| Rule 1 | IF relative_humidity IS less (0.3) AND temperature IS boiling (0.9) THEN sunshine_hour IS high ; | 0.3 |
| Rule 2 | IF cloudiness IS overcast (0) THEN sunshine_hour IS low ; | 0 |
| Rule 3 | IF relative_humidity IS normal (0.7) AND cloudiness IS partly_cloudy (0.5) THEN sunshine_hour IS normal ; | 0.5 |

Figure 3.15: Execution of a sample FCL with the given rules and inputs

Rule 1, Rule 2, and Rule 3 are selected to predict sunshine hour because they contain sunshine hour in the consequent part of the rule. Then, the membership values of the antecedent part are calculated, and the minimum membership value is specified for each rule. In the next step, the output results on the fuzzy membership set are determined, and the center of gravity method is used to find the sunshine hour as the final result. We show an example fuzzy association rule execution in FCL in Figure 3.16.



IF relative_humidity_in IS *overmuch_in* AND temperature_in IS *low_in* THEN rainfall_out IS *very_high_out*;

Figure 3.16: Execution of a sample fuzzy association rule

In this example, if the actual relative humidity is 0.72 and the actual temperature is 9.1 Celsius, then the rainfall will be 6.83 mm. The marking of input and output values on the membership function is also shown in the picture. Here, relative humidity and the actual temperature values are given as input, and FIS executes the rule and generates the rainfall output.

# CHAPTER 4

# FSOLAP QUERY MANAGEMENT

This section describes the architecture and query types that support fuzzy spatiotemporal queries on spatial OLAP-based structures. We mentioned querying data with the MDX query on the cube and fetching a sub-cube [50, 65]. The data selection process may not be so simple that it is limited to fetching sub-cubes by making only a few dimension restrictions on the MDX query. Depending on the complexity of the application domain, complex queries that include hierarchical attributes should also be supported. It is essential to provide hierarchical query support via structures defined in the designed metadata. SOLAP stores numeric and alphanumeric data in a hierarchical structure and allows hierarchical querying and analytics of this data. However, this is insufficient for spatiotemporal applications since complex queries are required. In our study, imprecise and fuzzy flexible queries on spatiotemporal data are considered as complex queries. Since spatiotemporal applications are complex applications, it is expected to flexibly support complex imprecise and fuzzy queries. In this context, FSOLAP provides data analytics on fuzzy data and effectively supports various types of fuzzy spatial and temporal queries. Fuzzy spatial queries are used to retrieve fuzzy spatial entities and their relationships; an example of a fuzzy spatial query in the form of the verbal language can be represented as follows:

**Query:** *Find the applicable cities for the installation of a wind power plant.*

The fuzzy rule set contains the following rule with linguistic terms about suitable locations for wind power plants.

```
if city.windspeed is high and city.position is south
                        then city.windpower is high
```

MDX queries, which are used for efficient data querying in OLAP databases, make flexible and fast queries on grouped data, unlike classical SQL queries [11]. Users can query OLAP cubes on analysis servers and develop various client applications with MDX queries. Although MDX does not emerge as a standard query language, it is widely used and generally accepted by companies that developed products with multidimensional query support over time. The query typed in the MDX format is as follows:

```
WITH MEMBER
  [Station.StationHierarchy].[Station Region]
       .[AKDENIZ REGION].[Around Somewhere in Akdeniz] AS
AGGREGATE(
  FILTER(
    [Station.StationHierarchy].[Station Region].members,
    fuzzify geo(st transform([Station].currentmember.PROPERTIES("geom"),
        8937, 6492), st transform(st geomfromtext("POINT(36.7871 34.2200)")
        , 8937, 6492))= "AROUND")
  ), geom=st transform(st geomfromtext("POINT(36.7871  34.2200)"), 4326,
      2991)
SELECT
    FILTER(  fuzzify measure([Measures].[wind speed]) ,
        fuzzify measure(AVG([Measures].currentmember.wind speed)) ="HIGH")),
    FILTER( fuzzify geo([Station].currentmember. PROPERTIES("geom"),
                [Station].[Station Region].PROPERTIES("geom"))="SOUTH")
     ON COLUMNS,

    [Station.StationHierarchy].[Station Region].[AKDENIZ REGION].children ,
    [Station.StationHierarchy].[All Stations].[AKDENIZ REGION].[Around
        Somewhere in Akdeniz]
   ON ROWS
FROM [MeteorologicalCube]
WHERE ([DateDimension1.Date Hierarchy 0].[All Dates])
```

This query determines the stations located in the south with high wind speed by filtering. Since there is no time restriction in the verbal query, the data in the entire time range is queried. The criteria in the query filter are determined by taking into account the fuzzy rule set regarding the installation of wind turbines. An expert defines these expert rules into the system and uses the fuzzy association rules that the framework automatically generates over the dataset during this process. The expert case mentioned here is a user with application domain knowledge. For our example, an expert is a wise person who has deep knowledge about meteorological events, explains the relations of weather events with each other, and can devise rules in this direction. Defining well-defined rules during this person's use of the system will be a factor that will increase the performance of the system.

Linguistic terms shown in the filtering section of the MDX query have been implemented to enable the use of imprecise criteria as a part of fuzziness. Accordingly, we developed fuzzify_geo and fuzzify_measure methods for fuzzifying spatial and non-spatial measurement data to provide the fuzzy capability to the query. These methods are used depending on whether the attributes handled in the query are spatial or not. The fuzzify_geo method is used for spatial features of the query. Thus, while providing fuzzy assets for spatial attributes in the MDX query, the fuzzify_measure method is similarly developed for non-spatial characteristics. While dealing with hierarchies and relations on spatial features, topological relations such as covers, inside, around, etc., are examined for two different spatial information in the fuzzify_geo method. The proposed study handles these methods as a fuzzy extension of MDX queries. We use GeoMondrian SOLAP Server [66] in this study, and query operations on this server are accomplished with the support of the geomondrian.jar Java library. This Java library is modified for fuzzy querying. The fuzzify _geo and fuzzify_measure methods are implemented in the base classes such as Parser.java, Query.java, and MondrianServerImpl.java. These developed method names are defined in the MondrianServerImpl.java class, including the Filter, Member, Where, etc., keywords used in the queries. Using these specified keywords, the Parser.java class enables the query to be split into its components. The Query.java class manages this process, and as a result, the sections and parameters of the query are determined. In addition, fuzzify _geo and fuzzify_measure methods should be integrated with the fuzzy module to perform their functions in the query at runtime. In this way, the fuzzy module allows the API method parameters to be fuzzified. While assembling the MDX query by the query processor, the relevant operator uses the fuzzy query parameters and fuzzy query criteria. The query is sent to the fuzzy module to fuzzify the features during execution, and the query structure becomes MDX form. Spatial features are also fuzzified during interrogation via the fuzzy module. The spatial functions provided by PostGIS are used for geometric calculations and relationships on spatial characteristics.

In the FSOLAP framework, we typically achieve query management through two main structures, as shown in Figure 4.1. One of these is the data layer, where we prepare, format, and query data. The other is the query module, which contains the frontend presented to the user for querying and query management components.

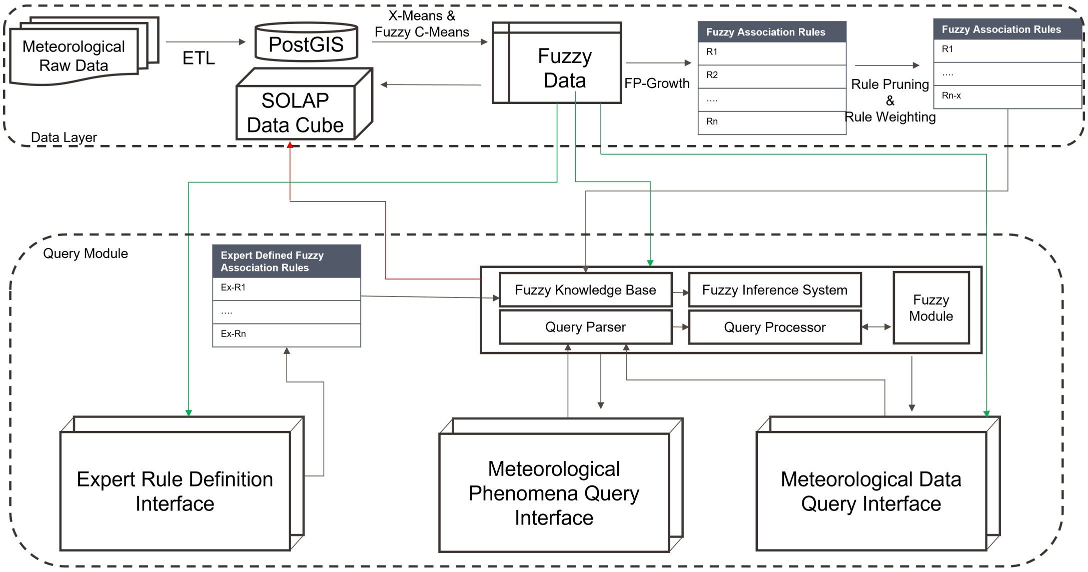
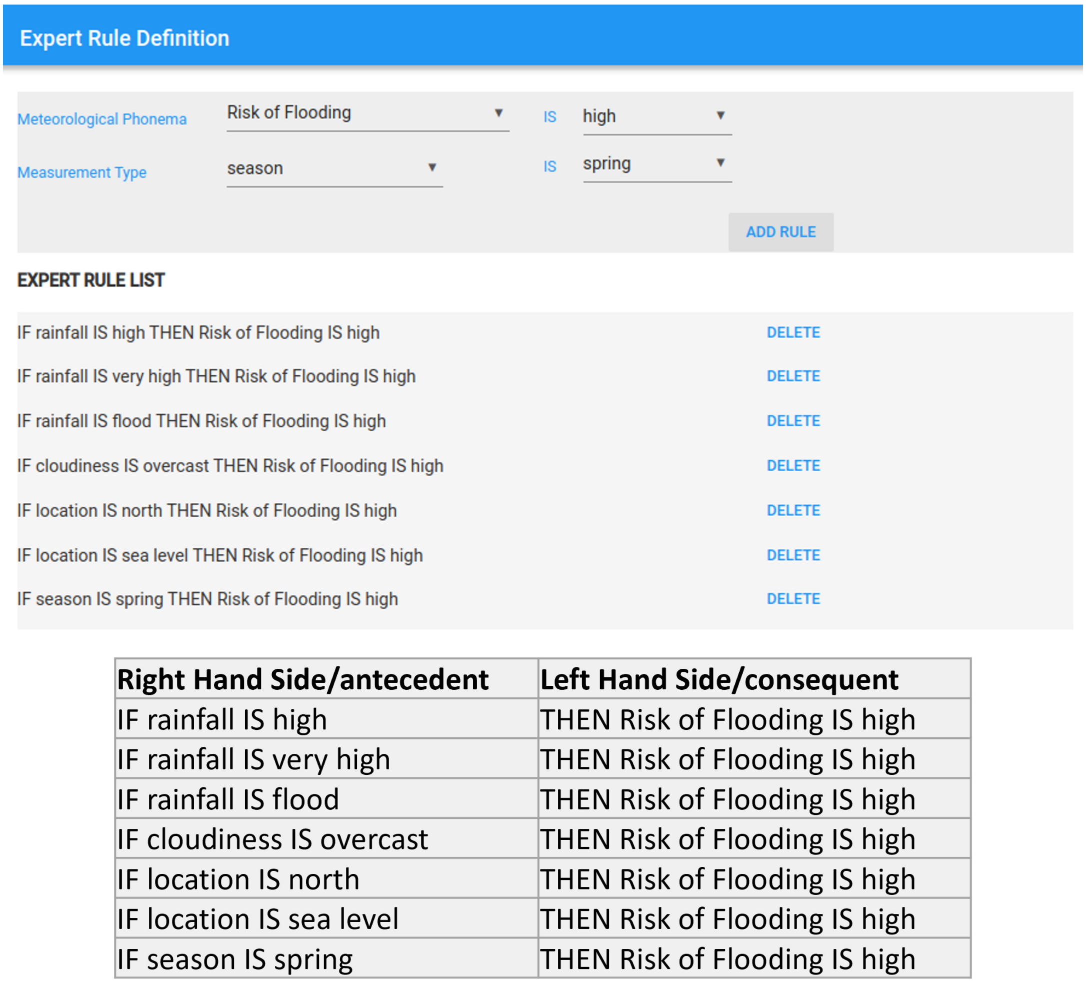


Figure 4.1: FSOLAP query management.

## 4.1 Query Module

The Query Module (QM) is responsible for query management in the FSOLAP framework. Figure 4.1 shows the sub-components of the query module. These are query interface (QIn), query parser (QPr), query processor (QPc), fuzzy module (FM), fuzzy knowledge base (FKB), and fuzzy inference system (FIS). When using the query module, users enter their queries into the framework with the support of the query interface. Query parser allows to parse and make sense of the entered query. The query processor enables the execution of the query. The fuzzy module supports handling the fuzzy parts of the query, while FKB and FIS support inference-specific operations.

Users can query a meteorological phenomenon or a meteorological measurement while querying. There are two query interfaces to support them. In order to query the metrological phenomenon, domain experts must define the association rules regarding this phenomenon in the framework. For this purpose, the rules regarding the meteorological phenomenon can be defined with the expert rule definition interface shown in Figure 4.2.



Expert Rule Definition

Meteorological Phonema: Risk of Flooding ▾ IS high ▾

Measurement Type: season ▾ IS spring ▾

EXPERT RULE LIST

IF rainfall IS high THEN Risk of Flooding IS high DELETE

IF rainfall IS very high THEN Risk of Flooding IS high DELETE

IF rainfall IS flood THEN Risk of Flooding IS high DELETE

IF cloudiness IS overcast THEN Risk of Flooding IS high DELETE

IF location IS north THEN Risk of Flooding IS high DELETE

IF location IS sea level THEN Risk of Flooding IS high DELETE

IF season IS spring THEN Risk of Flooding IS high DELETE

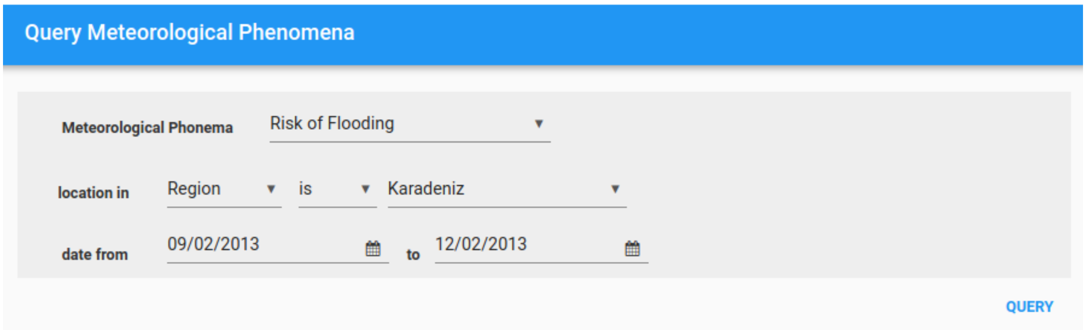
| Right Hand Side/antecedent | Left Hand Side/consequent |
|---|---|
| IF rainfall IS high | THEN Risk of Flooding IS high |
| IF rainfall IS very high | THEN Risk of Flooding IS high |
| IF rainfall IS flood | THEN Risk of Flooding IS high |
| IF cloudiness IS overcast | THEN Risk of Flooding IS high |
| IF location IS north | THEN Risk of Flooding IS high |
| IF location IS sea level | THEN Risk of Flooding IS high |
| IF season IS spring | THEN Risk of Flooding IS high |

Figure 4.2: Expert rule definition UI.

The expert selects the meteorological attribute and fuzzy class, then defines the fuzzy association rule on the phenomenon definition page. This rule is related to the antecedent part of the selected meteorological event. The defined fuzzy association rule is stored in FKB. Figure 4.3 shows the meteorological phenomenon query page, and the defined rules are used here.

In addition, the user can query meteorological data by selecting the attribute and the spatial and temporal criteria using the interface, as shown in Figure 4.4. The result page represents the query results in a list and demonstrates the spatial information on a map.

In the meteorological phenomenon inquiry process, the query processor selects the

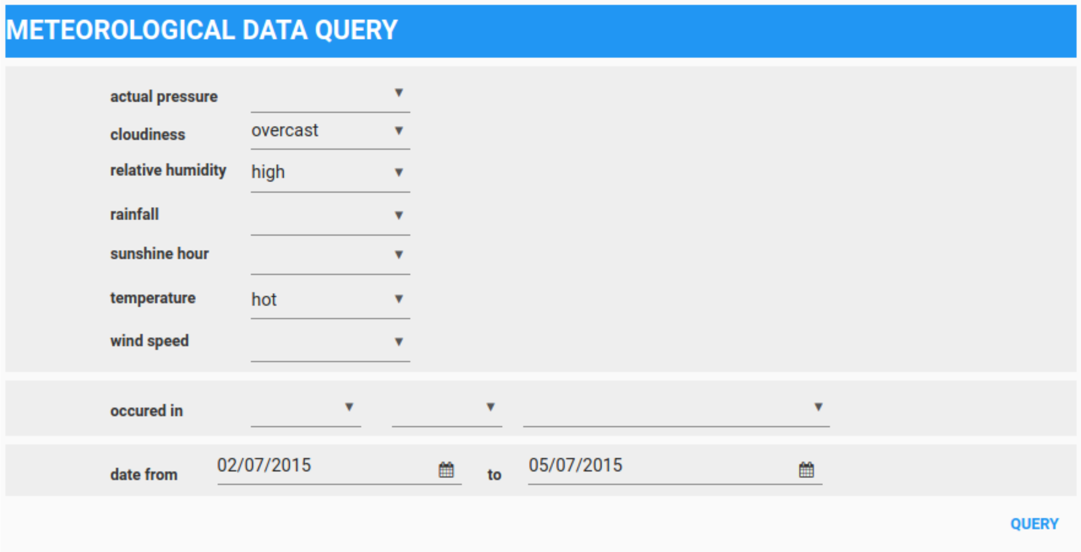


Meteorological Phonema: Risk of Flooding

location in: Region is Karadeniz

date from: 09/02/2013 to 12/02/2013

Figure 4.3: Meteorological phenomena query UI.



actual pressure:

cloudiness: overcast

relative humidity: high

rainfall:

sunshine hour:

temperature: hot

wind speed:

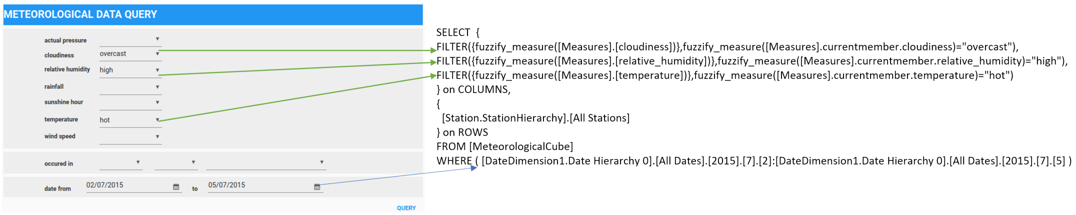
occured in:

date from: 02/07/2015 to 05/07/2015

Figure 4.4: Meteorological data query UI.

association rules of the relevant event from the FKB. In the antecedent part of these rules, fuzzy attributes and classes are determined and used as query criteria. The user can select the spatial and temporal conditions into the requirements of the MDX query. The query processor fetches the query results after executing the built MDX query on the SOLAP server. Again, the result page displays query results in a list and shows spatial information on a map. Figure 4.5 represents how the selected criteria are used in the interface when building the MDX query.

The QPr component parses and interprets the user query and determines which elements will process the query. The QPc module is a subcomponent reliable for running the query on the related systems and collecting and displaying the results. In



METEOROLOGICAL DATA QUERY

actual pressure:
cloudiness: overcast
relative humidity: high
rainfall:
sunshine hour:
temperature: hot
wind speed:
occured in:
date from: 02/07/2015 to: 05/07/2015

```
SELECT  {
FILTER({fuzzify_measure([Measures].[cloudiness])},fuzzify_measure([Measures].currentmember.cloudiness)="overcast"),
FILTER({fuzzify_measure([Measures].[relative_humidity])},fuzzify_measure([Measures].currentmember.relative_humidity)="high"),
FILTER({fuzzify_measure([Measures].[temperature])},fuzzify_measure([Measures].currentmember.temperature)="hot")
} on COLUMNS,
{
  [Station.StationHierarchy].[All Stations]
} on ROWS
FROM [MeteorologicalCube]
WHERE ( [DateDimension1.Date Hierarchy 0].[All Dates].[2015].[7].[2]:[DateDimension1.Date Hierarchy 0].[All Dates].[2015].[7].[5] )
```

Figure 4.5: Sample MDX of meteorological data query.

other words, the QPc component plays a coordinating role in query processing. QPc communicates and interacts between the SOLAP, the FIS, and the fuzzy module. It acquires user queries, analyzes them, sends requests to the SOLAP and/or to the FKB/FM, retrieves the results, and transmits them to the query interface.

The fuzzy module is the component that provides fuzzification and defuzzification operations. These operations are related to crisp-to-fuzzy or fuzzy-to-crisp transformations. In this module, we perform fuzzy clustering to generate membership classes and determine membership values using the FCM algorithm. The number of clusters is necessary for the FCM as a parameter. Therefore, we operated X-means clustering to determine the appropriate number of clusters. Then we cross-check the cluster with elbow [67] and silhouette [68] methods. In addition, we store the definitions of uncertain types, similarity relations, and membership functions in the fuzzy data map.

The fuzzy knowledge base (FKB) is the component that produces and stores fuzzy association rules. We first fuzzify the meteorological data on SOLAP, then generate fuzzy association rules with the FP-growth algorithm and store them in the FKB. After rule generation, we prune the resulting extensive list of rules using a confidence-measure-based pruning method [69] for performance improvement. We use the rules in the FKB in the case of inference as input for the FIS.

We utilize the FIS to support prediction-type queries. While executing the predictive-type query, the fuzzy association rule required for each criterion is requested from the FKB and sent to the FIS. In addition, the FM provides the fuzzy membership classes and membership values required for the values in the query as input to the FIS. This interface works as follows. $A' = F(x_0)$, where $x_0$ is a crisp value defined in the input universe $\cup$, $A_0$ is a fuzzy set defined in the same universe, and $F$ is a fuzzifier operator. The FIS is based on the application of the generalized modus ponens, an extension of the classical modus ponens proposed by Zadeh, where:

$$\frac{\text{(If X is A then Y is B)} \cap \text{(X is A')}}{\text{(Y is B')}} \tag{4.1}$$

where $X$ and $Y$ are linguistic variables, $A$ and $B$ are fuzzy sets, and $B'$ is the output fuzzy set inferred. To achieve this, the system firstly obtains the degree of matching of each rule by applying a conjunctive operator, and then infers the output fuzzy sets by means of a fuzzy implication operator. The FIS produces the same number of output fuzzy sets as the number of rules collected in the FKB.

The SOLAP server acts as a database server for objects and provides an application that stores measurement results, including spatiotemporal hierarchies, and supports MDX query types. After the ETL process and fuzzification, we insert the meteorological data into the spatial OLAP server. SOLAP server stores these data as spatial, temporal, and measurement-value hierarchies. The spatial hierarchy has region, city, and station breakdowns. SOLAP server can achieve the spatial hierarchy with a foreign key, as in classical relational databases, or with a minimum bounded rectangle (MBR) structure supporting the spatial structure. The temporal hierarchy is organized according to year, month, and day divisions. Furthermore, each measurement result is available in a hierarchical structure in SOLAP.

We modified the GeoMondrian SOLAP server and extended the MDX query form to support fuzzy queries. When querying, the user generally asks for fuzzy spatial or non-spatial objects that meet the conditions of the predefined rules within a specified time interval. We can evaluate the rules by examining the topological relations between fuzzy regions and objects. To support this, the fuzzify_measure and fuzzify_geo methods are implemented in the MDX query processor of the SOLAP server. The fuzzify_measure method uses the hierarchy for the non-spatial attributes, while the fuzzify_geo method uses the hierarchy for the spatial attributes.

The Algorithm 4 represents the implementation of the queries, and we define some sample queries in Section 4.2.

**Algorithm 4** The generic query evaluation algorithm

```
Input: The user $query$ with set of column members $CLN$ and predicates $PR$
Output: Set of retrieved/predicted objects $RSL$
  Initialization :
  $FT_p \leftarrow \{\}$    //fuzzy membership terms
  $FAR \leftarrow \{\}$    //fuzzy association rules
  $SP_t \leftarrow \{\}$    //spatial terms
  $NSP_t \leftarrow \{\}$    //non-spatial terms, measurement
  $D_s \leftarrow \{\}$    //SOLAP data cube query result holder
  $S_O \leftarrow \{\}$    //satisfying-objects
  Retrieve and Parse ($query$)
  if query includes prediction predicate($PR$) then
      Send query to FKB with ($CLN$,$PR$)
      Transfer to FIS with ($CLN$,$PR$)
      $FAR \leftarrow$Retrieve fuzzy association rules from FKB with ($CLN$,$PR$)
      $FT_p \leftarrow$Retrieve fuzzy memberships from FM with ($CLN$,$PR$)
      $SP_t \leftarrow$Defuzzify spatial predicates with ($CLN$)
      $NSP_t \leftarrow$Defuzzify non-spatial predicates with ($PR$)
      $D_s \leftarrow$Query spatial temporal data from SOLAP with ($SP_t$,$NSP_t$)
      $S_O \leftarrow$Make prediction with ($FAR$, $FT_p$,$D_s$)
      return $S_O$
  else
      if query is spatial then
          $SP_t \leftarrow$Defuzzify spatial predicates with ($CLN$)
          $NSP_t \leftarrow$Defuzzify non-spatial predicates with ($PR$)
          $D_s \leftarrow$Query spatial temporal data from SOLAP with ($SP_t$,$NSP_t$)
          $S_O \leftarrow$Fuzzify satisfying objects with ($D_s$)
          return $S_O$
      else
          $NSP_t \leftarrow$Defuzzify non-spatial predicates with ($PR$)
          $D_s \leftarrow$Query spatial temporal data from SOLAP with ($NSP_t$)
          $S_O \leftarrow$Fuzzify satisfying objects with ($D_s$)
          return $S_O$
      end if
  end if
```

## 4.2 Supported Query Types

We represented the architecture of the proposed environment for fuzzy spatiotemporal querying in the previous section. We give detailed information about handling the various query types employing the shown components in the followings.

### 4.2.1 Fuzzy Non-Spatial Query

This query type asks for fuzzy data not dealing with spatial attributes. The QM, the FM, and the SOLAP server components are working in the execution step and the query flow is given in Figure 4.6:

1. The QM retrieves the user query, parses it, and sends it to the FM.
2. The QM asks the SOLAP server for data using the query. The objects retrieved by the QM are sent to the FM component to fuzzify the result.
3. Fuzzified query results are displayed in the QM component.

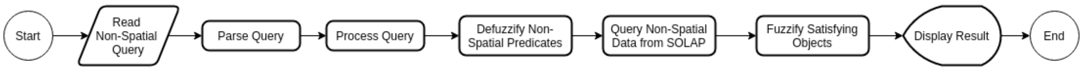
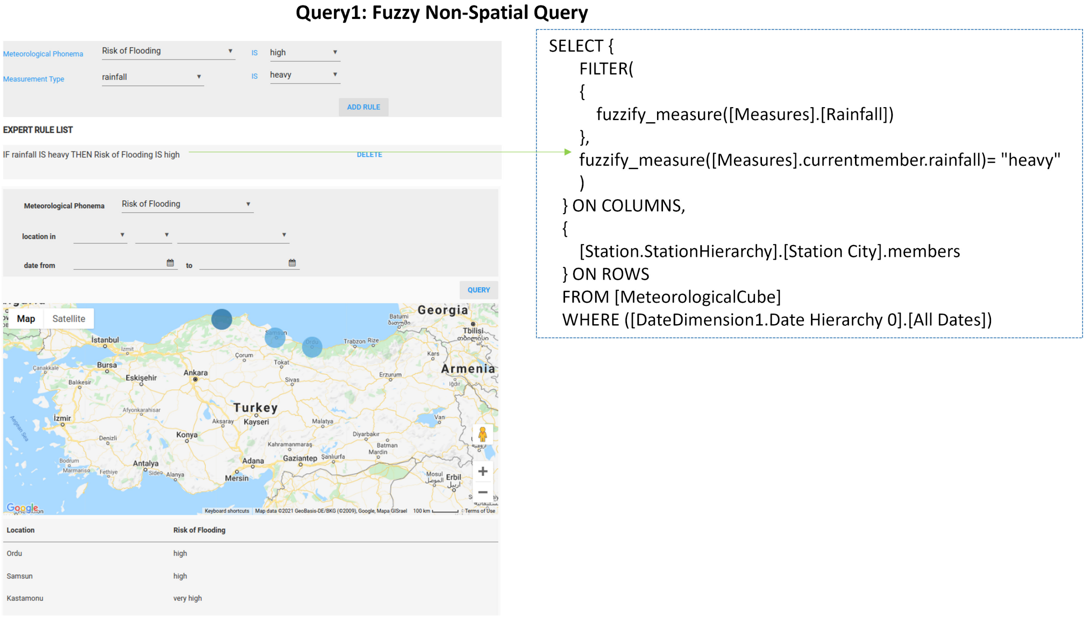

Start
Read Non-Spatial Query
Parse Query
Process Query
Defuzzify Non-Spatial Predicates
Query Non-Spatial Data from SOLAP
Fuzzify Satisfying Objects
Display Result
End

Figure 4.6: Fuzzy non-spatial query flow.

