T.C.

MARMARA UNIVERSITY

INSTITUTE FOR GRADUATE STUDIES IN

PURE AND APPLIED SCIENCES

A DATA-MINING BASED FRAUD DETECTION SYSTEM FOR HEALTH INSURANCE COMPANIES

Cüneyt AŞUK

THESIS

FOR THE DEGREE OF MASTER OF SCIENCE

IN

COMPUTER ENGINEERING

SUPERVISOR

Assoc. Prof. Dr. Melih KIRLIDOĞ

İSTANBUL 2010

T.C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES

A DATA-MINING BASED FRAUD DETECTION SYSTEM FOR HEALTH INSURANCE COMPANIES

Cüneyt AŞUK
(141100320060098)

THESIS

FOR THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SUPERVISOR
Assoc. Prof. Dr. Melih KIRLIDOĞ

İSTANBUL 2010

ACKNOWLEDGEMENT

It is my great honor and pleasure to gratefully thank to my “Master of Science Thesis” advisor, Assoc. Prof. Dr. Melih KIRLIDOĞ, for his tremendous support and help at every step of the study; from beginning to the end. I would also like to express my sincere appreciation to my advisor, Dr. KIRLIDOĞ, for his wise guidance and patience which made this dissertation possible. Dr. Melih KIRLIDOĞ provided the key technical insights, critically read through several drafts of this work and, made himself readily accessible.

I would like to express my sincere appreciation to my employers, general manager Yalçın TARKAN, software director Mutlu UĞURSAL for their technical supports. Their apprehensions, suggestions and recommendations made everything better and easier.

I would like to thank to my family for their endless support during my studies. They supported me for post-graduate education and I overcame all difficulties with the help of their trust and encouragement.

I wish to express gratitude to all academic and non-academic members of Marmara University Computer Engineering Department. It is a great honor to have chance to study with highly qualified members of this department.

January, 2010 **Cüneyt AŞUK**

CONTENTS

SAYFA

ACKNOWLEDGEMENT…………………………………………...i

CONTENTS……………………………….………….........ii

ÖZET……………………………………….......................iv

ABSTRACT………………………………….……….…........vi

SYMBOLS......…………………………..……..…........viii

ABBREVIATIONS…………………………….……........ix

FIGURES.........…………….....………………………......x

TABLES ………………………...…………………….........xiii

CHAPTER I. INTRODUCTION AND AIM……………….......1

I.1. INTRODUCTION……………………….......................................1

I.2. OBJECTIVE OF THE THESIS……………………….……..….......3

I.3. STRUCTURE OF THE THESIS……………………….……..……..4

CHAPTER II. GENERAL BACKGROUND…………...………….....5

II.1 KNOWLEDGE DISCOVERY IN DATABASE..........…………………..5

II.1.1 Preprocessing...7

II.1.2 Data Mining...10

II.1.2.1 Basic Data Mining Tasks..12

II.1.2.1.1 Classification...12

II.1.2.1.2 Regression..13

II.1.2.1.3 Time Series Analysis..14

II.1.2.1.4 Prediction...14

II.1.2.1.5 Clustering..14

II.1.2.1.6 Summarization..16

II.1.2.1.7 Association Rules..16

II.1.2.1.8 Sequence Discovery..16

II.1.2.2 Data Mining Issues..16

II.1.2.2.1 Anomaly Detection...17

II.1.3 Pattern Evaluation and Knowledge Presentation................................22

II.2 FRAUD DETECTION..........…………………...………………………..23

CHAPTER III. THE STUDY ………………….....……………….....35

III.1 PROBLEM DEFINITION....…………………...……………………........35

III.2 ORACLE DATABASE 11G..………………...……………………......36

III.3 ORACLE DATA MINING....………………...……………………......37

III.4 DATA PREPROCESSING.......……………...…………………….....38

III.4.1 Extract Transform Load..38

III.4.1.1 Extraction 40

III.4.1.2 Transform 42

III.4.1.3 Load...43

III.5 SUPPORT VECTOR MACHINE.......………...……………………....45

III.6 DATABASE...........................…………….…...………………………46

III.6.1 Database Overview...46

III.6.2 Database Structure...46

III.7 DATA MINING APPLICATION..............................………………..56

III.7.1 Fraud Type 1...61

III.7.2 Fraud Type 2...76

III.7.3 Fraud Type 3...83

CHAPTER IV. RESULTS AND DISCUSSIONS………………......97

CHAPTER V. CONCLUDING REMARKS

AND RECOMMENDATION..................................110

REFERENCES……………………...…………………………..…114

CURRICULUM VITAE…………………...……………………..119

ÖZET

SAĞLIK SİGORTASI ŞİRKETLERİ İÇİN VERİ MADENCİLİĞİ TABANLI SUİSTİMAL TESPİT SİSTEMİ

Sigortacılık tamamen risk ve olasılık üzerine çalışan bir sistemdir. Sigorta şirketleri kaza, hastalık ve buna benzer durumlar yüzünden insanların uğrayacağı maddi kayıpların riskini üstlendiklerini kabul ederler. Bu maddi kayıpların riskini üstlenmelerine karşılık müşterilerinden aylık belirli bir ücret alırlar. Bu ücrete sigorta primi denir. Sağlık sigortalıcılığı da aynı şekilde çalışır. Sigorta şirketi her ay sigorta primi ödeme karşılığında böbrek nakli, kanser tedavisi gibi sağlık sorunları sebebiyle oluşacak finansal riskleri üstlendiğini kabul eder.

Sağlık sektörünün en büyük problemi suistimallerdir. Sağlık suistimalleri, kişilerin veya kurumların haksız kazanç sağlamak adına kasıtlı olarak yaptıkları hile , yanlış beyan ve benzeri sahtekarlıklardır. Sigorta sistemine müşteri olarak giren insanların sayısının çok artması ile tıbbi suistimallerin sağlık sigorta sektörüne verdiği zarar gittikçe büyümektedir. 2007 yılında Amerika'da 4 milyar hasar sağlık sigorta şirketlerine gelmiş ve 2.26 trilyon dolar sağlık sektöründe harcanmıştır. Bu hasarların bir kısmının hileli olduğu tartışılmaz bir gerçektir. Bu hileli hasarlar tüm hasarlar göz önünde tutulduğunda az bir orana sahip olsalar da büyük bir maliyete sahiptirler. Sonuç olarak sağlık sigortacılığı sektörü suistimal tespit sistemine ihtiyaç duymaktadır.

Bu araştırmada sağlık sigorta şirketleri için veri madenciliği tabanlı suistimal tespit sistemi incelenmektedir. Bu sistem için öncelikle bir veri tabanı geliştirilmiştir. Sağlık sigorta sisteminden tüm kayıtlar bu veri tabanına aktarılmıştır. İncelemek üzere sağlık sigortacılığında üç suistimal tipi belirlenmiştir. Bunlar sigorta şirketinin ödemesi gereken miktardan daha fazla ödeme yapılan hasarlar, dört gün öncesine kadar doktor muayenesi ya da ameliyatı olmayan ilaç alımları ve ortalama vaka fiyatı, tüm sağlık merkezlerinin ortalamasından fazla olan ya da sigortalılara elden ödenen hasarların ortalamasının tüm sağlık merkezlerince sigortalılara elden ödenen hasarların ortalamasından fazla olan sağlık merkezleridir. Son suistimal tipinde sağlık merkezleri hastane ve eczane diye ayrılıp, her biri için ayrı model geliştirilmiştir. Daha sonra tek sınıflı destek vektör makineleri tabanlı suistimal tespit modelleri farklı kernel fonksiyonları ile şüpheli kayıt içermeyen veriler üzerinde oluşturulmuştur. Kullanılan kernel fonksiyonları da linear ve Gaussian kernel fonksiyonlarıdır. Bu modeler üretildikten sonra tamamı veri tabanında bulunan tüm kayıtlar üzerinde çalıştırılmıştır. Bu modellerin oluşturulmasında ve veri tabanının tamamı üzerinde çalıştırılmasında Oracle şirketinin veri madenciliği için geliştirdiği araç kullanılmıştır. Linear kernel fonksiyon tabanlı modellerin birinci ve ikinci suistimal tipleri için , Gaussian kernel fonksiyon tabanlı modelin de üçüncü suistimal tipi için uygun olduğu görülmüştür. Deneysel sonuçlar önerilen sistemin şüpheli kayıtlar üzerinde uygulandığında etkili sonuçlar verdiğini göstermektedir.

Ocak, 2010 **Cüneyt AŞUK**

ABSTRACT

A DATA MINING BASED FRAUD DETECTION SYSTEM FOR HEALTH INSURANCE COMPANIES

Insurance is all about risk and probabilities. With insurance, insurance company agrees to assume risk of incurring serious financial loss due to an accident, illness, bad luck, or other specified means. In exchange for this service, the insurance company charges their insured monthly premiums to help offset the cost of protecting him against this potential financial loss. Health insurance works the same way. Every month, insured pays a premium. In exchange for this premium, insurance company agrees to assume the financial risk if, for example, insured requires a kidney transplant or takes medical treatment.

The biggest problem of the health insurance sector is medical fraud and abuse. Health care fraud is an intentional deception or misrepresentation that the individual or entity makes knowing that the misrepresentation could result in some unauthorized benefit to the individual, or the entity or to some other party. With the increasing number of insured people, medical fraud began to compose great damage to the sector. In 2007, $2.26 trillion was spent on health care and more than 4 billion health insurance claims were processed in the United States. It is an undisputed reality that some of these health insurance claims are fraudulent. Although they constitute only a small fraction, those fraudulent claims carry a very high price tag. Because of this health insurance sector needs a fraud detection system.

In this research a fraud detection system based on data mining for health insurance companies is presented. First of all a new database was created. Huge data from health insurance system were then imported to the database. Three health insurance fraud types were determined to investigation. These are claims whose paid amount is greater than its invoice amount that insurance company will pay, transactions which is medicine taking without doctor inspection or surgical operation in four days, and health centers whose average incident cost bigger than the average incident cost of all health centers and rate of the payment to directly insured is bigger than the average payment to directly insured rate of all health centers. In last fraud type, health centers were separated into two types as hospital and pharmacy, and models were created for both of them. Fraud detection models based on one class support vector machine algorithm with different kernel function were created on non suspicious data. Kernel functions that were used to create these models are Linear and Gaussian kernel functions. After creating the models, they are applied to the whole data. Oracle data mining tool is used to create and apply model. It was seen that linear kernel function based models are suitable for first and second fraud types and Gaussian kernel function based model is suitable for third fraud type. Experimental results show that the proposed system gave effective results when applied to suspicious data sets.

January, 2010 **Cüneyt AŞUK**

SYMBOLS

G	**:** Temporal Graph
s	**:** Source Vertex
v	**:** Vertex
C(FP)	**:** Cost of False Positive
C(FN)	**:** Cost of False Negative
p(N)	**:** Probability of Negative
p(P)	**:** Probability of Positive
(TP)	**:** True Positive Ratio
(FP)	**:** False Positive Ratio

ABBREVIATIONS

AI	: Artificial of Intelligence
ANSI	: American National Standards Institute
ETL	: Extract Transform Load
ERP	: Enterprise Resource Planning
GAO	: The United States Goverment Accountability Office
HCFA	: Health Care Financing Administration
HIC	: Health Insurance Commission
HMO	: Health Maintenance Organizations
KDD	: Knowledge discovery in database
KL	: Kullback-Leibler
KLOD	: KL divergence for outlier detection
MLP	: Multilayer Neural Networks
NHCAA	: The National Health Care Anti-Fraud Association
ODM	: Odminer (Oracle data mining tool)
OLAP	: Online Analytical Processing
OLTP	: Online Transaction Processing
OCSVM	: One Class Support Vector Machine
OFP	: Outlier-Finding Process
PAM	: Partitioning Around Medoids
pdf	: Probability Density Function
RDBMS	: Relational Database Management System
ROC	: Receiver Operating Characteristics
SDEL	: Sequentially Discounting Laplace Estimation
SDEM	: Sequentially Discounting Expectation and Maximizing
SMP	: Symmetric Multiprocessing
SQL	: Structured Query Language
SVM	: Support Vector Machine
XML	: Extensible Markup Language

FIGURES

PAGES

Figure II.1 The Evolution of Database System Technology..................................6

Figure II.2 Steps of Knowledge Discovery in Database..8

Figure II.3 Overall Procedure of the OCSVM Method..20

Figure II.4 Example Of A Clinical Instance And The Corresponding Temporal Graph..28

Figure II.5 Examples of the Subtraction Operation..28

Figure II.6 Two Example Temporal Graphs Of Size 3..29

Figure II.7 Two Candidate Temporal Graphs Resulting From Joining G1 And G2 In Figure II.3..29

Figure II.8 Fraud Detection Scheme By Using Sub-models With Feedback Connections...32

Figure II.9 Example Of TP-FP Plane..33

Figure III.1 Data Reconciliation...40

Figure III.2 ER Diagram of the Database..47

Figure III.3 Creating a User...56

Figure III.4 Import Data...57

Figure III.5 Import Data Results...57

Figure III.6 Grant Privileges..58

Figure III.7 Oracle Data Miner Bin Directory..59

Figure III.8 Oracle Data Miner Choose Connection Dialog..................................59

Figure III.9 New Connection Dialog..60

Figure III.10 Oracle Data Miner Choose Connection Dialog..................................60

Figure III.11 Oracle Data Miner Main Screen...61

Figure III.12 Activity Build Source Table SQL Statement for Fraud Type 1..........63

Figure III.13 Activity Apply Source Table SQL Statement for Fraud Type 1.........63

Figure III.14 Activity Creating Wizard..64

Figure III.15 Selecting Case Table..65

Figure III.16 Selecting Source Table Columns...65

Figure III.17 Activity Name window..66
Figure III.18 Advanced Setting Dialog..66
Figure III.19 Activity Build Solution Window...67
Figure III.20 Selecting Build Activity Window..68
Figure III.21 Selecting Apply Data Source Window...68
Figure III.22 Selecting Supplemental Columns Window..69
Figure III.23 Entering Apply Activity Name Window..69
Figure III.24 SQL statement for counting suspicious data.......................................71
Figure III.25 SQL statement for counting correctly predicted case.........................71
Figure III.26 Result Viewer Screen...71
Figure III.27 Constant Input Warning Message..75
Figure III.28 Null Input Warning Message..75
Figure III.29 Activity Build Source View SQL Statement for Fraud Type 2..........78
Figure III.30 Activity Apply Source View SQL Statement for Fraud Type 2.........79
Figure III.31 SQL Statement for Taking Health Centers' Statistics.........................84
Figure III.32 SQL Statement for Creating Hospital Source Table...........................85
Figure III.33 SQL Statement for Updating Hospital Source Table..........................86
Figure III.34 SQL Statement for Updating Hospital Source Table as Suspicious...86
Figure III.35 SQL Statement for Creating Hospital Activity Build Source for Fraud Type 3..87
Figure III.36 SQL Statement for Creating Pharmacy Source Table.........................91
Figure III.37 SQL Statement for Updating Pharmacy Source Table........................91
Figure III.38 SQL Statement for Updating Pharmacy Source Table as Suspicious..92
Figure III.39 SQL Statement for Creating Pharmacy Activity Build Source for Fraud Type 3..92
Figure IV.1 SQL Statement for Getting Top Most Probability Claims for Fraud Type 1..99
Figure IV.2 SQL Statement for Getting Top Most Probability Claims for Fraud Type 2..101
Figure IV.3 SQL Statement for Calculation of the Number of Claims of Same User..102
Figure IV.4 SQL Statement for Getting Top Most Probability Hospitals for Fraud Type 3..104

Figure IV.5 SQL Statement for Getting Top Most Probability Pharmacy for Fraud Type 3..106

TABLES

PAGES

Table II.1 Time Line of Data Mining Development..11

Table II.2 Data Mining Tasks..12

Table II.3 Classification Algorithms Comparison...13

Table II.4 Clustering Algorithms Comparison..15

Table II.5 Sample Results of a Fraud Detection Method Based On Clustering Algorithms...34

Table III.1 User Table..48

Table III.2 Claim Table..49

Table III.3 Claim Detail Table...50

Table III.4 Policy Table...51

Table III.5 Agency Table...52

Table III.6 Salesman Table..53

Table III.7 Person Table..54

Table III.8 Health Center Tables...55

Table III.9 Models Results for Fraud Type 1..76

Table III.10 Models Results for Fraud Type 2..83

Table III.11 Statistical Information of Health Centers..84

Table III.12 Hospital Models Results for Fraud Type 3...90

Table III.13 Pharmacy Models Results for Fraud Type 3.......................................96

Table IV.1 Top Ten Claims for Fraud Type 1...100

Table IV.2 Top Ten Claims for Fraud Type 2...102

Table IV.3 Numbers of Claims of Same Users...102

Table IV.4 Top Ten Hospitals for Fraud Type 3...104

Table IV.5 Top Ten Pharmacies for Fraud Type 3..106

CHAPTER I

INTRODUCTION AND AIM

I.1 INTRODUCTION

Health insurance is a general term for insurance against loss by sickness or bodily injury. It typically includes coverage for expenses such as doctor visits and hospital stays, and can cover normal and preventive care such as check-ups, prenatal and baby care. It is sometimes used more broadly to include insurance covering disability or long-term nursing or custodial care needs. It is the state primary body that, enhances the health conditions, and eliminates the hazards which threat the public health, and obtains health services.

The concept of health insurance was proposed in 1694 by Hugh the Elder Chamberlen from the Peter Chamberlen family. In the late 19th century, "accident insurance" began to be available, which operated much like modern disability insurance. This payment model continued until the start of the 20th century in some jurisdictions (like California), where all laws regulating health insurance actually referred to disability insurance.

Accident insurance was first offered in the United States by the Franklin Health Assurance Company of Massachusetts. This firm, founded in 1850, offered insurance against injuries arising from railroad and steamboat accidents. Sixty organizations were offering accident insurance in the US by 1866, but the industry consolidated rapidly soon thereafter. While there were earlier experiments, the origins of sickness coverage in the US effectively date from 1890. The first employer-sponsored group disability policy was issued in 1911.

Before the development of medical expense insurance, patients were expected to pay all other health care costs out of their own pockets, under what is known as the fee-for-service business model. During the middle to late 20th century, traditional disability insurance evolved into modern health insurance programs. Today, most comprehensive private health insurance programs cover the cost of routine, preventive, and emergency health care procedures, and also most prescription drugs, but this was not always the case.

Hospital and medical expense policies were introduced during the first half of the 20th century. On a pre-paid basis, eventually leading to the development During the 1920s, individual hospitals began offering services to individuals of Blue Cross organizations. The predecessors of today's Health Maintenance Organizations (HMOs) originated beginning in 1929, through the 1930s and on during World War II.

Health care became an important sector in developed countries. The governments which provide their economical and social developments have built several programs for covering the health expenses and encouraged private sectors to build insurance companies. Besides, the governments keep pushing the employers to insure their employees so that the employees get better life standards. Because governments know that the raise in the ratio of insured employees is an important indicator for development of countries. Consequently, the number of people integrated to the insurance sector has been greatly enhanced. But this enhancement caused some big problems, including medical fraud and abuse.

Health care fraud is an intentional deception or misrepresentation that the individual or entity makes knowing that the misrepresentation could result in some unauthorized benefit to the individual, or the entity or to some other party. Medical fraud can be made in different ways. It is possible for affiliates, employers or medical professionals to make medical fraud.

For example medical professionals tricks include individuals obtaining subsidized or fully-covered prescription pills that are actually unneeded and then selling them on the black market for a profit, billing by practitioners for care that they never rendered, filing duplicate claims for the same service rendered, altering the dates, description of services, or identities of members or providers, billing for a non-covered service as a covered service, modifying medical records, intentional incorrect reporting of diagnoses or procedures to maximize payment, use of unlicensed staff, accepting or giving kickbacks for member referrals, waiving member co-pays, and prescribing additional or unnecessary treatment. Also affiliates can commit health care fraud by providing false information when applying for programs or services, forging or selling prescription drugs, using transportation benefits for non-medical related purposes, and loaning or using another’s insurance card.

With the increasing number of insured people, medical fraud began to compose great damage to the sector. The United States Government Accountability Office (GAO) estimates that $1 out of every $7 spent on Medicare is lost to fraud and abuse and that in 1998 alone, Medicare lost nearly $12 billion to fraudulent or unnecessary claims [1]. According to a report published by the General Accounting Office in the US, healthcare fraud and abuse costs the US as much as 10% of its annual spending on healthcare, representing US$ 100 billion per year [2]. Also according to the National Health Care Anti-Fraud Association's estimation, at least $51 billion is lost to health care fraud in calendar year 2003 [3]. In 2007, $2.26 trillion was spent on health care and more than 4 billion health insurance claims were processed in the United States. It is an undisputed reality that some of these health insurance claims are fraudulent. Although they constitute only a small fraction, those fraudulent claims carry a very high price tag.