**Query 1:** *Find all the cities at risk of flooding.*

The query is expressed in MDX, which is an OLAP query language which provides a specialized syntax for querying and manipulating the multidimensional data stored in OLAP cubes [70]. While it is possible to translate some of these queries into traditional SQL, this would frequently require the synthesis of clumsy SQL expressions, even for elementary MDX expressions. Furthermore, many OLAP vendors have used MDX, and it has become the standard for OLAP systems. While it is not an open standard, it is embraced by a wide range of OLAP vendors. Therefore, we extended MDX with fuzzy operators and wrote the query specified above in MDX form, using the query parameters shown in Figure 4.7.



Query1: Fuzzy Non-Spatial Query

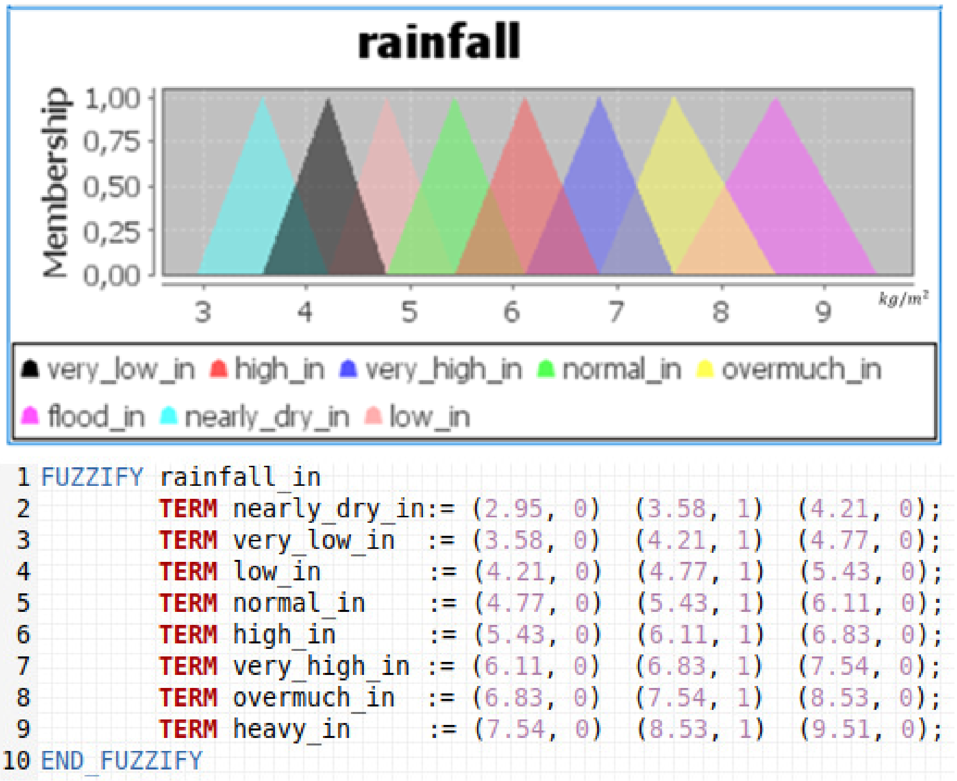
EXPERT RULE LIST

IF rainfall IS heavy THEN Risk of Flooding IS high

| Location | Risk of Flooding |
|---|---|
| Ordu | high |
| Samsun | high |
| Kastamonu | very high |

```
SELECT {
     FILTER(
     {
        fuzzify_measure([Measures].[Rainfall])
     },
     fuzzify_measure([Measures].currentmember.rainfall)= "heavy"
     )
  } ON COLUMNS,
  {
     [Station.StationHierarchy].[Station City].members
  } ON ROWS
  FROM [MeteorologicalCube]
  WHERE ([DateDimension1.Date Hierarchy 0].[All Dates])
```

Figure 4.7: Fuzzy non-spatial query.



```
FUZZIFY rainfall_in
        TERM nearly_dry_in:= (2.95, 0)  (3.58, 1)  (4.21, 0);
        TERM very_low_in   := (3.58, 0)  (4.21, 1)  (4.77, 0);
        TERM low_in        := (4.21, 0)  (4.77, 1)  (5.43, 0);
        TERM normal_in     := (4.77, 0)  (5.43, 1)  (6.11, 0);
        TERM high_in       := (5.43, 0)  (6.11, 1)  (6.83, 0);
        TERM very_high_in  := (6.11, 0)  (6.83, 1)  (7.54, 0);
        TERM overmuch_in   := (6.83, 0)  (7.54, 1)  (8.53, 0);
        TERM heavy_in      := (7.54, 0)  (8.53, 1)  (9.51, 0);
END_FUZZIFY
```

Figure 4.8: Rainfall membership classes.

To query the database, we first need to defuzzfy the fuzzy expression part of the query. The query processor requests the FM to defuzzify the fuzzy expression in the query. The fuzzy term is defuzzified according to the fuzzy membership function, as shown in Figure 4.8.

The *heavy* class in the query has a triangular-shaped membership function defined by the triple (7.5, 8.5, 9.5) that overlaps the membership function of the *overmuch* class in the range [7.5, 8.5]. In this case, the *heavy* class includes measurements between 8.0 and 9.5. The query processor of the GeoMondrian rearranges the MDX query and executes it in the SOLAP server. As a result of the query on the SOLAP server, the results matching the searched criteria contain the searched data. We again fuzzify the crisp values in the resulting data with the help of the FM. Here, the fuzzification subcomponent in the FM includes a triangular or trapezoidal membership function for each measurement result. It generates fuzzy class and membership values as output, using the crisp value of input from the relevant membership function. Finally, the results are displayed to the user, including fuzzy terms. For our example, we show the Rec1 and Rec4 records in Table 4.1 as the query result that meets the criteria.

Table 4.1: Sample data for rainfall in database

| Record-ID | City | Date | Crisp Val. | Fuzzy Val. |
|---|---|---|---|---|
| Rec1 | Ankara | 19 August 2016 | 8.6 | heavy (0.7) |
| Rec2 | Konya | 19 August 2016 | 4.9 | low (0.7) |
| Rec3 | Adana | 19 August 2016 | 4.1 | very-low (0.6) |
| Rec4 | Rize | 19 August 2016 | 8.8 | heavy (0.8) |

Suppose we execute this query in a relational database. In that case, we need to thoroughly scan all records, because it is necessary to calculate the rainfall value and find the queried value by grouping based on the city within the station measurement records. The cost of scanning all the data and grouping them is critical; the query execution time is related to the number of records in the database. In the FSOLAP environment, it is not necessary to access all records for the objects that satisfy the query criteria, due to the help of the hierarchical structure. The calculation of the measurements of the cities with which the stations are connected does not imply such a cost. Therefore, the cost of searching rainy stations is limited to the number of stations registered in the database, and the query execution time is less than the relational database query execution time.

### 4.2.2 Fuzzy Spatial Query

Fuzzy spatial queries allow the user to interrogate fuzzy spatial objects and their relationships. The QM, the FM, and the SOLAP server components are employed to fetch query results, as shown in Figure 4.9. The user asks for the objects that have topological relations with the entities under inquiry.

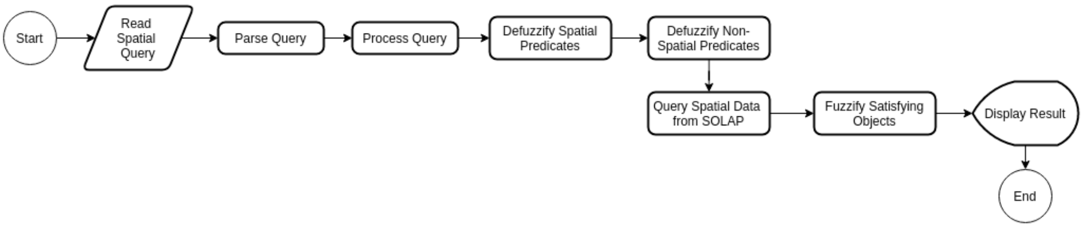

Start
Read Spatial Query
Parse Query
Process Query
Defuzzify Spatial Predicates
Defuzzify Non-Spatial Predicates
Query Spatial Data from SOLAP
Fuzzify Satisfying Objects
Display Result
End

Figure 4.9: Fuzzy spatial query flow.

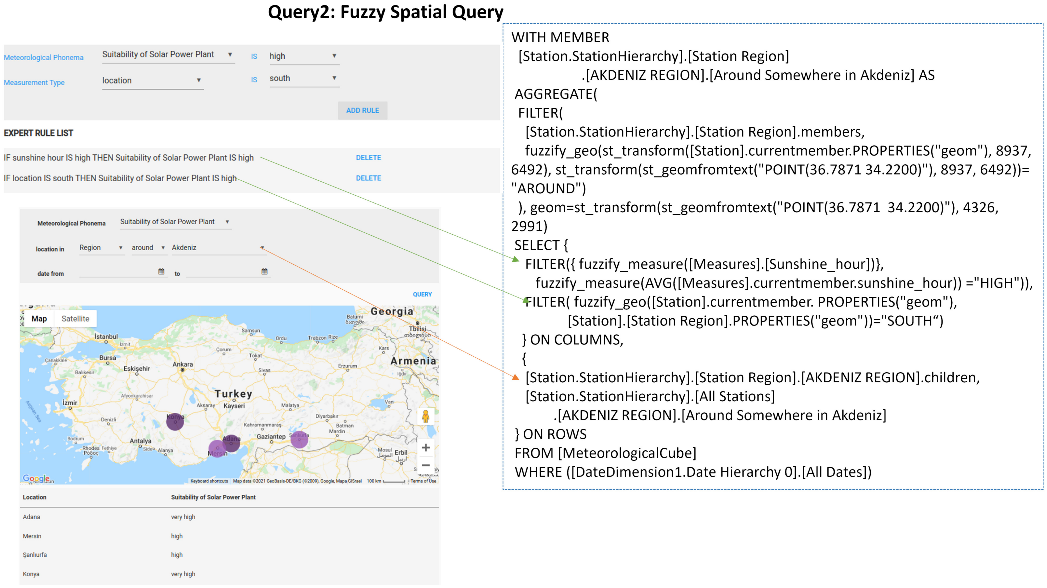
**Query 2:** *Retrieve the appropriate cities in south for the installation of a solar power plant*

A fuzzy rule definition uses linguistic values, as shown below in the FKB regarding suitable places for solar power plants.

```
    if city.sunshinehour is high and city.position is south
                                                then city.solarpower is high
```

Figure 4.10 shows how we implemented the MDX query with the parameters entered from the query interface.

In this query, regions in the south of Turkey with a very high sunshine duration are considered. The intersection of areas with positionally high sunshine hour and south fields are taken into account. We explained the operational structure of the *fuzzify_measure* method in the previous query. Here, the *fuzzify_geo* method is also



**Query2: Fuzzy Spatial Query**

Meteorological Phonema: Suitability of Solar Power Plant IS high

Measurement Type: location IS south

ADD RULE

EXPERT RULE LIST

IF sunshine hour IS high THEN Suitability of Solar Power Plant IS high DELETE

IF location IS south THEN Suitability of Solar Power Plant IS high DELETE

Meteorological Phonema: Suitability of Solar Power Plant

location in: Region around Akdeniz

date from to

QUERY

| Location | Suitability of Solar Power Plant |
|---|---|
| Adana | very high |
| Mersin | high |
| Şanlıurfa | high |
| Konya | very high |

```
WITH MEMBER
  [Station.StationHierarchy].[Station Region]
          .[AKDENIZ REGION].[Around Somewhere in Akdeniz] AS
 AGGREGATE(
  FILTER(
   [Station.StationHierarchy].[Station Region].members,
   fuzzify_geo(st_transform([Station].currentmember.PROPERTIES("geom"), 8937,
6492), st_transform(st_geomfromtext("POINT(36.7871 34.2200)"), 8937, 6492))=
"AROUND")
  ), geom=st_transform(st_geomfromtext("POINT(36.7871  34.2200)"), 4326,
2991)
 SELECT {
   FILTER({ fuzzify_measure([Measures].[Sunshine_hour])},
     fuzzify_measure(AVG([Measures].currentmember.sunshine_hour)) ="HIGH")),
   FILTER( fuzzify_geo([Station].currentmember. PROPERTIES("geom"),
           [Station].[Station Region].PROPERTIES("geom"))="SOUTH“)
  } ON COLUMNS,
  {
   [Station.StationHierarchy].[Station Region].[AKDENIZ REGION].children,
   [Station.StationHierarchy].[All Stations]
       .[AKDENIZ REGION].[Around Somewhere in Akdeniz]
 } ON ROWS
 FROM [MeteorologicalCube]
 WHERE ([DateDimension1.Date Hierarchy 0].[All Dates])
```

Figure 4.10: Fuzzy spatial query.

used. This method is run on the FM and determines the overlap relation between two geometric objects given as parameters. There are as many accesses in the query process as the number of stations in the database. On the other hand, the execution time for the relational database query, given in the following, can be longer due to the averaging of sunshine hour measurements and joining these with the stations.

```
SELECT c.name 1, r.month, r.day, AVG(sunshine hour)
FROM met data rainfall r, tr city c,
     meteorological station3 s, tr region rg
WHERE s.id=r.station id AND s.city id=c.gid
  AND rg.id=c.region id AND c.region id in(5,7)
GROUP BY c.name 1, r.month, r.day HAVING AVG(sunshine hour) 7
```

In this query, cities with an average daily sunshine duration of more than seven hours are regarded as having a high sunshine duration. These cities are in the Mediterranean and Southeastern Anatolia regions in the south of the country.

### 4.2.3 Fuzzy Spatiotemporal Query

In this type of query, the user asks for the fuzzy spatial objects that meet the conditions of the predefined rules within a specified time interval. The rules can be evaluated by an examination of the topological relations between fuzzy regions and fuzzy objects.

The query flow is shown in Figure 4.11.

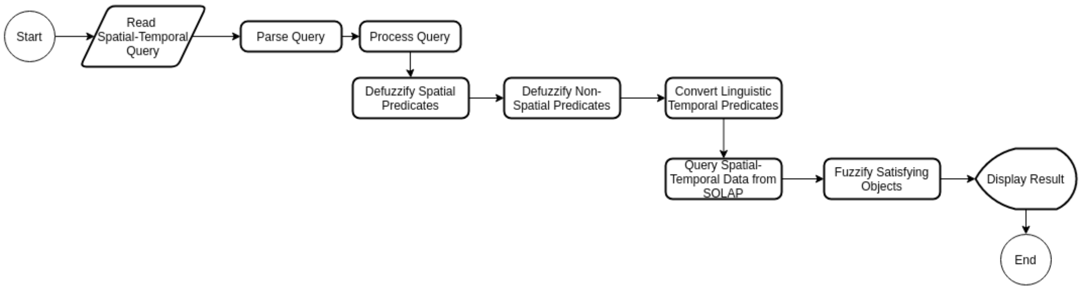

Start
Read Spatial-Temporal Query
Parse Query
Process Query
Defuzzify Spatial Predicates
Defuzzify Non-Spatial Predicates
Convert Linguistic Temporal Predicates
Query Spatial-Temporal Data from SOLAP
Fuzzify Satisfying Objects
Display Result
End

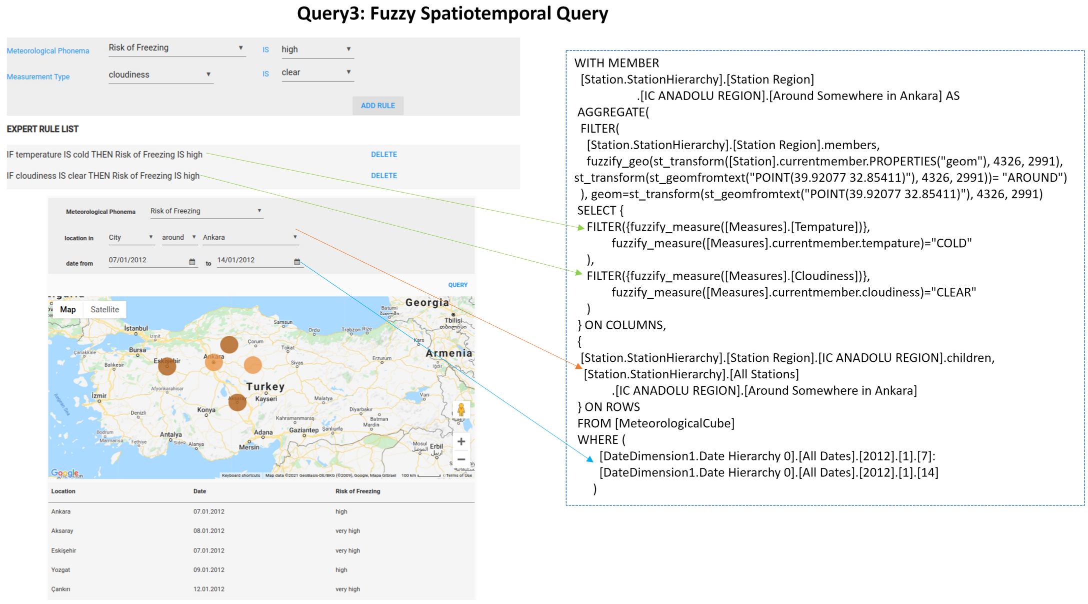
Figure 4.11: Fuzzy spatiotemporal query flow.

**Query 3:** *Retrieve locations around Ankara that were at high risk of freezing between 7 January 2012 and 14 January 2012.*

The FKB contains the following fuzzy rule definition that uses linguistic values regarding freezing events.

```
    if city.temperature is cold and city.cloudiness is clear
                                    then city.freezerisk is high
```

The query syntax's implementation in MDX is represented in Figure 4.12.



Query3: Fuzzy Spatiotemporal Query

Meteorological Phonema: Risk of Freezing IS high

Measurement Type: cloudiness IS clear

EXPERT RULE LIST

IF temperature IS cold THEN Risk of Freezing IS high

IF cloudiness IS clear THEN Risk of Freezing IS high

Meteorological Phonema: Risk of Freezing

location in: City around Ankara

date from: 07/01/2012 to 14/01/2012

| Location | Date | Risk of Freezing |
|---|---|---|
| Ankara | 07.01.2012 | high |
| Aksaray | 08.01.2012 | very high |
| Eskişehir | 07.01.2012 | very high |
| Yozgat | 09.01.2012 | high |
| Çankırı | 12.01.2012 | very high |

```
WITH MEMBER
 [Station.StationHierarchy].[Station Region]
                .[IC ANADOLU REGION].[Around Somewhere in Ankara] AS
 AGGREGATE(
  FILTER(
   [Station.StationHierarchy].[Station Region].members,
   fuzzify_geo(st_transform([Station].currentmember.PROPERTIES("geom"), 4326, 2991),
st_transform(st_geomfromtext("POINT(39.92077 32.85411)"), 4326, 2991))= "AROUND")
  ), geom=st_transform(st_geomfromtext("POINT(39.92077 32.85411)"), 4326, 2991)
 SELECT {
   FILTER({fuzzify_measure([Measures].[Tempature])},
          fuzzify_measure([Measures].currentmember.tempature)="COLD"
   ),
   FILTER({fuzzify_measure([Measures].[Cloudiness])},
          fuzzify_measure([Measures].currentmember.cloudiness)="CLEAR"
   )
 } ON COLUMNS,
 {
  [Station.StationHierarchy].[Station Region].[IC ANADOLU REGION].children,
  [Station.StationHierarchy].[All Stations]
          .[IC ANADOLU REGION].[Around Somewhere in Ankara]
 } ON ROWS
 FROM [MeteorologicalCube]
 WHERE (
    [DateDimension1.Date Hierarchy 0].[All Dates].[2012].[1].[7]:
    [DateDimension1.Date Hierarchy 0].[All Dates].[2012].[1].[14]
   )
```

Figure 4.12: Fuzzy spatiotemporal query.

In addition to the previous query, we can make more specific queries using date attribute conditions. The handling of the fuzzy predicates in the query operation is the same as for the fuzzy spatial query. For the distance attribute, the membership classes in the fuzzy data map are NEAR, CLOSE, and AROUND. We create these fuzzy classes by calculating the paired distances for the geometric data of the stations and applying fuzzy clustering of these values. However, the date predicate greatly reduces the amount of data to be retrieved from the database. As we mentioned earlier, this situation, which requires a full scan of an index-less relational database, is easily handled using the temporal hierarchy in the SOLAP environment. The execution time of the query depends on the number of stations in the database. Relational database systems must be fully searched for temperature and cloudiness between the given dates. In this case, the query execution time is proportional to the number of records and the number of stations in the database.

### 4.2.4 Fuzzy Spatiotemporal Predictive Query

This type of query asks for fuzzy spatial relations and a specified time with inference. The QM, the FM, the FIS, the FKB, and the SOLAP server components are employed to fetch query results, and the query flow is shown in Figure 4.13. The QM retrieves the user query, parses it, and sends it to the FM for defuzzification. If the QM detects the inference operand in the query, it sends the conditions to the FKB for inference. When the FKB receives the request from the QM, it determines the fuzzy association rules and sends them to the FIS, and the FIS obtains membership classes/functions from the fuzzy data map subcomponent. The FIS makes predictions with the given parameters and the collected knowledge, and then it sends the inference back to the QM.

**Query 4:** *Is there a possibility of a windstorm around Izmir during the last week of December?*

The FKB contains the following rules for meteorological events that occur depending on wind speed.

```
if station.windspeed is high then city.storm occurrence is possible
if station.windspeed is high and actual pressure is low
                             then city.storm occurrence is high possible
```

Unlike other query types, the antecedent part of the association rules is not used in the

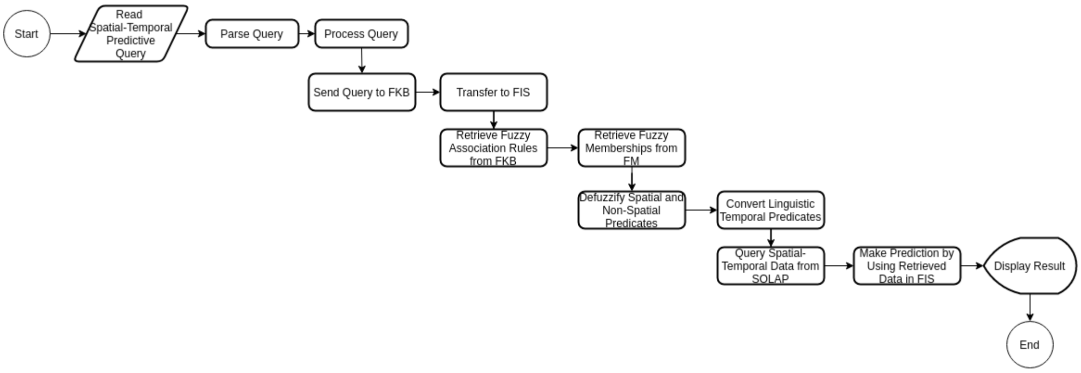
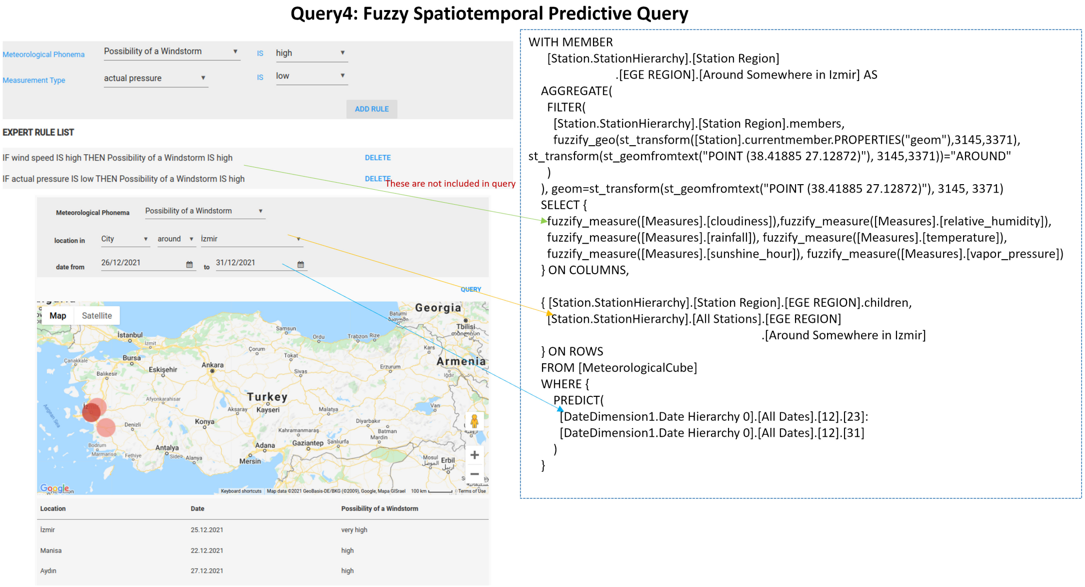

Start
Read Spatial-Temporal Predictive Query
Parse Query
Process Query
Send Query to FKB
Transfer to FIS
Retrieve Fuzzy Association Rules from FKB
Retrieve Fuzzy Memberships from FM
Defuzzify Spatial and Non-Spatial Predicates
Convert Linguistic Temporal Predicates
Query Spatial-Temporal Data from SOLAP
Make Prediction by Using Retrieved Data in FIS
Display Result
End

Figure 4.13: Fuzzy spatiotemporal predictive query flow.

FKB as a criterion when considering predictive queries. Since the purpose here is to predict the conditions that are the antecedents of the meteorological phenomenon in question, we do not include these fields in the query. Other fuzzy attributes are used as criteria in the MDX query. In addition, the spatial and temporal criteria entered into the interface are used for querying. When the QM detects the *PREDICT* expression in the query, it recognizes that the query requires an inference mechanism. The MDX query constructed with the criteria entered into the meteorological phenomenon query UI is illustrated in Figure 4.14.



**Query4: Fuzzy Spatiotemporal Predictive Query**

Meteorological Phonema: Possibility of a Windstorm IS high

Measurement Type: actual pressure IS low

EXPERT RULE LIST

IF wind speed IS high THEN Possibility of a Windstorm IS high

IF actual pressure IS low THEN Possibility of a Windstorm IS high

These are not included in query

Meteorological Phonema: Possibility of a Windstorm

location in: City around İzmir

date from: 26/12/2021 to 31/12/2021

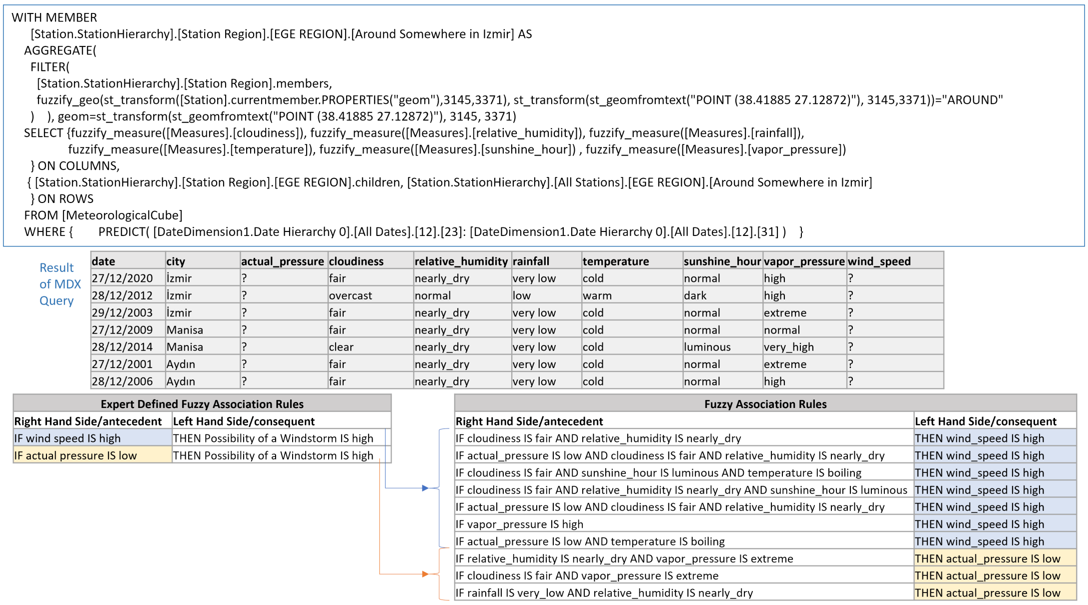
| Location | Date | Possibility of a Windstorm |
|---|---|---|
| İzmir | 25.12.2021 | very high |
| Manisa | 22.12.2021 | high |
| Aydın | 27.12.2021 | high |

```
WITH MEMBER
   [Station.StationHierarchy].[Station Region]
                  .[EGE REGION].[Around Somewhere in Izmir] AS
  AGGREGATE(
   FILTER(
    [Station.StationHierarchy].[Station Region].members,
    fuzzify_geo(st_transform([Station].currentmember.PROPERTIES("geom"),3145,3371),
st_transform(st_geomfromtext("POINT (38.41885 27.12872)"), 3145,3371))="AROUND"
   )
  ), geom=st_transform(st_geomfromtext("POINT (38.41885 27.12872)"), 3145, 3371)
  SELECT {
   fuzzify_measure([Measures].[cloudiness]),fuzzify_measure([Measures].[relative_humidity]),
   fuzzify_measure([Measures].[rainfall]), fuzzify_measure([Measures].[temperature]),
   fuzzify_measure([Measures].[sunshine_hour]), fuzzify_measure([Measures].[vapor_pressure])
  } ON COLUMNS,

  { [Station.StationHierarchy].[Station Region].[EGE REGION].children,
   [Station.StationHierarchy].[All Stations].[EGE REGION]
                                         .[Around Somewhere in Izmir]
  } ON ROWS
  FROM [MeteorologicalCube]
  WHERE {
    PREDICT(
     [DateDimension1.Date Hierarchy 0].[All Dates].[12].[23]:
     [DateDimension1.Date Hierarchy 0].[All Dates].[12].[31]
    )
  }
```

Figure 4.14: Fuzzy spatiotemporal predictive query.