The National Health Care Anti-Fraud Association (NHCAA) estimates conservatively that 3% of all health care spending-or $68 billion-is lost to health care fraud. That’s more than the gross domestic product of 120 different countries, including Iceland, Ecuador, and Kenya [4]. Other estimates by government and law enforcement agencies place the loss due to health care fraud as high as 10 percent of US nation’s annual health care expenditure-or a staggering $226 billion-each year [5]. These losses impact patients by increasing insurance premiums, governments by increasing the health care costs and taxpayers by increasing the taxes.

The goal of this research is to support the sector by developing a robust system for fraud detection and prevention.

I.2 OBJECTIVE OF THE RESEARCH

In this research, a data mining based fraud detection system is designed in order to determine suspicious transactions for health insurance companies. In this case-study, odminer (ODM) which is an Oracle data mining tool is used.

I.3 STRUCTURE OF THE RESEARCH

The contents of the research as follows:

- *Chapter 1 Introduction and Aim:* This chapter provides an introduction to the problem, objective of the thesis and illustrates the main structure of the thesis.

- *Chapter 2 General Background:* This chapter contains a general review about fraud detection system types. A brief literature research supplies necessary information for this chapter.

- *Chapter 3 The Study:* In this chapter, all steps and strategies of design of a data mining based fraud detection system for insurance companies. This is a case-study section and Chapter 3 is the main framework of the thesis.

- *Chapter 4 Results and Discussions:* This chapter presents all results of the study and supplies discussions.

- *Chapter 5 Concluding Remarks and Recommendations:* In this chapter, recommendations for future studies over similar subjects are contained.

CHAPTER II

GENERAL BACKROUND

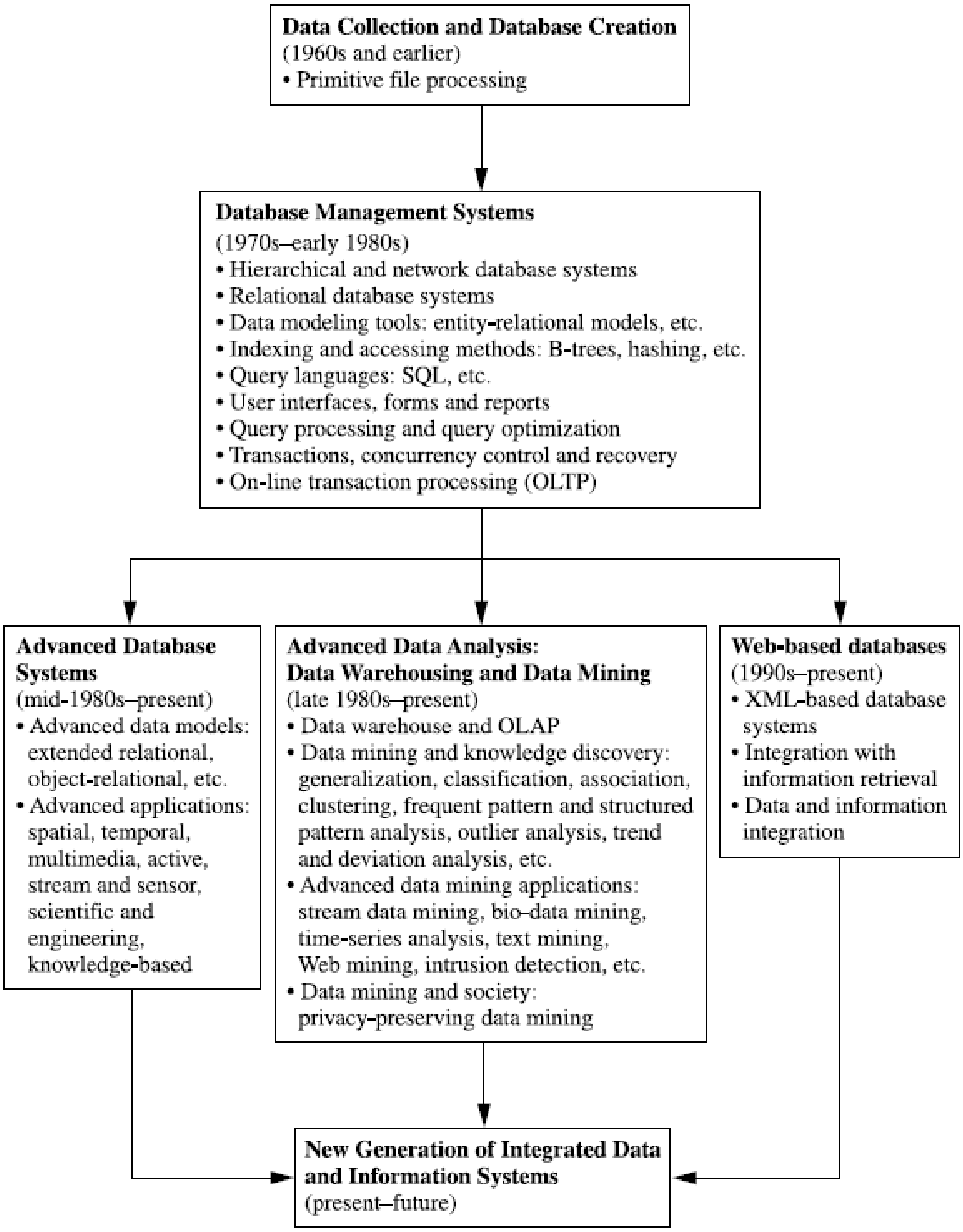
II.1 KNOWLEDGE DISCOVERY IN DATABASE

There is a big evolution of information technology. Also the database system industry has witnessed an evolutionary path in the development of the following functionalities: data collection and database creation, data management (including data storage and retrieval, and database transaction processing), and advanced data analysis (involving data warehousing and data mining). **Figure II.1** shows the evolution of database system technology [6].

Since the 1960s, database and information technology has been evolving systematically from primitive file processing systems to sophisticated and powerful database systems. The research and development in database systems since the 1970s has progressed from early hierarchical and network database systems to the development of relational database systems (where data are stored in relational table structures), data modeling tools, and indexing and accessing methods.

Database technology since the mid-1980s has been characterized by the popular adoption of relational technology and an upsurge of research and development activities on new and powerful database systems. These promote the development of advanced data models such as extended-relational, object-oriented, object-relational, and deductive models. Heterogeneous database systems and Internet-based global information systems such as the World Wide Web (WWW) have also emerged and play a vital role in the information industry.

The steady and amazing progress of computer hardware technology in the past three decades has led to large supplies of powerful and affordable computers, data collection equipment, and storage media. This technology provides a great boost to the database and information industry, and makes a huge number of databases and information repositories available for transaction management, information retrieval, and data analysis. [6]



Data Collection and Database Creation
(1960s and earlier)
• Primitive file processing
Database Management Systems
(1970s–early 1980s)
• Hierarchical and network database systems
• Relational database systems
• Data modeling tools: entity-relational models, etc.
• Indexing and accessing methods: B-trees, hashing, etc.
• Query languages: SQL, etc.
• User interfaces, forms and reports
• Query processing and query optimization
• Transactions, concurrency control and recovery
• On-line transaction processing (OLTP)
Advanced Database Systems
(mid-1980s–present)
• Advanced data models: extended relational, object-relational, etc.
• Advanced applications: spatial, temporal, multimedia, active, stream and sensor, scientific and engineering, knowledge-based
Advanced Data Analysis: Data Warehousing and Data Mining
(late 1980s–present)
• Data warehouse and OLAP
• Data mining and knowledge discovery: generalization, classification, association, clustering, frequent pattern and structured pattern analysis, outlier analysis, trend and deviation analysis, etc.
• Advanced data mining applications: stream data mining, bio-data mining, time-series analysis, text mining, Web mining, intrusion detection, etc.
• Data mining and society: privacy-preserving data mining
Web-based databases
(1990s–present)
• XML-based database systems
• Integration with information retrieval
• Data and information integration
New Generation of Integrated Data and Information Systems
(present–future)

Figure II.1 The Evolution of Database System Technology [6].

Data can now be stored in many different kinds of databases and information repositories and the amount of data kept in computer files and databases is growing at a phenomenal rate. At the same time, the users of these data are expecting more sophisticated information from them. For example, a marketing manager is no longer satisfied with a simple listing of marketing contacts; he wants detailed information about customers' past purchases as well as predictions of future purchases. Simple structured query languages are not adequate to support these types of demands. Knowledge discovery in database (KDD) steps in to solve these needs.

KDD is the overall process of converting raw data into useful information [7]. This process' steps are;

1) Data Cleaning
2) Data Integration
3) Data Selection
4) Data Transformation
5) Data Mining
6) Pattern Evaluation and Knowledge Presentation

Figure II.2 shows the steps of KDD [6].

II.1.1 Preprocessing

Four steps of KDD are different forms of preprocessing, where the data are prepared to data mining. Today's real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogeneous sources. Low-quality data will lead to low-quality mining results. Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data. There are six methods to fill missing values: Ignore the tuple, fill in the missing value manually, use a global constant to fill in the missing value, use the attribute mean to fill in the missing value, use the attribute mean to fill in the missing value, and use the most probable value to fill in the missing value. Ignore the tuple is usually done when the class label is missing. This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably. Filling the missing value manually is time consuming and may not be feasible given a large data set with many missing values.

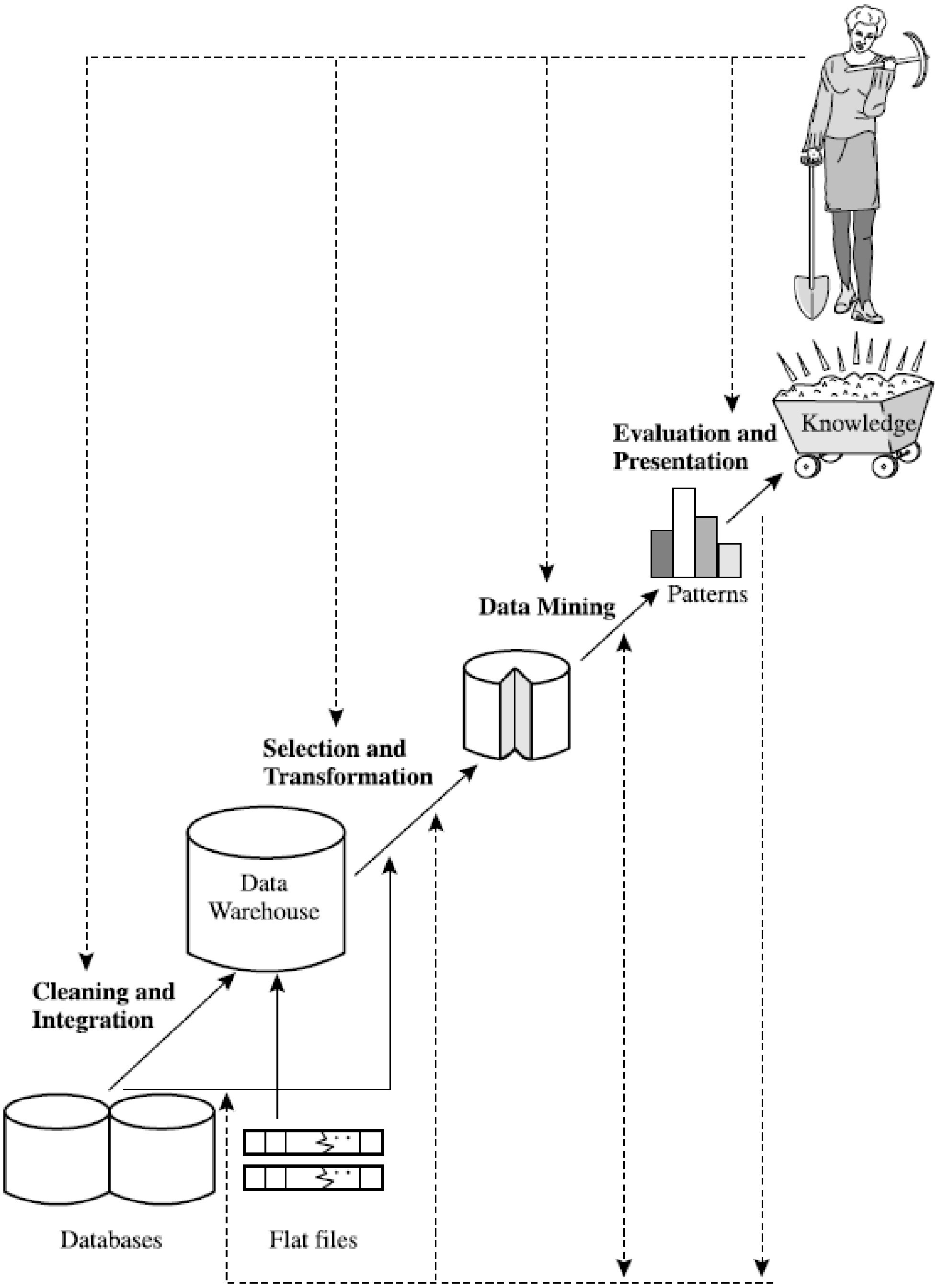


Figure II.2 Steps of Knowledge Discovery in Database [6].

Use a global constant to fill in the missing value: Replace all missing attribute values by the same constant, such as a label like “Unknown”. If missing values are replaced by, say, “Unknown,” then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common. Although this method is easy, it is not foolproof. Use the attribute mean to fill in the missing value, use the attribute mean for all samples belonging to the same class as the given tuple and use the most probable value to fill in the missing value are the other filling missing value techniques. Use the most probable value to fill in the missing value method is a most popular method. It uses the most information from the present data to predict missing values.

Noise is a random error or variance in a measured variable. Data smoothing is the act of using statistical techniques to remove irrelevant data points from the dataset. There are three methods for this activity: binning method, regression method, and clustering method. Binning method smoothes a sorted data value by consulting its neighborhoods which are the values around it. The sorted values are distributed into a number of "buckets," or bins. Because binning methods consult the neighborhood of values, they perform local smoothing. Data can be smoothed by fitting the data to a function, such as with regression. Linear regression involves finding the "best" line to fit two attributes (or variables), so that one attribute can be used to predict the other. Multiple linear regressions is an extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface.

Outliers may be detected by clustering, where similar values are organized into groups, or "clusters." Intuitively, values that fall outside of the set of clusters may be considered outliers.

Database can contain inconsistent values. Regardless of the cause of the inconsistent values, it is important to detect and, if possible, correct such problems. Some types of the inconsistencies are easy to detect. In other cases, it can be necessary to consult an external source of information. Once an inconsistency has been detected, it is sometimes possible to correct the data. The correction of an inconsistency requires additional or redundant information.

Data integration combines data from multiple sources into a coherent data store. These sources may include multiple databases, data cubes, or flat files. There are a number of issues to consider during data integration. Schema integration and object matching can be used. There are three main issues in data integration. One of them is entity identification problem. It is the matching up of equivalent real world entities from multiple data sources. Other important issue is redundancy. An attribute (such as annual revenue, for instance) may be redundant if it can be "derived" from another attribute or set of attributes. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set. A third important issue in data integration is the detection and resolution of data value conflicts. An attribute in one system may be recorded at a lower level of abstraction than the “same” attribute in another. It is the most encountered situation.

Data selection includes tasks which are get relevant data to the analysis from the database.

In data transformation, the data are transformed or consolidated into forms appropriate for mining. Data transformation can involve aggregation, generalization, normalization, attribute construction.

II.1.2 Data Mining

Data mining is the process of automatically discovering useful information in large data repositories. Data mining is a fifth step of KDD. Data mining techniques are deployed to scour large databases in order to find novel and useful patterns that might otherwise remain unknown. The current evolution of data mining functions and products is the result of years of influence from many disciplines, including databases, information retrieval, statistics, algorithms and machine learning. A major trend in the database community is to combine results from these seemingly different disciplines into one unifying data or algorithmic approach. The ultimate goal of this evolution is to develop a big picture view of the area that will facilitate integration of the various types of applications into real world user domains [8].

Data mining related areas are artificial intelligence**,** information retrieval, databases, and statistics. **Table II.1** shows developments in these areas leading to the current view of data mining [8].

Table II.1 Time Line of Data Mining Development [8]

Time	Area	Contribution
Late 1700s	Stat	Bayes theorem of probability
Early 1900s	Stat	Regression Analysis
Early 1920s	Stat	Maximum likelihood estimate
Early 1940s	AI	Neural Networks
Early 1950s		Nearest Neighbor
Early 1950s		Single Link
Late 1950s	AI	Perceptron
Late 1950s	Stat	Resampling, bias reduction, jackknife estimator
Early 1960s	AI	ML Started
Early 1960s	DB	Batch reports
Mid 1960s		Decision Trees
Mid 1960s	Stat	Linear Models for classification
	IR	Similarity measures
	IR	Clustering
	Stat	Exploratory data analysis (EDA)
Late 1960s	DB	Relational data model
Early 1970s	IR	SMART IR system
Mid 1970s	AI	Genetic algorithms
Late 1970s	Stat	Estimation with incomplete data
Late 1970s	Stat	K-means clustering
Early 1980s	AI	Kohonen self-organizing map
Mid 1980s	AI	Decision tree algorithms
Early 1990s	DB	Association rule algorithms
		Web and search engines
1990s	DB	Data warehousing
1990s	DB	Online analytic processing (OLAP)

Data mining tasks are generally divided into two major categories; predictive and descriptive tasks. The objective of predictive tasks is to predict the value of a particular attribute based on the values of other attributes. The attribute to be predicted is commonly known as a target, while the attributes used for making the prediction are known as the explanatory. Predictive tasks are classification, regression, time series analysis, prediction. The objective of descriptive tasks is to derive patterns that summarize the underlying relationships in data. Descriptive data mining tasks are often exploratory in nature and frequently require post processing techniques to validate and explain the results. Descriptive tasks are clustering, summarization, association rules, sequence discovery. **Table II.2** shows the data mining tasks.

Table II.2 Data Mining Tasks

PREDICTIVE TASKS	DESCRIPTIVE TASKS
Classification	Clustering
Regression	Summarization
Time Series Analysis	Association Rules
Prediction	Sequence Discovery

Data mining functions can also be classified as supervised and unsupervised functions. Supervised functions are used to predict a value, they require the specification of a target (known outcome). Targets are either binary attributes indicating yes/no decision (buy/don't buy) or multi-class targets indicating a preferred alternative (color of sweater). Naïve Bayes for classification is a supervised mining algorithm. Unsupervised functions are used to find the intrinsic structure, relations, or affinities in data. Unsupervised mining does not use a target. Clustering algorithms can be used to find naturally occurring groups in data.

II.1.2.1 Basic Data Mining Tasks

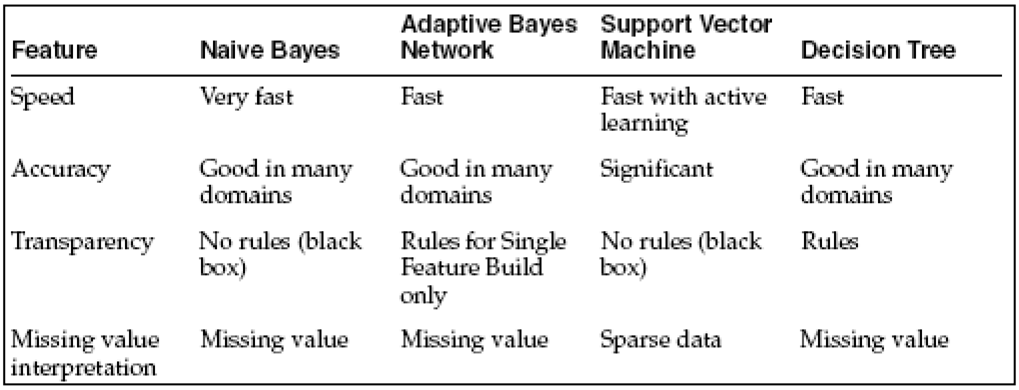
II.1.2.1.1 Classification

Classification of a collection consists of dividing the items that make up the collection into categories or classes. In the context of data mining, classification is done using a model that is built on historical data. The goal of predictive classification is to accurately predict the target class for each record in new data, that is, data that is not in the historical data.