We previously mentioned that the fuzzy association rules which are expert-defined are stored in the FKB. The fuzzy association rules defined for the relevant phenomenon are chosen in the meteorological phenomenon inquiry. The antecedent of each rule is used to look for the fuzzy attribute and membership class found in the consequent part of the fuzzy association rules. In other words, the rules which include these antecedents in the FKB are selected as a consequence of the rules in the fuzzy association rules, and this process is demonstrated in Figure 4.15.



```
WITH MEMBER
    [Station.StationHierarchy].[Station Region].[EGE REGION].[Around Somewhere in Izmir] AS
   AGGREGATE(
    FILTER(
     [Station.StationHierarchy].[Station Region].members,
     fuzzify_geo(st_transform([Station].currentmember.PROPERTIES("geom"),3145,3371), st_transform(st_geomfromtext("POINT (38.41885 27.12872)"), 3145,3371))="AROUND"
    )    ), geom=st_transform(st_geomfromtext("POINT (38.41885 27.12872)"), 3145, 3371)
   SELECT {fuzzify_measure([Measures].[cloudiness]), fuzzify_measure([Measures].[relative_humidity]), fuzzify_measure([Measures].[rainfall]),
            fuzzify_measure([Measures].[temperature]), fuzzify_measure([Measures].[sunshine_hour]) , fuzzify_measure([Measures].[vapor_pressure])
    } ON COLUMNS,
   { [Station.StationHierarchy].[Station Region].[EGE REGION].children, [Station.StationHierarchy].[All Stations].[EGE REGION].[Around Somewhere in Izmir]
    } ON ROWS
   FROM [MeteorologicalCube]
   WHERE {       PREDICT( [DateDimension1.Date Hierarchy 0].[All Dates].[12].[23]: [DateDimension1.Date Hierarchy 0].[All Dates].[12].[31] )    }
```

Result of MDX Query

| date | city | actual_pressure | cloudiness | relative_humidity | rainfall | temperature | sunshine_hour | vapor_pressure | wind_speed |
|---|---|---|---|---|---|---|---|---|---|
| 27/12/2020 | İzmir | ? | fair | nearly_dry | very low | cold | normal | high | ? |
| 28/12/2012 | İzmir | ? | overcast | normal | low | warm | dark | high | ? |
| 29/12/2003 | İzmir | ? | fair | nearly_dry | very low | cold | normal | extreme | ? |
| 27/12/2009 | Manisa | ? | fair | nearly_dry | very low | cold | normal | normal | ? |
| 28/12/2014 | Manisa | ? | clear | nearly_dry | very low | cold | luminous | very_high | ? |
| 27/12/2001 | Aydın | ? | fair | nearly_dry | very low | cold | normal | extreme | ? |
| 28/12/2006 | Aydın | ? | fair | nearly_dry | very low | cold | normal | high | ? |

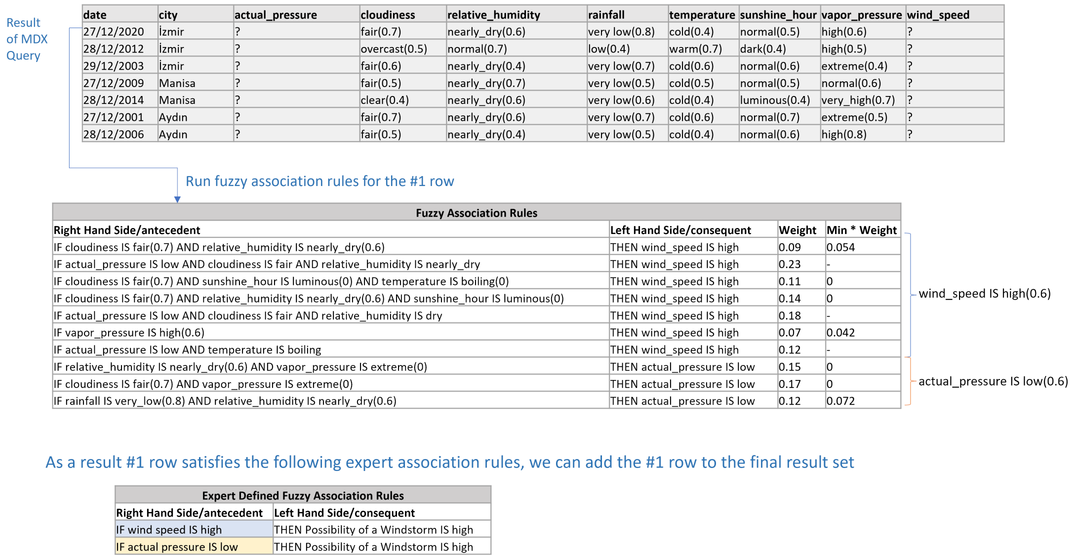
| Expert Defined Fuzzy Association Rules | |
|---|---|
| Right Hand Side/antecedent | Left Hand Side/consequent |
| IF wind speed IS high | THEN Possibility of a Windstorm IS high |
| IF actual pressure IS low | THEN Possibility of a Windstorm IS high |

| Fuzzy Association Rules | |
|---|---|
| Right Hand Side/antecedent | Left Hand Side/consequent |
| IF cloudiness IS fair AND relative_humidity IS nearly_dry | THEN wind_speed IS high |
| IF actual_pressure IS low AND cloudiness IS fair AND relative_humidity IS nearly_dry | THEN wind_speed IS high |
| IF cloudiness IS fair AND sunshine_hour IS luminous AND temperature IS boiling | THEN wind_speed IS high |
| IF cloudiness IS fair AND relative_humidity IS nearly_dry AND sunshine_hour IS luminous | THEN wind_speed IS high |
| IF actual_pressure IS low AND cloudiness IS fair AND relative_humidity IS nearly_dry | THEN wind_speed IS high |
| IF vapor_pressure IS high | THEN wind_speed IS high |
| IF actual_pressure IS low AND temperature IS boiling | THEN wind_speed IS high |
| IF relative_humidity IS nearly_dry AND vapor_pressure IS extreme | THEN actual_pressure IS low |
| IF cloudiness IS fair AND vapor_pressure IS extreme | THEN actual_pressure IS low |
| IF rainfall IS very_low AND relative_humidity IS nearly_dry | THEN actual_pressure IS low |

Figure 4.15: Fuzzy spatiotemporal predictive query execution: step 1.

We create inferences for each row fetched from the MDX query by running the rules selected from the fuzzy association rule set in the FIS, as shown in Figure 4.16.

The minimum value is calculated by multiplying the results by the weight value of each association rule. The same fuzzy class result is determined by taking the maximum value among the minimum values. If the result value meets the expected criteria, the relevant MDX query result row is marked as satisfied. The results marked as satisfied are shown on the results list and the map.



Result of MDX Query

| date | city | actual_pressure | cloudiness | relative_humidity | rainfall | temperature | sunshine_hour | vapor_pressure | wind_speed |
|---|---|---|---|---|---|---|---|---|---|
| 27/12/2020 | İzmir | ? | fair(0.7) | nearly_dry(0.6) | very low(0.8) | cold(0.4) | normal(0.5) | high(0.6) | ? |
| 28/12/2012 | İzmir | ? | overcast(0.5) | normal(0.7) | low(0.4) | warm(0.7) | dark(0.4) | high(0.5) | ? |
| 29/12/2003 | İzmir | ? | fair(0.6) | nearly_dry(0.4) | very low(0.7) | cold(0.6) | normal(0.6) | extreme(0.4) | ? |
| 27/12/2009 | Manisa | ? | fair(0.5) | nearly_dry(0.7) | very low(0.5) | cold(0.5) | normal(0.5) | normal(0.6) | ? |
| 28/12/2014 | Manisa | ? | clear(0.4) | nearly_dry(0.6) | very low(0.6) | cold(0.4) | luminous(0.4) | very_high(0.7) | ? |
| 27/12/2001 | Aydın | ? | fair(0.7) | nearly_dry(0.6) | very low(0.7) | cold(0.6) | normal(0.7) | extreme(0.5) | ? |
| 28/12/2006 | Aydın | ? | fair(0.5) | nearly_dry(0.4) | very low(0.5) | cold(0.4) | normal(0.6) | high(0.8) | ? |

Run fuzzy association rules for the #1 row

| Fuzzy Association Rules | | | |
|---|---|---|---|
| **Right Hand Side/antecedent** | **Left Hand Side/consequent** | **Weight** | **Min * Weight** |
| IF cloudiness IS fair(0.7) AND relative_humidity IS nearly_dry(0.6) | THEN wind_speed IS high | 0.09 | 0.054 |
| IF actual_pressure IS low AND cloudiness IS fair AND relative_humidity IS nearly_dry | THEN wind_speed IS high | 0.23 | - |
| IF cloudiness IS fair(0.7) AND sunshine_hour IS luminous(0) AND temperature IS boiling(0) | THEN wind_speed IS high | 0.11 | 0 |
| IF cloudiness IS fair(0.7) AND relative_humidity IS nearly_dry(0.6) AND sunshine_hour IS luminous(0) | THEN wind_speed IS high | 0.14 | 0 |
| IF actual_pressure IS low AND cloudiness IS fair AND relative_humidity IS dry | THEN wind_speed IS high | 0.18 | - |
| IF vapor_pressure IS high(0.6) | THEN wind_speed IS high | 0.07 | 0.042 |
| IF actual_pressure IS low AND temperature IS boiling | THEN wind_speed IS high | 0.12 | - |
| IF relative_humidity IS nearly_dry(0.6) AND vapor_pressure IS extreme(0) | THEN actual_pressure IS low | 0.15 | 0 |
| IF cloudiness IS fair(0.7) AND vapor_pressure IS extreme(0) | THEN actual_pressure IS low | 0.17 | 0 |
| IF rainfall IS very_low(0.8) AND relative_humidity IS nearly_dry(0.6) | THEN actual_pressure IS low | 0.12 | 0.072 |

As a result #1 row satisfies the following expert association rules, we can add the #1 row to the final result set

| Expert Defined Fuzzy Association Rules | |
|---|---|
| **Right Hand Side/antecedent** | **Left Hand Side/consequent** |
| IF wind speed IS high | THEN Possibility of a Windstorm IS high |
| IF actual pressure IS low | THEN Possibility of a Windstorm IS high |

Figure 4.16: Fuzzy spatiotemporal predictive query execution: step 2.

## 4.3 Fuzzy Spatial Aggregation

Aggregation plays an essential role in knowledge discovery across the collected data. As the amount of data stored in data warehouses continues to expand, the most important and frequently accessed data can benefit from aggregation, making it feasible to generate summaries. When the data is aggregated, it can be queried quickly instead of requiring all the processing cycles to access each underlying atomic data row and aggregate it in execution time when it is queried or accessed. This section explains the extension of the framework with fuzzy spatial aggregation to generate fuzzy summaries for knowledge discovery. The fuzzy spatial aggregation method is proposed and examined in the framework's query model, and generated summaries are represented as a product of the data aggregation process.

Data analysis and querying can be accomplished with a general perspective to extract interesting information from the data. In this approach, it is beneficial to visualize the results graphically. OLAP can be successfully used to present data mining results to data analysts as it provides the appropriate infrastructure for easy integration of hierarchical data and visualization. This situation makes it necessary to use OLAP in data analysis and querying. Because OLAP offers an environment where data can be modeled and viewed in multiple dimensions, it is suitable for managing hierarchies and aggregations required for data mining.

OLAP arises in the literature because it provides efficient, effective, flexible data summarization and hierarchical structure, and simple computation of aggregations. For this reason, users often use OLAP applications to acquire a higher-level aggregated view of data to understand trends and make decisions, such as analysts and managers.

Data mining and OLAP are mainly used jointly in the case of OLAP operations selecting a sub-cube. Several data mining algorithms are used to clarify various data analytics questions because a data cube demonstrates logically grouped views and aggregations at different levels appropriate for these questions [71].

OLAP operators are used to manipulating data along the multiple dimensions. These operators are related to the hierarchies and are primarily used to increase or decrease the level of aggregation. Commonly used OLAP operations which are associated with the level of aggregations are as follows [72, 73, 74]:

- Drill-down: Reduce the level of aggregation, and increase the level of details. This operation is the converse of roll-up.
- Roll-up: Expand the level of aggregation or apply a group-by operation on dimensions. The station dimension can be rolled up into city-region-all.

Hierarchies include levels and help summarize specific data with more general intent. For instance, the "station" dimension can be organized as a "city-region-all" hierarchy in Figure 4.17(a). Aggregations are accomplished over the measures by dimensions or their hierarchies, as displayed in Figure 4.17(b). All dimensions have at least one higher hierarchical level, the "all" level.

In this context, Zhou et al. [75] have proposed an efficient polygon amalgamation method for merging spatial objects. This method is about the computation of aggregation for spatial measures. In another study, Prasher and Zhou [76] proposed multi-resolution amalgamation in which they dynamically performed aggregation for spatial data cube generation. They changed the resolutions of the region to keep spatial data at much higher resolutions and re-classified amalgamation objects into

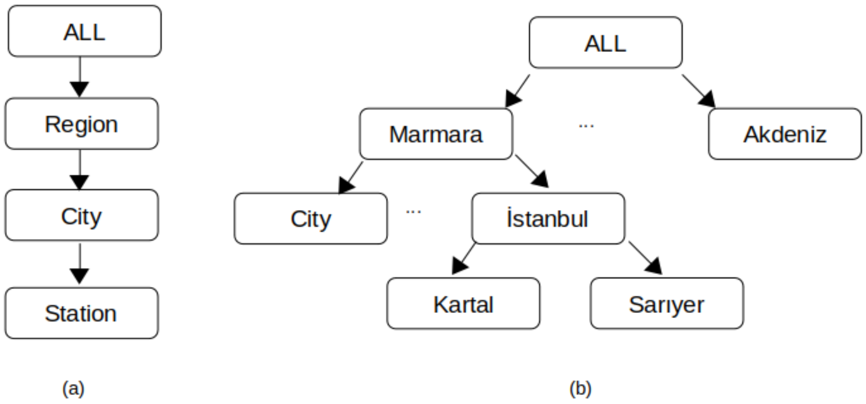


Figure 4.17: Spatial hierarchy concept of the station

high-level objects that form a new spatial layer. Papadias et al. [77] have presented a method for storing aggregated results in the spatial index to combine with the materialization technique.

Petry and Yager [78] recently explored aspects of applying interval-based fuzzy sets (IVFS) for Covid-19 contact tracing. They represented the aggregation of IVFSs and provided information measures to guide the aggregation. They proposed two approaches: averaging the fuzzy intervals and merging them. In their study, they also considered the application of IVFS in spatial data. They used aggregation to involve this to minimum bounding rectangles for Geographic Information System(GIS) by determining the distance and area of IVFS data.

Kacprzyk et al. [79, 80] performed trend analysis by calculating linguistic summaries on time-series data with a fuzzy quantifier-driven aggregation approach. They exhibited the straight line segments of a piecewise linear approximation of time series, and proposed summaries of time series refer to the outlines of trends identified. In their study, the summaries are represented as frequency-based and duration-based summaries protoforms which are defined as an abstract prototype of a linguistically quantified proposition.

Laurent also studied [73] aggregation to compute the degree to which the aggregated cell belongs to the cube. The proposal is about calculating the arithmetic compilation of the membership values of aggregated cells. This study represents an approach in which the arithmetic averages of the measurement data belonging to the regions in the same membership class are aggregated at a higher level.

Specifying the aggregation method used in constructing summary information is a critical issue. Related to this, some researchers studied the performance of several aggregation methods on different data sets and proposed literature reviews on the subject of aggregation.

Nowacka et al. [81] compared various aggregation operators such as average, OWA operator [82], induced OWA operator (IOWA), Leximin, and Leximax using their proposed fuzzy information retrieval model. They advised some guidelines for selecting values of various parameters to choose the best-suited aggregation operator.

Within the scope of this study, while performing the aggregation process as represented in Figure 4.18, an approach is driven in the form of obtaining the totals and calculating the averages by making spatial weighting for each record of the relevant measurement value.

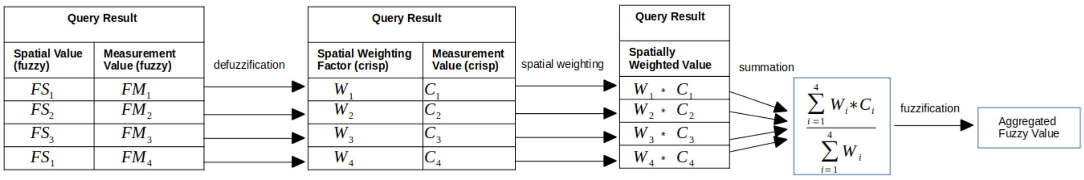


Figure 4.18: Fuzzy spatial aggregation process

One commonly used defuzzification technique is the Center of gravity (COG) / Centroid of Area (COA) Method. This method supplies a crisp value based on the center of gravity of the fuzzy set. The total area of the membership function distribution used to represent the combined control action is divided into several sub-areas. The area and the center of gravity or centroid of each sub-area are calculated, and then the summation of all these sub-areas is taken to find the defuzzified value for a discrete fuzzy set.

Suppose that we have query result set $R = \{r_1, r_2, ..., r_n\}$ which includes tuples of fuzzy spatial membership class and its value, fuzzy measurement membership class and its membership value (i.e. tuple of rainfall record $r =< west(0.5), high(0.6) >$). To aggregate these query results, firstly, we need to defuzzify the fuzzy measurement and spatial values in each record. The defuzzified value $x^*$ using COG is defined as:

$$x^* = \frac{\sum_{i=1}^{N} A_i.x_i}{\sum_{i=1}^{N} A_i} \tag{4.2}$$

Here $N$ indicates the number of records, $A_i$ is the area of the membership function, and $x_i$ represents the centroid of area, respectively, of $i^{th}$ record. Similarly, the fuzzy spatial value of the tuple is defuzzified as $S_i$. The weight of the precise value calculated after defuzzification of the spatial data differs according to the query typed. If the spatial property of the query result is within the scope of a discrete set, a weighting factor of 1 or 0 can be used. In another case, if the spatial feature is related to distance, the weighting factor can be calculated as increasing or inversely proportional to distance. The defuzzified measurement values are multiplied by each tuple's spatially weighted factor (W) to weight the aggregation with spatial attributes. The product results are summed and divided by the sum of the spatial weights to calculate the aggregated precise value.

$$Agg(R) = \frac{\sum_{i=1}^{N} W_i.x_i^*}{\sum_{i=1}^{N} W_i} \tag{4.3}$$

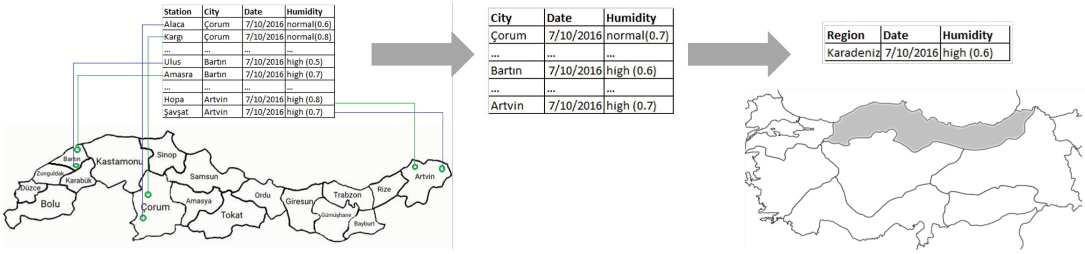
This equation calculates the precise value of aggregation where $x_i^*$ is the defuzzified measurement value, $W_i$ is the spatially weighted factor of the record tuple. The precise result is fuzzified by applying the fuzzy membership function of the relevant measurement as a parameter.

$$Aggregation(R) = fuzzify(Agg(R)) \tag{4.4}$$

The whole process of aggregation can be formulated as follows:

$$Aggregation(R) = fuzzify(\frac{\sum_{i=1}^{N} W_i.\frac{\sum_{i=1}^{N} A_i.x_i}{\sum_{i=1}^{N} A_i}}{\sum_{i=1}^{N} W_i}) \tag{4.5}$$

Columns having non-hierarchical data in the OLAP data cube are member properties. It is beneficial to obtain new data by making calculations from the data in these columns if the queries need it. For instance, derived variables can be produced by these member properties, i.e., Average Humidity = Total Humidity/Number of Records. The derived variables do not take up space in the database because they are calculated on the fly. Therefore, they help reduce the size of the database and the consolidation time, despite the small overhead on runtime. In this case, spatial aggregation, such as region merge or map overlay, can be performed for them. Here, spatially weighted aggregation method is introduced for fuzzy spatial data cube in which membership values of the regions are fetched on the query execution for the aggregated region. For this purpose, the aggregate operator is used in the Multidimensional Expression (MDX) query as a counterpart to the group-by operator in the traditional Structured Query Language (SQL) query. The aggregated value of the relevant level is calculated by grouping the results obtained as a result of the MDX query. As a first step, the membership classes and values in the highest level data are selected. Then, the Center of Gravity (COG) method is applied for the defuzzification of each fuzzy measurement and spatial values. The weighting factor for each tuple is calculated using the defuzzified spatial value. The defuzzified measurement value is weighted with the weighting factor. All weighted values are summed and averaged with the sum of defuzzified spatial values. This process calculated a single aggregated precise value. The final precise value is fuzzified with the fuzzy function of the relevant measure. So, membership value is calculated by computing the average of the membership values of the data in a single membership class obtained as a result. Thus, the membership class and its value for the appropriate level are calculated, and the aggregation process is completed. Figure 4.19 shows an example of aggregation as follows. First, we aggregate the station data to calculate the city data, then aggregate the city data and compute the region data.



| Station | City | Date | Humidity |
|---|---|---|---|
| Alaca | Çorum | 7/10/2016 | normal(0.6) |
| Kargı | Çorum | 7/10/2016 | normal(0.8) |
| ... | ... | ... | ... |
| Ulus | Bartın | 7/10/2016 | high (0.5) |
| Amasra | Bartın | 7/10/2016 | high (0.7) |
| ... | ... | ... | ... |
| Hopa | Artvin | 7/10/2016 | high (0.8) |
| Şavşat | Artvin | 7/10/2016 | high (0.7) |

| City | Date | Humidity |
|---|---|---|
| Çorum | 7/10/2016 | normal(0.7) |
| ... | ... | ... |
| Bartın | 7/10/2016 | high (0.6) |
| ... | ... | ... |
| Artvin | 7/10/2016 | high (0.7) |

| Region | Date | Humidity |
|---|---|---|
| Karadeniz | 7/10/2016 | high (0.6) |

Figure 4.19: Aggregation of the average humidity for the Karadeniz Region in 2016

The pseudo-code for fuzzy spatial aggregation is given below in Algorithm 5. The complexity of the algorithm is $O(n*h)$, n being the count of the tuples in the database and h being the average number of hierarchical levels for dimensions.

**Algorithm 5** Algorithm of aggregation

```
Input: $R = \{r_1, ..., r_K\}$ is a set of record tuples
Output: $aggregated_{fuzzy}$: tuple of aggregated membership class and its value
for $i \leftarrow 1$ to $K$ do
    $x_i^* \leftarrow$ defuzzify $r_i$.fuzzy-measure-value
    $S_i \leftarrow$ defuzzify $r_i$.fuzzy-spatial-value
    $W_i \leftarrow$ weight $S_i$
    $weighted_{val} \leftarrow W_i * x_i^*$
    $total_{weight} \leftarrow total_{weight} + weighted_{val}$
    $total_{def} \leftarrow total_{def} + weighted_{val}$
end for
$aggregated_{precise} \leftarrow total_{def} / total_{weight}$
$aggregated_{fuzzy} \leftarrow$ fuzzify $aggregated_{precise}$
return $aggregated_{fuzzy}$
```

Aggregated data can also be used to generate summaries to provide an overview of the application domain. This summary information is a verbal expression and is very useful for a human being. Yager [83] presented an early proposal on fuzzy summaries, and then researchers [80, 84] established more applicable forms, and studies were assembled on them for a long time by making improvements. A fuzzy summary can be broadly defined as "$P$ *objects are* $S$ : $t$". Here P is a quantifier represented by a fuzzy set, and $S$ is the summarizer also represented by another fuzzy set, $t$ is the degree of truth of the summary. Using fuzzy labels containing natural linguistic terms instead of numeric values is more effective in human understanding. For example, an expression such as "Sky is partly cloudy" is more understandable than "The sky is 3/8 cloudy". Expressing in a natural language instead of numerical terms can be beneficial for humans to understand because communication in a natural language is more practical for human beings. Here, the approach of expressing summaries using natural language does not aim to substitute classical statistical analysis but offers an additional form of human intelligibility, straightforward inferences, and simple handling of data description.

The performing of the aggregation operation in a sample query is clarified as follows.

### 4.3.1 Spatial Hierarhical Aggregation

In this aggregation type, the data are aggregated by considering the spatial characteristics. As a spatial feature, geometric coverage between the upper and sub-area is considered. This relationship between data is evaluated with the help of their MBR information.

**Hierarhical Query:** *Retrieve the average rainfall of 2016 for all regions.*

The linguistically expressed query is written in MDX format as follows. Here the query for the aggregation operation is managed with the aggregate keyword. While performing spatial aggregation for this process, a minimum bounding rectangle (MBR) definition containing all regions is given as a parameter. Since the region-based criterion in the query is whether cities belong to the geographical regions, fuzzy spatial membership is either 1 or 0. Therefore the weighting factor of each tuple is 1 because all tuples in the result set are covered by only one region.

$$W_x = \begin{cases} 1 & \text{if } x \in \text{specific region,} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

During the query process, firstly, the rainfall data of the stations covered by the MBR passed as a parameter are collected. Then, the aggregate operation mentioned in Algorithm 5 for each hierarchy level is applied from bottom to top.

```
 WITH MEMBER
 [Station.StationHierarchy].[Station Region].[ALL] AS
AGGREGATE(
 FILTER(
   [Station.StationHierarchy].[Station Region].members,
   fuzzify geo(st transform([Station].currentmember.PROPERTIES("geom")),
       st transform(st geomfromtext("POINT(26 42)"), 1540, 667))= "IN")
)
SELECT    fuzzify measure([Measures].[rainfall])    ON COLUMNS,
          [Station.StationHierarchy].[Station Region].children    ON ROWS
FROM [MeteorologicalCube]
WHERE ([DateDimension1.Date Hierarchy 0].[2016])
```

The data retrieved by the query in the first step are shown in Table 4.2. In addition, region information is given in Table 4.3, and sample station data is represented in Table 4.4. The geom column in the tables contains the polygon definition that determines the geometric boundaries of the relevant region or station. Since there are too many points in the polygon definition, this area is described with three points.

Table 4.2: A sample of retrieved rainfall data by the MDX query.

| **StationID** | **Day** | **Month** | **Year** | **Rainfall** |
|---|---|---|---|---|
| 17050 | 12 | 3 | 2016 | low (0.4) |
| 17050 | 7 | 5 | 2016 | normal (0.6) |
| 17050 | 30 | 9 | 2016 | normal (0.8) |
| 17130 | 6 | 1 | 2016 | low (0.5) |
| 17130 | 3 | 4 | 2016 | normal (0.7) |
| 17130 | 8 | 8 | 2016 | nearly-dry (0.5) |
| ... | ... | ... | ... | ... |
| 18980 | 14 | 2 | 2016 | high (0.4) |
| 18980 | 2 | 5 | 2016 | low (0.5) |
| 18980 | 21 | 12 | 2016 | normal (0.4) |
| 19112 | 11 | 1 | 2016 | normal (0.6) |
| 19112 | 6 | 4 | 2016 | high (0.5) |
| 19112 | 25 | 10 | 2016 | low (0.6) |
| ... | ... | ... | ... | ... |

Table 4.3: Sample data for station in database.

| **StationID** | **Station Name** | **Geom** |
|---|---|---|
| 19112 | İstanbul Kartal | Polygon ... |
| 18980 | İstanbul Sarıyer | Polygon ... |
| 17130 | Ankara Keçiören | Polygon ... |
| 17050 | Edirne Merkez | Polygon ... |

The membership classes and function in the rainfall aggregation example are given in Figure 4.20.