A classification task begins with build data (also known as training data) for which the target values (or class assignments) are known. Different classification algorithms use different techniques for finding relations between the predictor attributes' values and the target attribute's values in the build data. These relations are summarized in a model; the model can then be applied to new cases with unknown target values to predict target values. A classification model can also be applied to data that was held aside from the training data to compare the predictions to the known target values; such data is also known as test data or evaluation data. The comparison technique is called testing a model, which measures the model's predictive accuracy. The application of a classification model to new data is called applying the model, and the data is called apply data or scoring data. Applying a model to data is often called scoring the data [9].

There are four main classification algorithms: Decision Tree Algorithm, Naive Bayes Algorithm, Adaptive Bayes Network Algorithm, and Support Vector Machine Algorithm. **Table II.3** shows comparison of classification algorithms [10].

Table II.3 Classification Algorithms Comparison [10]



Feature	Naive Bayes	Adaptive Bayes Network	Support Vector Machine	Decision Tree
Speed	Very fast	Fast	Fast with active learning	Fast
Accuracy	Good in many domains	Good in many domains	Significant	Good in many domains
Transparency	No rules (black box)	Rules for Single Feature Build only	No rules (black box)	Rules
Missing value interpretation	Missing value	Missing value	Sparse data	Missing value

II.1.2.1.2 Regression

Regression is used to map a data item to a real valued prediction variable. In actuality, regression involves the learning of the function that does this mapping. Regression assumes that the target data fit into some known type of functions and then determines the best function of this type that models the given data.

Regression models are similar to classification models. The difference between regression and classification is that regression deals with numerical or continuous target attributes, whereas classification deals with discrete or categorical target attributes. In other words, if the target attribute contains continuous (floating-point) values or integer values that have inherent order, a regression technique can be used. If the target attribute contains categorical values, that is, string or integer values where order has no significance, a classification technique is called for.

Support Vector Machine, Active Learning, and One Class Support Vector Model can be used for regression models.

II.1.2.1.3 Time Series Analysis

With the time series analysis, the value of an attribute is examined as it varies over time. The values usually are obtained as evenly spaced time points. A time series plot is used to visualize the time series. There are three basic functions performed in time series analysis. In one case, distance measures are used to determine the similarity between different time series. In the second case, the structure of the line is examined to determine its behavior. A third application would be to use the historical time series plot to predict future values.

II.1.2.1.4 Prediction

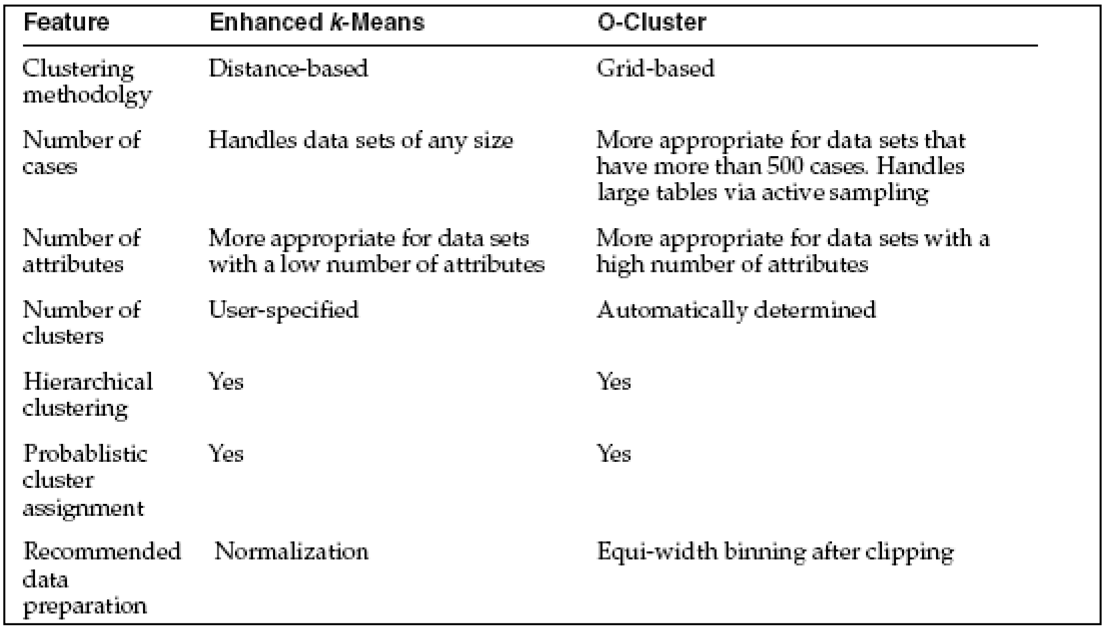
Many real world data mining application can be seen as predicting future data states based on past and current data. Prediction can be viewed as a type of classification. The difference is that predicting is predicting a future state rather than a current state. Prediction applications include flooding, speech recognition, machine learning, and pattern recognition. Although future values may be predicted using time series analysis or regression techniques, other approaches may be used as well.

II.1.2.1.5 Clustering

Clustering is similar to classification except that the groups are not predefined, but rather defined by the data alone. Clustering is alternatively referred to as unsupervised learning or segmentation. It can be thought of as partitioning or segmenting the data into groups that might or might not be disjointed. The clustering is usually accomplished by determining the similarity among the data on predefined attributes. The most similar data are grouped into clusters. Since the clusters are not predefined, a domain expert often requires interpreting the meaning of the created clusters. Clustering analysis identifies clusters embedded in the data. A cluster is a collection of data objects that are similar in some sense to one another. A good clustering method produces high-quality clusters to ensure that the inter-cluster similarity is low and the intra-cluster similarity is high; in other words, members of a cluster are more like each other than they are like members of a different cluster.

Clustering aims to find useful groups of clusters where usefulness is defined by the goals of the data analysis. There are five different types of clusters: Well separated, prototype based, graph based, density based, and conceptual clusters. Well separated cluster is a set of objects in which each object is closer to every other object in the cluster than to any object not in the cluster. Prototype based cluster is a set of objects in which each object is closer to the prototype that defines the cluster than to the prototype of any other cluster. If the data is represented as a graph where the nodes are objects and the links represent connections among objects then a cluster can be defined as a graph based. Density based cluster is a dense region of objects that is surrounded by a region of low density. Conceptual cluster is a set of objects that share some property. There are two main clustering algorithms; K-Means Algorithm and O-Cluster Algorithm. The main characteristics of the enhanced k-means and O-Cluster algorithms are summarized in **Table II.4** [10].

Table II.4 Clustering Algorithms Comparison [10]



Feature	Enhanced *k*-Means	O-Cluster
Clustering methodolgy	Distance-based	Grid-based
Number of cases	Handles data sets of any size	More appropriate for data sets that have more than 500 cases. Handles large tables via active sampling
Number of attributes	More appropriate for data sets with a low number of attributes	More appropriate for data sets with a high number of attributes
Number of clusters	User-specified	Automatically determined
Hierarchical clustering	Yes	Yes
Probablistic cluster assignment	Yes	Yes
Recommended data preparation	Normalization	Equi-width binning after clipping

II.1.2.1.6 Summarization

Summarization maps data into subsets with associated simple descriptions. Summarization is also called characterization or generalization. It extracts or derives representative information about the database. This may be accomplished by actually retrieving portions of the data. Alternatively summary type information can be derived from the data. The summarization succinctly characterizes the contents of the database.

II.1.2.1.7 Association Rules

Association refers to the data mining task of discovering interesting relationships hidden in large data sets. The uncovered relationships can be represented in the form of association rules. Association rule is a model that identifies specific types of data association. These are not causal relationships. They do not represent any relationship inherent in the actual data or in the real world. However association rules can be used to assist retail store management in effective advertising, and marketing.

II.1.2.1.8 Sequence Discovery

Sequential analysis or sequence discovery is used to determine sequential patterns in data. These patterns are based on a time sequence of actions. These patterns are similar to associations in that data are found to be related, but the relationship is based on time.

II.1.2.2 Data Mining Issues

There are many important implementation issues associated with data mining. These are human interaction, over fitting, outliers, interpretation of results, visualization of results, large datasets, high dimensionality, multimedia data, missing data, irrelevant data, noisy data, changing data, integration, and application. These issues should be addressed by data mining algorithms and products.

II.1.2.2.1 Anomaly Detection

Anomaly detection consists of identifying novel or anomalous patterns [9]. These non-conforming patterns are often referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants in different application domains. Of these, anomalies and outliers are two terms used most commonly in the context of anomaly detection; sometimes interchangeably. Anomaly detection finds extensive use in a wide variety of applications [11]. The importance of anomaly detection is due to the fact that anomalies in data translate to significant and often critical actionable information in a wide variety of application domains. Detecting outliers or anomalies in data has been studied in the statistics community as early as the 19th century [12]. Over time, a variety of anomaly detection techniques have been developed in several research communities. Many of these techniques have been specifically developed for certain application domains, while others are more generic. An Outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism [13]. It is a data object that does not comply with the general behavior of the data, it can be considered as noise (One person's noise could be another person's signal) or exception, which is quite useful in rare events analysis [6].

There are three common causes of anomalies: data from different classes, natural variation, and data measurement or collection error. An object may be different from other objects because it is of a different type or class. Many data sets can be modeled by statistical distributions where the probability of a data object decreases rapidly as the distance of the object from the center of the distribution increases. The systems utilize the data collection in an organization’s data storages, which may be from legacy systems, databases or files which are in different format and structure. Therefore, it is important that the data is formatted and structured to suit the data activities to produce significant results [14]. So errors in the data collection or measurement process are another anomaly sources. During data collection, error records are also identified through the use of table column constraints and references to linked descriptive tables. Through constant feedback of the errors identified, the data quality at source can be improved, preferably during data capture [15]. A straightforward anomaly detection approach, therefore, is to define a region representing normal behavior and declare any observation in the data which does not belong to this normal region as an anomaly. But several factors make this apparently approach very challenging. Defining a normal region which encompasses every possible normal behavior is very difficult. When anomalies are the result of malicious actions, the malicious adversaries often adapt themselves to make the anomalous observations appear like normal, thereby making the task of defining normal behavior more difficult. In many domains normal behavior keeps evolving and a current notion of normal behavior might not be sufficiently representative in the future. The exact notion of an anomaly is different for different application domains. Availability of labeled data for training/validation of models used by anomaly detection techniques is usually a major issue. Often the data contains noise which tends to be similar to the actual anomalies and hence is difficult to distinguish and remove.

There are three main techniques of anomaly detection: Model based techniques, proximity based techniques and density based techniques. Many anomaly detection techniques first build a model of the data. Anomalies are objects that do not fit the model very well. Models can be created by clustering. If the model is a set of clusters, anomalies are the objects that do not strongly belong to any clusters. Also classification techniques can be used for building a model. Proximity based techniques define a proximity measure between objects .Anomalous objects are those that are distant from most of the other objects. Many of proximity based techniques are based on distances so they are referred to as distance based outlier detection techniques. Density based techniques compute density of the objects and considers objects anomalous that are in regions of low density are relatively distant from their neighbors.

There is no single universally applicable or generic outlier detection approach as discussed in [16, 17, 18] .Different approaches have been reported by researches for anomaly detection. Mongiovi et al. [19] proposed a time series analysis approach for anomalies detection in water supply network analysis. A novel approach based on Data Mining techniques, which allows realizing early fault diagnosis, is shown. Wei et al. [20] proposed a system which is based on the recently introduced idea of time series bitmaps. This system allows users to efficiently navigate through a time series of arbitrary length and identify portions that require further investigation. Zhang et al. [21] proposed a method for network anomaly detection based on one class support vector machine (OCSVM). The method consists of two main steps: first is the detector training, the training data set is used to generate the OCSVM detector which is able to learn the nominal profile of the data, and the second step is to detect the anomalies in the performance data with the trained detector. No prior knowledge of the normal and abnormal data is required for this method. This algorithm achieves the automated detection of the anomalies and can also support and complement the decisions provided by the current rule-based system. **Figure II.3** shows the overall procedure of OCSVM method [21]. Deshmeh et al. [22] proposed a novel technique for identifying anomalous instances in a horizontal data distribution scenario .This technique involves using a sampling technique which depends upon the clustering scheme that best fits the data, using an association rule analyzer which extracts both local and global anomaly detection association rules, and employing a set of local and global predictors that use subsets of attributes of the original dataset, mapping them on several target values. The distance between the predicted and actual attribute values of data instances is used as the source for the association rule extraction.

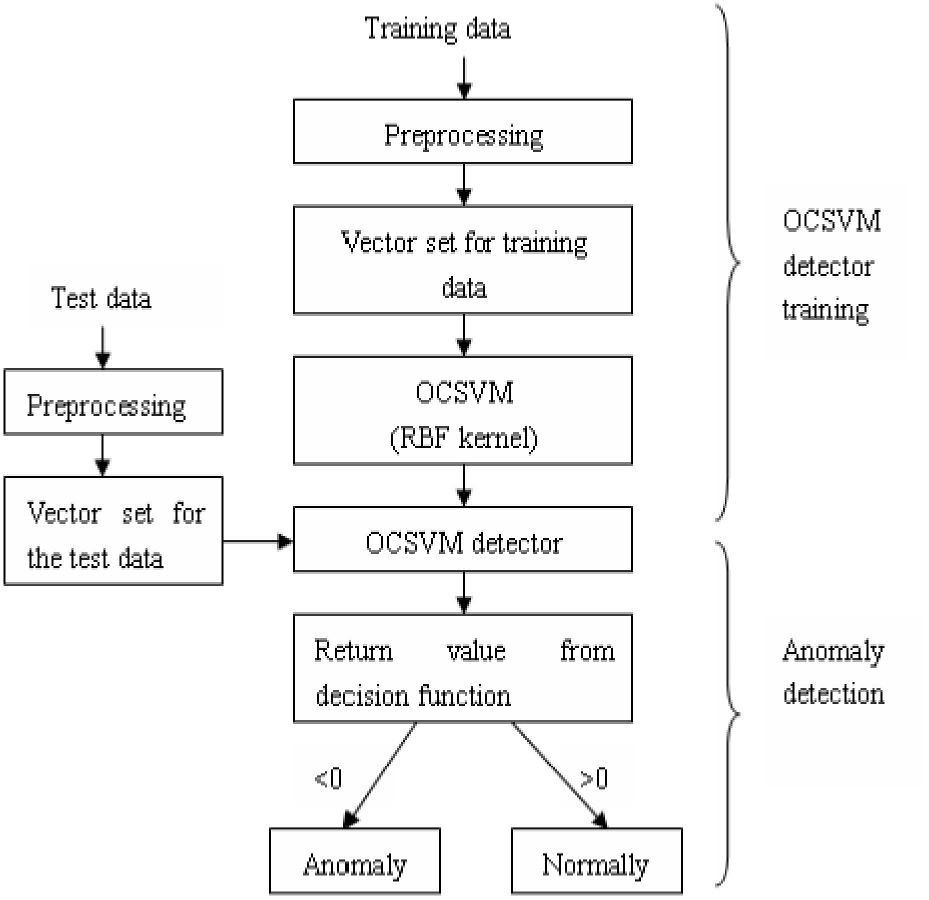


Figure II.3 Overall Procedure of the OCSVM Method [21]

The contributions of this technique are: a) distributed anomaly detection, where both data and process are distributed, only a limited form of sharing is allowed and no single entity is allowed to observe the whole data, in anyway, b) solving the problem in cases where concept drifts might occur, c) providing a solution which is able to handle potential dishonesty from participating entities, d) using association rules for anomaly detection, while maintaining the speed required in anomaly detection which is necessary in various applications. Oh et al. [23] proposed a new outlier detection method based on Kullback-Leibler (KL) divergence. The original concept of KL divergence was designed as a measure of distance between two distributions. This technique derived from Markov blanket algorithm where redundant and irrelevant features are removed based on KL divergence. KL divergence for outlier detection (KLOD) achieved higher or comparable performance than Mahalanobis distance based method and one-class SVM.

Knorr et al. [24] introduced a distance based approach in which outliers are those objects for which there are less than k points within a given threshold in the input data set. Angiulli et al. [25] proposed a distance- based outlier detection method which finds the top outliers and provides a subset of the data set, called outlier detection solving set, which can be used to predict if new unseen objects are outliers. Al-Zoubi [26] proposed an outlier detection method based on clustering algorithm. Partitioning Around Medoids (PAM) clustering algorithm is performed. Small clusters are determined and considered as outlier clusters. The rest of outliers are found in the remaining clusters based on calculating the absolute distances between the medoid of the current cluster and each of the points in the same cluster.

Cutsem et al. [27] present a method based on fuzzy clustering. In order to test the absence or presence of outliers, two hypotheses are used. However, the hypotheses do not account for the possibility of multiple clusters of outliers. Jiang et. al. [28] presented a two-phase method to detect outliers. In the first phase, the authors proposed a modified k-means algorithm to cluster the data, and then, in the second phase, an Outlier-Finding Process (OFP) is proposed. The small clusters are selected and regarded as outliers, using minimum spanning trees.

Hewahi et al. [29] proposed a novel approach for Class Outliers Mining based on the K nearest neighbors using distance-based similarity function to determine the nearest neighbors. They proposed a novel definition for Class Outlier and a ranking score that is Class Outlier Factor (COF) to measure the degree of being a Class Outlier for an object. Kumpulainen et al. [30] proposed the importance of scaling in distance based methods and the possibility to incorporate a priori knowledge of the relative importance of the variables by scaling. They investigated the effects of scaling on unsupervised distance-based anomaly detection. They present a scaling method that is solely based on a priori expert knowledge. They demonstrate the method using two radio interface performance measurements from a mobile telecommunication network.

Cansado et al. [31] presented an application of artificial of intelligence (AI) technology to the problem of automatic detection of anomalous records in a large database. The main goals of this study are an effective detection of the records that are potentially anomalous, a suitable selection of the subset of attributes that explains what makes a record anomalous, an efficient implementation that allows us to scale the approach to large databases. The representational power of Gaussian mixture models, together with an optimized version of the expectation maximization algorithm, and caching strategies throughout the whole implementation, provided an efficient algorithm for Bayesian network structure-learning. The results of algorithm on real databases indicate that it is possible to use a simple, yet powerful, model for estimating the joint probability density function (PDF) of the domain variables. They showed that this joint PDF can be used to effectively detect anomalies as low likelihood elements. They also showed that the joint probability factorization provided by the BN can help in the detection of rare objects.

There are several applications for which anomalies are of considerable interest. Intrusion detection, industrial damage detection, image processing, anomaly detection in text data, sensor networks, and fraud detection are some of them.

II.1.3 Pattern Evaluation and Knowledge Presentation

Pattern evaluation, which is the sixth step of KDD, identifies the truly interesting patterns representing knowledge based on some fascinating measures. At seventh step of KDD, which is knowledge presentation, visualization and knowledge representation techniques are used to present the mined knowledge to the user. The use of visualization techniques allows users to summarize, extract, and grasp more complex results than more mathematical or text type descriptions of the results. Visualization techniques include graphical, geometric, icon-based, pixel-based, hierarchical and hybrid. Traditional graph structures including bar charts, pie charts, histograms and line graphs may be used. Geometric techniques include the box plot and scatter diagram techniques. Using figures, colors, or other icons can improve the presentation of the results. With pixel-based techniques each data value is shown as a uniquely colored pixel. Hierarchical techniques divide the display area into regions based on data values by rank. [8]

II.2 FRAUD DETECTION

Fraud detection means detecting the fraud in the shortest period. It is troublesome occupation. There are a lot of works on fraud detection in the literature and practice. For example The Consortium to Combat Medical Fraud, a joint project of the Coalition Against Insurance Fraud, the National Health Care Anti-Fraud Association and the National Insurance Crime Bureau was formed in 2008 to develop new strategies to combat fraud committed by medical providers. Other participants include insurance companies, state insurance fraud bureaus and the FBI.

Great medical knowledge is required for detecting the fraudulent activities. Lots of health insurance companies use medical experts to detect abnormal activities. Medical experts have huge experiences and knowledge. They control the activities and determine frauds. But the increase in the number of insured people causes databases with huge sizes. So traditional fraud detection which medical experts review all health activity and designate abnormal ones is getting impractical. It takes too much time and they omit big portion of fraud. Also they cannot detect new types of fraud. Their experiences are not sufficiently qualified to notice these new types of fraud. Because of these handicaps, more complicated methods and algorithms that help experts to detect fraud and abuse from large database are required. More sophisticated antifraud systems incorporating a wide array of statistical methods are needed and are being developed for effective fraud detection. The major advantages of these systems include automatic learning of fraud patterns from data; specification of “fraud likelihood” for each case, so that efforts for investigating suspicious cases can be prioritized; and identification of new types of fraud which were not previously documented.