Table 4.4: Region data in database.

| RegionID | Region Name | Geom |
|---|---|---|
| 1 | Marmara | Polygon ... |
| 2 | Doğu Anadolu | Polygon ... |
| 3 | Ege | Polygon ... |
| 4 | Güneydoğu Anadolu | Polygon ... |
| 5 | İç Anadolu | Polygon ... |
| 6 | Karadeniz | Polygon ... |
| 7 | Akdeniz | Polygon ... |

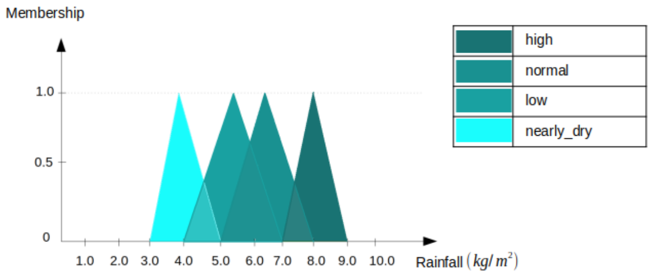
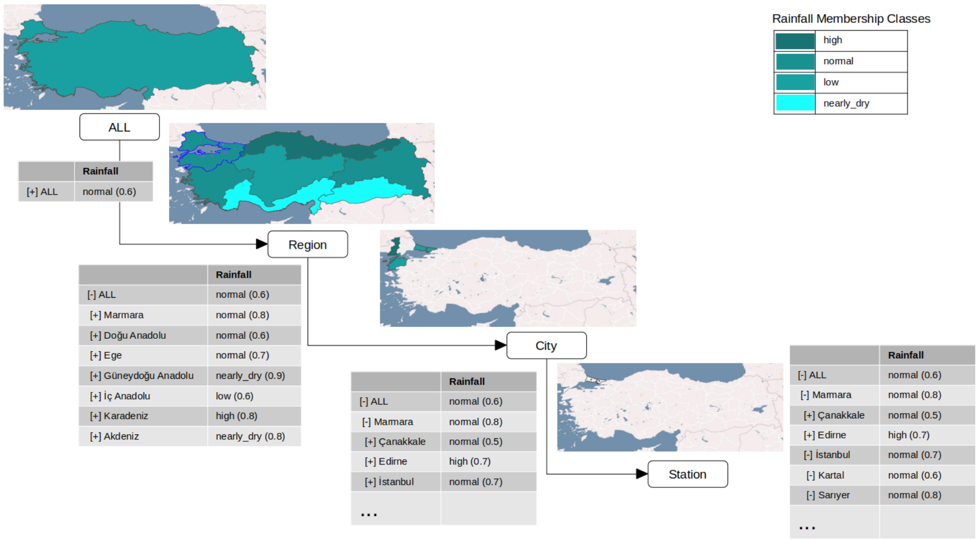


Figure 4.20: Membership classes and function of rainfall

The details of processing the aggregation using the data in these tables are shown in Figure 4.21.

After the MDX query retrieves the data in Table 4.2, aggregation with Algorithm 5 is performed, and the aggregated values for station hierarchy are firstly calculated, then related cities aggregation is computed. Finally, the region and "ALL" hierarchies are figured due to aggregation, respectively. As an advantage of using the OLAP structure, the "ALL" hierarchy is expanded with a drill-down operation downstream. Thus, the region breakdown and aggregated values of each region are displayed. Similarly, the drill-down process makes expansion from cities to stations possible. This drill-down can be accomplished in the city where the relevant station is located to reach the station information at the lowest level.

As shown in the example, firstly, the aggregated value is displayed for all regions, then the aggregated values are displayed according to the hierarchical order towards



| | Rainfall |
|---|---|
| [+] ALL | normal (0.6) |

| | Rainfall |
|---|---|
| [-] ALL | normal (0.6) |
| [+] Marmara | normal (0.8) |
| [+] Doğu Anadolu | normal (0.6) |
| [+] Ege | normal (0.7) |
| [+] Güneydoğu Anadolu | nearly_dry (0.9) |
| [+] İç Anadolu | low (0.6) |
| [+] Karadeniz | high (0.8) |
| [+] Akdeniz | nearly_dry (0.8) |

| | Rainfall |
|---|---|
| [-] ALL | normal (0.6) |
| [-] Marmara | normal (0.8) |
| [+] Çanakkale | normal (0.5) |
| [+] Edirne | high (0.7) |
| [+] İstanbul | normal (0.7) |
| ... | |

| | Rainfall |
|---|---|
| [-] ALL | normal (0.6) |
| [-] Marmara | normal (0.8) |
| [+] Çanakkale | normal (0.5) |
| [+] Edirne | high (0.7) |
| [-] İstanbul | normal (0.7) |
| [-] Kartal | normal (0.6) |
| [-] Sarıyer | normal (0.8) |
| ... | |

Figure 4.21: An example of fuzzy spatial aggregation

the lower levels, and the fuzzy spatial aggregation process is completed.

The following fuzzy summaries in the form of "$P\, objects\, are\, S\, :\, t$" can be generated using the aggregation results.

- For example, a fuzzy summary we could generate from this aggregation query is: The rainfall in Edirne in 2016 is high(0.7).

- The second example for a fuzzy summary of this aggregation query is: The rainfall of the Marmara Region in 2016 is normal(0.8).

### 4.3.2 Conceptual Aggregation

In this aggregation type, assuming a spatial region is determined as a concept, it is ensured that we aggregate the data in the area covered by this region. Here, the region determined as a concept is classified into fuzzy $\alpha - cut$ parts. The $\alpha - cut$ value is assessed in the spatial weighting of the data in the aggregation process. The formal definitions of $\alpha - cut$ is given in previous sections.

The following linguistic form of query can be given as an example where spatial weights are considered.

**Conceptual Query:** *In which hinterland cities did the freezing occur in 2015 winter?*

Freezing is a meteorological phenomenon that occurs in low temperatures under cloudless. Continentality is one of the critical factors influencing this phenomenon. For this reason, the situation of being in the internal region is a factor that increases the risk of the event occurring. For this reason, spatial weighting is taken into account in this event.

The hinterland expression (the concept used in this query) is a spatial indicator and exhibits the interior of the country, as shown in Figure 4.22. This term also comprises the continentality of the location. As we mentioned in the section where we explained the topological relations, we can show a spatial region as $\alpha - cut$ levels.

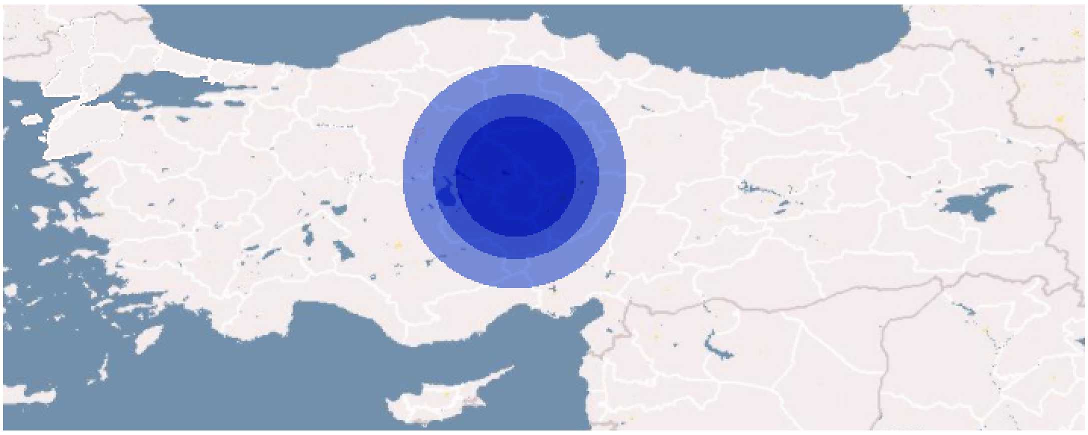


Figure 4.22: $\alpha - cut$ levels of the fuzzy hinterland attribute

These $\alpha - cut$ levels contain transitions as a reflection of membership values. For the hinterland attribute, three $\alpha - cut$ levels are determined as the center, near, and far as shown in Figure 4.23. Here, we take the map's center point as a reference while determining the $\alpha - cut$ levels for the definition of the inner zone and consider the stations located in the circular area 300 km away from this point. For the $\alpha - cut$ levels, the values of 75-150-225-300 shown in Figure 4.23 are set as limiting thresholds.

Stations outside the lowest $\alpha - cut$ level do not meet the expected threshold, so they contain the value of zero spatially and are not included in the result list. The measurements of the stations in the result list are subjected to the aggregation process,

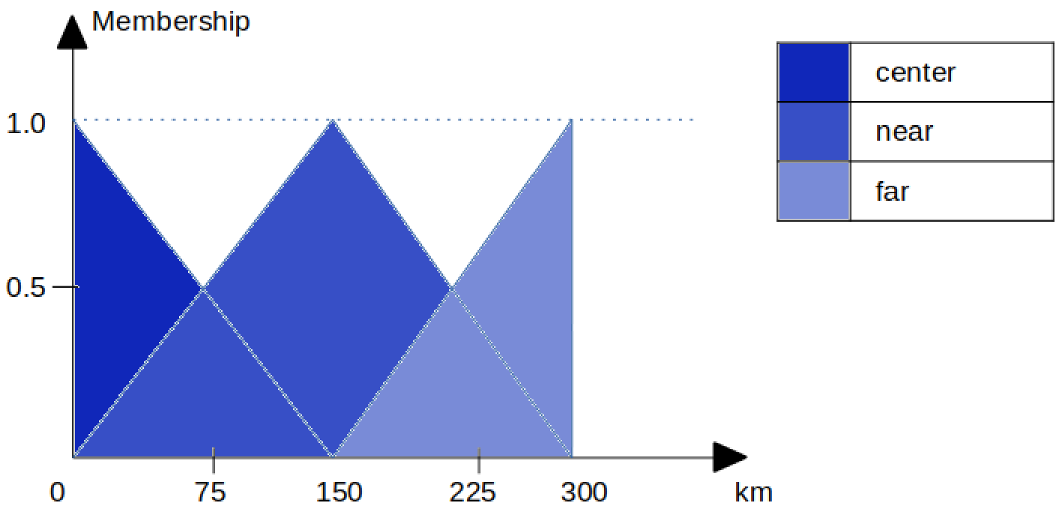


Figure 4.23: Membership function and classes of the hinterland attribute

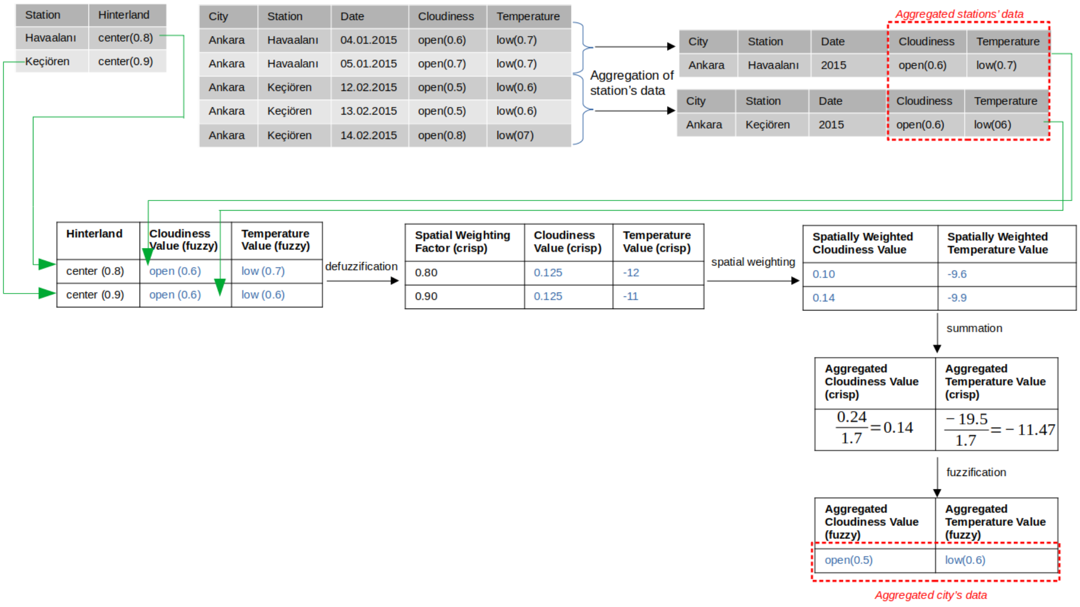
and the final value is evaluated. The spatial weighting factor of this query can be formulated as follows:

$$W_x = \frac{D_{max} - D_x}{D_{max}} \tag{4.7}$$

Here, $D_{max}$ is the maximum precise value of the spatial attribute, and $D_x$ is the precise value of the related record. In this query, there is a situation where the spatial weight decreases as far as the center. Because of this, it is seen that the weighting equation falls inversely with the distance. Spatial weighting for values in the result list is calculated by multiplying with the weighting factor of the defuzzified membership value at the $\alpha - cut$ level to which it is relevant. Thus, the measurement value of a station with a center membership class is more effective than a far station.

Here, we aggregate each city's cloudiness and temperature values. While performing the aggregation process, we use the measurements of the stations belong the cities by considering the spatial weights of the stations. In our example, two stations belong to the city of Ankara. The data of these stations are aggregated to calculate the measurement of the city of Ankara. In the aggregation process, the measurement value of the station, which is closer to the center of the hinterland circle, has a more significant effect on the result. This operation is about weighting the station's measurement with its spatial attribute.

The execution of the sample conceptual query and the aggregation process steps are illustrated in Figure 4.24.



| Station | Hinterland |
|---|---|
| Havaalanı | center(0.8) |
| Keçiören | center(0.9) |

| City | Station | Date | Cloudiness | Temperature |
|---|---|---|---|---|
| Ankara | Havaalanı | 04.01.2015 | open(0.6) | low(0.7) |
| Ankara | Havaalanı | 05.01.2015 | open(0.7) | low(0.7) |
| Ankara | Keçiören | 12.02.2015 | open(0.5) | low(0.6) |
| Ankara | Keçiören | 13.02.2015 | open(0.5) | low(0.6) |
| Ankara | Keçiören | 14.02.2015 | open(0.8) | low(07) |

| City | Station | Date | Cloudiness | Temperature |
|---|---|---|---|---|
| Ankara | Havaalanı | 2015 | open(0.6) | low(0.7) |

| City | Station | Date | Cloudiness | Temperature |
|---|---|---|---|---|
| Ankara | Keçiören | 2015 | open(0.6) | low(06) |

| Hinterland | Cloudiness Value (fuzzy) | Temperature Value (fuzzy) |
|---|---|---|
| center (0.8) | open (0.6) | low (0.7) |
| center (0.9) | open (0.6) | low (0.6) |

| Spatial Weighting Factor (crisp) | Cloudiness Value (crisp) | Temperature Value (crisp) |
|---|---|---|
| 0.80 | 0.125 | -12 |
| 0.90 | 0.125 | -11 |

| Spatially Weighted Cloudiness Value | Spatially Weighted Temperature Value |
|---|---|
| 0.10 | -9.6 |
| 0.14 | -9.9 |

| Aggregated Cloudiness Value (crisp) | Aggregated Temperature Value (crisp) |
|---|---|
| $\frac{0.24}{1.7}=0.14$ | $\frac{-19.5}{1.7}=-11.47$ |

| Aggregated Cloudiness Value (fuzzy) | Aggregated Temperature Value (fuzzy) |
|---|---|
| open(0.5) | low(0.6) |

Figure 4.24: Execution steps of aggregation for the sample conceptual query

We show how the cloudiness and temperature measurement results of the two stations are weighted with the hinterland values of the stations, and the aggregation is calculated. Also, the spatial weighting and aggregation of the temperature values of the two stations are demonstrated in this figure. When the results produced with these two aggregations for each city meet the "cloudiness is open" and "temperature is low" criteria, the cities with these results are included in the freezing city list.

Fuzzy summaries in the form of "$P\ objects\ are\ S\ :\ t$" can be generated using the aggregation results and listed as follows.

- Hinterland position of Ankara is center(0.6).
- Cloudiness of Ankara in 2015 winter is open(0.5).
- Temperature of Ankara in 2015 winter is low(0.6).

### 4.3.3 Prediction over Aggregated Data

As we explain that FSOLAP is capable of making inferences with the support of the FIS it contains. In summary, X-Means clustering is applied to precise measurement results, and the appropriate number of clusters is determined. The defined number of clusters and precise measurement data are fuzzified with FCM. The fuzzy association rule set is generated using the FP-Growth method on the fuzzy measurement results. Repetitive or unnecessary rules in this set are eliminated by rule pruning. After weighting the remaining rules with Rule Power Factor, the final minimized and the weighted fuzzy association rule set is produced. In addition to the membership classes, membership values, and membership functions generated by the FCM, the association rule set is used to construct the FIS.

In addition to the fuzzy association rule set generated by the measurement data, the expert-defined rule set for meteorological phenomena can also be created in FSOLAP by the meteorological domain expert. The domain expert uses the produced fuzzy association rules while defining the expert-defined rules. This study explains the process of making inferences using aggregated data with the following spatial fuzzy query sample.

**Predictive Query-1:** *According to the precipitation and temperature averages of the summer season, which cities are at high risk of drought in the coming years?*

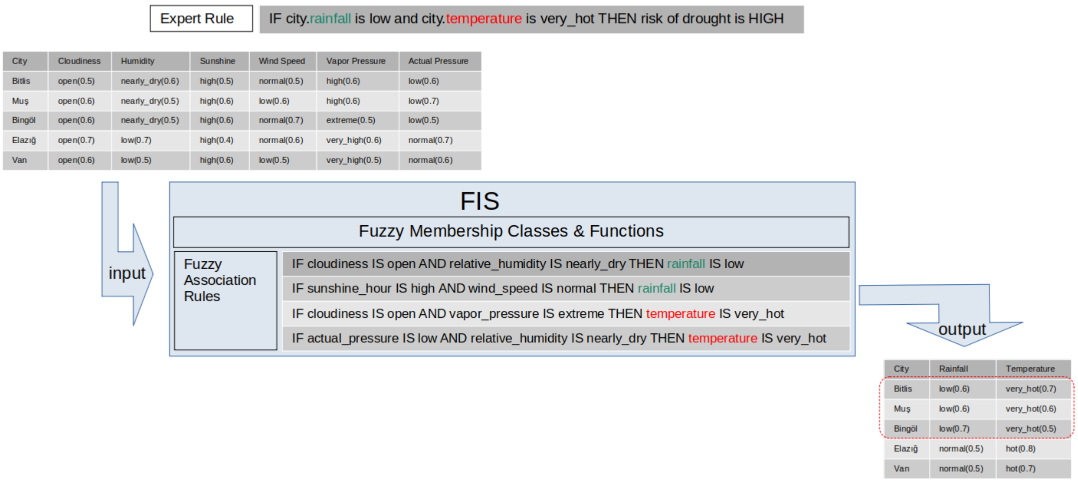
Meteorological drought is when precipitation falls below the average for a certain period. It is the difference between the annual, seasonal, or monthly precipitation totals from the norm. The high temperature and the lack of precipitation in the summer increase the risk of drought. The domain expert can define this meteorological phenomenon by following rule in the expert-defined rule set.

```
IF city.rainfall IS low AND city.temperature IS very high
    THEN risk of drought IS high
```

When creating this rule, the domain expert uses the generated fuzzy membership classes and values using measurement data stored in the FIS. The rainfall and temperature measurements in the antecedent section of the expert-defined rule are the predictive part of inference. The rules containing these fields are determined and executed in the consequent part of the fuzzy association rules. That is, rainfall and temperature values are missing in the testing records, and the remaining measurements are given as input to the fuzzy association rules. A sample of fuzzy association rules is represented below. Predicted rainfall and temperature values are generated as output of the FIS.

```
IF cloudiness IS open AND relative humidity IS nearly dry
      THEN rainfall IS low
IF sunshine hour IS high AND wind speed IS normal
      THEN rainfall out IS low
IF cloudiness is open AND vapor pressure is extreme
      THEN temperature IS very high
IF actual pressure IS low AND relative humidity IS nearly dry
      THEN temperature IS very high
```

In the example rule set above, the measurements in the form of cloudiness, relative humidity, sunshine hour, etc., are first aggregated based on summer period measurements. After these aggregated data are given as input to the rules, rainfall and temperature measurements are predicted as output. Figure 4.25 shows the process of the prediction over aggregated data.



IF city.rainfall is low and city.temperature is very_hot THEN risk of drought is HIGH

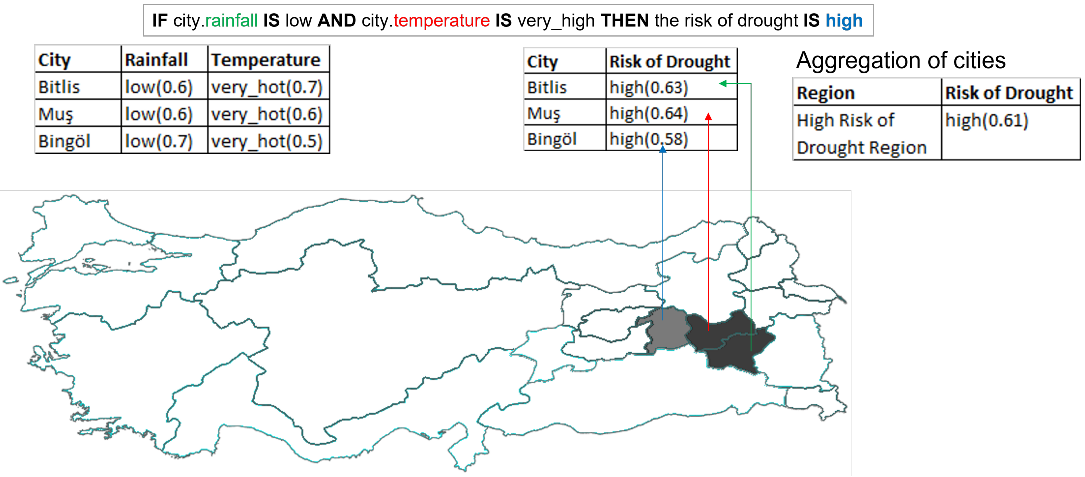
| City | Cloudiness | Humidity | Sunshine | Wind Speed | Vapor Pressure | Actual Pressure |
|---|---|---|---|---|---|---|
| Bitlis | open(0.5) | nearly_dry(0.6) | high(0.5) | normal(0.5) | high(0.6) | low(0.6) |
| Muş | open(0.6) | nearly_dry(0.5) | high(0.6) | low(0.6) | high(0.6) | low(0.7) |
| Bingöl | open(0.6) | nearly_dry(0.5) | high(0.6) | normal(0.7) | extreme(0.5) | low(0.5) |
| Elazığ | open(0.7) | low(0.7) | high(0.4) | normal(0.6) | very_high(0.6) | normal(0.7) |
| Van | open(0.6) | low(0.5) | high(0.6) | low(0.5) | very_high(0.5) | normal(0.6) |

IF cloudiness IS open AND relative_humidity IS nearly_dry THEN rainfall IS low
IF sunshine_hour IS high AND wind_speed IS normal THEN rainfall IS low
IF cloudiness IS open AND vapor_pressure IS extreme THEN temperature IS very_hot
IF actual_pressure IS low AND relative_humidity IS nearly_dry THEN temperature IS very_hot

| City | Rainfall | Temperature |
|---|---|---|
| Bitlis | low(0.6) | very_hot(0.7) |
| Muş | low(0.6) | very_hot(0.6) |
| Bingöl | low(0.7) | very_hot(0.5) |
| Elazığ | normal(0.5) | hot(0.8) |
| Van | normal(0.5) | hot(0.7) |

Figure 4.25: Prediction over aggregated data execution

In our example query, all meteorological measurements except rainfall and temperature are aggregated into the city hierarchy for the summer period, and results are obtained. These results are given as input to the FIS, and rainfall and temperature predictions are produced. The results in the form of "rainfall is low" and "temperature is high," which are the result of the inferences, provide the necessary conditions for drought. Figure 4.26 represents the cities in the query result satisfy what we are looking for in our query.



**IF** city.rainfall **IS** low **AND** city.temperature **IS** very_high **THEN** the risk of drought **IS** **high**

| City | Rainfall | Temperature |
|---|---|---|
| Bitlis | low(0.6) | very_hot(0.7) |
| Muş | low(0.6) | very_hot(0.6) |
| Bingöl | low(0.7) | very_hot(0.5) |

| City | Risk of Drought |
|---|---|
| Bitlis | high(0.63) |
| Muş | high(0.64) |
| Bingöl | high(0.58) |

Aggregation of cities

| Region | Risk of Drought |
|---|---|
| High Risk of Drought Region | high(0.61) |

Figure 4.26: Cities that are at high risk of drought in the coming years

We can also generate the following fuzzy summaries using the predictive aggregate query results.

- The risk of drought in the coming years for Bitlis is high(0.5).
- The predicted rainfall of Muş is low(0.6).
- The predicted temperature of Bingöl is very hot(0.5).

As another example query, the predictive query containing the fuzzy spatial concept, as in the conceptual query, is shown as follows. In this query, the rainfall in the Eastern Black Sea Region is predicted with the given fuzzy spatial predictive aggregate query.

**Predictive Query-2:** *How is the rain going to be in the lowlands of the East Black Sea Region?*

Orographic precipitation occurs in the Eastern Black Sea Region. Also known as slope precipitation, the rise of an air mass along a slope causes it to cool gradually. This situation reduces the maximum humidity, causing the air to become saturated with moisture. The lowland, the concept used in this query, is a spatial indicator and is related to the altitude of the location. The amount of precipitation a place receives in the region varies depending on the altitude. The humid air from the sea gets cold as it rises on the slopes parallel to the sea with the effect of the wind and leaves its moisture as precipitation. This situation reveals the relationship between altitude with precipitation. Since the humid air does not cool enough in places at very low altitudes, the amount of rainfall is less, and at higher altitudes, relatively more precipitation occurs. At very high altitudes, although the amount of precipitation is low because the clouds evacuate the moisture in it, the precipitation form changes due to the low temperature and turns into solid form in the form of hail or snow. The domain expert can define the probability of rain by following the expert-defined rule.

```
IF city.temperature IS normal AND city.relative humidity IS overmuch
        THEN the probability of rain IS high
```

Depending on the domain expert-defined rule above, we use the following example fuzzy association rules in FKB in the prediction process. We execute these rules to predict the temperature and relative humidity measurements in the rule defined by the domain expert.

```
IF cloudiness IS mostly cloudy AND vapor pressure IS high
        THEN temperature IS normal
IF actual pressure IS normal AND sunshine hour IS normal
        THEN temperature IS normal
IF actual pressure IS high AND cloudiness IS mostly cloudy
        THEN relative humidity IS much
```

Figure 4.27 shows the fuzzy membership functions we assembled for predicting relative humidity and temperature measurements in the sample fuzzy association rules.

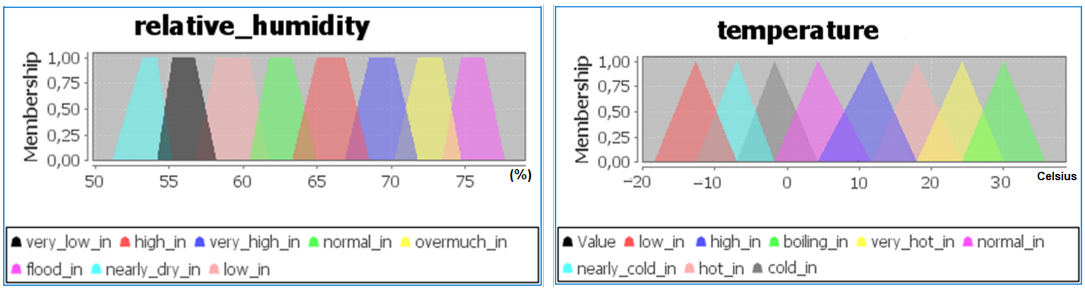


Figure 4.27: Membership functions of the relative humidity and temperature

For the fuzzy spatial concept in the form of *lowland*, the membership function of the altitude feature is represented in Figure 4.28.

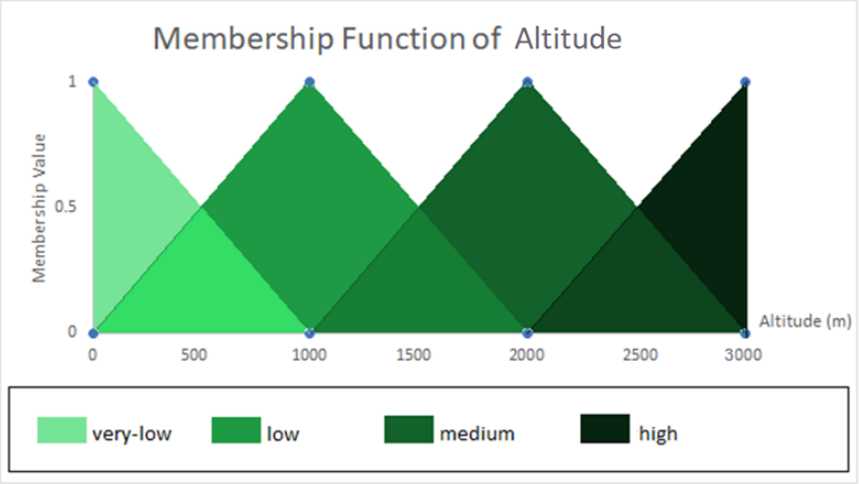
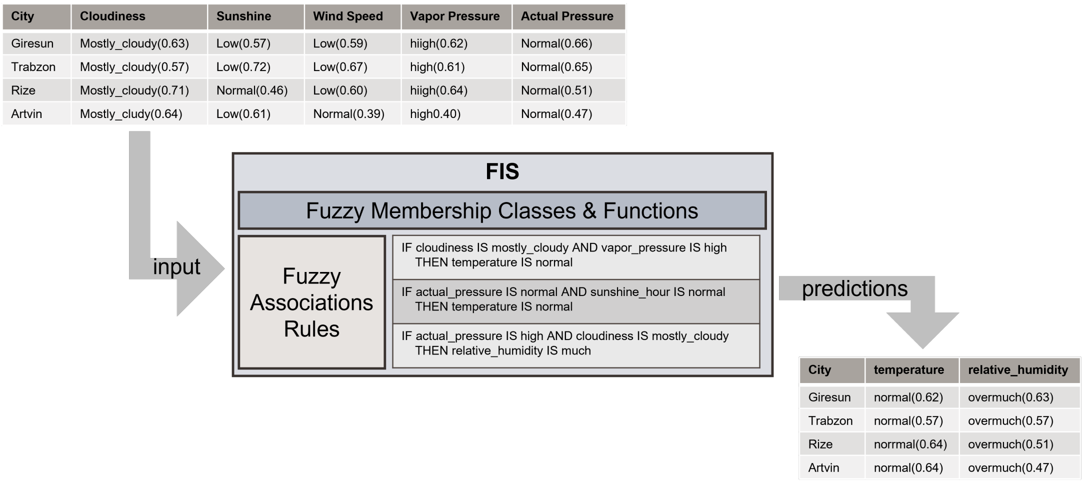


Figure 4.28: Membership functions of the altitude

In the first step of the fuzzy spatial predictive aggregate query, we start by querying the measurement values for input to the fuzzy association rules. These measurement values include station-based daily measurements. These values are weighted by considering the altitude values of the stations. The weighting process is in the form of multiplying each station's data by the weight of the altitude of the relevant station. This multiplication is applied by using precise values. The weighted station-based measurement values are aggregated, and the measurement values at the city level are calculated. Thus, we compute aggregated measurement values as necessary inputs for prediction. These values are sent to the FIS, the relevant association rules are executed, and the prediction results are generated. As a result of this process, we acquire predicted measurement results on a city basis, as shown in Figure 4.29.