Service providers, including doctors, hospitals, ambulance companies, and laboratories; insurance subscribers, including patients and patients’ employers; and insurance carriers, who receive regular premiums from their subscribers and pay health care costs on behalf of their subscribers, including governmental health departments and private insurance companies may be involved in the commission of health care fraud [32]. According to which party commits the fraud, fraud behaviors can be classified as (a) service providers’ fraud like billing services that are not actually performed; unbundling, billing each stage of a procedure as if it were a separate treatment; up coding, billing more costly services than the one actually performed; performing medically unnecessary services solely for the purpose of generating insurance payments; misrepresenting non-covered treatments as medically necessary covered treatments for the purpose of obtaining insurance payments; and falsifying patients' diagnosis and/or treatment histories to justify tests, surgeries, or other procedures that are not medically necessary, (b) insurance subscribers' fraud like falsifying records of employment/eligibility for obtaining a lower premium rate; filing claims for medical services which are not actually received; and using other persons' coverage or insurance card to illegally claim the insurance benefits, (c) insurance carriers' fraud like falsifying reimbursements; and falsifying benefit/service statements. Among these three types of fraud, the one committed by service providers' accounts for the greatest proportion of the total health care fraud and abuse. Although the vast majority of service providers are honest and ethical, the few dishonest ones may have various possible ways to commit fraud on a very broad scale, thus posing great damage to the health care system [33].

Different techniques have been reported by researches for health care fraud detection. Viveros et al. [34] examined the effectiveness of two data mining techniques, association rules and neural segmentation, when applied to large databases in a health insurance information system. They have shown that data mining algorithms can be used successfully on large, real customer data , with reasonable execution time. Moreover, they have also shown that these algorithms can result in quantifiable benefits for the interested organization, helping identify specific actions to be taken. They provided a classification of general practitioners into groups of various sizes reflecting the nature and style of their practices. The new subgroups will allow greatly improved monitoring of practice patterns. Given that the business goal of health insurance corporations is modifying the behavior of practitioners towards best practice, neural segmentation provides the means of understanding and monitoring the behavior of the various subgroups in an application such as general practice. Sokol et al. [35] examined precursory tasks, which must be performed prior to the actual data mining, customer discussions, data extraction and cleaning, transformation of the database, and basic statistics and visualization of the information of the data. They describe the tasks performed in support of a project for HCFA (Health Care Financing Administration). They reported their goal setting discussions with the representatives from the HCFA and the Office of the Inspector General in a project for detecting service providers' fraud from the HCFA claim data. The discussions led to six focused types of fraud to be identified, including Ambulance Services, Skilled Nursing Facilities, Laboratory Services, Psychiatric Services, Home Health Services, and new or expanded benefits under the Balanced Budget Act of 1997. They reported that different digit formats were used to represent the same physician's unique identification number in the HCFA data, making automatic computer algorithms fail to link the claims involving the same physician. Yamanishi et al. [36] used unsupervised method is called SmartSifter to detect outliers in the pathology dataset provided by the HIC of Australia. They listed features for detecting the fraud committed by medical test laboratories. These features include the percentage distributions over several test categories (chemical, microbiology, and immunology), the number of different patients dealt with, and the frequency of tests performed. SmartSifter uses a probabilistic model to represent the underlying data-generating mechanism. In the probabilistic model, a histogram is used to represent the probability distribution of categorical variables; for each bin of the histogram, a finite mixture model is used to represent the probability distribution of continuous variables. When a new case is coming, SmartSifter updates the probabilistic model by employing an SDLE (Sequentially Discounting Laplace Estimation) and an SDEM (Sequentially Discounting Expectation and Maximizing) algorithm to learn the probability distributions of the categorical and continuous variables, respectively. A score is given to this new case, measuring how much the probabilistic model has changed since the last update. A high score indicates that this new case may be an outlier. William et al. [37] used the C5.0 algorithm in detecting insurance subscribers' fraud for the Health Insurance Commission (HIC) of Australia. Due to the huge amount of data (40,000 insurance subscribers), they faced the challenge that an overly complex decision tree with thousands of rules was generated, which made it difficult to interpret. To address this challenge, they proposed a three-step so-called "divide and conquer" procedure. First, a clustering algorithm was applied to divide all insurance subscribers' profiles into groups. Second, a decision tree was built for each group and then converted into a set of rules. Finally, each rule was evaluated by establishing a mapping from the rule to a measure of its significance using simple summary statistics such as the average number of claims made in that cluster and the average size of the claims; then extremes (either defined by experts or compared with the overall average) would be signaled for further investigation. Through this procedure, the number of rules was significantly reduced to 280 summed over ten clusters. Riedinger et al. [38] created an expert system assisting in detecting service providers' fraud whose name is Electronic Fraud Detection (EFD). It includes several steps. First, discriminating features are defined by experts. Then, the information gain for a provider is computed. The information gain measures how different the distribution of the provider is from that of all the peers taken together. Computation of the information gain can be greatly simplified under the assumption of independently normally distributed features. Finally, the providers are plotted as points on a 2-D space with one axis representing the information gain and the other representing total dollars paid to the providers. The frontier points on the plot are highlighted and the corresponding providers are considered to be suspicious. Liou et al. [39] employed three data mining techniques; logistic regression, neural networks, and classification trees, to detect fraudulent healthcare providers in the Taiwan NHI database based on their submitted claims. They used the SPSS application Clementine 7 to implement three data mining algorithms. All three approaches detect the fraudulent and abusive medical care institutions. In terms of overall accuracy, the classification tree is superior to the logistic regression and neural network models. The high rates of correct identification indicate that the selected variables can identify hospitals submitting irregular medical claims. The algorithm that features the lowest ratio of wrongly identified normal institutions could be implemented at the lowest cost. In light of this fact, the C5.0 tree is the optimal detection model. Yang et al. [40] proposed a fraud detection model based on clinical pathway. They intend to determine the ways of medical activities and find abnormal activities with the help of their characteristics. Clinical pathways (or integrated care pathways) defined as 'multidisciplinary care plans, in which diagnosis and therapeutic intervention are performed by physicians, nurses, and other staff for a particular diagnosis or procedure' [41]. Clinical pathway involves some rules. They consist of ordered activities. For example, the pathway of cholecystectomy begins with the preadmission process, which mainly involves preadmission testing and anesthesia consultation, goes though several assessments, surgery, and physician orders, and ends with a follow-up visit at the surgeon's office [42]. Fallowing the clinical pathway has important place on fraud detection. An abnormal activity which does not obey the pathways is marked as fraud. For example if a patient has an operation without any diagnosing it is likely a fraud. The most critical part of this system is identifying the pathways. System must control all types of ways. Generally blood analysis or any other analysis have done once time. But some cases doctors send patient to blood analysis more than one to make diagnosis definite. System must consider these detail parts. This technique uses the medical activity data. It gathers the data to create a clinical instance. In insurance company database the detail of clinical instance are scattered. For example the name, address, medical id of a doctor that concern with affiliate, data of activities, result of analysis are in different tables. To detect ways of clinical activities, system must combine these data. Also system must recourse to medical experts to understand the data and put them in order. To modeling the system, they intend to discover structure of pattern. Clinical instance contains medical activities. They can be blood analysis, blood pressure measuring, medicine treatment and surgical operation. These activities can execute sequentially, concurrently or repeatedly. Some of the activities are execute more than one. So to determine structure of pattern exactly they have to examine the base of the medical activities. To reach their goal they apply structure pattern mining techniques [43]. They define clinical instance I which involves activities, temporal graph G of a clinical instance I which involves set of activities and set of edges. **Figure II.4 A** shows the instance and **Figure II.4 B** shows its temporal graph. A temporal graph G is said to be frequent if it is supported by no less than s% of the clinical instances, where s% is a user-defined minimum support threshold. System must determine all frequent temporal graphs to discover structure of pattern. System can join the temporal graph to determine candidate graph. The new graph that created by joining two frequent temporal graph is a frequent temporal graph too.

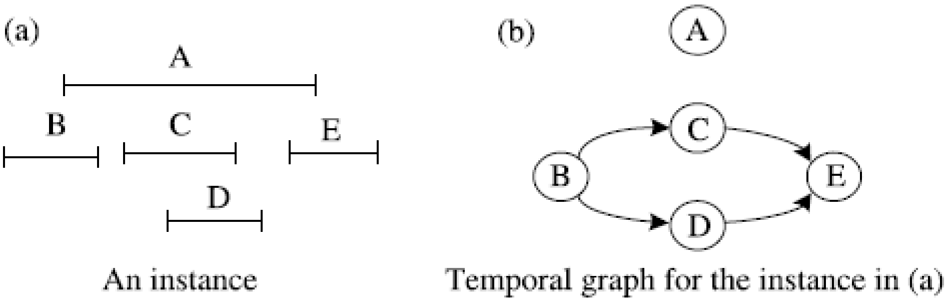


Figure II.4 Example Of A Clinical Instance And The Corresponding Temporal Graph [40].

But joining the temporal graph can cause a redundant candidate graph. To eliminate candidate graphs they use subtraction operation which says let G be a temporal graph and v be a vertex in G. The operation of subtracting v from G, denoted as GK{v}, deletes v and its associated edges from G. In addition, transitive edges via v are reconstructed by connecting each source vertex of incoming edges of v to each destination vertex of outgoing edges of v.

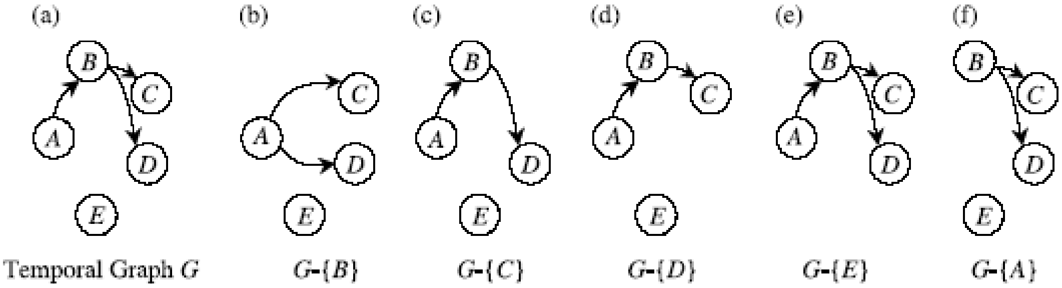


Figure II.5 Examples of the Subtraction Operation [40].

Figure II.5 shows the result of subtraction operation. To create a candidate temporal graph, system must decide if two frequent temporal graphs can be joined. To reach this decision it uses the definition which says two temporal graphs Gi and Gj are said to be joinable if there exists a source vertex s (with no incoming edges) in Gi and a sink vertex e (with no outgoing edges) in Gj such that Gi – {s} = Gj – {e}.

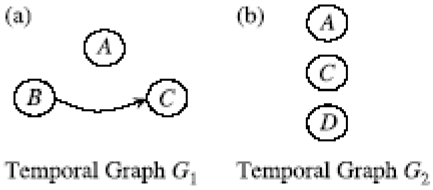


Figure II.6 Two Example Temporal Graphs Of Size 3 [40].

Two temporal graphs shown in **Figure II.6** are joinable. When system joins two temporal graphs the joint set exists. It involves two candidate temporal graphs. One of them is union of two graphs, the other one is union of them and adding a path from source vertex to sink vertex if there does not exist a path between each other.

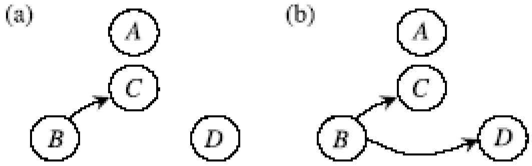


Figure II.7 Two Candidate Temporal Graphs Resulting From Joining G1 and G2 in Figure II.6 [40]

Figure II.7 shows the join set of G1 and G2 that are shown in **Figure II.6**. By joining the temporal graph system determine candidate temporal graph and also eliminate redundant temporal graph. So system determines the structure of pattern with these graphs. After determining the structure of patterns, system intends to select pattern feature. Frequent structure patterns are regarded as features in this system. So the there are too many features. Some of them are irrelevant or redundant. These types of features decrease the efficiency. Therefore system must eliminate these features. There are two main feature selection models. One of them is wrapper model which examines all features one by one and calculate their estimated accuracy than select the one of them which has highest estimated accuracy. The other one is filter model. There are three most well-known filter models. Their names are RELIEF, FOCUS and Markov blanket filter. Markov blanket filter selects optimal subset of feature .It eliminates feature which does not give additional information to a subset of features. Because of the huge number of feature that system has, it selects Markow blanket filter. First of all system must generate translated variables to eliminate features. A translated example e of an instance I is a pair (f, c), where f and c are a feature vector and a class label, respectively. fZ(f1,f2,.,fn) denotes an assignment of a set of Boolean features FZ(F1,F2,.,Fn), in which feature fj is set to 1 if and only if instance I supports the corresponding pattern of Fj. c is an assignment of a categorical variable C. After generating translated example system designates the ancestor features. Then system starts to control if the features support the instances from descendant features. System controls the features children if it does not have children or all of its children support the instances that feature pass the elimination. But if one of the features children does not support both instance it fails. This operation goes on until all features control. The remaining features form the detection model. After detection model forms, system takes medical cases and finds the fraudulent activities by using this detection model. Ortega et al. [44] proposed a fraud detection system based on multilayer neural networks. They examine fraud and abuse with their occasions. This algorithm's approach is analyzing the person who can cause this fraud. There are three actors in medical claim. The affiliates who pay an additional contribution for a specific health plan to insurance company and the fee determines the level of coverage, medical professionals who are doctors, nurses, medical experts and employers who employ affiliates at their company. This method intends to analyze these actors one by one with the help of multilayer neural networks (MLP). It examines all of them and gets an idea about actors. The association of these analyze gives a helpful data to detect fraud. Fraud detection based on multilayer neural networks use information about affiliates, medical professionals, and employers like age, sex, name of affiliate, identification of medical professional, type of claim, and identification of employer. This technique gets these data and separates them into affiliate data, employer data and medical professional data. System based on multilayer neural network gets two types of data. One of them is abusive medical claims which shows fraudulent pattern and the other one is rejected modified or approved medical claim by medical experts. This technique uses divide and conquer strategy. It classifies the data into four group which are medical claim, affiliate, medical professional and employer data. It determines the characteristics property of four types. It reduces these properties by choosing the most diagnostic property from correlated properties but if two properties which complete them are both taken. System gets historical data since 12 months backward. They intend to form a model that controls the medical claims and gives the fraud probability to each of them. System concentrates on medical claim actors. So system forms four sub-models by multilayer neural networks. Initially system modeled by two-layer neural networks but it changed it to multilayer neural network because two layer neural network showing high variance. Each sub-model created by 10 multilayer neural networks (MLP). Each sub-model has its special datasets. Affiliate sub-model interacts with affiliate database which is created by all affiliate data, like employer sub-model, medical professional sub-model and medical claim sub-model has their own data. These data are divided into three set. One of them is training set. MLPs use this set to designate networks weight. With the help of training operation MLPs decrease the wrong results. MLPs find out its error by validation data. MLPs evaluate the generalization performance by test data. This techniques name is early stopping. System generated from these four sub-models and the connections. Each sub-model connects with all other sub-models. The inputs of sub-models are feature vectors. Before the modeling the sub-model these feature vectors were computed. Each sub-model gets the input and gives a n estimated result. The outputs of each sub-model are inputs of other sub-model. These inputs gave more information to other sub-model for estimating the true result. This is a feedback mechanism. Medical claim model is executed daily because it takes claim every day. The other sub-models executed once monthly. All sub-model renew itself monthly by executing training operations. System adds new samples to their datasets. These samples are combination of equal number of fraudulent and normal cases. With the help of this operation, sub-models are updated and they can determine new types of fraud and abuse.

Figure II.8 shows the scheme of fraud detection system based on multilayer neural networks. This fraud detection system takes place on insurance companies' general system. Fraud detection system based on neural networks works at all night. It investigates all medical claims and gives them a fraud probability. This system works as a filter. Experts decide which claims they will concentrate on with the help of the results of it. This detection system cause some costs. The most important one is persons' who are controlling the claims salaries. System determines the probability threshold point.

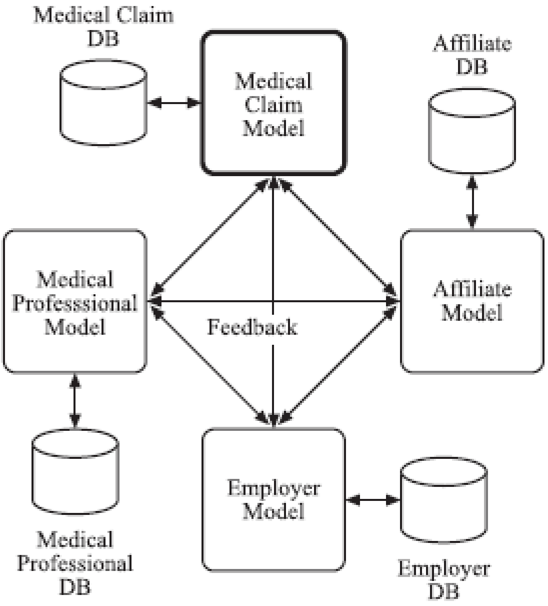


Figure II.8 Fraud Detection Scheme By Using Sub-models With Feedback Connections [44].

The claims whose fraud probability is under that point are not controlled. Determining the threshold point is a critical issue. Because if system determines high point, too many fraud can be overlooked. Also if system determines lower than it must be, redundant effort may expand. System estimated threshold value by using receiver operating characteristics (ROC) curves and iso-performans line.ROC curve plots the true positive (TP) ratio against the false positive (FP) ratio at different threshold value. True positive is a unit of correct fraud classification and false positive is a unit of false fraud classifications. Iso-performance line is a line on the TP-FP plane whose slope is defined by an equation that shown in **Formula 2.1**. p(N) and p(P) are the prior probabilities of obtaining a negative and a positive example respectively, and FN is the false negative rate. The C(FP) and C(FN) represent the costs of a FP and a FN error, respectively.

$$\frac{TP_2 - TP_1}{FP_2 - FP_1} = \frac{p(N)C(FP)}{p(P)C(FN)}, \qquad (2.1)$$

The intersection of iso-performance and ROC curve gives the optimum threshold point. **Figure II.9** is an example of TP – FP plane [40].

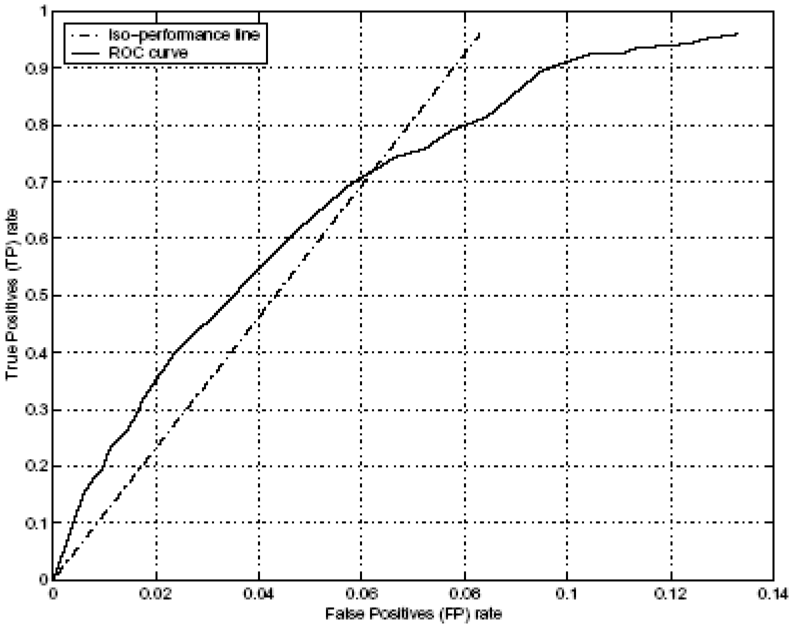
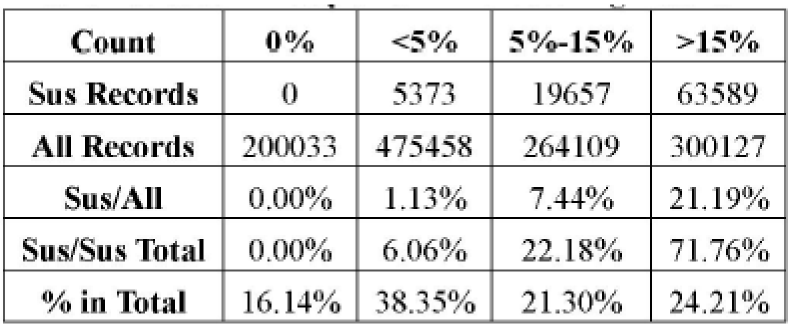


Figure II.9 Example of TP-FP Plane [40].