In the second step of the query, after the predicted values of the cities are calculated, we use this data to execute the association rule defined by the domain expert and make a meteorological phenomenon prediction for the cities. We employ the prediction results we generated in the first step of each city as input in this step. Therefore, we obtain the prediction results for the probability of rain that we are searching for in the query on a city basis. As shown in Figure 4.30, we compute the value at the regional level by aggregating the city-based results. This aggregation is in the form of subjecting the results of the cities to the COG method of the fuzzy membership classes.



| City | Cloudiness | Sunshine | Wind Speed | Vapor Pressure | Actual Pressure |
|---|---|---|---|---|---|
| Giresun | Mostly_cloudy(0.63) | Low(0.57) | Low(0.59) | hiigh(0.62) | Normal(0.66) |
| Trabzon | Mostly_cloudy(0.57) | Low(0.72) | Low(0.67) | high(0.61) | Normal(0.65) |
| Rize | Mostly_cloudy(0.71) | Normal(0.46) | Low(0.60) | hiigh(0.64) | Normal(0.51) |
| Artvin | Mostly_cludy(0.64) | Low(0.61) | Normal(0.39) | high0.40) | Normal(0.47) |

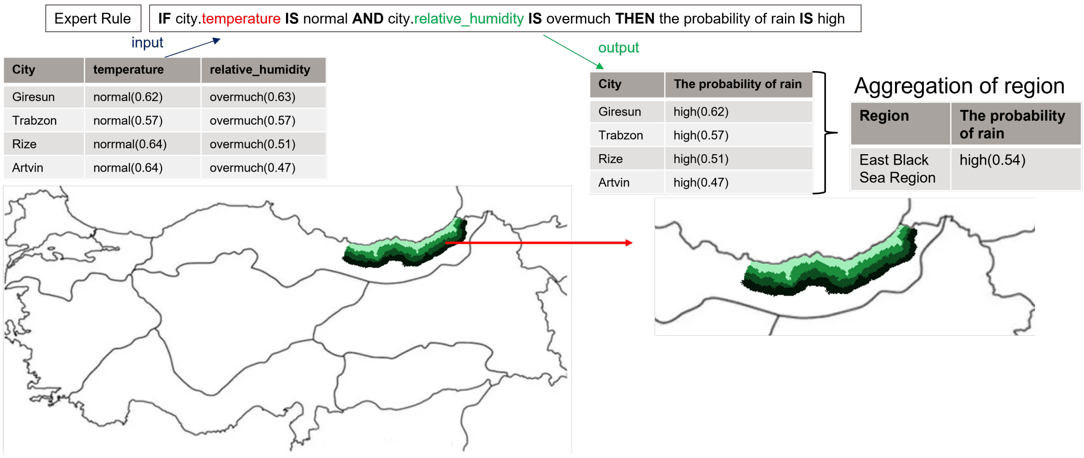
IF cloudiness IS mostly_cloudy AND vapor_pressure IS high THEN temperature IS normal

IF actual_pressure IS normal AND sunshine_hour IS normal THEN temperature IS normal

IF actual_pressure IS high AND cloudiness IS mostly_cloudy THEN relative_humidity IS much

| City | temperature | relative_humidity |
|---|---|---|
| Giresun | normal(0.62) | overmuch(0.63) |
| Trabzon | normal(0.57) | overmuch(0.57) |
| Rize | norrmal(0.64) | overmuch(0.51) |
| Artvin | normal(0.64) | overmuch(0.47) |

Figure 4.29: Execution of the fuzzy spatial predictive aggregate query



Expert Rule: IF city.temperature IS normal AND city.relative_humidity IS overmuch THEN the probability of rain IS high

| City | temperature | relative_humidity |
|---|---|---|
| Giresun | normal(0.62) | overmuch(0.63) |
| Trabzon | normal(0.57) | overmuch(0.57) |
| Rize | norrmal(0.64) | overmuch(0.51) |
| Artvin | normal(0.64) | overmuch(0.47) |

| City | The probability of rain |
|---|---|
| Giresun | high(0.62) |
| Trabzon | high(0.57) |
| Rize | high(0.51) |
| Artvin | high(0.47) |

| Region | The probability of rain |
|---|---|
| East Black Sea Region | high(0.54) |

Figure 4.30: Aggregation of the rainfall for the East Black Sea Region

Finally, the following fuzzy summaries can be generated using the fuzzy spatial predictive aggregate query results.

- The probability of rain for the East Black Sea Region is high(0.54).
- The probability of rain for Giresun is high(0.62).

# CHAPTER 5

# CASE STUDY: METEOROLOGICAL APPLICATION

After building the FSOLAP framework, it is needed to validate and gain valuable insights. For this purpose, this study uses real meteorological data obtained from the Turkish Meteorological Office. This meteorological data is massive with spatiotemporal information. Meteorological applications naturally include location and time information and also inherently have fuzziness. Therefore, systems involving fuzzy spatial and temporal phenomena are required.

## 5.1 Study Area

In this study, collected meteorological data in Turkey is used for testing. The geographic position of Turkey is in 26° to 45° East longitudes and 36° to 42° North latitudes. It also has seven geographical regions divided concerning their location, climate, agricultural diversities, topography, human habitat, flora and fauna, transportation, etc. These are the Central Anatolia, Mediterranean, Marmara, Black Sea, Eastern Anatolian, Southeastern Anatolia, and Aegean regions.

Gathered data contains measures from 1970 to 2017 collected from 1161 meteorological observation stations chosen from various cities, taking into account the distribution of the measurements throughout the country. Fig.5.1 displays a sample of cities of the observation stations.

The sample of meteorological stations is given in Table 5.1 as follows.

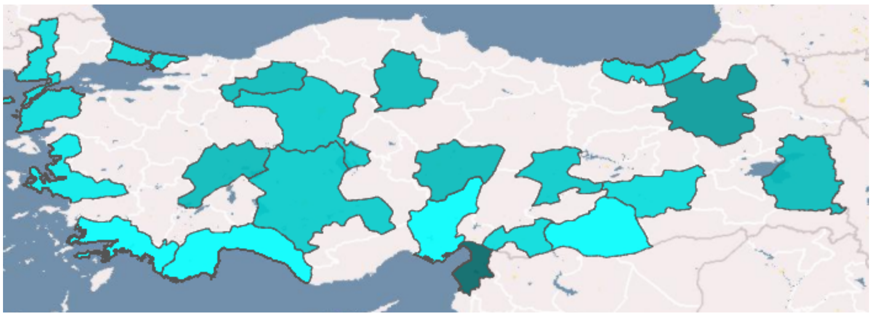


Figure 5.1: Sample cities of the meteorological stations

Table 5.1: The sample of meteorological measurement stations

| Station<br>No | Station<br>Name | City | Town | Latitude<br>(°) | Longitude<br>(°) | Altitude<br>(m) |
|---|---|---|---|---|---|---|
| 17026 | Sinop | Sinop | Merkez | 41.02 | 35.15 | 32 |
| 17091 | Sivas | Sivas | Merkez | 39.80 | 36.89 | 1600 |
| 17137 | Elmadağ | Ankara | Elmadağ | 39.79 | 32.97 | 1807 |
| 17262 | Kilis | Kilis | Merkez | 36.70 | 37.11 | 640 |
| 17681 | Zile | Tokat | Zile | 40.29 | 35.89 | 719 |

## 5.2 Data Sets

This study uses daily meteorological measurements data collected between 1970 and 2007. This data is in text-based files obtained from the meteorology office. Although there are many types of meteorological measurements, the study is carried out by taking into account nine different measures. The measurement types studied are as follows: relative humidity, temperature, rainfall, cloudiness, actual pressure, the direction of the wind, speed of the wind, vapor pressure, and sunshine hour. Table 5.2 shows the files received from meteorology, data types, and measurement units for each file.

The station file has no of the station, name of the station, city of the station, and coordinates of the station such as longitude, latitude, altitude points. Other files are in CSV format and include daily measures from 01.01.1970 to 01.01.2017. Each record of the file consists of station id, measurement type, measurement date, and

Table 5.2: Measurement Files and Details

| Filename | Description | Units |
|---|---|---|
| stations-info.txt | measurement station info | station name and coordinates |
| vapor-pressure.xlsx | vapor pressure measurements (daily) | hectopascal (1 hPa = 100 Pa) |
| sunshine(daily).xlsx | sunshine hours measurements (daily) | hours |
| temperature(average).xlsx | temperature (daily average) | celsius |
| wind-direction-speed.xlsx | wind direction and speed (daily max) | meter per second and direction |
| pressure(average).xlsx | actual pressure (daily average) | hectopascal (1 hPa = 100 Pa) |
| cloudness.xlsx | cloudness (daily average) | 8 octa |
| humidity(average).xlsx | relative humidity (daily average) | percentage |
| wind-speed(average).xlsx | wind speed (daily average) | meter per second |
| rainfall(manual).xlsx | rainfall (daily-manual total) | kg per meter square |
| rainfall(omgi).xlsx | rainfall (daily-omgi total) | kg per meter square |

measurement value.

Table 5.3 shows sample data for the actual pressure, and there is also a similar structure in the files for other measurements. The station number and name in the table are related to the records in the station table. The date column contains the measurement date, and the actual pressure column includes the measurement value.

Table 5.3: Actual pressure measurement data

| Station No | Station Name | Date | Actual Pressure (hPa ) |
|---|---|---|---|
| 17037 | Trabzon | 6 March 1971 | 1016.6 |
| 17199 | Malatya | 2 January 2011 | 954.3 |
| 17300 | Antalya | 1 January 1990 | 1013.5 |
| 17172 | Van | 23 May 1995 | 826.6 |
| 17070 | Bolu | 28 April 1982 | 922.3 |

Structured and spatially hierarchical data tables in PostGIS are shown in Fig. 5.2.

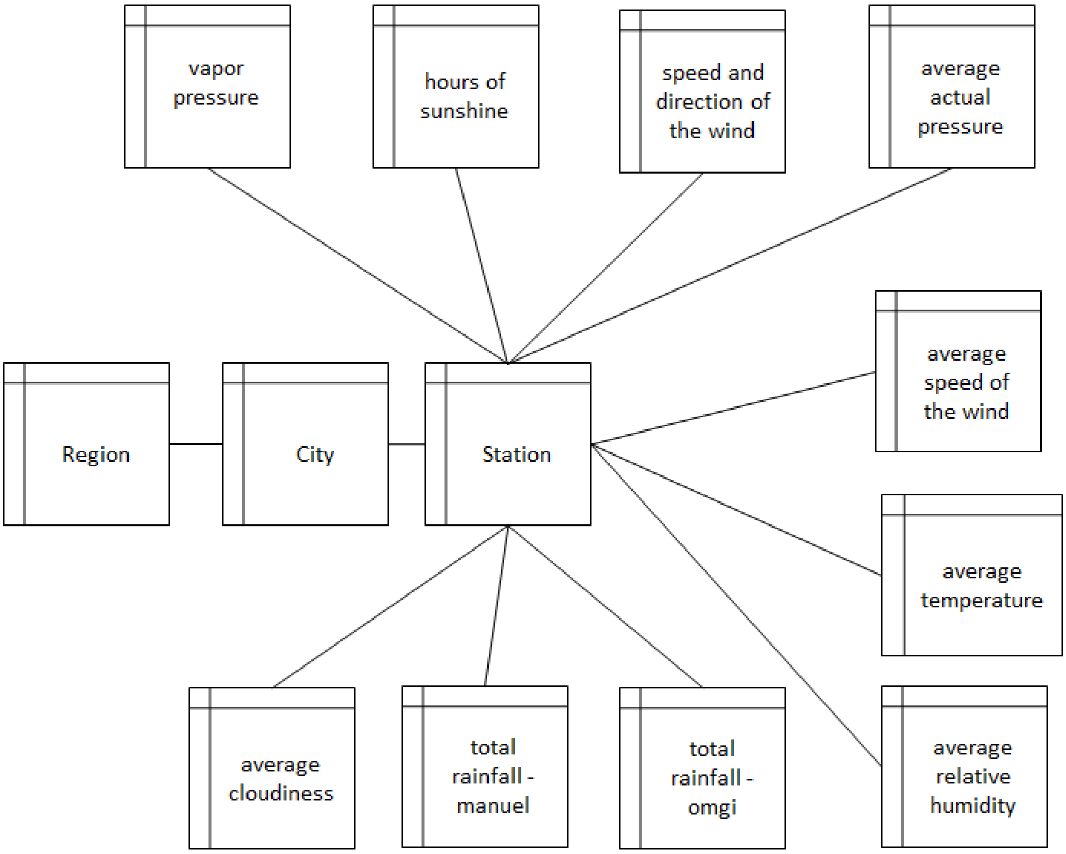


Figure 5.2: PostGIS database tables.

# CHAPTER 6

# EXPERIMENTAL RESULTS

In this study, the results of the experimental analysis of the framework using a meteorological dataset are discussed. In this section, the nature of the data is explained, and the relationships between the meteorological measurements are explored. Additionally, the FSOLAP framework is tested with the data to prove the validity of the approach and predictive analytics cases are performed to help instruct future steps. Query performance of the FSOLAP is also tested with the given environment and results are explained.

## 6.1 Prediction Accuracy Performance of the Framework

Tests are conducted using meteorological data to demonstrate the predictive performance of the proposed framework. Fig. 6.1 shows the general overview of FSOLAP-based prediction and its evaluation steps. This illustration represents how to make a generic FSOLAP-based prediction over the basic steps. Firstly, the fuzzy spatial cube is built after the fuzzification of structured training data. Therefore, fuzzy spatial data is stored in fuzzy spatial OLAP cube. The data required for the relevant meteorological phenomena on FSOLAP is retrieved with the MDX query. A fuzzy association rule set is generated from the fuzzy data. FIS is constructed by combining fuzzy membership classes, membership functions, and association rules. Predictions are made by processing the test data with FIS. FIS takes precise values as input and produces precise values as output. We cluster the predicted values to measure accuracy in the confusion matrix.

An example of applying FSOLAP based prediction and outcome of evaluation steps

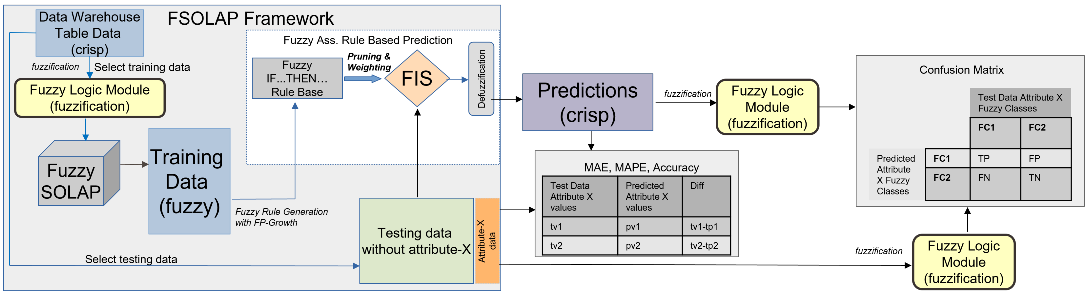
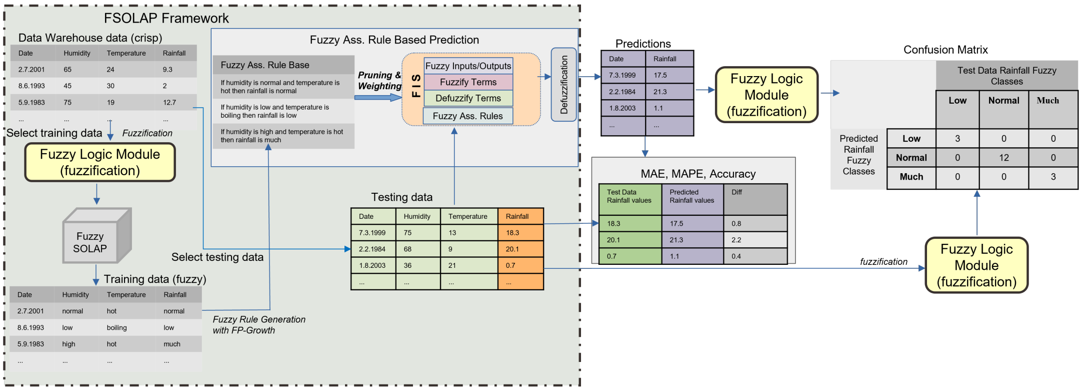

Data Warehouse Table Data (crisp)
FSOLAP Framework
fuzzification
Select training data
Fuzzy Logic Module (fuzzification)
Fuzzy SOLAP
Training Data (fuzzy)
Fuzzy Ass. Rule Based Prediction
Fuzzy IF...THEN... Rule Base
Pruning & Weighting
FIS
Defuzzification
Fuzzy Rule Generation with FP-Growth
Testing data without attribute-X
Attribute-X data
Select testing data
Predictions (crisp)
fuzzification
Fuzzy Logic Module (fuzzification)
MAE, MAPE, Accuracy
Test Data Attribute X values
Predicted Attribute X values
Diff
tv1
pv1
tv1-tp1
tv2
pv2
tv2-tp2
Confusion Matrix
Test Data Attribute X Fuzzy Classes
FC1
FC2
Predicted Attribute X Fuzzy Classes
FC1
TP
FP
FC2
FN
TN
fuzzification
Fuzzy Logic Module (fuzzification)

Figure 6.1: General view of FSOLAP based prediction and evaluation steps

is illustrated in Fig. 6.2. This figure shows a sample fuzzy dataset produced after fuzzification of precise data. This fuzzy data is stored in the fuzzy SOLAP cube. An example fuzzy association rule set generated from fuzzified data is also shown. In addition to the MAE, MAPE, and Accuracy values calculated by comparing the predicted data with the actual data, the accuracy computation of the data clustered in the confusion matrix is also demonstrated.



Data Warehouse data (crisp)

| Date | Humidity | Temperature | Rainfall |
|---|---|---|---|
| 2.7.2001 | 65 | 24 | 9.3 |
| 8.6.1993 | 45 | 30 | 2 |
| 5.9.1983 | 75 | 19 | 12.7 |
| ... | ... | ... | ... |

Training data (fuzzy)

| Date | Humidity | Temperature | Rainfall |
|---|---|---|---|
| 2.7.2001 | normal | hot | normal |
| 8.6.1993 | low | boiling | low |
| 5.9.1983 | high | hot | much |
| ... | ... | ... | ... |

Fuzzy Ass. Rule Base

If humidity is normal and temperature is hot then rainfall is normal

If humidity is low and temperature is boiling then rainfall is low

If humidity is high and temperature is hot then rainfall is much

Testing data

| Date | Humidity | Temperature | Rainfall |
|---|---|---|---|
| 7.3.1999 | 75 | 13 | 18.3 |
| 2.2.1984 | 68 | 9 | 20.1 |
| 1.8.2003 | 36 | 21 | 0.7 |
| ... | ... | ... | ... |

Predictions

| Date | Rainfall |
|---|---|
| 7.3.1999 | 17.5 |
| 2.2.1984 | 21.3 |
| 1.8.2003 | 1.1 |
| ... | ... |

MAE, MAPE, Accuracy

| Test Data Rainfall values | Predicted Rainfall values | Diff |
|---|---|---|
| 18.3 | 17.5 | 0.8 |
| 20.1 | 21.3 | 2.2 |
| 0.7 | 1.1 | 0.4 |

Confusion Matrix

| | | Test Data Rainfall Fuzzy Classes | | |
|---|---|---|---|---|
| | | Low | Normal | Much |
| Predicted Rainfall Fuzzy Classes | Low | 3 | 0 | 0 |
| | Normal | 0 | 12 | 0 |
| | Much | 0 | 0 | 3 |

Figure 6.2: An example application of FSOLAP based prediction and evaluation

In this study, the confusion matrix is utilized to summarize prediction results and measure the effectiveness of the FSOLAP framework. The confusion matrix compares the predictions of the target attribute and the actual values. It describes the performance of a model on a set of test data for which the correct values are known. And it also visualizes the accuracy of an algorithm. We can interpret the performance of our classification model using a confusion matrix. At the same time, with the help of the confusion matrix, we can find our different metric values and evaluate our model success in detail according to the process we do. The confusion matrix is a square matrix where the column represents the actual values, and the row depicts the predicted value of the model and vice versa. More specifically, a confusion matrix presents how a classification model becomes confused while making predictions. An exemplary matrix (model) will have large values across the diagonal and small values off the diagonal. Measuring a confusion matrix provides better insight into what our classification model is getting correct and what types of errors it creates.

Let us define the terminology and derivations from a confusion matrix as follows. The accuracy, or rate, is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

Recall is a metric that shows how much of the operations we need to estimate as positive, we estimate as positive. A high recall points that the class is identified correctly (small FN).

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

On the other hand, precision shows how many of the values we estimated as positive are actually positive. High precision means a bar labeled as positive is definitely positive (small FP).

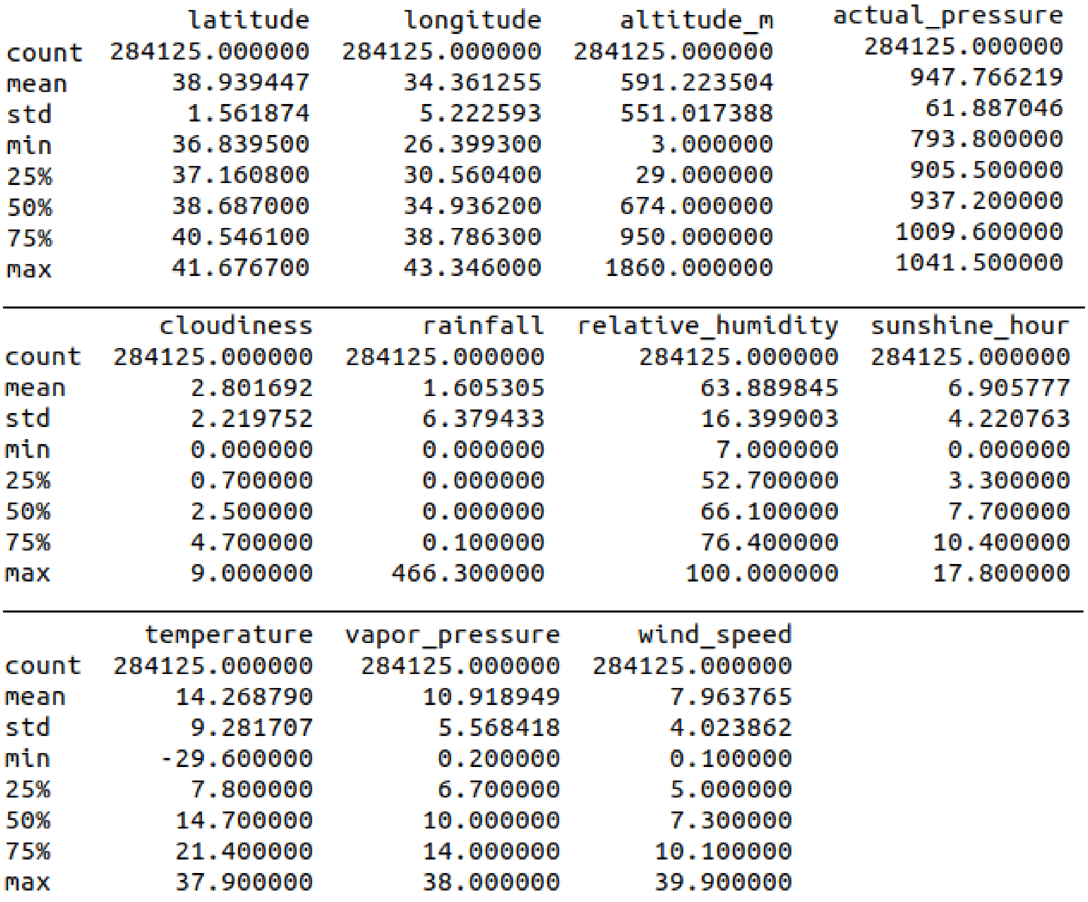
$$Precision = \frac{TP}{TP + FP} \tag{6.3}$$

High recall and low precision point out that most positive examples are correctly recognized (small FN), but many false positives exist. On the other hand, low recall and high precision show that many positive examples are missed(high FN), but those we predict as positive are indeed positive (small FP).

After having precision and recall, F-measure can also be calculated by using both. F-measure or F1 Score value shows us the harmonic average of precision and recall values. It uses a harmonic instead of arithmetic mean because we should not ignore extreme cases.

$$F - Measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{6.4}$$

Before making predictions with FSOLAP, the descriptive statistics of the dataset, which represent those that summarize the central tendency, dispersion, and shape of the distribution of the dataset, excluding NaN values, are explained. The summary of statistics about each feature in the dataset contains min, max, count, mean, std, and lower/upper percentiles of 50. The 50 percentile is identical to the median, as shown in Fig. 6.3.



```
            latitude      longitude     altitude_m    actual_pressure
count  284125.000000  284125.000000  284125.000000      284125.000000
mean       38.939447      34.361255     591.223504         947.766219
std         1.561874       5.222593     551.017388          61.887046
min        36.839500      26.399300       3.000000         793.800000
25%        37.160800      30.560400      29.000000         905.500000
50%        38.687000      34.936200     674.000000         937.200000
75%        40.546100      38.786300     950.000000        1009.600000
max        41.676700      43.346000    1860.000000        1041.500000
```

```
          cloudiness       rainfall  relative_humidity  sunshine_hour
count  284125.000000  284125.000000      284125.000000  284125.000000
mean        2.801692       1.605305          63.889845       6.905777
std         2.219752       6.379433          16.399003       4.220763
min         0.000000       0.000000           7.000000       0.000000
25%         0.700000       0.000000          52.700000       3.300000
50%         2.500000       0.000000          66.100000       7.700000
75%         4.700000       0.100000          76.400000      10.400000
max         9.000000     466.300000         100.000000      17.800000
```

```
         temperature  vapor_pressure     wind_speed
count  284125.000000   284125.000000  284125.000000
mean       14.268790       10.918949       7.963765
std         9.281707        5.568418       4.023862
min       -29.600000        0.200000       0.100000
25%         7.800000        6.700000       5.000000
50%        14.700000       10.000000       7.300000
75%        21.400000       14.000000      10.100000
max        37.900000       38.000000      39.900000
```

Figure 6.3: Summary of the meteorological dataset

A heat map is demonstrated as another tool to analyze the overall view of the dataset. The heat map is a two-dimensional visual representation of data. It displays the individual values in a matrix as colors. Fig. 6.4 shows high-level relations on the dataset. Similar features are shown as blue boxes (interval 0 to 1), and dissimilar features are given as red boxes (range is -1 to 0). There is a remarkable high dissimilarity between actual pressure and altitude, as shown in the heat map.

Another overall operation on data in this study is the clustering of each meteorological

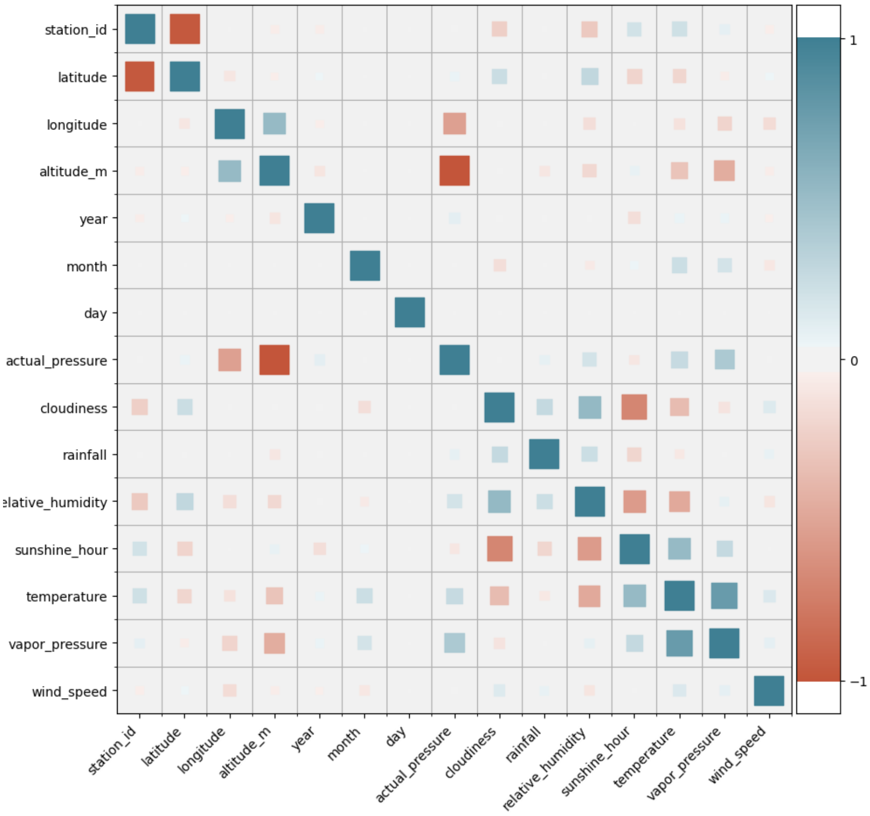


Figure 6.4: Heat map of the meteorological dataset

attribute. In this study, X-means clustering is executed to specify the proper number of clusters. After that the elbow and silhouette methods are applied to cross-check the X-means result. Elbow is a method that examines the ratio of variance explained as a function of the number of clusters [67]. Another technique is the silhouette which refers to interpretation and consistency within data clusters. The technique offers a concise graphical representation of how well each object has been classified [68]. The output of elbow and silhouette methods about the optimum number of clusters for actual pressure data is given in Fig. 6.5. The correct number of clusters is six for actual pressure as we got it by utilizing X-means clustering. We do the same cross-check for relative humidity, sunshine hour, temperature features to verify X-means clustering results.