So fraud detection system based on neural network takes all claims and calculates their fraud probability. After this operation experts examine separated claims. Peng et al. [45] proposed a fraud detection system based on clustering algorithms. They intend to group the huge database. This system separates the data by their characteristics. All data which are same cluster are similar and which are different clusters are completely different. After the clustering data, system determines clusters' fraud data ratio. This knowledge is too helpful to detect fraud and abuse. Fraud detection based on clustering algorithms use information about affiliates, medical professionals, and employers like age, sex, name of affiliate, identification of medical professional, type of claim, and identification of employer. System based on clustering methods examines data types of variables in database. They are important because they are important for cluster methods. This technique determines irrelevant, redundant and correlated data. Irrelevant data for example identification numbers which have no effect on fraud detection eliminated. Also redundant data like city and state address which are getting from other variables and correlated which affect the result are eliminated. This technique transforms the non-numerical data into numerical type because clustering methods gets numerical data.

This technique intends to cluster the data than calculate the ratio of fraud records to the number of all records in these clusters. After all process it gives useful information to the medical experts. This system does not have clear knowledge about dataset so it can not define the class that will form. Consequently this system uses clustering algorithms. Because they give good information about data distribution and they are useful if you do not have class labels. Clustering algorithms get the number of cluster that will be formed from the user. But system cannot determine how many clusters will be formed. Therefore system runs the clustering methods many times with different input of cluster numbers. After that, experts study on the results and determine which cluster number is desirable. To determine the ratio of fraud claims at formed cluster, system takes dataset whose fraud elements are previously known from insurance company. System constitutes clusters on this data with calculated cluster number. After that system calculates fraud ratios in all clusters. These values are very important for detecting fraudulent data. Experts determine which clusters and types of data they have to concentrate on. They omit some clusters and data groups with the help of this knowledge. **Table II.5** shows an example of results that fraud system based on clustering algorithms gives [45]. Second , third and fourth columns represent the clusters whose ratio of having fraud data are zero , less than five , between five and fifteen and bigger than fifteen. With this result experts determine which clusters they have to control and which clusters they can omit.

Table II.5 Sample Results of a Fraud Detection Method Based On Clustering Algorithms [45]



Count	0%	<5%	5%-15%	>15%
Sus Records	0	5373	19657	63589
All Records	200033	475458	264109	300127
Sus/All	0.00%	1.13%	7.44%	21.19%
Sus/Sus Total	0.00%	6.06%	22.18%	71.76%
% in Total	16.14%	38.35%	21.30%	24.21%

CHAPTER III

THE STUDY

III.1 PROBLEM DEFINITION

Health insurance is an important sector in all countries. There are too many affiliates, employers, employees and medical professionals in this sector. Consequently, big amount of money pass through hands. Also there are a lot of fraud types in this sector. Affiliates, employees and professionals derive too much improper personal benefit. This loss of money has bad effects on the national economy. It increases the insurance premiums, the health care costs and the taxes. Insurance companies attend to detect and prevent misuses. Most of insurance companies try to detect frauds with the help of traditional method. In this method, companies nominate medical experts to detect fraud. They take lots of medical transaction groups from insurance company’s operational system. These transaction groups are chosen randomly.   Medical experts investigate these transaction groups and try to find fraudulent events. This technique causes great waste of time and money. Insurance companies have to nominate too many medical experts because every day thousands of medical events are entered into operational systems. Because of the necessity of choosing medical transaction groups randomly, medical experts needlessly investigate too many valid transaction groups. This takes too long time. Besides, medical experts investigate too many transactions and they omit big portion of fraud events because of the disparity of interest. Medical experts find fraud and abuse events with the help of their experiences. They master a lot of fraudulent events and types in medical sector. When they investigate medical event transaction group, they categorize transaction in fraud type which they thought that the transaction is close similar. But medical experts cannot master every type of fraud and abuse. This causes lots of fraud and abuse not to be noticed. Because of these handicaps, medical insurance sector needs a more complicated system which helps medical expert to detect fraud and abuse from large operational systems and databases.

The purpose of this thesis is to create a complicated, feasible fraud detection system. It will analyze whole operational system data and allocate some of them. It will give the opportunity to detect the actors who are in fraudulent activity and it will give some other types of result to the system user. This system will not detect the fraud events; it will detach the medical events which are suspicious. Medical experts take the results of the system and investigate these medical events. Then they will adjudge that those events are fraudulent or not. This system is not a decision unit; it is only decision support system.

III.2 ORACLE DATABASE 11G

Oracle 11g is the latest version of the Oracle database. It was designed for grid computing. Oracle Corporation started beta testing Oracle database 11g in September 2006 and announced the new release on 11 July 2007. It works for the management of enterprise information. Oracle Database 11g is designed to be deployed on everything from small blade servers to the biggest SMP servers and clusters of all sizes. Oracle Database 11g can be used to power transaction processing, data warehousing, and content management applications because it can manage data from traditional business information to XML and 3D spatial information. Oracle added new features in the Oracle Database 11g version. Some of them are; new data types like binary xml type, DICOM medical images, virtual columns in tables, new pivot and unpivot operations, compound triggers, new continue statement, dynamic SQL enhancements, support for rolling upgrades, partition advisor, new composite partition types like Range/Range, List/Range, List/Hash, and List/List, partitioning by virtual columns, ability to apply many patches on-line without downtime, support for case sensitive and multi-byte passwords, encrypt backups, incremental backup on physical readable physical standby, secure files, online application upgrades, and "duality" between SQL and XML.

III.3 ORACLE DATA MINING

Oracle Data Mining is scalable, powerful, data mining software built into the Oracle Database. Traditional approaches for data mining include:

a) Client-server tools that operate on data extracted from the corporate database into local client file systems, or

b) Web-integrated tools that mine data accessed across loosely coupled systems, or

c) Mining engines that are packaged with, but not fully integrated into the database.

These products involve data transfer from the database (either the mission-critical OLTP system or a data warehouse) to the analytics engine, possibly with several data transformation engines/steps intervening between the two entities. They also involve subsequent transfer of results (models, summarizations) to the production system for deployment (i.e. for scoring new data and/or analysis). Besides the obvious cost of managing such multiple, dedicated islands of data, the lack of consolidation and data movement between these islands renders the overall system inefficient at providing a holistic view of enterprise data, and a system that is inherently insecure.

A fundamental strength of Oracle Data Mining is that its mining server is a cohesive, integral component of the RDBMS platform, designed to leverage and extend all the powerful capabilities of the database engine for scalability, security, performance, and availability over incremental releases of the product. All analytical functions and powerful mining algorithms like Support Vector Machines, Association Rules, and Decision Trees operate directly on the data stored in the database and are implemented based on the database kernel primitives. The resulting mining models are stored and managed in the database, and they can be used to score new data and/or retrieve summarizations and analysis through industry-standard SQL and widely used Oracle interfaces and tools. The same Oracle11g platform offers complementary technologies for BI such as relational and multi-dimensional OLAP, analytic/aggregate/window/forecasting functions, data partitioning, parallelism, materialized views, ETL (extract-translate-load), bit-mapped indexes, and star schemas. The Oracle Application Server 11g supports BI Reporting, development tools and Portals. Along with Oracle’s traditional OLTP strengths, all these parts combined to enable consolidation of mission-critical data into one common, secure, scalable Oracle11g BI platform.

ODM is available as an option to the Oracle Database 11g Enterprise Edition. It consists of Data Miner – a GUI wizard for business analysts; Native Oracle SQL Prediction functions; PL/SQL packages for building and deploying models; and a standards-based Java API for J2SE/J2EE-based application development. ODM is integrated into Oracle BI tools such as Discoverer and third-party BI tools from SPSS™, SAP™, Inforsense™ and others.

III.4 DATA PREPROCESSING

Today's real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogeneous sources. Companies have valuable data lying around throughout their networks that needs to be moved from one place to another. The only problem is that the data lies in different formats in various resources. The data can come from any source i.e., a mainframe application, an ERP application, a CRM tool, a flat file, and an Excel spreadsheet even a message queue. All these types of data must be transformed into a single suitable format and stored in large repository. Extract Transform Load (ETL) process is a type of data preprocessing. ETL process is used for feeding data into the data warehouse system. The process gets its source data from OLTP and other input systems as well as data warehouses. Hence, it is important to focus on ETL in a data mining applications that uses data warehouse as the input system. The next section will elaborate the ETL.

III.4.1 EXTRACT TRANSFORM LOAD

The goal of extract, transform and load process is to provide a single, authoritative source for data that support decision making. Ideally, this data layer is detailed, historical, normalized, comprehensive, timely and quality controlled.

1) Detailed: The data are detailed (rather than summarized) providing maximum flexibility for various user communities to structure the data to best suit our needs.
2) Historical: The data are periodic to provide a historical perspective.

3) Normalized: The data are fully normalized. Normalized data provide greater integrity and flexibility of use the denormalized data. Denormalization is not necessary to improve performance because reconciled data are usually accessed periodically using batch processes.
4) Comprehensive: Reconciled data reflect an enterprise wide perspective whose design conforms to the enterprise data model.
5) Timely: Data need not be real time, however data must be current enough so that decision making can react in time.
6) Quality Controlled: Reconciled data must be of unquestioned quality and integrity because they are summarized into the data marts and used for decision making.

These characteristics of reconciled data are quite different from the typical operational data from which they are derived. Operational data are typically detailed, but they differ strongly in the other four dimensions. Operational data are transient rather than historical. Operational data are not normalized. Depending on their roots operational data may never have been normalized or may have been denormalized for performance reasons. Rather than being comprehensive, operational data are generally restricted in scope to a particular application. Operational data are often of poor quality, with numerous types of inconsistencies and errors. The data reconciliation process is responsible for transforming operational data to reconciled data. Because of the sharp differences between these two types of data, data reconciliation is a difficult technically challenge. Data reconciliation occurs in two stages during the process of filling our system. First stage is an initial load and second stage is subsequent updates, which is normally performed on a periodic basis, to keep the system current and to expand our system. Data reconciliation can be visualized as a process, shown in **Figure III.1**, consisting of three steps which are extract, transform and load [46].

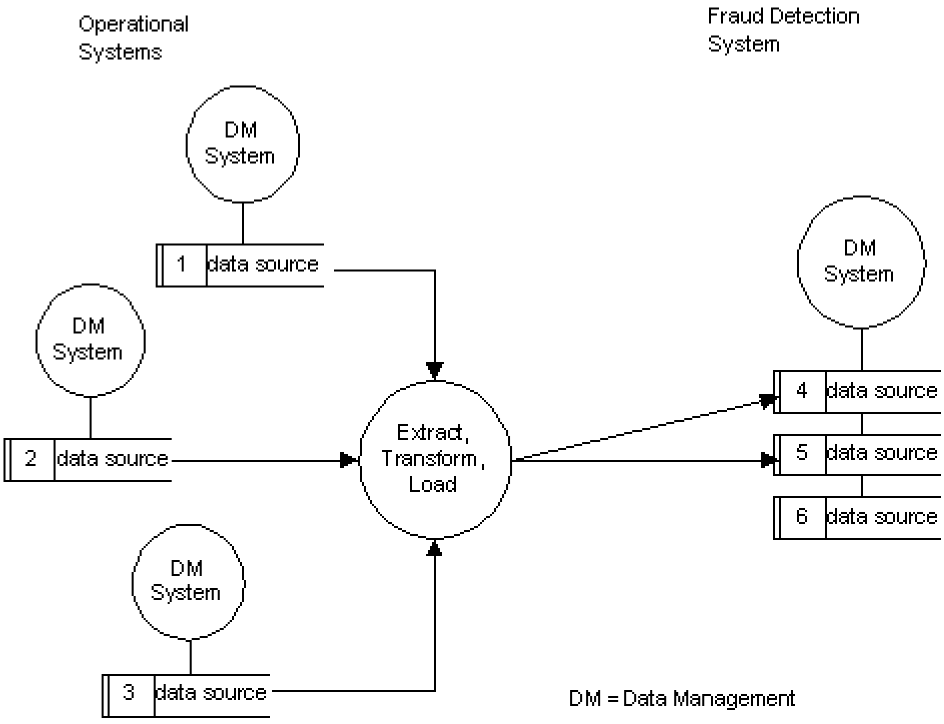


Figure III.1 Data Reconciliation [46]

III.4.1.1 EXTRACTION

Capturing the relevant data from the source files and databases used the fill the system is typically called extracting. Usually, not all of the data contained in the various operational systems are required, but just a subset. Extracting the subset of data is based on an extensive analysis of both the source and target systems, which is best performed by a team directed by data administration and composed of both end users.

The two generic types of data extracts are static extract and incremental extract. Static extract is used to fill the system initially, and incremental extract is used for ongoing system maintenance. It is a method of capturing a snapshot of the required source data at a point in time. The view of the source data is independent of the time at which it was created.

Incremental extract captures only the changes that have occurred in the source data since the last capture. The most common method is log capture. Recall that the database log contains after images that record the most recent changes to database records. With log capture, only after images that are logged after the last capture are selected from the log.

One of the steps that have to be done carefully is qualifying which system's or other data sources' record to use for extraction into the staging area. A major criterion is the quality of the data in the source systems. Quality depends on clarity of data naming, so the system designer know exactly what data exist in a source system, completeness and accuracy of business rules enforced by a source system, which directly affects the accuracy of data; also, the business rule in the source should match the rules to be used in the system, format of data which common formats across sources help to match related data.

Cleanse is generally accepted practice that the role of the ETL process is to identify erroneous data, not fix them. Fixes should be made in the appropriate source systems, so such erroneous data, created by systematic procedural mistakes, do not reoccur. Rejected data are eliminated from further ETL steps and will be reprocessed in the next feed from the relevant source system.

The type of data cleansing required depends on the quality of data in the source system. Besides fixing the types of the problems identified earlier, other common cleansing task include decoding the data to make them understandable , reformatting and changing data types and performing other functions to put data from each source into the target system format ready for transformation, adding timestamps to distinguish values for the same attribute over time, converting between different units of measure, generating primary keys for each row of a table, matching and merging separate extractions into one table or file and matching data to go into the same row of the generated table, logging errors detected, fixing those errors, and reprocessing corrected data without creating duplicate entries, finding missing data to complete the batch of the data necessary for subsequent loading. The order in which different data sources are processed many matter. For example it may be necessary to process customer data from a sales system before new customer demographic data from an external system can be matched to customers.

Once the data are cleansed in the staging area, data are ready for transformation.

III.4.1.2 TRANSFORM

Data transformation is at the very center of the data reconciliation process. It is the component of the data reconciliation that converts the data from the format of source operational system to the format of the target system. Data transformation accepts data from the data capture component, then maps the data to the format of reconciled data layer, and then passes them to the load and index component. Data transformation may range from a simple change in data format or representation to a highly complex exercise in data integration. In general, the goal of the data scrubbing is to correct errors in data values in the source data, whereas the goal of data transformation is to convert the data format from the source to the target system. It is essential to scrub the data before they are transformed because if there are errors in the data before they are transformed, the errors will remain in the data after transformation. Data transformation encompasses a variety of different functions. These functions may be classified broadly into two categories; record level functions and field level functions. Operating on a set of records, such as a file or table, are the most important record level functions are selection, joining, normalization and aggregation.

Selecting is the process of partitioning data according to predefined criteria. Selection is used to extract the relevant data from the source systems that will be used to fill the target system. In fact, selection is typically a part of the capture function discussed earlier.

Joining combines data from various sources into a single table or view. Joining data is an important function because it is often necessary to consolidate data from various sources.

Normalization is the process of decomposing relations with anomalies to produce smaller, well structured relations. As indicated earlier, source data in operational systems are often denormalized. The data must therefore be normalized as part of data transformation.

Aggregation is the process of transforming data from a detailed level to a summary level. Once we have tables that are ready for loading into the target system we can perform summary calculations (aggregations) and store this summary data to enable quicker running of queries. When creating our dimensional data model it is essential that good paths of aggregation form part of the design of the dimension tables.

A field level function converts data from a given format in a source record to a different format in a target record. There are two types of field level functions: single field and multifield.

A single field transformation converts data from a single source field to a single target field. There are two basic methods for performing a single field transformation: algorithmic and table lookup.

A multifield transformation converts data from one or more source fields to one or more target fields. It also contains two types. Many to one transformation takes data from more than one source field and fill the one target field. One to many transformation takes data from one source field an fill more than one target field.

III.4.1.3 LOAD

The last step in filling the target system is to load the selected data into the target system and to create the necessary index. Two basic modes for loading data to the target system are refresh and update.

Refresh mode is an approach to filling the target system that employs bulk rewriting of the target data at periodic intervals. That is, the target data are written initially to fill target system. Then at periodic intervals the target system is rewritten, replacing the previous contents. This mode has become less popular.

Update mode is an approach in which only changes in the source data are written to the target system. To support the periodic nature of target system data, these new records are usually written to the target system without overwriting or deleting previous records.

Refresh mode is generally used to fill the target system when it is first created. Update mode is then generally used for ongoing maintenance of the target system.

The important point of updating process is determining how frequently to update the target system. One of the most opinions for updating the data is updating the target system data as frequently as is practical. Infrequent updating causes massive loads and users to wait for new data. Near real time loads are necessary for active target system but may be inefficient and unnecessary for most target systems. Daily updates are sufficient for most organizations. However, daily updates make it impossible to react to some changing conditions, such as reprising or changing purchase orders for slow moving items. The industry trend is towards updates several times a day and near real time, and less use of more infrequent refresh intervals, such as monthly.

Loading data into the target system typically means appending new rows to tables in the target system. It may also mean updating existing rows with new data, and it may mean purging identified data from the target system that have become obsolete due to age or that were incorrectly loaded in a prior load operation. Data may be loaded from the staging area into the target system by sql commands, special load utilities provided by the target system or third party renders, custom written routines coded by the target system administrators.

ETL process must be done in proper way because significant operational problems can occur with improperly designed ETL systems. This process can have various hurdles which must be taken care while performing the same, some of the challenges are:

1. The biggest challenge to ETL process is to save time because ETL process is time consuming process so research should be done to improve the speed of ETL.
2. The scalability of an ETL system across the lifetime of its usage needs to be established during analysis. This includes understanding the volumes of data that will have to be processed within Service Level Agreements.
3. The challenge is to select the data elements required to support business users' analysis, exploration and reporting processes. Extraction process has to select only that data which can be used for decision support system.
4. If the Data volume is very large then ETL process took a lot of time to get complete.
5. Once data is extracted and transformed the other challenge is to store data in architecture which support execution of complex query and retrieval of data in less time.
6. Missing and invalid values are also a big challenge to ETL.

III.5 SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships [47]. It is a suite of algorithms that are used for classification algorithms. SVM provides coefficients that are useful in understanding the relationship and patterns in the dataset. The SVM algorithms differ from the other algorithms by their adaptability for diverse types of data.

One class support vector machine is one of the SVM (OCSVM) algorithms. The OCSVM algorithm maps input data into a high dimensional feature space (via a kernel) and iteratively finds the maximal margin hyperplane which best separates the training data from the origin. The OCSVM may be viewed as a regular two-class SVM where all the training data lies in the first class, and the origin is taken as the only member of the second class [48].

Oracle Data Mining’s Support Vector Machines (SVM) algorithm is actually a suite of algorithms, adaptable for use with a variety of problems and data. By swapping one kernel for another, SVM can fit diverse problem spaces. Oracle Data Mining supports two kernels, Linear and Gaussian [49].

Data records with N attributes can be thought of as points in N-dimensional space, and SVM attempts to separate the points into subsets with homogeneous target values; points are separated by hyper planes in the linear case, and in the non-linear case (Gaussian) by non-linear separators. SVM finds the vectors that define the separators giving the widest separation of classes (the “support vectors”). This is easy to picture in the case of N = 2; then the solution defines a straight line (linear) or a curve (non-linear) separating the differing classes of points in the plane. SVM solves regression problems by defining an N-dimensional “tube” around the data points, determining the vectors giving the widest separation. SVM can emulate some traditional methods, such as linear regression and neural nets, but goes far beyond those methods in flexibility, scalability, and speed.

III.6 DATABASE

In this section, design of the database structure used in this thesis will be explained. At the beginning, a short overview describes the logical database structure and then all the table definitions are given.

III.6.1 DATABASE OVERVIEW

This section explains the database structure used in this thesis. Some of the database items are brifly described. There are seven main items of database structure. Two of them are policy and claim which are the products of health insurance companies.The other five items are salesman, agency, person, healthcenter which are the users of the health insurance system and user which is the employees of the health insurance system. **Figure III.2** shows the ER Diagram of the database.

III.6.2 DATABASE STRUCTURE

Database has seven main types of items. Each type of main items has more than one table. Policy, policy type, policy status, product, proposal, commission type, currency tables are under policy item. Claim, claim detail, benefit, benefit type, claim status reason, claim status, claim type tables are under claim item. Salesman, salesman type and salesman team tables are under salesman item. Agency, agency type, tax office and city tables are under agency item. Person, gender, education, marital status, occupation, risk group and person relation type tables are under person item. Health center, health center type, doctor degree, specialty, bank, bank branch, province and district tables are under health center item. User, department and user type tables are under user item.

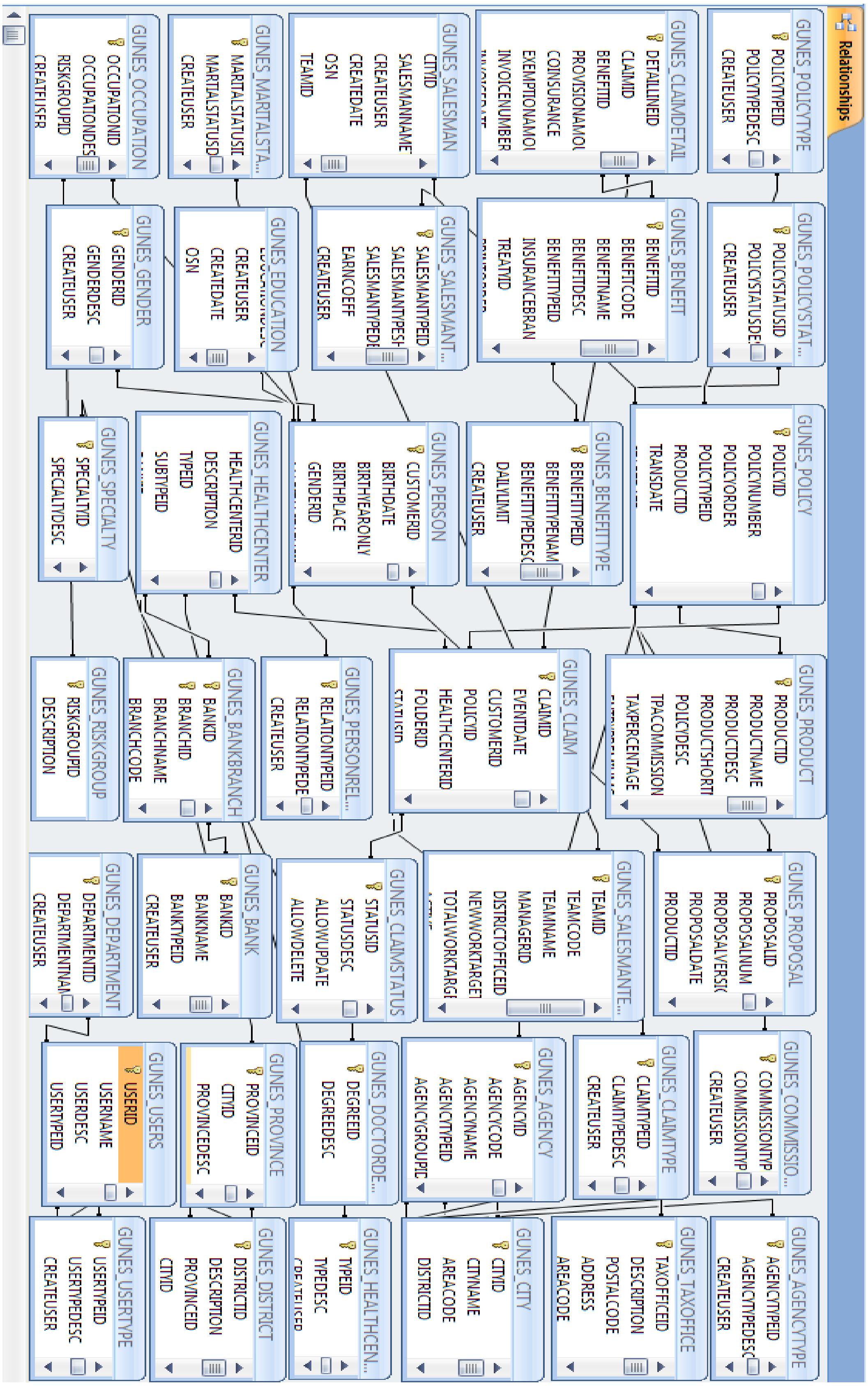
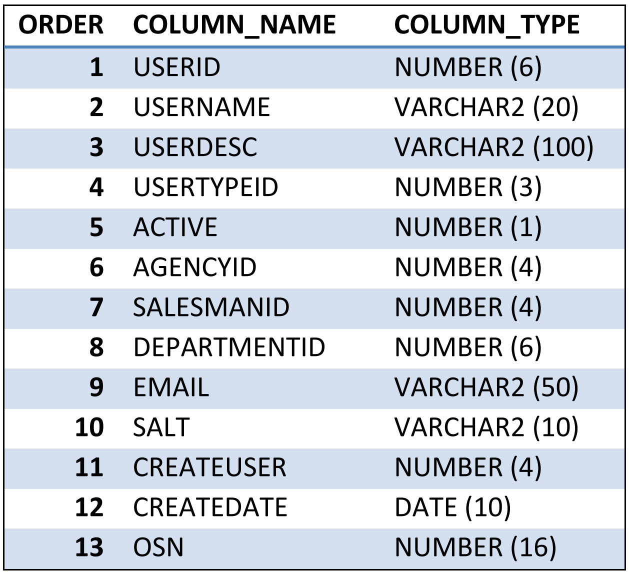


Figure III.2 ER Diagram of the Database

Users table contains the informations of people who are working in insurance companies.This table is important too because some of the fraudulent events are done by insurance companies employers. **Table III.1** shows user table details.

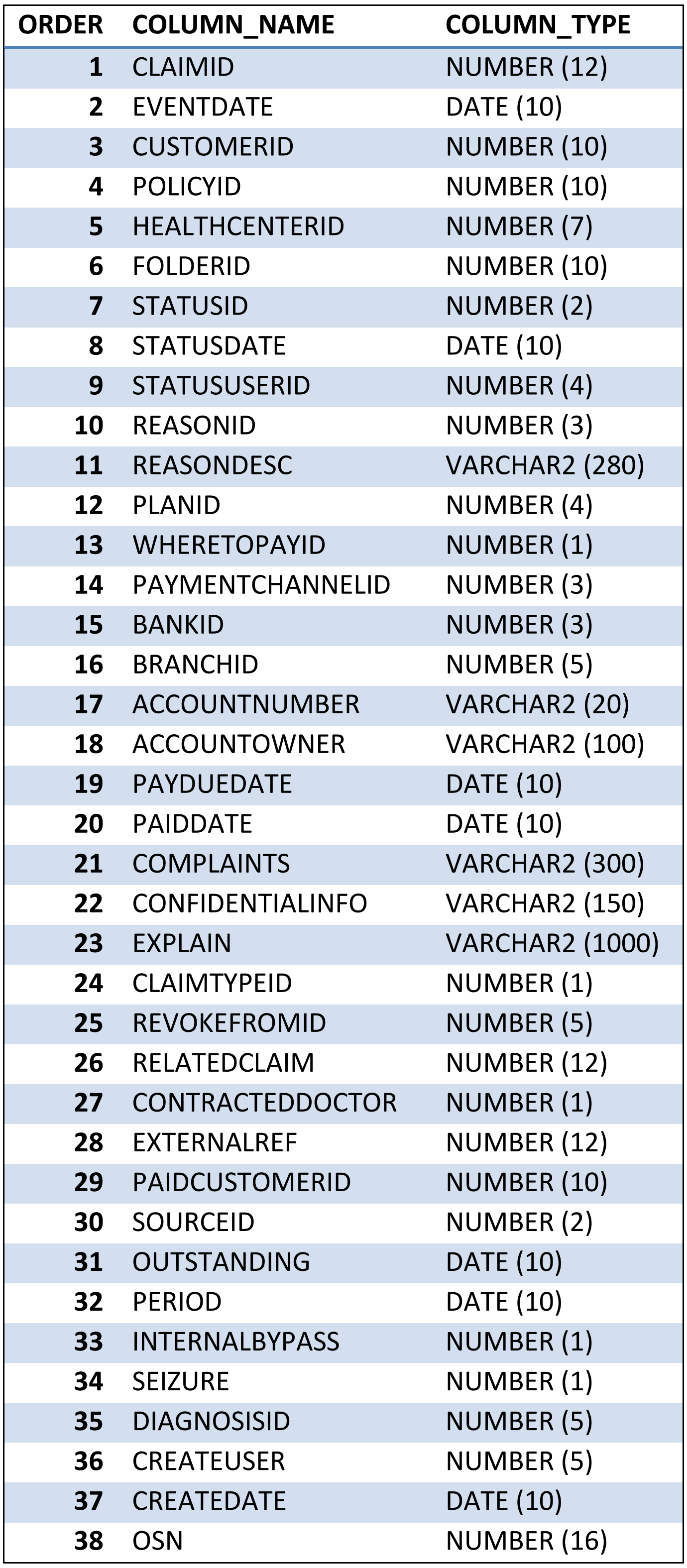
Table III.1 User Table



ORDER	COLUMN_NAME	COLUMN_TYPE
1	USERID	NUMBER (6)
2	USERNAME	VARCHAR2 (20)
3	USERDESC	VARCHAR2 (100)
4	USERTYPEID	NUMBER (3)
5	ACTIVE	NUMBER (1)
6	AGENCYID	NUMBER (4)
7	SALESMANID	NUMBER (4)
8	DEPARTMENTID	NUMBER (6)
9	EMAIL	VARCHAR2 (50)
10	SALT	VARCHAR2 (10)
11	CREATEUSER	NUMBER (4)
12	CREATEDATE	DATE (10)
13	OSN	NUMBER (16)

Claim table contains the events that health insurance company will pay to the insuree. For example if a insured person is treated , one record is kept in this table. Eventdate , policyid, healthcenterid, customerid are kept in this table.

Table III.2 Claim Table



ORDER	COLUMN_NAME	COLUMN_TYPE
1	CLAIMID	NUMBER (12)
2	EVENTDATE	DATE (10)
3	CUSTOMERID	NUMBER (10)
4	POLICYID	NUMBER (10)
5	HEALTHCENTERID	NUMBER (7)
6	FOLDERID	NUMBER (10)
7	STATUSID	NUMBER (2)
8	STATUSDATE	DATE (10)
9	STATUSUSERID	NUMBER (4)
10	REASONID	NUMBER (3)
11	REASONDESC	VARCHAR2 (280)
12	PLANID	NUMBER (4)
13	WHERETOPAYID	NUMBER (1)
14	PAYMENTCHANNELID	NUMBER (3)
15	BANKID	NUMBER (3)
16	BRANCHID	NUMBER (5)
17	ACCOUNTNUMBER	VARCHAR2 (20)
18	ACCOUNTOWNER	VARCHAR2 (100)
19	PAYDUEDATE	DATE (10)
20	PAIDDATE	DATE (10)
21	COMPLAINTS	VARCHAR2 (300)
22	CONFIDENTIALINFO	VARCHAR2 (150)
23	EXPLAIN	VARCHAR2 (1000)
24	CLAIMTYPEID	NUMBER (1)
25	REVOKEFROMID	NUMBER (5)
26	RELATEDCLAIM	NUMBER (12)
27	CONTRACTEDDOCTOR	NUMBER (1)
28	EXTERNALREF	NUMBER (12)
29	PAIDCUSTOMERID	NUMBER (10)
30	SOURCEID	NUMBER (2)
31	OUTSTANDING	DATE (10)
32	PERIOD	DATE (10)
33	INTERNALBYPASS	NUMBER (1)
34	SEIZURE	NUMBER (1)
35	DIAGNOSISID	NUMBER (5)
36	CREATEUSER	NUMBER (5)
37	CREATEDATE	DATE (10)
38	OSN	NUMBER (16)

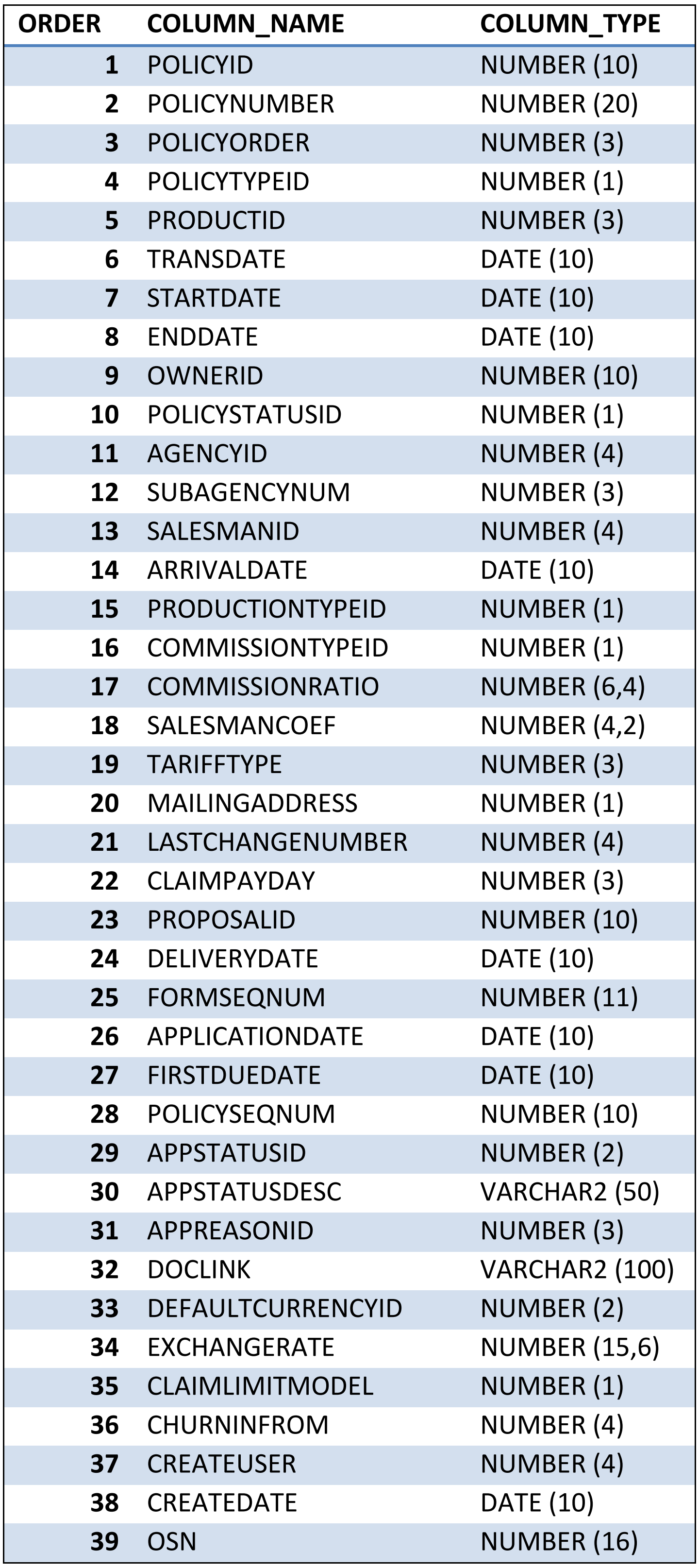
Table III.2 shows claim table details. ClaimDetail table contains the events detail. For example, when the person visited the hospital , how the doctor treated him/her and what additional reports (like tomography) the doctor requested. At some stage the patient may buy drugs from pharmacy. These three events are kept in this table. **Table III.**3 shows claim detail table details.

Table III.3 Claim Detail Table

ORDER	COLUMN_NAME	COLUMN_TYPE
1	DETAILLINEID	NUMBER (12)
2	CLAIMID	NUMBER (12)
3	BENEFITID	NUMBER (4)
4	PROVISIONAMOUNT	NUMBER (15,6)
5	COINSURANCE	NUMBER (4,2)
6	EXEMPTIONAMOUNT	NUMBER (15,6)
7	INVOICENUMBER	NUMBER (15)
8	INVOICEDATE	DATE (10)
9	INVOICEAMOUNT	NUMBER (15,6)
10	LIMITVALUE	NUMBER (15,6)
11	INVOICEEXCLUSION	NUMBER (15,6)
12	PAYABLEAMOUNT	NUMBER (15,6)
13	PAIDAMOUNT	NUMBER (15,6)
14	NETAMOUNT	NUMBER (15,6)
15	VATRATE	NUMBER (2)
16	DAYCOUNT	NUMBER (3)
17	ANNUALLIMITGROUP	NUMBER (2)
18	STARTDATE	DATE (10)
19	FINISHDATE	DATE (10)
20	CREATEUSER	NUMBER (4)
21	CREATEDATE	DATE (10)
22	OSN	NUMBER (16)

Policy table contains the policy details. ProductId, policyStartDate, policyEndDate, policyStatus, salesmanId, agencyId are important columns in this table. **Table III.4** shows the policy table details.

Table III.4 Policy Table



ORDER	COLUMN_NAME	COLUMN_TYPE
1	POLICYID	NUMBER (10)
2	POLICYNUMBER	NUMBER (20)
3	POLICYORDER	NUMBER (3)
4	POLICYTYPEID	NUMBER (1)
5	PRODUCTID	NUMBER (3)
6	TRANSDATE	DATE (10)
7	STARTDATE	DATE (10)
8	ENDDATE	DATE (10)
9	OWNERID	NUMBER (10)
10	POLICYSTATUSID	NUMBER (1)
11	AGENCYID	NUMBER (4)
12	SUBAGENCYNUM	NUMBER (3)
13	SALESMANID	NUMBER (4)
14	ARRIVALDATE	DATE (10)
15	PRODUCTIONTYPEID	NUMBER (1)
16	COMMISSIONTYPEID	NUMBER (1)
17	COMMISSIONRATIO	NUMBER (6,4)
18	SALESMANCOEF	NUMBER (4,2)
19	TARIFFTYPE	NUMBER (3)
20	MAILINGADDRESS	NUMBER (1)
21	LASTCHANGENUMBER	NUMBER (4)
22	CLAIMPAYDAY	NUMBER (3)
23	PROPOSALID	NUMBER (10)
24	DELIVERYDATE	DATE (10)
25	FORMSEQNUM	NUMBER (11)
26	APPLICATIONDATE	DATE (10)
27	FIRSTDUEDATE	DATE (10)
28	POLICYSEQNUM	NUMBER (10)
29	APPSTATUSID	NUMBER (2)
30	APPSTATUSDESC	VARCHAR2 (50)
31	APPREASONID	NUMBER (3)
32	DOCLINK	VARCHAR2 (100)
33	DEFAULTCURRENCYID	NUMBER (2)
34	EXCHANGERATE	NUMBER (15,6)
35	CLAIMLIMITMODEL	NUMBER (1)
36	CHURNINFROM	NUMBER (4)
37	CREATEUSER	NUMBER (4)
38	CREATEDATE	DATE (10)
39	OSN	NUMBER (16)

Agency table contains agencies which find customers and sell insurance companies policies to them. This table is important for us because we will try to determine agencies whose policies are fraudulent. **Table III.5** shows the agency table details.