In our study, it is essential to determine the appropriate number of clusters. The number of clusters should be appropriate because it is a factor that affects the accuracy and the performance of the prediction. The Silhouette method is performed to control

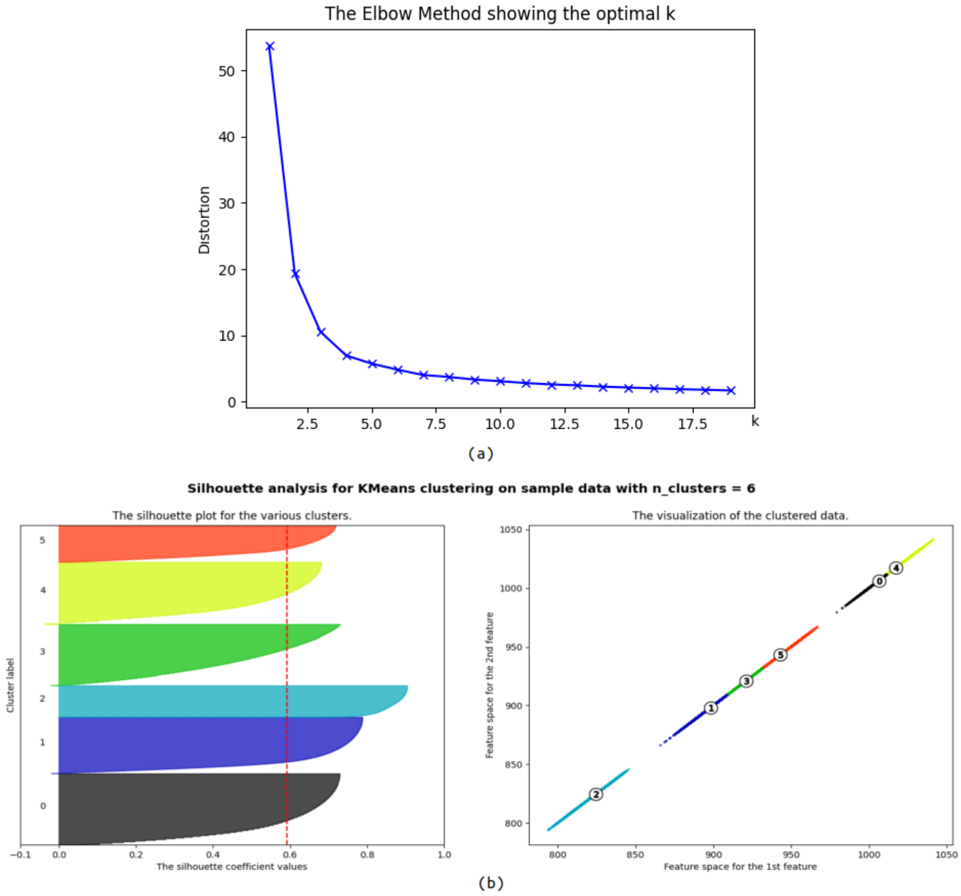


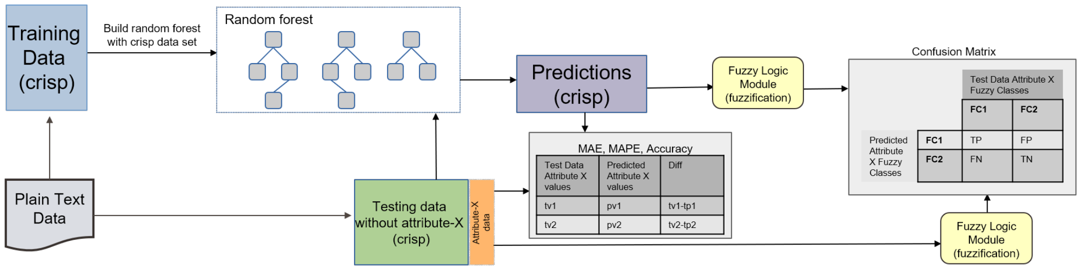
Figure 6.5: (a) Elbow graph and (b) Silhouette graph of actual pressure data

the appropriateness of the number of clusters. While using the method, precise measurement values are given as input, and the appropriate number of clusters is acquired as output.

We let the model learn the answers during training to predict the target feature using all other features. The model learns the expected relationship between all the features and the target value during the training. The learned relations are used to predict a test dataset in the model evaluation time. The prediction results are compared with the known values to measure how accurate the model is by using the actual values in the test dataset. In the data preparation process before the analysis, the data is split into two groups training data and test data. Generally, random selection is performed while grouping the data so that a particular part of the data does not have an effect. There are nearly 15 M rows in the meteorological dataset, consisting of measurements from different stations. 5-fold cross-validation is applied, and results are averaged to produce a single estimation. Therefore, 80% of the data is selected for training, and the remaining 20% is used for testing.

After introducing the general view of data and splitting the dataset into the train and test datasets, the model is trained with the training dataset using SVM, Random Forest (RF), Fuzzy Random Forest (FRF), and the FSOLAP framework. SVM is one of the supervised learning methods generally used in classification problems. It draws a line to separate points placed on a plane. It aims to have this line at the maximum distance for the points of both classes. Random forest is a classification model that tries to make more accurate classification by producing more compatible models using multiple decision trees [85]. They are ensembles of decision trees, each decision tree constructed by using a subset of the features used to classify a given population (they are sub-trees, prevent outliers). These decision trees vote on classifying a given input data instance, and the random forest bootstraps these votes to choose the best prediction. The fuzzy random forest is an ensemble based on fuzzy decision trees. It is a multiple classifier system based on a forest of fuzzy decision trees. This strategy integrates the robustness of multiple classifier systems, the power of randomness to increase the diversity of trees, and the flexibility of fuzzy logic and fuzzy sets for imperfect data management [86]. We represent the general view of prediction and result evaluation steps for the random forest model in Fig. 6.6. While making predictions with random forest, we first separate the text data into two groups training and testing data. Then, we construct a forest containing n decision trees that we have determined using the training data. We make predictions by giving the test data we have defined before to the resulting forest as input. To compare the predicted data with the actual data, we calculate MAE, MAPE, and Accuracy values, and we measure accuracy in the confusion matrix by fuzzy clustering the data. Thus, we make a fair comparison with the FSOLAP-based prediction results.

The evaluation steps of an example application of random forest prediction are demonstrated in Fig. 6.7, a visual summary of this process. Here, the training data model is executed to learn the relationships between the features and the targets. The next step is to determine how good the model is. For this purpose, predictions on the test data are made, and keep in mind that the model is never let to see the test answers. Then a comparison between the predictions and the known values is completed. Finally,

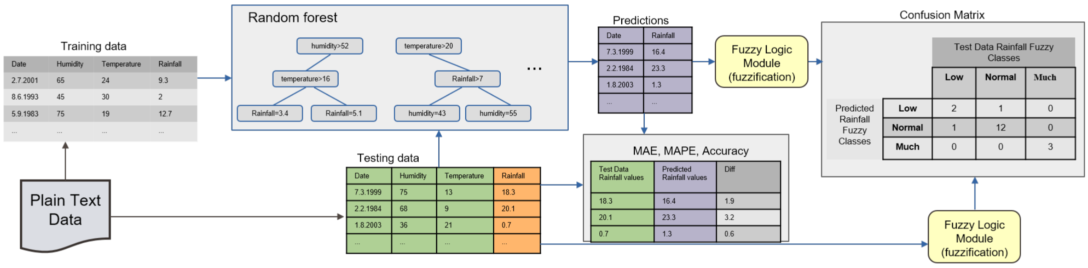

Training Data (crisp)
Build random forest with crisp data set
Random forest
Predictions (crisp)
Fuzzy Logic Module (fuzzification)
Confusion Matrix
Plain Text Data
Testing data without attribute-X (crisp)
Attribute-X data
MAE, MAPE, Accuracy
Fuzzy Logic Module (fuzzification)

| Test Data Attribute X values | Predicted Attribute X values | Diff |
|---|---|---|
| tv1 | pv1 | tv1-tp1 |
| tv2 | pv2 | tv2-tp2 |

| Predicted Attribute X Fuzzy Classes | | Test Data Attribute X Fuzzy Classes | |
|---|---|---|---|
| | | FC1 | FC2 |
| | FC1 | TP | FP |
| | FC2 | FN | TN |

Figure 6.6: General view of Random Forest model prediction and result evaluation

the result is fuzzified to use them in confusion matrices or directly use the result to calculate the mean absolute error.



Training data

| Date | Humidity | Temperature | Rainfall |
|---|---|---|---|
| 2.7.2001 | 65 | 24 | 9.3 |
| 8.6.1993 | 45 | 30 | 2 |
| 5.9.1983 | 75 | 19 | 12.7 |
| ... | ... | ... | ... |

Predictions

| Date | Rainfall |
|---|---|
| 7.3.1999 | 16.4 |
| 2.2.1984 | 23.3 |
| 1.8.2003 | 1.3 |
| ... | ... |

Testing data

| Date | Humidity | Temperature | Rainfall |
|---|---|---|---|
| 7.3.1999 | 75 | 13 | 18.3 |
| 2.2.1984 | 68 | 9 | 20.1 |
| 1.8.2003 | 36 | 21 | 0.7 |
| ... | ... | ... | ... |

MAE, MAPE, Accuracy

| Test Data Rainfall values | Predicted Rainfall values | Diff |
|---|---|---|
| 18.3 | 16.4 | 1.9 |
| 20.1 | 23.3 | 3.2 |
| 0.7 | 1.3 | 0.6 |

Confusion Matrix

| Predicted Rainfall Fuzzy Classes | | Test Data Rainfall Fuzzy Classes | | |
|---|---|---|---|---|
| | | Low | Normal | Much |
| | Low | 2 | 1 | 0 |
| | Normal | 1 | 12 | 0 |
| | Much | 0 | 0 | 3 |

Figure 6.7: A sample application of Random Forest model for prediction and evaluation of results

In the first case, the prediction of the actual pressure measurement is analyzed. This operation starts with the SVM model, then uses the random forest, fuzzy random forest, and finally, our FSOLAP framework. While exploring a feature, i.e., actual pressure, the importance of the feature is checked for the model. In Fig. 6.8, it is interesting to observe that the most important feature is altitude. Here, we should note that the actual pressure and altitude have high dissimilarity in the heat map, as shown in Fig. 6.4.

The confusion matrices of prediction results of the actual pressure are prepared for each model, as shown in Fig. 6.9. Here is a comparative analysis of the predicted values of the SVM, RF, FRF, and FSOLAP-based prediction processes with the actual values.

ML models produce precise prediction results from precise test inputs. On the other hand, FSOLAP produces fuzzy prediction results from crisp test data. In compar-

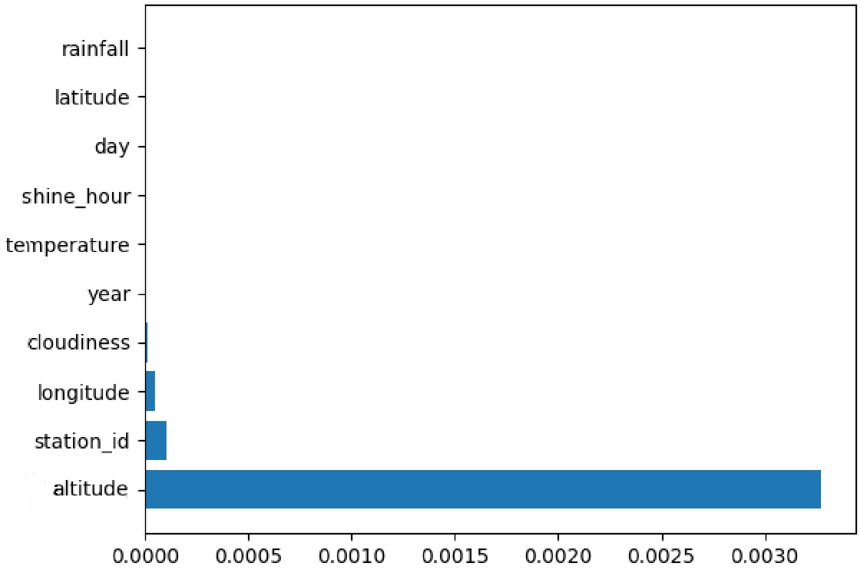
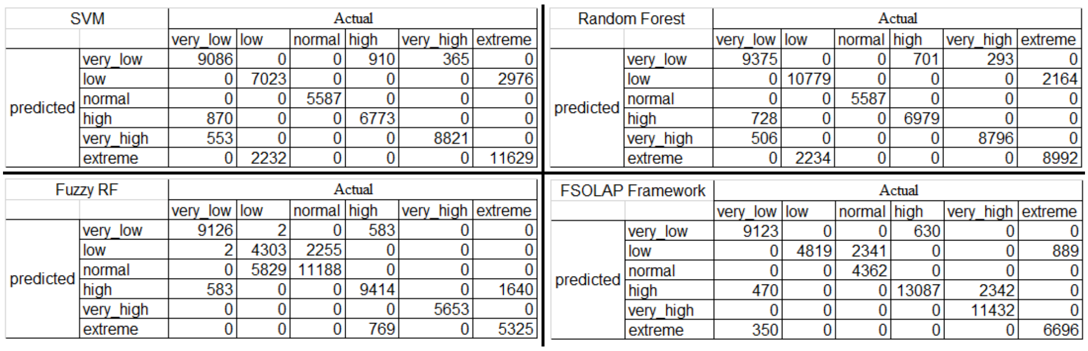


Figure 6.8: Feature/Variable importance of SVM based prediction for actual pressure

ing the results obtained, we use MAE, MAPE, and accuracy evaluations made with precise values, as we mentioned before. Since it already produces precise prediction results with ML models, these values are used directly in MAE, MAPE, and accuracy calculations. The fuzzy predictions made by FSOLAP are defuzzified with the defuzzification process, and precise values are calculated. Using these values, MAE, MAPE, and accuracy computations are performed for FSOLAP. MAE, MAPE, and accuracy values calculated with the results of ML models are compared with the MAE, MAPE, and accuracy calculations produced with the prediction results produced by FSOLAP.

In order to compare the results produced by the ML models with the results produced by FSOLAP using the confusion matrix, the results produced by the ML models are fuzzified with the fuzzification process. In this case, the results produced by both ML models and FSOLAP are fuzzified. The fuzzification process also fuzzifies the actual values of the testing data. Thus, we use the actual fuzzified data and the predicted results to create a confusion matrix by considering the membership classes.

Table 6.1 represents the statistical analysis for the predictive analytics, which are computed using confusion matrices and the given terminology definitions at the beginning of this section.



| SVM | | Actual | | | | | |
|---|---|---|---|---|---|---|---|
| | | very_low | low | normal | high | very_high | extreme |
| predicted | very_low | 9086 | 0 | 0 | 910 | 365 | 0 |
| | low | 0 | 7023 | 0 | 0 | 0 | 2976 |
| | normal | 0 | 0 | 5587 | 0 | 0 | 0 |
| | high | 870 | 0 | 0 | 6773 | 0 | 0 |
| | very_high | 553 | 0 | 0 | 0 | 8821 | 0 |
| | extreme | 0 | 2232 | 0 | 0 | 0 | 11629 |

| Random Forest | | Actual | | | | | |
|---|---|---|---|---|---|---|---|
| | | very_low | low | normal | high | very_high | extreme |
| predicted | very_low | 9375 | 0 | 0 | 701 | 293 | 0 |
| | low | 0 | 10779 | 0 | 0 | 0 | 2164 |
| | normal | 0 | 0 | 5587 | 0 | 0 | 0 |
| | high | 728 | 0 | 0 | 6979 | 0 | 0 |
| | very_high | 506 | 0 | 0 | 0 | 8796 | 0 |
| | extreme | 0 | 2234 | 0 | 0 | 0 | 8992 |

| Fuzzy RF | | Actual | | | | | |
|---|---|---|---|---|---|---|---|
| | | very_low | low | normal | high | very_high | extreme |
| predicted | very_low | 9126 | 2 | 0 | 583 | 0 | 0 |
| | low | 2 | 4303 | 2255 | 0 | 0 | 0 |
| | normal | 0 | 5829 | 11188 | 0 | 0 | 0 |
| | high | 583 | 0 | 0 | 9414 | 0 | 1640 |
| | very_high | 0 | 0 | 0 | 0 | 5653 | 0 |
| | extreme | 0 | 0 | 0 | 769 | 0 | 5325 |

| FSOLAP Framework | | Actual | | | | | |
|---|---|---|---|---|---|---|---|
| | | very_low | low | normal | high | very_high | extreme |
| predicted | very_low | 9123 | 0 | 0 | 630 | 0 | 0 |
| | low | 0 | 4819 | 2341 | 0 | 0 | 889 |
| | normal | 0 | 0 | 4362 | 0 | 0 | 0 |
| | high | 470 | 0 | 0 | 13087 | 2342 | 0 |
| | very_high | 0 | 0 | 0 | 0 | 11432 | 0 |
| | extreme | 350 | 0 | 0 | 0 | 0 | 6696 |

Figure 6.9: Confusion matrices of prediction for actual pressure

Table 6.1: Results of Statistical Analysis for Actual Pressure

| | SVM | RF | FRF | FSOLAP |
|---|---|---|---|---|
| accuracy | 0.8586 | **0.8791** | 0.791 | 0.8753 |
| AUC | 0.9177 | **0.9273** | 0.8611 | 0.8905 |
| recall | 0.86 | **0.89** | 0.80 | 0.88 |
| F1 | 0.87 | **0.89** | 0.80 | 0.88 |
| precision | 0.87 | **0.89** | 0.82 | 0.93 |

According to Table 6.1, The performance results of all the models are close to each other except for the fuzzy random forest. The fuzzy random forest has the lowest accuracy, AUC, recall, F1 score, and precision. Therefore, the proposed model of FSOLAP is relatively accurate and scalable because it has close scores to the random forest model, which is the best model among all. Here, the actual pressure data has outliers, and it is unbalanced, as shown in the histogram of data in Fig. 6.10. Random forest manages outliers by basically binning them. It is also indifferent to non-linear features. It has strategies for balancing errors in class population unbalanced datasets. It tries to minimize the overall error rate, so when there is an unbalanced dataset, the larger class may get a low error rate while the smaller class will have a more significant error rate. As a result, the random forest model is a little bit better than the FSOLAP framework.

The second prediction is about the relative humidity, and the statistical results are given in Table 6.2.

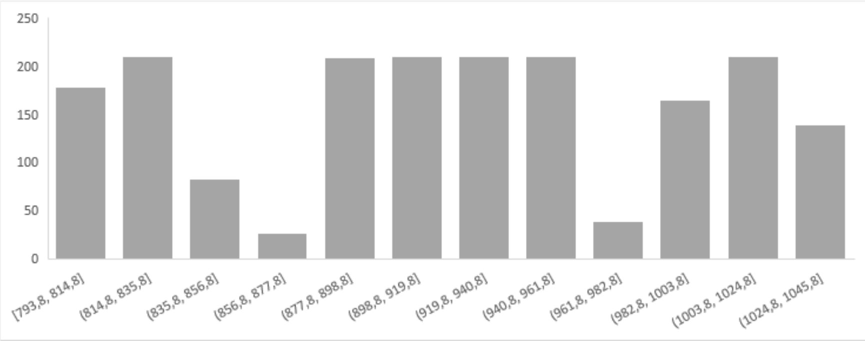


Figure 6.10: Histogram of actual pressure data

Table 6.2: Results of Statistical Analysis for Relative Humidity

| | SVM | RF | FRF | FSOLAP |
|---|---|---|---|---|
| accuracy | 0.7904 | 0.8152 | 0.8202 | **0.8358** |
| AUC | 0.8343 | 0.8712 | 0.8661 | **0.8813** |
| recall | 0.79 | 0.82 | 0.83 | **0.84** |
| precision | 0.80 | 0.83 | 0.84 | **0.86** |
| F1 Score | 0.79 | 0.82 | 0.83 | **0.84** |

FSOLAP has the highest scores in this case, but the remaining models are also accurate as they have scores close to our model. The relative humidity data histogram is shown in Fig. 6.11, which has a normal distribution. Therefore, FSOLAP performs significantly better than others with normally distributed data.

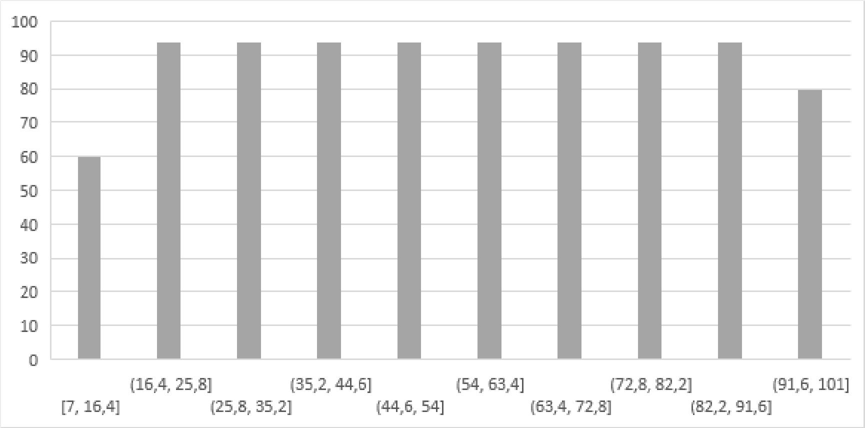


Figure 6.11: Histogram of relative humidity data

Another analysis is made with the sunshine hour feature and the computed statistics are presented in Table 6.3. As the results show, all the models have low scores compared to the previous analyses. Our model and the Random Forest model are the winners with close scores; on the other hand, SVM and Fuzzy Random Forest do not perform well.

Table 6.3: Results of Statistical Analysis for Sunshine Hour

| | SVM | RF | FRF | FSOLAP |
|---|---|---|---|---|
| accuracy | 0.6196 | 0.7125 | 0.6205 | **0.7210** |
| AUC | 0.7318 | 0.8002 | 0.6689 | **0.7951** |
| recall | 0.62 | 0.72 | 0.60 | **0.72** |
| precision | 0.61 | 0.71 | 0.60 | **0.75** |
| F1 Score | 0.61 | 0.71 | 0.60 | **0.72** |

The last prediction is made for the temperature measurement, and Table 6.4 shows the results of the statistical analysis.

Table 6.4: Results of Statistical Analysis for Temperature

| | SVM | RF | FRF | FSOLAP |
|---|---|---|---|---|
| accuracy | 0.9174 | 0.9248 | 0.9127 | **0.9202** |
| AUC | 0.9462 | 0.9471 | 0.9139 | **0.9524** |
| recall | 0.92 | 0.89 | 0.91 | **0.92** |
| precision | 0.92 | 0.89 | 0.91 | **0.92** |
| F1 Score | 0.92 | 0.89 | 0.91 | **0.92** |

In this last case, all models are good without any exceptions. The Random Forest also has scores similar to our model in the normally distributed temperature data, as shown in Fig. 6.12.

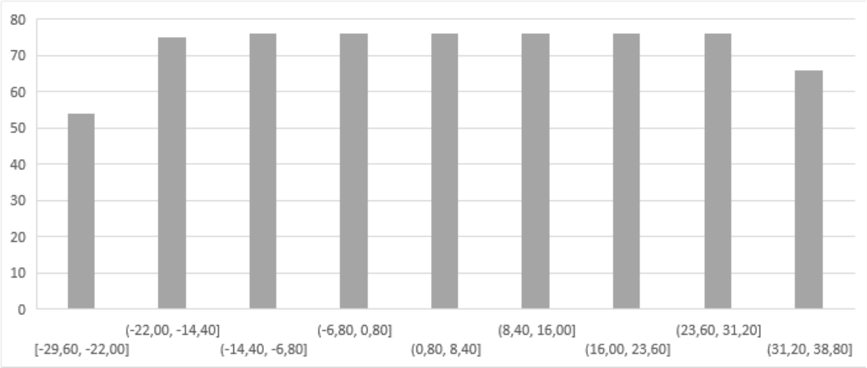


Figure 6.12: Histogram of temperature data

Finally, it is represented that FSOLAP-based prediction is quite accurate and scalable according to the results of our statistical analysis. Therefore, the success of the FSOLAP is validated with several comparisons with well-known models such as SVM, Random Forest, and Fuzzy Random Forest.

The prototype application is tested with the specifications, technology, and tools in the following environment.

- *Development IDE*: Eclipse IDE 2021-06
- *Operating System*: Windows 10 x64 with Intel i5-7200U CPU and 16 GB RAM
- *Java*:11.0.13, Java HotSpot Client 64-bit VM 11.0.13+8
- *Python*: 3.8 for 64-bit Server
- *SOLAP Server*: GeoMondrian 1.0 Server
- *Database*: PostgreSQL 13.4
- *Fuzzy Inference System*: jFuzzyLogic.1.0.jar
- *Data Size*: approximately 12 GB data consisting of 1161 stations and 15 M records for each measurement (15 M × 10 measurement types).

The average CPU usage (ACU), the average memory usage (AMU), the model building time (MBT), and the model prediction time (MPT) are measured by running each prediction model, including the FSOLAP framework. The measurements in the form of model building time and model prediction time given in the study are to be thought of as computation time. A one-time model building operation is performed at the beginning to create the environment with raw data. Since prediction can be made more than once with the different datasets using the built model, the model prediction time is evaluated individually. These two operations provide a more detailed presentation of computation time. The performance test results for all models are shown in Table 6.5. We calculate the average values by taking the arithmetic average of the results we obtained in the tests performed five times.

Table 6.5: Performance test results of the prediction models

| Measurement Type | Performance Metric | SVM | Radom Forest | Fuzzy Random Forest | FSOLAP |
|---|---|---|---|---|---|
| Actual Pressure | ACU (%) | 31.7 | 36.6 | 34.6 | 27.3 |
| | AMU (mb) | 462 | 4625 | 6739 | 751 |
| | MBT (sec) | 1442 | 592 | 882 | 747 |
| | MPT (sec) | 102 | 57 | 79 | 38 |
| Relative Humidity | ACU (%) | 32.1 | 34.4 | 32.7 | 25.5 |
| | AMU (mb) | 1065 | 6173 | 6694 | 745 |
| | MBT (sec) | 1428 | 518 | 971 | 615 |
| | MPT (sec) | 127 | 65 | 89 | 29 |
| Sunshine Hour | ACU (%) | 32.3 | 35.1 | 34.1 | 25.9 |
| | AMU (mb) | 456 | 7975 | 6763 | 727 |
| | MBT (sec) | 1459 | 561 | 1061 | 709 |
| | MPT (sec) | 109 | 63 | 81 | 34 |
| Temperature | ACU (%) | 32.1 | 33.9 | 33.9 | 26.4 |
| | AMU (mb) | 369 | 3478 | 5812 | 755 |
| | MBT (sec) | 1549 | 565 | 1127 | 805 |
| | MPT (sec) | 113 | 65 | 80 | 32 |

The measurement values are evaluated in the table individually for each measurement type based on average CPU and memory usage, model building time, and model prediction time. Here, average CPU usage is the average CPU usage rate measured during model building and prediction. Similarly, the average memory usage is measured in megabytes (MB) in model building and prediction. The model is trained on the training dataset during the model building time. The model prediction time is about testing the built model on test data. In this context, the average CPU usage of the SVM, random forest, fuzzy random forest, and FSOLAP models are compared over the column chart, as shown in Fig. 6.13. In the graph, the CPU utilization rates are Random Forest, Fuzzy Random Forest, SVM, and FSOLAP. Building 1000 decision trees, finding results, and voting require more processing power than other models for the Random Forest and Fuzzy Random Forest models. The SVM method finds the best line to determine the hyperplane for which the maximum margin is the optimal hyperplane. Here, the SVM model needs computational power for tuning operation. FSOLAP requires computational power to select the relevant rule from the association rule set and apply fuzzification-defuzzification procedures for input/output data. This process demands less computational power compared to the operations of other models.

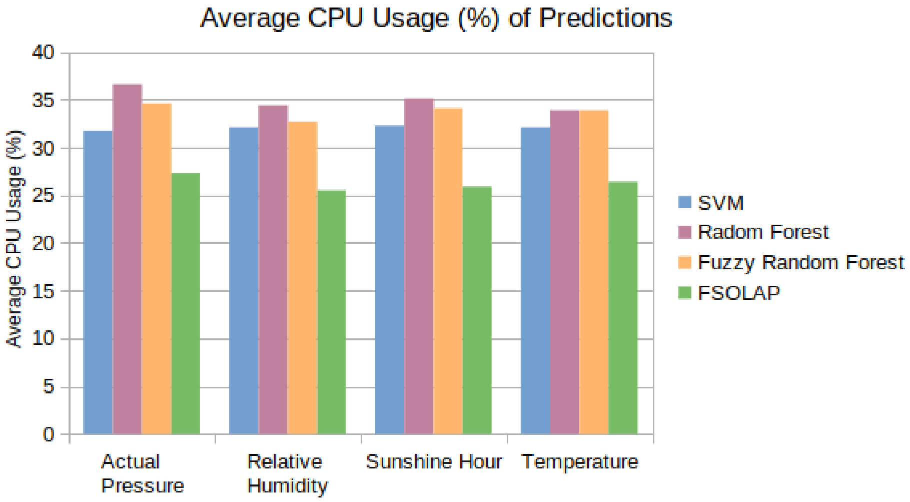


Figure 6.13: Average CPU usage of prediction models

Similar to the computational power requirement, the average memory usages of the models are also graphically shown during the prediction process in Fig.6.14. According to this chart, Random Forest builds a model that consumes the highest memory, and Fuzzy Random Forest also requires memory close to that of Random Forest. Both models require a high amount of memory because they build 1000 decision trees on memory. FSOLAP only keeps association rule set and fuzzy data map on memory, so it needs much less memory than these. Since the SVM model uses the measurement values on the memory and the line values for the classifying process, it uses the least amount of memory compared to all other models.