Table III.5 Agency Table

ORDER	COLUMN_NAME	COLUMN_TYPE
1	AGENCYID	NUMBER (4)
2	AGENCYCODE	VARCHAR2 (10)
3	AGENCYNAME	VARCHAR2 (50)
4	AGENCYTYPEID	NUMBER (2)
5	AGENCYGROUPID	NUMBER (4)
6	UPPERAGENCYID	NUMBER (4)
7	STARTDATE	DATE (10)
8	ENDDATE	DATE (10)
9	STATUSID	NUMBER (1)
10	TAXOFFICEID	NUMBER (4)
11	TAXNUMBER	NUMBER (10)
12	TAXOFFICE	VARCHAR2 (30)
13	DEPO	NUMBER (1)
14	DEPOSIT	NUMBER (15,6)
15	DEPOSITTYPE	NUMBER (1)
16	DISTRICTOFFICE	NUMBER (3)
17	PHONE	VARCHAR2 (15)
18	FAX	VARCHAR2 (15)
19	EMAIL	VARCHAR2 (50)
20	URL	VARCHAR2 (50)
21	ADDRESS	VARCHAR2 (120)
22	CITYID	NUMBER (3)
23	AGENCYNAMETOKEN	VARCHAR2 (50)
24	CREATEUSER	NUMBER (4)
25	CREATEDATE	DATE (10)
26	OSN	NUMBER (16)

Salesman table contains the people who buys policy from the insurance company. If a person insures another person, insured persons informations are kept here too. **Table III.6** shows salesman table details.

Table III.6 Salesman Table

ORDER	COLUMN_NAME	COLUMN_TYPE
1	SALESMANID	NUMBER (4)
2	SALESMANCODE	NUMBER (8)
3	SALESMANNAME	VARCHAR2 (50)
4	SALESMANTYPEID	NUMBER (2)
5	MANAGERID	NUMBER (4)
6	OFFICEID	NUMBER (3)
7	AGENCYID	NUMBER (4)
8	STARTDATE	DATE (10)
9	ENDDATE	DATE (10)
10	STATUSID	NUMBER (1)
11	PHONE	VARCHAR2 (15)
12	FAX	VARCHAR2 (15)
13	ADDRESS	VARCHAR2 (120)
14	CITYID	NUMBER (3)
15	TEAMID	NUMBER (5)
16	CREATEUSER	NUMBER (4)
17	CREATEDATE	DATE (10)
18	OSN	NUMBER (16)

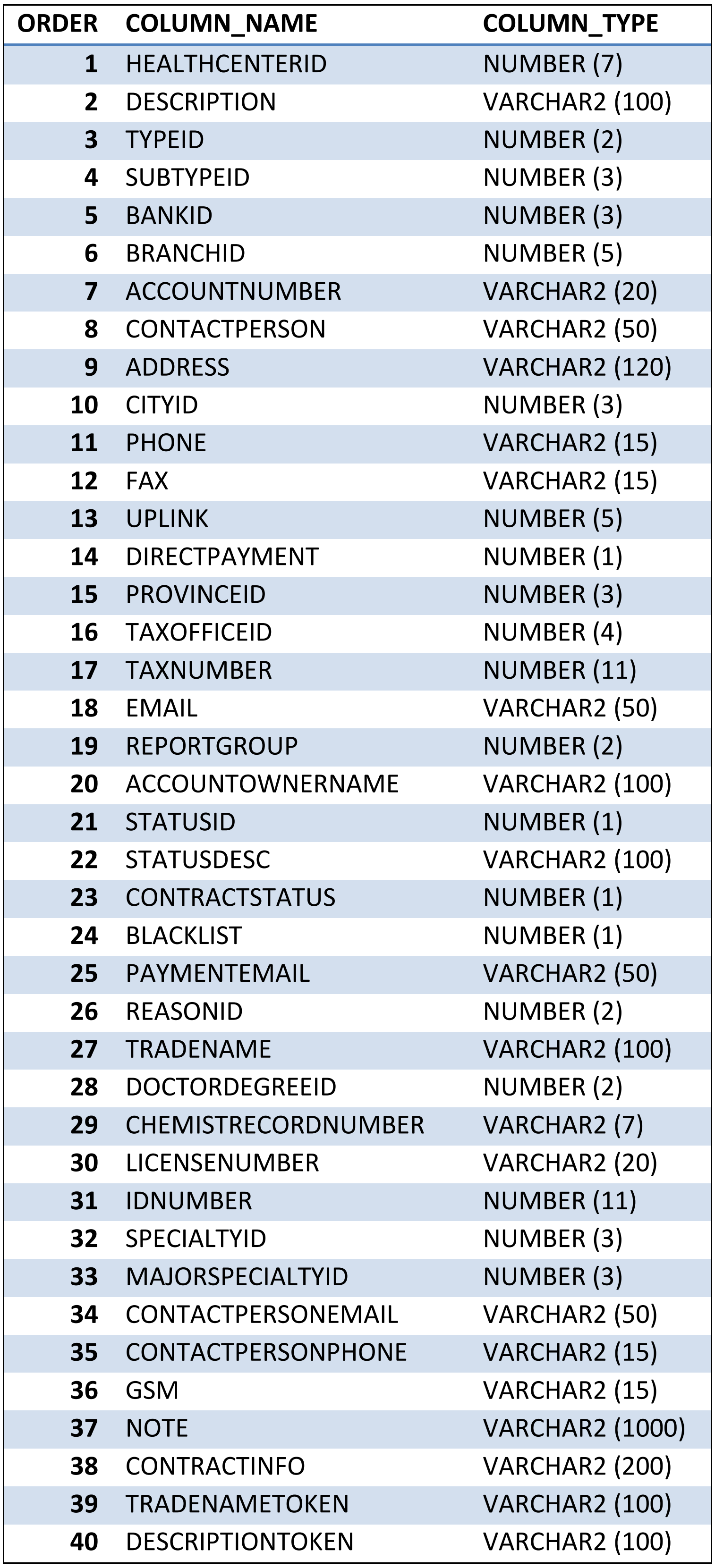
Person table contains all people in the health insurance sector. **Table III.7** shows person table details.

HealthCenter table contains all properties of healtcenters. This table is very important for us because we will use it for following where fraudulent event occurs. **Table III.8** shows health center table details.

Table III.7 Person Table

ORDER	COLUMN_NAME	COLUMN_TYPE
1	CUSTOMERID	NUMBER (10)
2	BIRTHDATE	DATE (10)
3	BIRTHYEARONLY	NUMBER (1)
4	BIRTHPLACE	VARCHAR2 (30)
5	GENDERID	NUMBER (1)
6	MARITALSTATUSID	NUMBER (1)
7	RELATIONTYPEID	NUMBER (1)
8	EDUCATIONID	NUMBER (2)
9	IDNUMBER	NUMBER (11)
10	SOCSECNUMBER	NUMBER (16)
11	SOCSECINSTID	NUMBER (1)
12	FATHERNAME	VARCHAR2 (45)
13	MOTHERNAME	VARCHAR2 (45)
14	MAIDENNAME	VARCHAR2 (25)
15	OCCUPATIONID	NUMBER (3)
16	EXTERNALREF	VARCHAR2 (20)
17	EXTERNALID	NUMBER (16)
18	EXTERNALGROUPID	NUMBER (10)
19	BLACKLIST	NUMBER (1)
20	BLACKLISTDESC	VARCHAR2 (50)
21	BLACKLISTUSERID	NUMBER (4)
22	BLACKLISTDATE	DATE (10)
23	VIP	NUMBER (1)
24	VIPDESC	VARCHAR2 (50)
25	VIPUSERID	NUMBER (4)
26	VIPDATE	DATE (10)
27	LIFETIMERENEWAL	NUMBER (1)
28	LTNDESC	VARCHAR2 (80)
29	LTNUSERID	NUMBER (4)
30	LTNDATE	DATE (10)
31	FIRSTINSUREDDATE	DATE (10)
32	ACQUIREDRIGHT	VARCHAR2 (200)
33	CARDNO	NUMBER (16)
34	PASSPORTNUMBER	VARCHAR2 (20)
35	CREATEUSER	NUMBER (4)
36	CREATEDATE	DATE (10)
37	OSN	NUMBER (16)

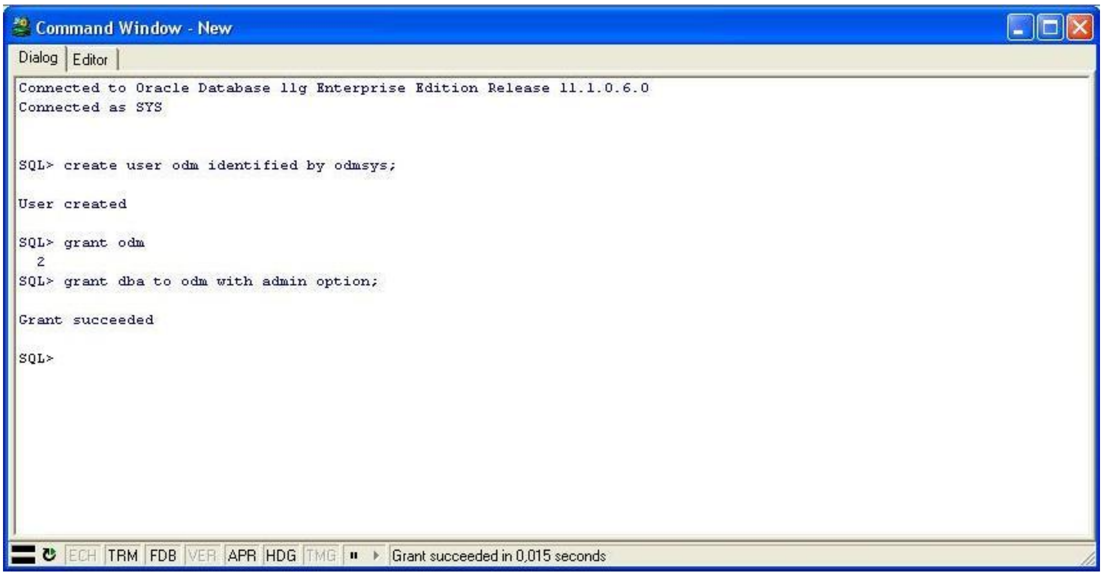
Table III.8 Health Center Table



ORDER	COLUMN_NAME	COLUMN_TYPE
1	HEALTHCENTERID	NUMBER (7)
2	DESCRIPTION	VARCHAR2 (100)
3	TYPEID	NUMBER (2)
4	SUBTYPEID	NUMBER (3)
5	BANKID	NUMBER (3)
6	BRANCHID	NUMBER (5)
7	ACCOUNTNUMBER	VARCHAR2 (20)
8	CONTACTPERSON	VARCHAR2 (50)
9	ADDRESS	VARCHAR2 (120)
10	CITYID	NUMBER (3)
11	PHONE	VARCHAR2 (15)
12	FAX	VARCHAR2 (15)
13	UPLINK	NUMBER (5)
14	DIRECTPAYMENT	NUMBER (1)
15	PROVINCEID	NUMBER (3)
16	TAXOFFICEID	NUMBER (4)
17	TAXNUMBER	NUMBER (11)
18	EMAIL	VARCHAR2 (50)
19	REPORTGROUP	NUMBER (2)
20	ACCOUNTOWNERNAME	VARCHAR2 (100)
21	STATUSID	NUMBER (1)
22	STATUSDESC	VARCHAR2 (100)
23	CONTRACTSTATUS	NUMBER (1)
24	BLACKLIST	NUMBER (1)
25	PAYMENTEMAIL	VARCHAR2 (50)
26	REASONID	NUMBER (2)
27	TRADENAME	VARCHAR2 (100)
28	DOCTORDEGREEID	NUMBER (2)
29	CHEMISTRECORDNUMBER	VARCHAR2 (7)
30	LICENSENUMBER	VARCHAR2 (20)
31	IDNUMBER	NUMBER (11)
32	SPECIALTYID	NUMBER (3)
33	MAJORSPECIALTYID	NUMBER (3)
34	CONTACTPERSONEMAIL	VARCHAR2 (50)
35	CONTACTPERSONPHONE	VARCHAR2 (15)
36	GSM	VARCHAR2 (15)
37	NOTE	VARCHAR2 (1000)
38	CONTRACTINFO	VARCHAR2 (200)
39	TRADENAMETOKEN	VARCHAR2 (100)
40	DESCRIPTIONTOKEN	VARCHAR2 (100)

III.7 DATA MINING APPLICATION

After the installation of Oracle 11G database and Oracle data miner, a user was created for data mining. Then administrator role was given to him. **Figure III.3** shows these activities.

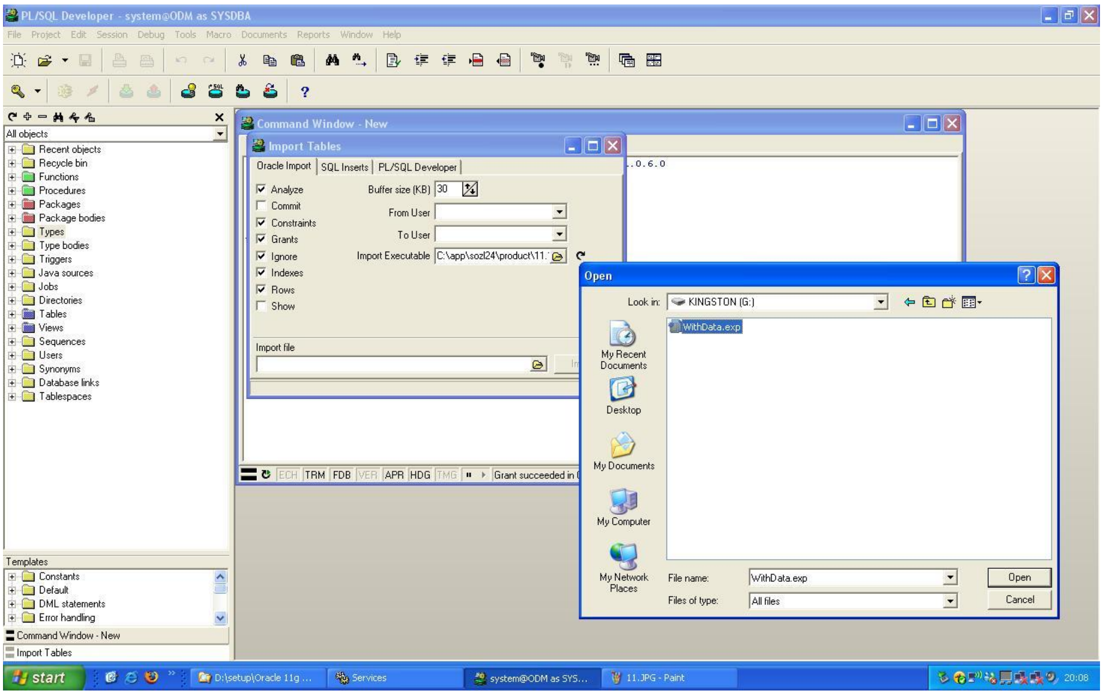


```
Connected to Oracle Database 11g Enterprise Edition Release 11.1.0.6.0
Connected as SYS


SQL> create user odm identified by odmsys;

User created

SQL> grant odm
  2
SQL> grant dba to odm with admin option;

Grant succeeded

SQL>
```

Grant succeeded in 0,015 seconds

Figure III.3 Creating a user

To start the data mining activities, our health insurance data was imported into the new database. To import data into the database, PL/SQL Developer language was used. Data source was selected then tables were imported. **Figure III.4** shows the import data step.



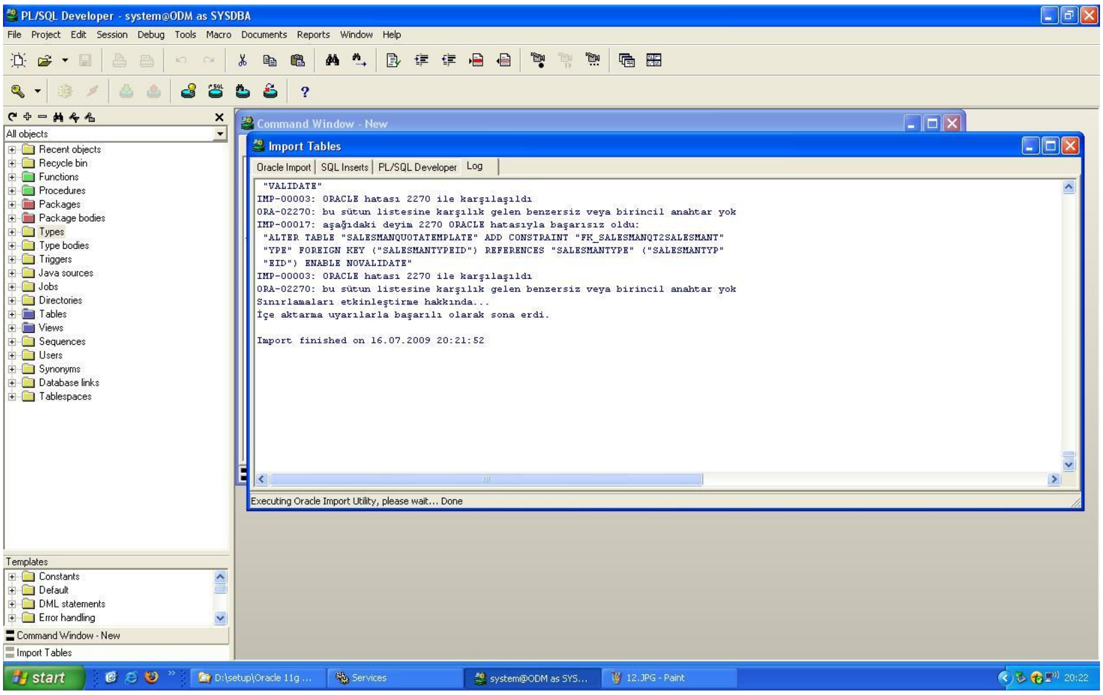
Import Tables — Oracle Import

☑ Analyze
□ Commit
☑ Constraints
☑ Grants
☑ Ignore
☑ Indexes
☑ Rows
□ Show

Buffer size (KB): 30
Import Executable: C:\app\sozl24\product\11.

Open — Look in: KINGSTON (G:) — WithData.exp
File name: WithData.exp
Files of type: All files

Figure III.4 Import Data



```
 "VALIDATE"
IMP-00003: ORACLE hatası 2270 ile karşılaşıldı
ORA-02270: bu sütun listesine karşılık gelen benzersiz veya birincil anahtar yok
IMP-00017: aşağıdaki deyim 2270 ORACLE hatasıyla başarısız oldu:
 "ALTER TABLE "SALESMANQUOTATEMPLATE" ADD CONSTRAINT "FK_SALESMANQT2SALESMANT"
 "YPE" FOREIGN KEY ("SALESMANTYPEID") REFERENCES "SALESMANTYPE" ("SALESMANTYP"
 "EID") ENABLE NOVALIDATE"
IMP-00003: ORACLE hatası 2270 ile karşılaşıldı
ORA-02270: bu sütun listesine karşılık gelen benzersiz veya birincil anahtar yok
Sınırlamaları etkinleştirme hakkında...
İçe aktarma uyarılarla başarılı olarak sona erdi.

Import finished on 16.07.2009 20:21:52
```

Executing Oracle Import Utility, please wait... Done

Figure III.5 Import Data Results

Figure III.5 shows the result of importing data step.

Oracle Corporation recommends a set of basic privileges for data mining. The following GRANT statements grant these privileges to a user named dmuser.

GRANT create mining model TO dmuser;

GRANT create procedure TO dmuser;

GRANT create session TO dmuser;

GRANT create table TO dmuser;

GRANT create sequence TO dmuser;

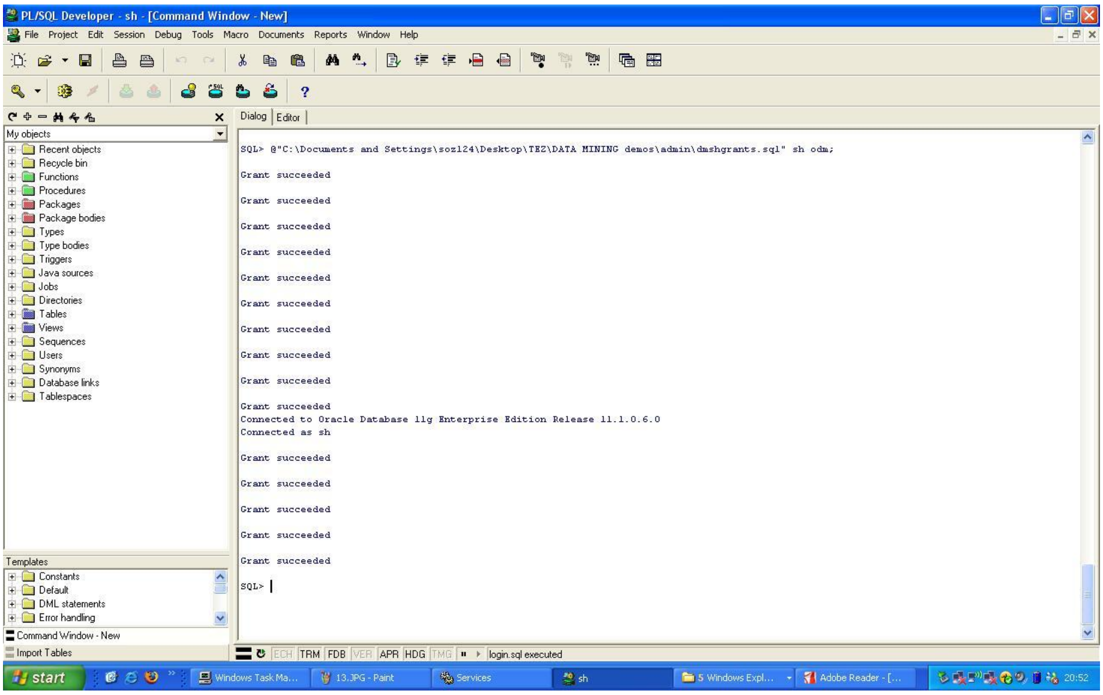
GRANT create view TO dmuser;

GRANT create job TO dmuser;

GRANT create type TO dmuser;

GRANT create synonym TO dmuser;

The CREATE MINING MODEL privilege is required for creating models. **Figure III.6** shows the grant privileges step.