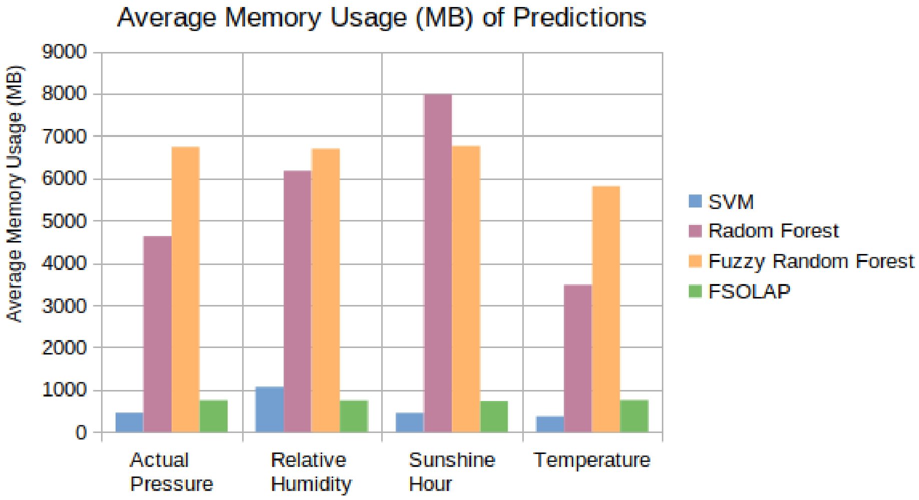


Figure 6.14: Average memory usage of prediction models

A comparison of the model building time of all models is made as part of the performance tests. The time spent between starting the process and building the model is represented for each model in Fig.6.15.

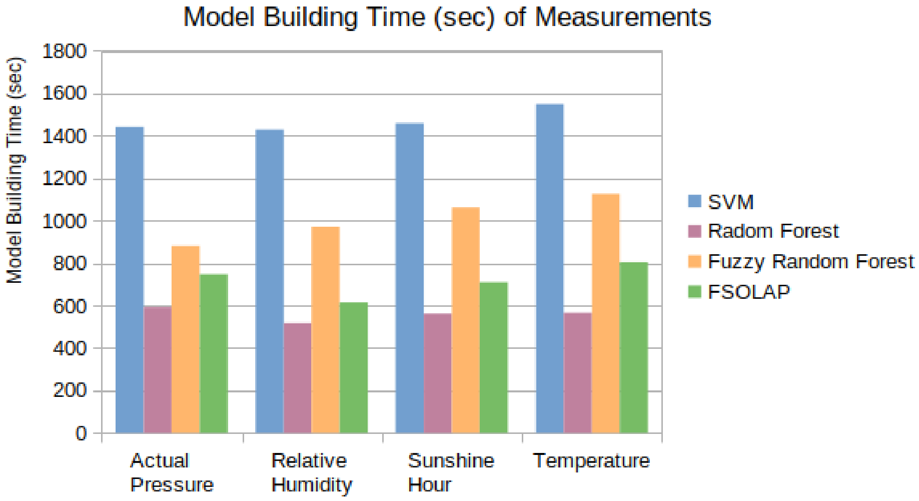


Figure 6.15: Average model building time of prediction models

SVM is the model with the longest model building time in the graph. SVM takes a long time to configure the best row detection during classification. FSOLAP wastes time in the process of selecting appropriate association rules and during the fuzzification and defuzzification process. The fuzzy random forest takes time in the fuzzification process and the creation of the decision tree. Random forest also takes a long time to build a decision tree, similar to its fuzzy type. But random forest takes less time than fuzzy random forest because no fuzzification process is required.

Finally, the comparison of the model prediction time is given in Fig.6.16. Again, like the model building time, SVM also requires more computation time for prediction (prediction time) as it depends on the number of support vectors and features. Predicting and voting on 1000 decision trees increase the prediction time in the Random Forest and Fuzzy Random Forest models. FSOLAP fuzzifies the input data and applies association rules for prediction during prediction time, and it takes the least execution time compared to other models.

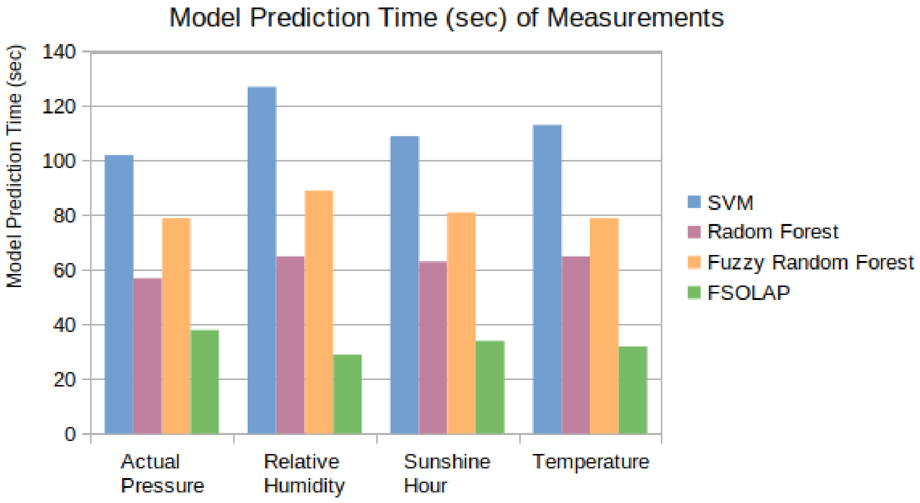


Figure 6.16: Average model prediction time of prediction models

When evaluating performance tests in general, SVM requires the highest execution time, although it requires low computing power and low memory. In addition, it performs poorly in terms of accuracy. Fuzzy random forest performs in more acceptable prediction time using high computational power, high memory usage, and has average accuracy performance. Random Forest performs faster than others in model building time with the highest computing power and memory, but offers high accuracy. FSOLAP does not require as much resource as others in terms of computing power and memory, and it also works at a reasonable model build time. In addition, it performs well in terms of prediction time. It also has a high level of performance in terms of accuracy. While considering all the parameters with the experimental results, FSOLAP is the preferable approach over other methods as it offers high accuracy and scalability with less resource usage.

## 6.2 Query Performance of the Framework

We measured the average CPU usage, memory usage, and execution time by running each query type in the fuzzy SOLAP-based framework and the PostgreSQL database. Here, average CPU usage is the average CPU usage rate measured during querying. Similarly, average memory usage is the average memory usage measured in megabytes (MB) during querying. The execution time is the average of the measurements obtained over several query runs.

First, we addressed some of the high-level factors that affect the query performance with regard to CPU usage, memory usage, and execution time. Data size directly affects the performance of the query because the query uses one or more tables with millions of rows or more. Joins are another factor affecting performance; if the query joins two tables, increasing the row count of the result set substantially, the query is likely to be slow. Aggregations also affect performance, as combining multiple rows to produce a result requires more computation than simply retrieving those rows.

In addition to obtaining this information, we also performed the roll-up function provided by SOLAP for aggregating with the UNION operator in relational database queries. In this case, aggregating N dimensions requires N such unions in an SQL query. Another essential issue to consider in terms of query performance is that of cross-tabulations. While SOLAP supports such operations naturally, SQL requires an even more complicated combination of unions and GROUP BY clauses for cross-tabulations. An N-dimensional cross-tabulation requires a $2^N$-way union of $2^N$ different GROUP BY operators to build the underlying representation. In most relational databases, this results in $2^N$ scans of the data and $2^N$ sorts or hashes.

The CPU usage for the queries was measured over several query runs, and the average CPU usage for all query types was calculated. The results are given in Table 6.6.

Table 6.6: Comparision of average CPU usages between FSOLAP and relational database SQL queries

| | **FSOLAP Query Ave. CPU Usage (%)** | **Relational Database SQL Query Ave. CPU Usage (%)** |
|---|---|---|
| Query1 | 29.2 | 33.7 |
| Query2 | 30.3 | 36.6 |
| Query3 | 30.1 | 31.3 |
| Query4 | 30.9 | Not Supported |

The average CPU usages of the FSOLAP-based query and the relational database query are compared in the column chart shown in Figure 6.17.

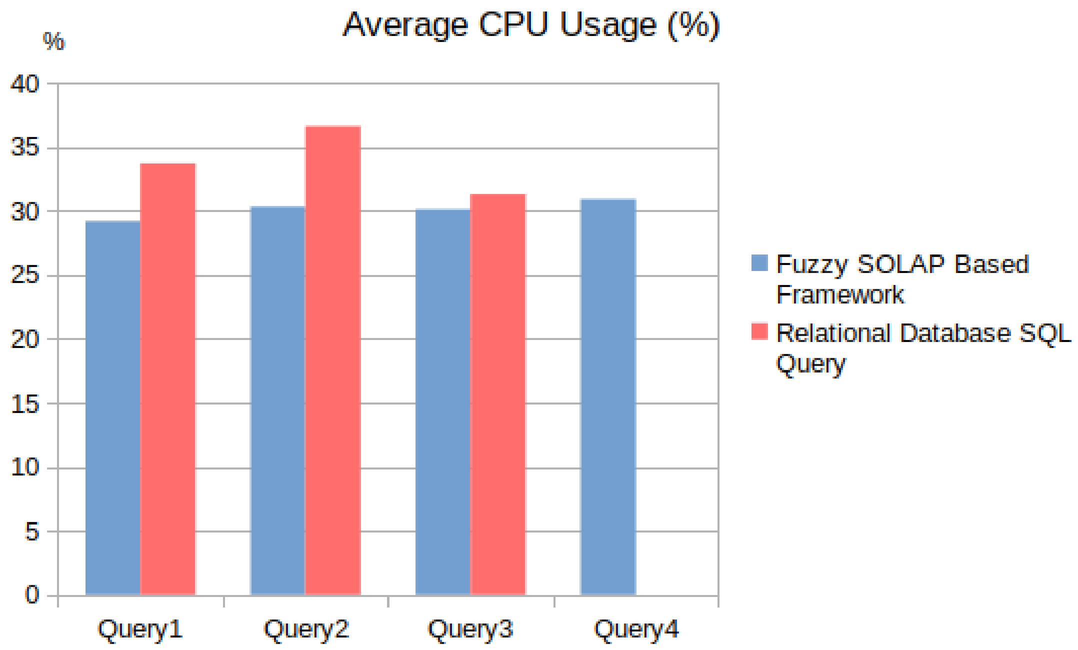


Figure 6.17: Average CPU usages of FSOLAP and relational database SQL queries.

Similar to the computational power requirement, the measurement results for the average memory usage are given in Table 6.7.

Table 6.7: Comparision of average memory usages between FSOLAP and relational database SQL queries

| | **FSOLAP Query Ave. Memory Usage (MB)** | **Relational Database SQL Query Ave. Memory Usage (MB)** |
|---|---|---|
| Query1 | 150 | 278 |
| Query2 | 228 | 330 |
| Query3 | 115 | 229 |
| Query4 | 217 | Not Supported |

The average memory usages of the queries are represented graphically in Figure 6.18. According to this chart, relational database queries consume more memory than FSOLAP-based queries.

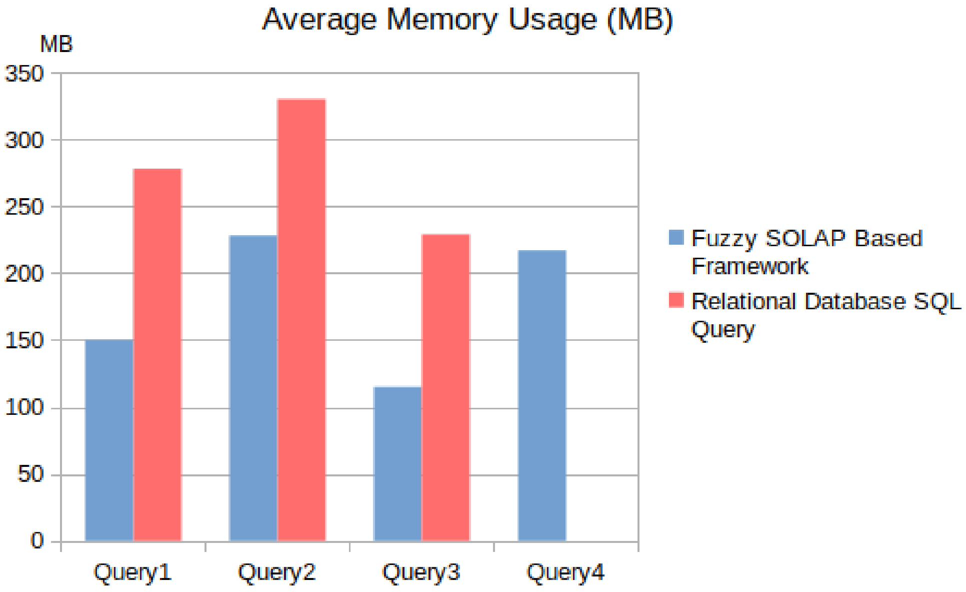


Figure 6.18: Average memory usage of FSOLAP and relational database SQL queries.

A comparison of the execution times of the queries was used as part of the performance testing, and the results are shown in Table 6.8.

Table 6.8: Comparison of average execution times between FSOLAP and relational database SQL queries

| | **FSOLAP Query Ave. Execution Time (ms)** | **Relational Database SQL Query Ave. Execution Time (ms)** |
|---|---|---|
| Query1 | 596,480 | 1,630,362 |
| Query2 | 257,054 | 643,642 |
| Query3 | 18,314 | 172,303 |
| Query4 | 183,717 | Not Supported |

We have shown the time spent between starting the query and finishing the query graphically for each query in Figure 6.19. The graph shows that relational database queries have a longer execution time.

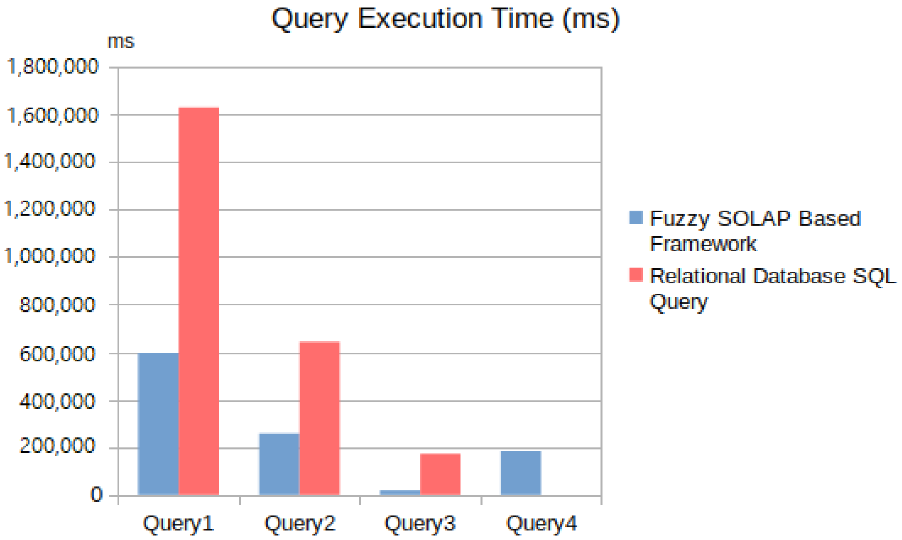


Figure 6.19: Execution times of FSOLAP and relational database SQL queries.

The implementation of Query 1 in the relational database requires the $having\, avg$ operation as an aggregation for all cities. This requires a great deal of CPU and memory resource usage. Along with these, it also causes a long query time. Query 2 requires $having\ avg$ as an aggregation along with a spatial search. A spatial data search uses index matches with the join operand in the query. This query requires more CPU and memory than other queries, but the query time is comparatively less than Query 1 since the query has a spatial restriction. Query 3, on the other hand, is better in terms of resource usage as it possesses additional time restrictions compared to Query 2, but it also takes less query time. The aggregation process in the queries involves the CPU usage, the union, and the join operands, affecting the memory usage. According to the query criteria, the amount of data in the query process determines the query time. When we evaluate the performance tests in general, we observe that FSOLAP-based query operations require fewer resources and less time than relational database queries. While we obtain adequate CPU and memory usage results, especially in queries containing spatial and temporal criteria, we obtain better results in terms of execution time. In addition, FSOLAP performs well in prediction-type queries, which are not supported for relational database queries.

Based on our experimental analysis and considering all the parameters mentioned, FSOLAP-based querying is preferred over relational database querying, as FSOLAP offers scalability with low resource usage.

### 6.3 Performance of the Aggregate Queries

In order to compare the FSOLAP-based aggregate query with the traditional SQL-based aggregate query, we construct sample queries for both query types. In Hierarchical Query, we have included the query in the MDX structure in the previous section to show the sunshine hour data measured in the Southeast Anatolia region by calculating hierarchically. The SQL query, which corresponds to obtaining the same result as this query, is built as follows. In this query, aggregation is performed with the GROUP BY operator, and hierarchical levels are assembled with the UNION operator.

```
 SELECT r.region name, t.year, avg(sunshine hour)
 FROM 2met data sunshine t, meteorological station3 s, tr city c,
     tr region r
 WHERE t.station id=s.id AND s.city id=c.gid AND c.region id=r.id
   AND r.region name= GUNEYDOGU ANADOLU REGION
 GROUP BY r.region name, t.year
UNION ALL
 SELECT c.name1, t.year, avg(sunshine hour)
 FROM 2met data sunshine t, meteorological station3 s, tr city c,
     tr region r
 WHERE t.station id=s.id AND s.city id=c.gid AND c.region id=r.id
   AND r.region name= GUNEYDOGU ANADOLU REGION
 GROUP BY c.name1, t.year
UNION ALL
 SELECT s.station name, t.year, avg(sunshine hour)
 FROM 2met data sunshine t, meteorological station3 s, tr city c,
     tr region r
 WHERE t.station id=s.id AND s.city id=c.gid AND c.region id=r.id
   AND r.region name= GUNEYDOGU ANADOLU REGION
 GROUP BY s.station name, t.year
```

In Conceptual Query, we retrieve the cities where the freezing occurs in the hinterland regions' open skies and low temperatures. While we can easily support complex queries in FSOLAP-based MDX queries, handling this complexity with traditional SQL-based queries is naturally challenging. So a linguistic term in the form of a hinterland region can be defined in FSOLAP; we can only meet this concept in SQL query as a region with specific boundaries. Although it does not fully replace the FSOLAP-based MDX query, a SQL-based query containing the desired results is constructed as follows.

```
SELECT c.name1, t.month,t.day, avg(t.temperature), avg(t2.cloudiness)
FROM 2met data temperature t, 2met data avg cloudiness t2,
    meteorological station3 s, tr city c, tr region r
WHERE t.station id=s.id AND s.city id=c.gid AND c.region id=r.id
 AND r.region name= IC ANADOLU REGION  AND t.station id=t2.station id
 AND t.year=t2.year AND t.month=t2.month AND t.day=t2.day
 AND t.temperature 4 AND t2.cloudiness 2 AND t.year=2015
GROUP BY c.name1, t.month, t.day order by t.month, t.day, c.name1
```

We measured the average CPU usage (ACU), average memory usage (AMU), and query execution time (QET) by running each query type in the fuzzy SOLAP-based framework and the PostgreSQL database. Here, average CPU usage is the average CPU usage rate measured during querying. Similarly, average memory usage is the average memory usage measured in megabytes (MB) during querying. The execution time is the average of the measurements obtained over several query runs.

The CPU usage for the queries was measured over several query runs, and the average CPU usage for all query types was calculated. The results are given in Table 6.9.

Table 6.9: Performance test results of the aggregation queries

| Query Type | Performance Metric | FSOLAP Based | SQL Based |
|---|---|---|---|
| Hierarchical Query | ACU (%) | 24.3 % | 31.2 % |
| | AMU (mb) | 2.1 mb | 3.8 mb |
| | QET (sec) | 12.6 sec | 36.9 sec |
| Conceptual Query | ACU (%) | 15.2 % | 22.3 % |
| | AMU (mb) | 1.6 mb | 2.3 mb |
| | QET (sec) | 8.3 sec | 16.6 sec |
| Predictive Query-1 | ACU (%) | 33.2 % | NA |
| | AMU (mb) | 2.8 mb | NA |
| | QET (sec) | 43.7 sec | NA |
| Predictive Query-2 | ACU (%) | 35.6 % | NA |
| | AMU (mb) | 2.9 mb | NA |
| | QET (sec) | 48.3 sec | NA |

Implementing a Hierarchical Query in the relational database requires the $group\ by\ avg$ operation to aggregate all stations and cities under the given region. Also, it needs to use the $union\ all$ operator to combine the result sets of region, city, and station $select$ statements. Using $group\ by\ avg$ and combining them requires a great deal of CPU and memory resource usage. Along with these, it also causes a long query time. Conceptual Query requires $group\ by\ avg$ as an aggregation along with a spatial search. A spatial data search uses index matches with the join operand in the query. This query requires less CPU and memory than the previous queries, and the query time is comparatively less than the Hierarchical Query since the query has a spatial restriction. On the other hand, Predictive Query-1 and Predictive Query-2 are only supported by FSOLAP and needs more resource usage as it possesses additional time for the prediction operation compared to FSOLAP-based Conceptual Query, so it also takes more query time. Here, the prediction part of the queries need more resources and time by using FIS to execute the fuzzy association rules for inference. Prediction Query-2 needs a little bit more CPU and time than Prediction Query-1. This situation is because Prediction Query-2 includes fuzzy operations on the spatial feature of the data to handle lowland conceptual terms. The aggregation process in the queries involves the CPU usage, the union, and the join operands, affecting the memory usage. According to the query criteria, the amount of data in the query process determines the query time. Figure 6.20 represents the ACU, AMU, and QET measurement results of Hierarchical Query, Conceptual Query, and Predictive Queries as a chart.

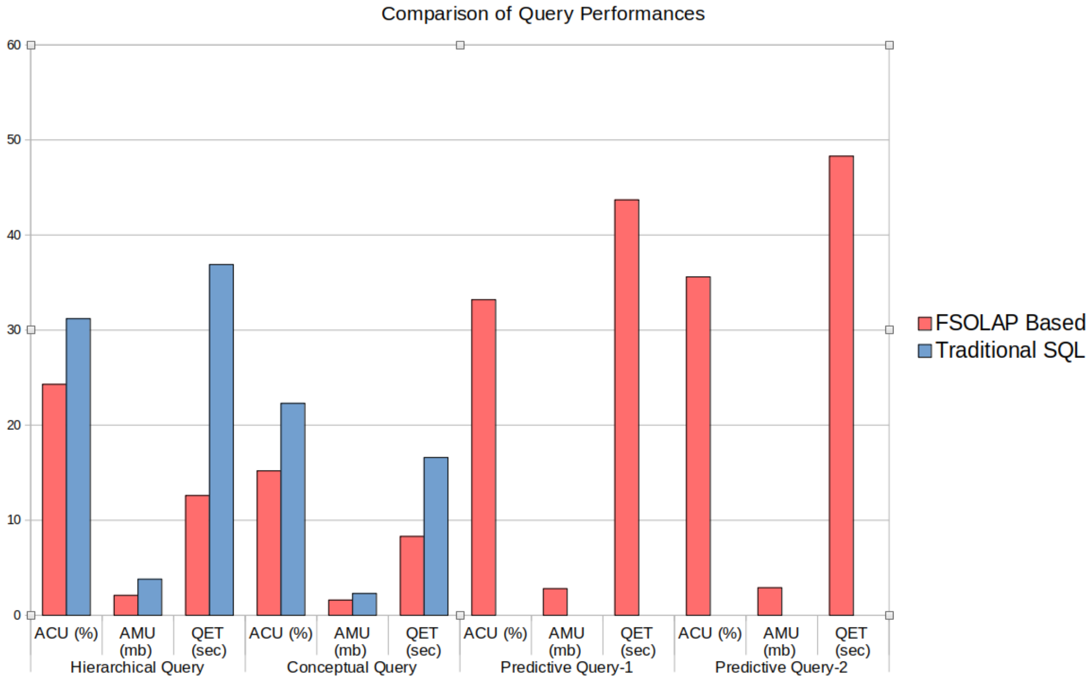
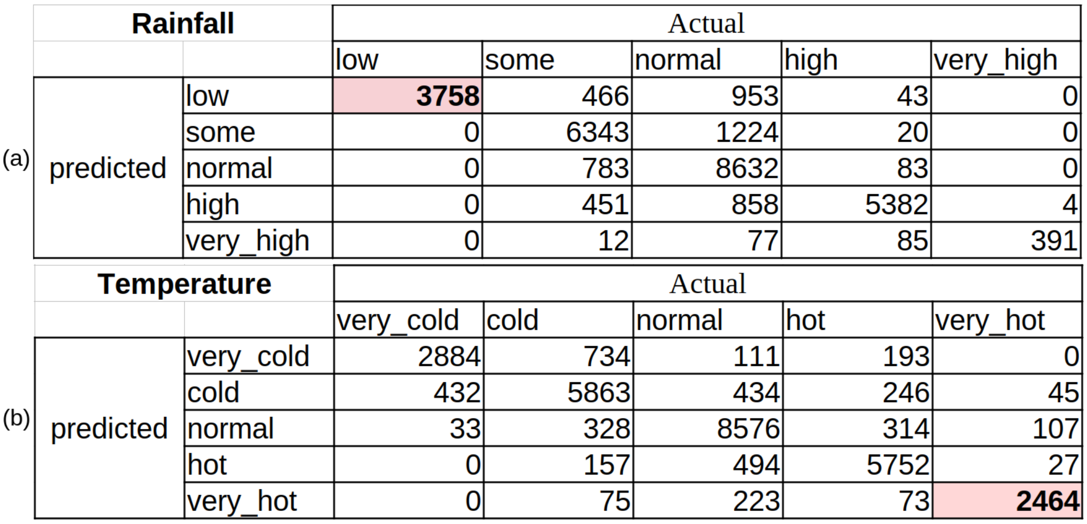


Figure 6.20: Comparison of Query Performances for FSOLAP-Based and Traditional SQL Queries

When we evaluate the performance tests in general, we observe that FSOLAP-based aggregate query operations require fewer resources and less time than relational database queries. While we obtain adequate CPU and memory usage results, especially in queries containing spatial and temporal criteria, we obtain better results in terms of execution time. In addition, FSOLAP performs well in prediction-type queries, which are not supported for relational database queries.

The confusion matrices of prediction results of the rainfall and temperature are prepared, as shown in Fig. 6.21. Here is a systematic comparison of the predicted values of the FSOLAP-based prediction processes with the actual values.



| | Rainfall | | Actual | | | |
|---|---|---|---|---|---|---|
| | | | low | some | normal | high | very_high |
| (a) | predicted | low | **3758** | 466 | 953 | 43 | 0 |
| | | some | 0 | 6343 | 1224 | 20 | 0 |
| | | normal | 0 | 783 | 8632 | 83 | 0 |
| | | high | 0 | 451 | 858 | 5382 | 4 |
| | | very_high | 0 | 12 | 77 | 85 | 391 |

| | Temperature | | Actual | | | | |
|---|---|---|---|---|---|---|---|
| | | | very_cold | cold | normal | hot | very_hot |
| (b) | predicted | very_cold | 2884 | 734 | 111 | 193 | 0 |
| | | cold | 432 | 5863 | 434 | 246 | 45 |
| | | normal | 33 | 328 | 8576 | 314 | 107 |
| | | hot | 0 | 157 | 494 | 5752 | 27 |
| | | very_hot | 0 | 75 | 223 | 73 | **2464** |

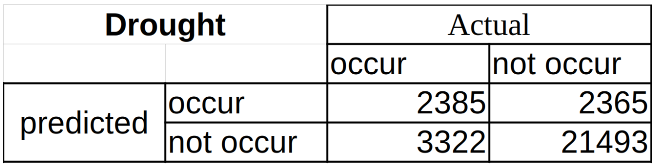
Figure 6.21: Confusion matrix of rainfall (a) and temperature (b)

Table 6.10 represents the statistical analysis for the predictive analytics, which are computed using confusion matrices and the given terminology definitions at the beginning of this section.

Table 6.10: Results of Statistical Analysis for Rainfall and Temperature

| | **Rainfall** | **Temperature** |
|---|---|---|
| accuracy | 0.8288 | 0.8637 |
| AUC | 0.8718 | 0.9069 |
| recall | 0.80 | 0.87 |
| precision | 0.79 | 0.87 |
| F1 | 0.82 | 0.88 |

According to Table 6.10, we predicted rainfall values with an accuracy of 82% and temperature values with an accuracy of 86%. Missing metrics for rainfall data in the sample dataset adversely impacted prediction accuracy. Using these two measurement types, we can simply determine the overall success of estimating cities that may experience drought as 82% based on the minimum accuracy value but this is not valid. Because we look at the "rainfall is low" and "temperature is very-high" conditions for drought prediction, these two conditions must be met in the same record. In this context, we created the confusion matrix in Figure 6.22 by defining "occur" for records that meet these two conditions and "not occur" for records that do not meet both simultaneously.



| **Drought** | | Actual | |
|---|---|---|---|
| | | occur | not occur |
| predicted | occur | 2385 | 2365 |
| | not occur | 3322 | 21493 |

Figure 6.22: Confusion matrix of drought

We represent the accuracy, recall, precision, and F1 values calculated on the data in the confusion matrix in Table 6.11.

Table 6.11: Results of Statistical Analysis for Drought

| | **Drought** |
|---|---|
| accuracy | 0.8076 |
| recall | 0.50 |
| precision | 0.41 |
| F1 | 0.45 |

The drought risk of the FSOLAP-based predictive aggregate query is calculated as 80%. This value is less than the accuracy of rainfall and temperature predictions because the resulting data that satisfies both conditions simultaneously is a subset of the resultant data that satisfies the conditions separately. For example, on 20 July 2015, we predicted rainfall as low and temperature as high for Van. While we predicted rainfall correctly for this result record, we expected the temperature wrong, so we could not accurately anticipate drought conditions. Therefore, the drought prediction accuracy falls below the rainfall and temperature prediction accuracy.

# CHAPTER 7

# DISCUSSION

This study composes the advantages of fuzzy and SOLAP concepts to develop a new framework based on fuzzy SOLAP (FSOLAP) and explains its inference capability. We have experienced the efficient and effective support of SOLAP on spatial and temporal hierarchical data in predictive analytics. In addition, it is shown that the fuzzy logic approach is appropriate for complex applications such as spatiotemporal with an example of a fuzzy query containing verbal language terms. Along with this, it is explained how fuzzy spatiotemporal prediction is performed with the inference capability that we have added to FSOLAP by the support of FIS. FSOLAP is shown to perform predictions accurately and efficiently using fewer resources, comparing it to well-known machine learning models such as SVM, RF, and FRF based on average CPU utilization, average memory usage, average model building time, and average prediction time. Performance results are close to each other in all models compared in terms of CPU usage. However, FSOLAP is slightly better than others. The reason for this is the tunings to generate an appropriate number of rules. Therefore, the absence of unnecessary and repetitive rules also prevents excessive processing load. Regarding memory usage, SVM and FSOLAP perform much better than the others. The main reason for this is the memory requirement since forest models operate with many tree combinations, while SVM processes use only data without any structure of holding data. FSOLAP, on the other hand, uses testing data, fuzzy membership classes, membership functions, and fuzzy association rules while making predictions, and all of these take up little memory space. Also, although FSOLAP lags behind the RF method in model building time, it performs better than the other models in prediction time. FSOLAP makes more accurate predictions with normally distributed data than with well-known machine learning techniques. We can therefore conclude that our model is reasonably accurate and scalable based on our experimental analysis on the meteorological dataset. We should also note that the performance of FSOLAP is related to the number of fired fuzzy association rules in prediction time. Since the fuzzy association rules selected to execute at the prediction time depend on the content of the data used for testing, this directly affects the performance.

In this study, we focused on SOLAP-based data mining and prediction, fuzzy inference and querying, fuzzy spatial and temporal queries, fuzzy data mining and querying, and fuzzy predictive analytics. We evaluated our framework for ease of use, scalability, and prediction performance. In the literature, [17] conducted a study combining fuzzy logic and spatial data mining to examine spatial correlations from imprecise spatial data and integrate them into their Geospatial Information Database (GIDB) systems. The authors examined directional or geometric relationships in soil types in the context of spatial correlation. Although [17] offer a new approach that performs inferencing using fuzzy logic with data mining and fuzzy spatial querying, their study does not support fuzzy temporal predictive queries. Also, their approach has no performance evaluation included in their article. Therefore, we cannot judge the scalability and prediction accuracy of their approach.

In another study, a decision-making prototype software based on multi-criteria analysis (MCA) was developed by [13]. This software is about solving complex decision-making problems and deals with hybrid analysis processes that handle complex multi-criteria situations on OLAP. The prototype software provides data mining and inference by combining OLAP and fuzzy logic and provides temporal and fuzzy queries. However, their proposals do not support fuzzy and predictive spatial queries and do not provide any performance tests in their study.

On the other hand, [18] presented a framework that supports SOLAP-based fuzzy data mining, inference, and fuzzy spatial querying, with their proposed system. They incorporated their structure with the fuzzy OLAP-based Intelligent Geographical Project (IGP), which provides decision support on a fuzzy spatial database. Although this system is a spatial data warehouse with fuzzy logic that provides fuzzy queries on OLAP, it contains shortcomings such as not supporting temporal and fuzzy predictive queries. Moreover, there is no easy-to-use interface for users, and no performance evaluation is done.

Working on the storage location assignment problem, [15] came up with a platform called Fuzzy Storage Assignment System (FSAS), which provides data mining with inference support with OLAP and fuzzy concepts. With the system they developed, they attempted to solve the storage location assignment problem by making more acceptable use of decision support data for the benefit of human knowledge. However, their proposed system does not support temporal and fuzzy predictive queries, and its performance has not been evaluated. The comparison between FSOLAP and related works according to their concepts and features is given in Table 7.1.

Table 7.1: Comparison of FSOLAP with Existing Studies

| Features and Concepts | GIDB [17] | OLAP MCA [13] | IGP [18] | FSAS [15] | FSOLAP |
|---|---|---|---|---|---|
| OLAP | | ✓ | ✓ | ✓ | ✓ |
| SOLAP | ✓ | | ✓ | | ✓ |
| Fuzziness | ✓ | ✓ | ✓ | ✓ | ✓ |
| Inference | ✓ | ✓ | ✓ | ✓ | ✓ |
| Data Mining | ✓ | ✓ | ✓ | ✓ | ✓ |
| Fuzzy Querying | ✓ | ✓ | ✓ | ✓ | ✓ |
| Temporal Querying | | ✓ | | | ✓ |
| Fuzzy Spatial Querying | ✓ | | ✓ | | ✓ |
| Fuzzy Predictive Querying | | | | | ✓ |
| Easy to Use | | ✓ | | ✓ | |
| Visualization | | ✓ | | ✓ | ✓ |
| Performance Evaluation | | | | | ✓ |

Considering the number of clusters, fuzzification, and fuzzy association rules of the data on the data marts are taken into account, it is challenging to manage online data. So, we do not have active learning (online learning) incorporated into our systems yet, that is, online data does not contribute to the existing knowledge of the framework.

When new data is given as input to the developed framework, the learning model needs to be updated considering this new data entered. Therefore, incorporating an active learning approach into our framework can be considered for future research work.

Since the motivation of our study is using fuzzy and spatial OLAP concepts together and making predictions over this structure, the studies carried out in this fuzzy spatial OLAP are considered related studies. We have included recent studies conducted to predict meteorological events such as drought [37, 38, 39], operational streamflow [40], rainfall [41], and reservoir operation under climate change [42]. We must mention that the recent related studies use statistical methods, ANN-based or ANFIS, to make predictions over one or two data types such as drought or precipitation. They do not propose any framework, as we do in this paper, to deal with fuzzy and multi-dimensional data types in the context of a fuzzy spatial OLAP system. More specifically, Mohamadi et al. [37] proposed a drought modeling using ANFIS with hybrid soft computing models. Additionally, Wu and Chau [41] studied with an ANN-based model to predict rainfall. Taormina and Chau [40] also researched with an ANN-based model to predict streamflow. Even though the ANN-based model is used in a wide variety of applications, including rule-based control systems, classification, and pattern matching, ANN has a drawback of computational complexity as it carries the curse of dimensionality. This weakness limits ANN to be used only with the applications having less number of inputs, whereas our datasets include several features.

SOLAP is suitable for the hierarchical form of spatiotemporal data, and fuzzy logic easily addresses the complex structure of spatial and temporal applications. Therefore, the FSOLAP framework brings them together and makes predictive analytics effective and efficient. However, the framework has some difficulties in terms of ease of use by naive users. Although the framework provides visual tools, a certain level of expertise is required to use these tools. As represented in the use case, data collection, ETL, and other prediction processes must be performed by the domain experts in the system. This situation makes it difficult for naïve users to use our framework, and therefore some possible improvements can be made in future work to ensure easy use of the framework and handle other complex applications such as air conditioning and maritime transport.

# CHAPTER 8

# CONCLUSIONS

In this study, a fuzzy SOLAP-based framework (FSOLAP) is proposed to conduct effective and efficient analyses, make inferences and support various queries on the large amount of daily growing data containing spatial and temporal features. While the natural hierarchical structure of spatiotemporal data is managed with SOLAP, the complexity of spatial-temporal applications is overcome with fuzzy logic. In addition, a fuzzy inference system (FIS) is also integrated into the framework to extend the inference capability required for predictive analytics. In the implementation part of this study; data is collected, cleaned, and structured with ETL operations, then SOLAP server construction and meta-data definitions are completed, MDX modifications are made to support fuzziness, and fuzzy classes are created, and finally, fuzzy association mining rules are generated.

Generated fuzzy association rules are pruned to make the framework work better. Then the remaining rules are weighted to improve their effects on the result. Subsequently, in the case study, FSOLAP was tested on real meteorological data. Similarly, we tested well-known machine learning techniques such as SVM, RF, and FRF on the same dataset and compared their results obtained with FSOLAP. This study claims that using fuzzy logic and SOLAP concepts for spatiotemporal applications would be efficient and effective. Thus, this assertion is confirmed both in the accuracy of the prediction and in the results of the performance tests. In addition, FSOLAP and some related studies are compared to determine if they provide specific features and concepts. FSOLAP is shown to have a more comprehensive feature set than similar studies. Improving the framework for ease of use for naïve users and allowing it to be used in other application areas, such as agriculture and maritime transport as future work.

While the analyzes and results provided in this study are reasonable, the FSOLAP framework still requires further research. It is observed that the proposed framework and random forest produce results close to each other during accuracy performance tests. More accurate results are obtained with Random Forest, especially if the data is unbalanced with outliers. It reduces overfitting in decision trees and helps to improve accuracy. However, despite handling outliers, it takes a lot of computational power and resources to build many trees to combine their outputs. It also takes a long time to predict because it combines many decision trees to determine the class. When the data size grows linearly, the performance of Random Forest decreases. On the other hand, the proposed framework requires reasonable computational power and appropriate resources when the data size is large. We may need to investigate outlier handling to minimize the overall error rate for better accuracy as a future contribution to the work of the proposed framework.

Comparing the forecast accuracy, model building time, prediction time, and memory requirements of the proposed framework with other products such as Numerical Weather Prediction (NWP) can be considered as future work.

In addition, this study improves the fuzzy querying capability to the FSOLAP framework. For this purpose, an extension is performed on the MDX query to support complex fuzzy queries. In this context, various types of spatial and temporal queries have become executable on SOLAP in MDX form with the FSOLAP framework. Furthermore, the model and methods of this proposal could be adapted or extended to different application environments, such as agriculture and habitat applications. Thus, a pre-warning system can be developed to predict the risk of seeing frost in a specific location and prevent the destruction of agricultural plants and/or the environment. We can also address the inadequacy of the framework in online learning within the scope of future studies. Considering the lack of datasets, the scope of our study, and the space limitations, we did not conduct further investigation on other applications. But they all can be considered for future studies.

# REFERENCES

[1] J. Han, "Towards on-line analytical mining in large databases," *ACM SIGMOD Record*, vol. 27, pp. 97—107, mar 1998.

[2] Y. P. Huang, L. J. Kao, and F. E. Sandnes, "Data mining and fuzzy inference based salinity and temperature variation prediction," in *2007 IEEE International Conference on Systems, Man and Cybernetics*, pp. 2074–2079, 2007.

[3] T. R. Sivaramakrishnan and S. Meganathan, "Association rule mining and classifier approach for quantitative spot rainfall prediction," *Journal of Theoretical and Applied Information Technology*, vol. 34, no. 2, pp. 173–177, 2011.

[4] M. Kaya and R. Alhajj, "Fuzzy olap association rules mining-based modular reinforcement learning approach for multiagent systems," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 2, pp. 326–338, 2005.

[5] J. G. Stell, "Part and complement: Fundamental concepts in spatial relations," *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 1–17, 2004.

[6] T. Cheng, M. Molenaar, and H. Lin, "Formalizing fuzzy objects from uncertain classification results," *International Journal of Geographical Information Science*, vol. 15, no. 1, pp. 27–42, 2001.

[7] P. Fisher, C. Arnot, R. Wadsworth, and J. Wellens, "Detecting change in vague interpretations of landscapes," *Ecological Informatics*, vol. 1, no. 2, pp. 163–178, 2006.

[8] B. Plewe, "The nature of uncertainty in historical geographic information," *Transactions in GIS*, vol. 6, no. 4, pp. 431–456, 2002.

[9] G. Bordogna, S. Chiesa, and D. Geneletti, "Linguistic modelling of imperfect spatial information as a basis for simplifying spatial analysis," *Information Sciences*, vol. 176, no. 4, pp. 366–389, 2006.

[10] K. Kianmehr, M. Kaya, A. M. ElSheikh, J. Jida, and R. Alhajj, "Fuzzy association rule mining framework and its application to effective fuzzy associative classification," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 6, pp. 477–495, 2011.

[11] S. Rivest, Y. Bédard, and M. Nadeau, "Solap: A new type of user interface to support spatio-temporal multidimensional data exploration and analysis," in *2003 Workshop ISPRS*, (Quebec, Canada), October 2003.

[12] K. Kumar, P. Radha Krishna, and S. Kumar De, "Fuzzy olap cube for qualitative analysis," in *Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing*, pp. 290–295, 2005.

[13] O. Boutkhoum and M. Hanine, "An integrated decision-making prototype based on olap systems and multicriteria analysis for complex decision-making problems," *Applied Informatics*, vol. 4, no. 1, 2017.

[14] C. Molina, B. Prados-Suárez, M. P. de Reyes, and C. Peña Yañez, "Improving the understandability of olap queries by semantic interpretations," in *Flexible Query Answering Systems* (H. L. Larsen, M. J. Martin-Bautista, M. A. Vila, T. Andreasen, and H. Christiansen, eds.), (Berlin, Heidelberg), pp. 176–185, Springer Berlin Heidelberg, 2013.

[15] C. Lam, S. Chung, C. Lee, G. Ho, and T. Yip, "Development of an olap based fuzzy logic system for supporting put away decision," *International Journal of Engineering Business Management*, vol. 1, pp. 1–13, 2009.

[16] R. Ďuračiová and J. Faixová Chalachanová, "Fuzzy spatio-temporal querying the postgresql/postgis database for multiple criteria decision making," in *Dynamics in GIscience* (I. Ivan, J. Horák, and T. Inspektor, eds.), (Cham), pp. 81–97, Springer International Publishing, 2018.

[17] R. Ladner, F. E. Petry, and M. A. Cobb, "Fuzzy set approaches to spatial data mining of association rules," *Transactions in GIS*, vol. 7, no. 1, pp. 123–138, 2003.

[18] P. David, S. Maria, and P. Ivo, "Fuzzy spatial data warehouse: A multidimensional model," *Decision Support Systems Advances in*, pp. 57–66, 2010.

[19] K. Zheng, X. Zhou, P. C. Fung, and K. Xie, “Spatial query processing for fuzzy objects,” *The VLDB Journal*, vol. 21, no. 5, pp. 729–751, 2012.

[20] N. Nurain, M. E. Ali, T. Hashem, and E. Tanin, “Group nearest neighbor queries for fuzzy geo-spatial objects,” in *Second International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data*, GeoRich’15, (New York, NY, USA), pp. 25–30, Association for Computing Machinery, 2015.

[21] A. Sözer, A. Yazıcı, H. Oğuztüzün, and F. E. Petry, *Querying Fuzzy Spatiotemporal Databases: Implementation Issues*, pp. 97–116. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[22] R. B. Messaoud, O. Boussaid, and S. Rabaseda, “Mining association rules in olap cubes,” in *2006 Innovations in Information Technology*, pp. 1–5, 12 2006.

[23] P. Cingolani and J. Alcalá-Fdez, “jfuzzylogic: a robust and flexible fuzzy-logic inference system language implementation,” in *2012 IEEE International Conference on Fuzzy Systems*, pp. 1–8, 2012.

[24] H. K. Soni, S. Sharma, and M. Jain, “Frequent pattern generation algorithms for association rule mining : Strength and challenges,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 3744–3747, 2016.

[25] C. González, L. Tineo, and A. Urrutia, “Fuzzy olap: A formal definition,” in *Advances in Computational Intelligence* (W. Yu and E. N. Sanchez, eds.), (Berlin, Heidelberg), pp. 189–198, Springer Berlin Heidelberg, 2009.

[26] S. Winter, “Topological relations between discrete regions,” in *Advances in Spatial Databases* (M. J. Egenhofer and J. R. Herring, eds.), (Berlin, Heidelberg), pp. 310–327, Springer Berlin Heidelberg, 1995.

[27] T. Takahashi, N. Shima, and F. Kishino, “An image retrieval method using inquiries on spatial relationships,” *Journal of Information Processing*, vol. 15, pp. 441—-449, jan 1992.

[28] M. A. Cobb and F. E. Petry, “Modeling spatial relationships within a fuzzy framework,” *Journal of the American Society for Information Science*, vol. 49, no. 3, pp. 253–266, 1998.

[29] H. Yang, M. Cobb, and K. Shaw, “A clips-based implementation for querying binary spatial relationships,” in *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, vol. 4, (Vancouver, Canada), pp. 2388–2393, 2001.

[30] A. Laurent, “Querying fuzzy multidimensional databases: Unary operators and their properties,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 11, pp. 31–45, sep 2003.

[31] T. Takahashi, N. Shima, and F. Kishino, “An image retrieval method using inquiries on spatial relationships,” *Journal of Information Processing*, vol. 15, pp. 441—-449, jan 1992.

[32] S. Winter, “Topological relations between discrete regions,” in *Advances in Spatial Databases* (M. J. Egenhofer and J. R. Herring, eds.), (Berlin, Heidelberg), pp. 310–327, Springer Berlin Heidelberg, 1995.

[33] H. Yang, M. Cobb, and K. Shaw, “A clips-based implementation for querying binary spatial relationships,” in *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, vol. 4, (Vancouver, Canada), pp. 2388–2393, 2001.

[34] M. A. Cobb and F. E. Petry, “Modeling spatial relationships within a fuzzy framework,” *Journal of the American Society for Information Science*, vol. 49, no. 3, pp. 253–266, 1998.

[35] S. Keskin and A. Yazıcı, “Modelling and designing spatial and temporal big data for analytics,” in *Computer and Information Sciences* (T. Czachórski, E. Gelenbe, K. Grochla, and R. Lent, eds.), (Cham), pp. 104–112, Springer International Publishing, 2018.

[36] R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” *ACM SIGMOD Record*, vol. 22, pp. 207–216, June 1993.

[37] S. Mohamadi, S. S. Sammen, F. Panahi, M. Ehteram, O. Kisi, A. Mosavi, A. N. Ahmed, A. El-Shafie, and N. Al-Ansari, “Zoning map for drought prediction

using integrated machine learning models with a nomadic people optimization algorithm," *Natural Hazards*, vol. 104, no. 1, p. 537–579, 2020.

[38] S. Shamshirband, S. Hashemi, H. Salimi, S. Samadianfard, E. Asadi, S. Shadkani, K. Kargar, A. Mosavi, N. Nabipour, K.-W. Chau, and et al., "Predicting standardized streamflow index for hydrological drought using machine learning models," *Engineering Applications of Computational Fluid Mechanics*, vol. 14, no. 1, p. 339–350, 2020.

[39] C. Zhao, Y. Huang, Z. Li, and M. Chen, "Drought monitoring of southwestern china using insufficient grace data for the long-term mean reference frame under global change," *Journal of Climate*, vol. 31, no. 17, p. 6897–6911, 2018.

[40] R. Taormina and K.-W. Chau, "Ann-based interval forecasting of streamflow discharges using the lube method and mofips," *Engineering Applications of Artificial Intelligence*, vol. 45, p. 429–440, 2015.

[41] C. Wu and K. Chau, "Prediction of rainfall time series using modular soft computing methods," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 3, p. 997–1007, 2013.

[42] M. Ehteram, S. F. Mousavi, H. Karami, S. Farzin, V. P. Singh, K.-w. Chau, and A. El-Shafie, "Reservoir operation based on evolutionary algorithms and multi-criteria decision-making under climate change and uncertainty," *Journal of Hydroinformatics*, vol. 20, no. 2, p. 332–355, 2018.

[43] S. Kohail and A. El-Halees, "Implementation of data mining techniques for meteorological data analysis," *International Journal of Information and Communication Technology Research (JICT)*, vol. 1, no. 3, pp. 96–100, 2011.

[44] S. Keskin, A. Yazıcı, and H. Oguztüzün, "Implementation of x-tree with 3d spatial index and fuzzy secondary index," in *Flexible Query Answering Systems*, vol. 7022 of *Lecture Notes in Computer Science*, pp. 72–83, Springer, 2011.

[45] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The r*-tree: An efficient and robust access method for points and rectangles," *ACM SIGMOD Record*, vol. 19, no. 2, p. 322–331, 1990.

[46] S. Berchtold, D. A. Keim, and H.-P. Kriegel, “The x-tree: An index structure for high-dimensional data,” in *Proceedings of the 22th International Conference on Very Large Data Bases*, VLDB ’96, (San Francisco, CA, USA), p. 28–39, Morgan Kaufmann Publishers Inc., 1996.

[47] S. Keskin and A. Yazıcı, “Fsolap: A fuzzy logic-based spatial olap framework for effective predictive analytics,” *Expert Systems with Applications*, vol. 213, p. 118961, 2023.

[48] PostGIS, “PostGIS:Spatial and Geographic Objects for PostgreSQL,” December 2021. Retrieved from `https://postgis.net`. Accessed December 12, 2021.

[49] M. Schneider, “A design of topological predicates for complex crisp and fuzzy regions,” in *Conceptual Modeling - ER 2001* (H. S.Kunii, S. Jajodia, and A. Sølvberg, eds.), (Berlin, Heidelberg), pp. 103–116, Springer Berlin Heidelberg, 2001.

[50] S. Keskin and A. Yazıcı, “Management of complex and fuzzy queries using a fuzzy solap-based framework,” in *Flexible Query Answering Systems: 14th International Conference, FQAS 2021, Bratislava, Slovakia, September 19–24, 2021, Proceedings*, (Berlin, Heidelberg), p. 109–126, Springer-Verlag, 2021.

[51] X. Tang, Y. Fang, and W. Kainz, “Fuzzy topological relations between fuzzy spatial objects,” in *Proceedings of the Third International Conference on Fuzzy Systems and Knowledge Discovery*, FSKD’06, (Berlin, Heidelberg), pp. 324––333, Springer-Verlag, 2006.

[52] F. B. Zhan and H. Lin, “Overlay of two simple polygons with indeterminate boundaries,” *Transactions in GIS*, vol. 7, no. 1, pp. 67–81, 2003.

[53] D. Pelleg and A. Moore, “X-means: Extending k-means with efficient estimation of the number of clusters,” in *In Proceedings of the 17th International Conference on Machine Learning*, (San Francisco, USA), pp. 727–734, Morgan Kaufmann, 2000.

[54] J. C. Bezdek, “A convergence theorem for the fuzzy isodata clustering al-

gorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no. 1, pp. 1–8, 1980.

[55] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53–87, 2004.

[56] Pentaho, "Pentaho Mondrian Documentation," December 2021. Retrieved from `https://mondrian.pentaho.com`. Accessed December 12, 2021.

[57] Spatialytics, "Spatialytics - geovisualization tool for spatial data," January 2021. Retrieved from `http://www.spatialytics.org`. Accessed January 14, 2021.

[58] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, (San Francisco, CA, USA), pp. 487—-499, Morgan Kaufmann Publishers Inc., 1994.

[59] B. Goethals, J. Muhonen, and H. Toivonen, "Mining non-derivable association rules," 04 2005.

[60] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo, "Finding interesting rules from large sets of discovered association rules," in *Proceedings of the Third International Conference on Information and Knowledge Management*, CIKM '94, (New York, USA), pp. 401—-407, Association for Computing Machinery, 1994.

[61] B. Goethals and J. Van den Bussche, "On supporting interactive association rule mining," in *Data Warehousing and Knowledge Discovery* (Y. Kambayashi, M. Mohania, and A. M. Tjoa, eds.), (Berlin, Heidelberg), pp. 307–316, Springer Berlin Heidelberg, 2000.

[62] R. T. Ng, L. V. Lakshmanan, J. Han, and A. Pang, "Exploratory mining and pruning optimizations of constrained associations rules," *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data - SIGMOD'98*, pp. 13–24, 1998.

[63] S. L. Salzberg, *C4.5: Programs for Machine Learning*, vol. 16. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1994.

[64] Ochin, S. Kumar, and N. Joshi, "Rule power factor: A new interest measure in associative classification," *Procedia Computer Science*, vol. 93, pp. 12–18, 2016.

[65] S. Keskin and A. Yazıcı, "Modeling and querying fuzzy solap-based framework," *ISPRS International Journal of Geo-Information*, vol. 11, no. 3, 2022.

[66] GeoMondrian, "GeoMondrian SOLAP Server," January 2021. Retrieved from `http://www.spatialytics.org`. Accessed January 14, 2021.

[67] D. Marutho, S. Hendra Handaka, E. Wijaya, and Muljono, "The determination of cluster number at k-mean using elbow method and purity evaluation on headline news," in *2018 International Seminar on Application for Technology of Information and Communication*, pp. 533–538, 2018.

[68] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[69] F. P. Pach, A. Gyenesei, S. Németh, P. Árva, and J. Abonyi, "Fuzzy association rule mining for the analysis of historical process data," *Acta Agraria Kaposváriensis*, vol. 10, no. 3, pp. 89–107, 2006.

[70] G. Spofford, S. Harinath, C. Webb, D. Huang, and F. Civardi, *MDX-Solutions, 2nd Edition*. Wiley, 2006.

[71] B. Kelkar, "Exploiting symbiosis between data mining and olap for business insights," *DM Direct Newsletter*, pp. 1270–1281, 2001.

[72] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Elsevier, 2006.

[73] A. Laurent, "Generating fuzzy summaries from fuzzy multidimensional databases," *Advances in Intelligent Data Analysis*, pp. 24–33, 2001.

[74] J. Han, "Olap mining: An integration of olap with data mining," *Data Mining and Reverse Engineering*, pp. 3–20, 1998.

[75] X. Zhou, D. Truffet, and J. Han, “Efficient polygon amalgamation methods for spatial olap and spatial data mining,” *Advances in Spatial Databases*, pp. 167–187, 1999.

[76] S. Prasher and X. Zhou, “Multiresolution amalgamation: Dynamic spatial data cube generation,” in *In Proceedings of Fifteenth Australian Database Conference*, pp. 103–111, 2004.

[77] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, “Efficient olap operations in spatial data warehouses,” *Advances in Spatial and Temporal Databases*, pp. 443–459, 2001.

[78] F. E. Petry and R. R. Yager, “Interval-valued fuzzy sets aggregation and evaluation approaches,” *Applied Soft Computing*, vol. 124, p. 108887, 2022.

[79] J. Kacprzyk, A. Wilbik, and S. Zadrożny, “Linguistic summarization of time series using a fuzzy quantifier driven aggregation,” *Fuzzy Sets and Systems*, vol. 159, no. 12, p. 1485–1499, 2008.

[80] J. Kacprzyk, A. Wilbik, and Z. Sławomir, “An approach to the linguistic summarization of time series using a fuzzy quantifier driven aggregation,” *International Journal of Intelligent Systems*, 2010.

[81] K. Nowacka, S. Zadrozny, and J. Kacprzyk, “An experimental comparison of various aggregation operators in a fuzzy information retrieval model,” *NAFIPS 2008 - 2008 Annual Meeting of the North American Fuzzy Information Processing Society*, 2008.

[82] R. R. Yager, “Families of owa operators,” *Fuzzy Sets and Systems*, vol. 59, no. 2, p. 125–148, 1993.

[83] R. R. Yager, “A new approach to the summarization of data,” *Information Sciences*, vol. 28, no. 1, p. 69–86, 1982.

[84] A. Laurent, “A new approach for the generation of fuzzy summaries based on fuzzy multidimensional databases1,” *Intelligent Data Analysis*, vol. 7, no. 2, p. 155–177, 2003.

[85] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 3, pp. 261–277, 2001.

[86] P. Bonissone, J. M. Cadenas, M. C. Garrido, and R. A. Díaz-Valladares, “A fuzzy random forest,” *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 729–747, 2010.

# CURRICULUM VITAE

**Surname, Name:** Keskin, Sinan

## EDUCATION

| Degree | Institution | Year of Graduation |
|---|---|---|
| M.S. | METU Computer Engineering | 2010 |
| B.S. | Hacettepe University Computer Engineering | 2006 |

## PUBLICATIONS

1. Keskin, S., and Yazıcı, A., "FSOLAP: A Fuzzy Logic-Based Spatial OLAP Framework for Effective Predictive Analytics." Expert Systems with Applications, vol. 213, pp. 118961, 2023.

2. Keskin, S., and Yazıcı, A., "Modeling and Querying Fuzzy SOLAP-Based Framework." ISPRS International Journal of Geo-Information, vol. 11, no. 3, pp. 191, 2022.

3. Keskin, S., and Yazıcı A., "Management of Complex and Fuzzy Queries Using a Fuzzy SOLAP-Based Framework." Flexible Query Answering Systems, LNAI 12871, pp. 109–126., 2021.

4. Keskin, S., and Yazıcı, A. "Modelling and Designing Spatial and Temporal Big Data for Analytics." Communications in Computer and Information Science, pp. 104–112, 2018.

5. Keskin, S., Yazıcı, A., and Oğuztüzün, H. "Implementation of X-tree with 3D spatial index and Fuzzy Secondary Index." Flexible Query Answering Systems, pp. 72–83, 2011.