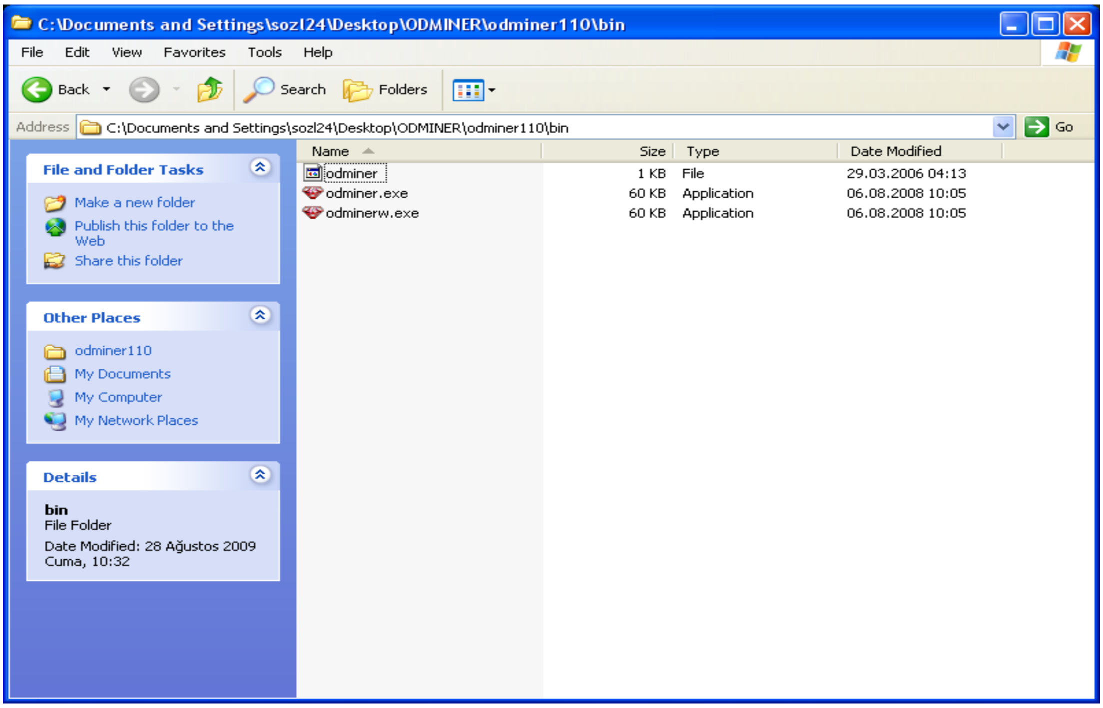


```
SQL> @"C:\Documents and Settings\soz124\Desktop\TEZ\DATA MINING demos\admin\dmshgrants.sql" sh odm;

Grant succeeded

Grant succeeded

Grant succeeded

Grant succeeded

Grant succeeded

Grant succeeded

Grant succeeded

Grant succeeded

Grant succeeded

Grant succeeded
Connected to Oracle Database 11g Enterprise Edition Release 11.1.0.6.0
Connected as sh

Grant succeeded

Grant succeeded

Grant succeeded

Grant succeeded

Grant succeeded

SQL>
```

login.sql executed

Figure III.6 Grant Privileges

To start Oracle Data Miner, odminerw.exe was run to launch Oracle Data Miner in bin directory which is in the Oracle Data Miner installation directory. **Figure III.7** shows Oracle Data Miner Bin directory.



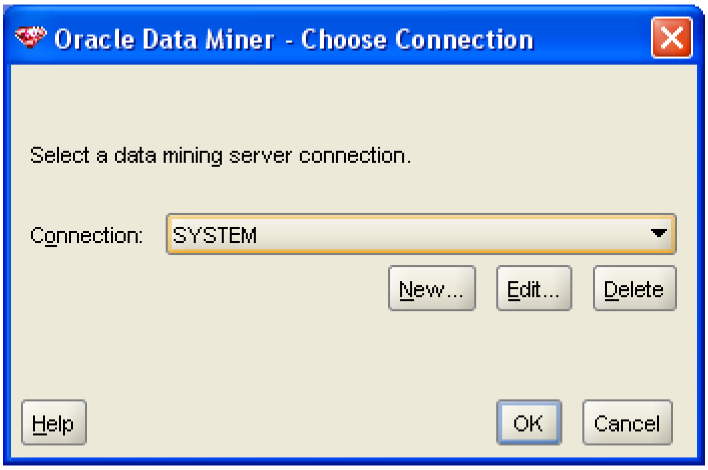
Address: C:\Documents and Settings\sozl24\Desktop\ODMINER\odminer110\bin

Name	Size	Type	Date Modified
odminer	1 KB	File	29.03.2006 04:13
odminer.exe	60 KB	Application	06.08.2008 10:05
odminerw.exe	60 KB	Application	06.08.2008 10:05

bin
File Folder
Date Modified: 28 Ağustos 2009 Cuma, 10:32

Figure III.7 Oracle Data Miner Bin Directory

When Oracle Data Miner was started, and the Choose Connection dialog appeared. To create a new connection, New button was chosen. **Figure III.8** shows the Choose Connection dialog.

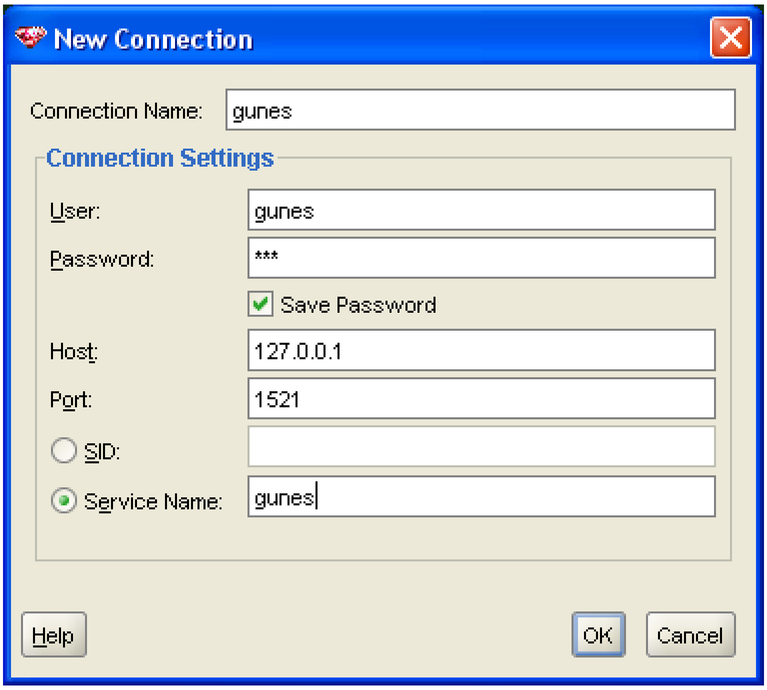


Select a data mining server connection.

Connection: SYSTEM

Figure III.8 Oracle Data Miner Choose Connection Dialog

The New Connection dialog appeared. **Figure III.9** shows new connection dialog.



New Connection

Connection Name: gunes

Connection Settings

User: gunes

Password: ***

☑ Save Password

Host: 127.0.0.1

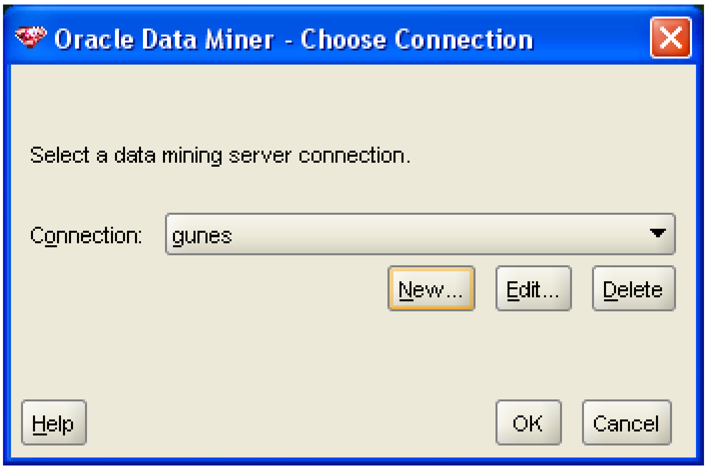
Port: 1521

○ SID:

● Service Name: gunes

Figure III.9 New Connection Dialog

The new connection was created and returned to the Choose Connection dialog. the connection that we created was selected, and Oracle Data Miner is launched using the specified connection. **Figure III.10** shows the connection dialog.



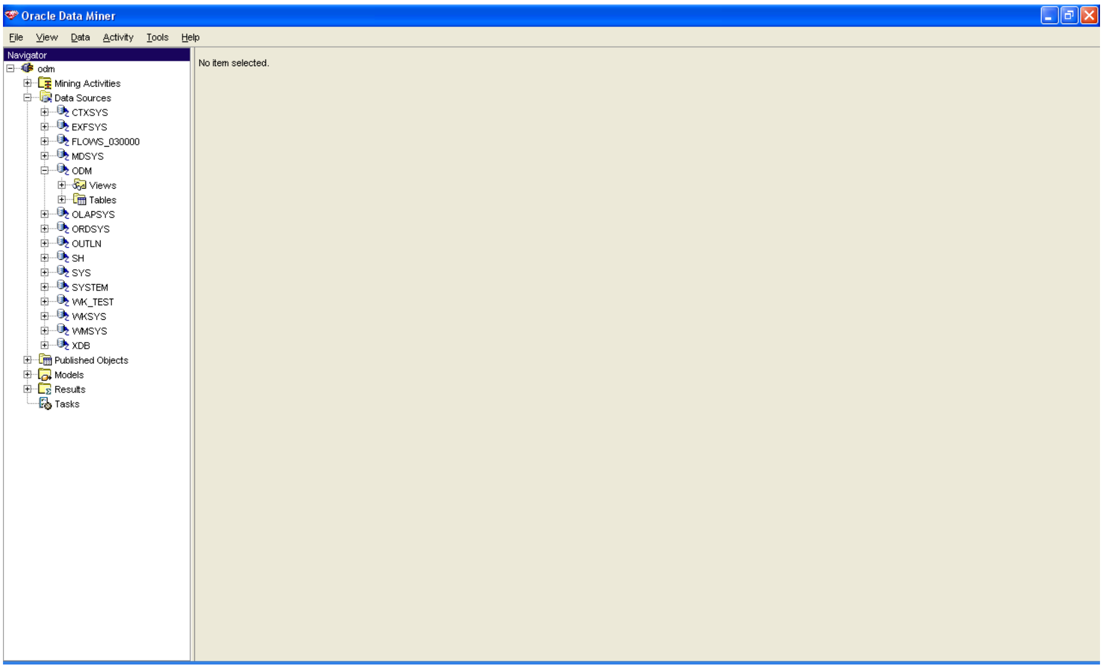
Oracle Data Miner - Choose Connection

Select a data mining server connection.

Connection: gunes

Figure III.10 Oracle Data Miner Choose Connection Dialog

On Oracle Data Miner screen, the database can be seen under Data Source part. **Figure III.11** shows Oracle Data Miner main screen.



No item selected.

Figure III.11 Oracle Data Miner Main Screen

After these steps, the fraud types that were going to be detected were decided. There are too many fraud types in health insurance sector. Some of them can be designated by a simple sql command, some of them can be designated complex sql command and the remainders can be designated with data mining applications.

Three fraud types were selected to analyze and work on it. These fraud types are “claims whose paid amount is greater than its invoice amount that insurance company will pay”, “transactions which involve medicine taking without doctor inspection or surgical operation within four days”, and “health centers whose average incident cost bigger than the average incident cost of all health centers and rate of the payment to directly insured is bigger than the average payment to directly insured rate of all health centers”. The following sections explain these fraud types in detail:

III.7.1 FRAUD TYPE 1

The first selected fraud type is “claims whose paid amount is greater than its invoice amount that insurance company will pay”. Health center prepare invoices for all transactions. According to insured’s policy, health insurance company pays the whole invoice amount or some portion of invoice amount. If the health insurance company pays more than its responsibility, this transaction is suspicious. Claim detail table keeps the claims’ benefit based details data. All transactions in one claim are kept in this table. Whole expense is kept on invoiceAmount column in this table. Coinsurance column keeps the insured's responsible payment ratio. PaidAmount column keeps the value that health insurance paid to the insured. So this suspicious event occurs if paid amount is bigger than multiplication of invoice amount and insurance company's responsibility ratio. **Formula 3.1** shows the formula of fraud detection type; where PA is paid amount, IA is invoice amount, and CI is coinsurance.

$$PA > IA * ((100 - CI) / 100) \tag{3.1}$$

Before building a data mining model, source tables have to be created. Missing values and inconsistent data have bad effects on data mining model and anomaly detection. In this fraud type, the important columns that designate the anomalous case are coinsurance, invoiceAmount and paidAmount. Missing and inconsistent data in these columns cannot be allowed. These data must be scanned out from source datasets. There are 832553 rows in claimDetail table. In this table 115 rows' coinsurance column has null value, 94 rows' invoiceAmount column has null value and 1878 rows' invoiceAmount column has zero value. Also 93 rows of this table have null value in paidAmount column and 26389 rows have zero value in paidAmount column. These missing values and inconsistent data exist because of the errors of operational system of insurance company or the mistakes of the user of the operational system. So source table were created from claim detail table where the paidAmount and invoiceAmount columns' values are not null and bigger than zero, and coinsurance column's value is not null. Also oracle data mining tool requires RISK column which represents suspicious cases. The data has been transformed so that the build data source consists only of records with RISK = 0, representing no risk. Deneme_sourceFull table was created as build source. **Figure III.12** shows the SQL statement for creating the build source table. The number of rows decreased from 832553 to 805207 in deneme_sourceFull table.

```
create or replace view deneme_sourceFull as
select c.*, 0 risk from CLAIMDETAIL c
where abs(nvl(c.invoiceamount,0)) > 0  and  abs(c.paidamount)
> 0 and c.coinsurance is not null
 and abs(c.invoiceamount)*((100-abs(c.coinsurance))/100) >=
abs(c.paidamount)
```

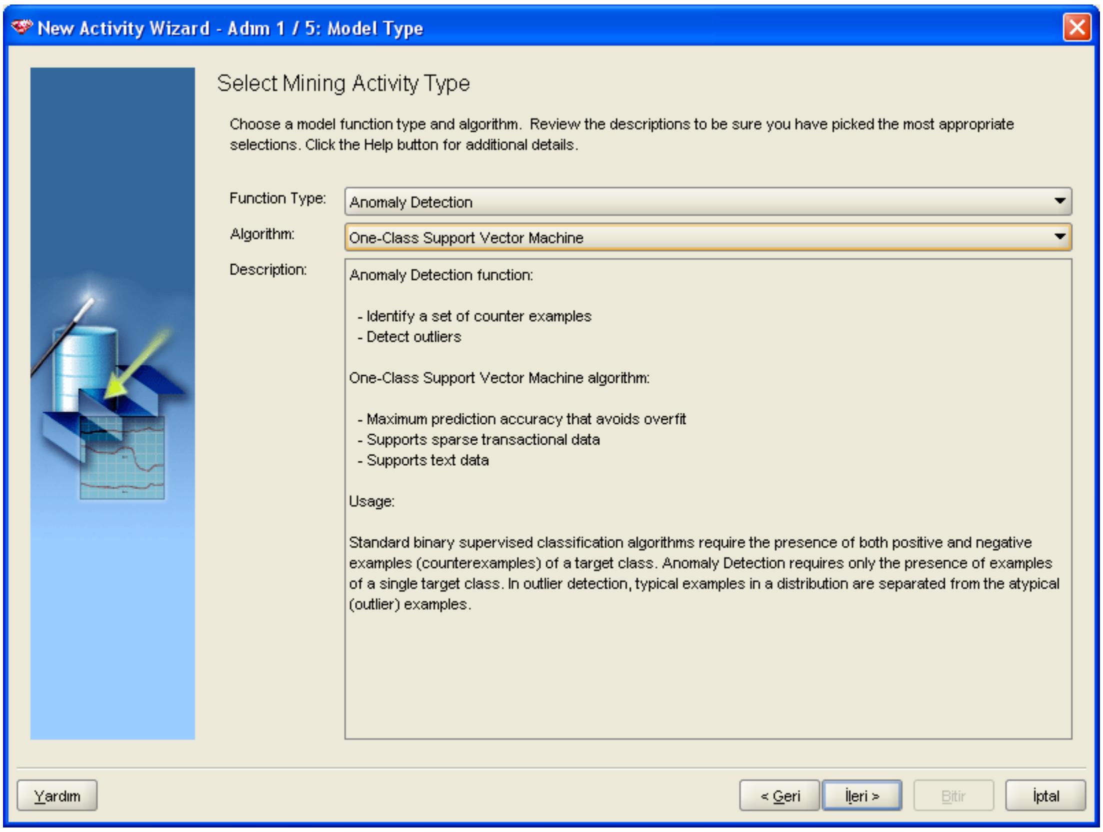
Figure III.12 Activity Build Source Table SQL Statement for Fraud Type 1

Then activity test data source was created. Oracle data mining tool requires a test source that has rarely suspicious data. So the test data source table has a few records with RISK = 1, representing the unusual cases that the Anomaly Detection model will try to find. Deneme_source_FullTest was created as activity source. **Figure III.13** shows the SQL statement for creating the activity apply source table. The number of rows in the deneme_souce_FullTest is 806056. As the number of rows in deneme_sourceFull, source table of build activity, is 805207, 849 new rows were added to source table of apply activity.

```
create or replace view deneme_source_FullTest as
 select c.*,
 case when
   abs(c.invoiceamount)*((100-abs(c.coinsurance))/100) >=
abs(c.paidamount) then
     0
   else
     1
 end risk
 from claimdetail c
 where abs(nvl(c.invoiceamount,0)) > 0  and  abs(paidamount) >
0 and c.coinsurance is not null
```

Figure III.13 Activity Apply Source Table SQL Statement for Fraud Type 1

After creating the source tables for data mining activity, build activity step was started. Build was chosen on the Activity pull-down menu and Anomaly Detection function and One-Class Support Vector Machine algorithm were selected. **Figure III.14** shows the activity create wizard. The modified source data deneme_sourceFull table is selected, and the attribute detailLineId column is designated as the unique identifier. **Figure III.15** shows the selecting case table window. Source tables' columns, which are related to detecting system, are selected to build activity model. **Figure III.16** shows the source column selection window. Notice that RISK has been automatically eliminated from the Build process because it has the constant value of 0. A name was entered for the activity. **Figure III.17** shows activity name window. There is an advanced settings button on the final page. The algorithm can be modified with choosing the kernel function type and setting the tolerance value and outlier rate. **Figure III.18** shows advanced settings dialog. Default settings were selected. In default settings, kernel function was left to odminer's option. It can select linear or Gaussian kernel functions. Also in default settings, tolerance value was set to 0.001, outlier rate was set to 0.1 and active learning option was set to enable. Then the build activity was finished. **Figure III.19** shows activity build solution window.



New Activity Wizard - Adım 1 / 5: Model Type

Select Mining Activity Type

Choose a model function type and algorithm. Review the descriptions to be sure you have picked the most appropriate selections. Click the Help button for additional details.

Function Type: Anomaly Detection

Algorithm: One-Class Support Vector Machine

Description: Anomaly Detection function:

- Identify a set of counter examples
- Detect outliers

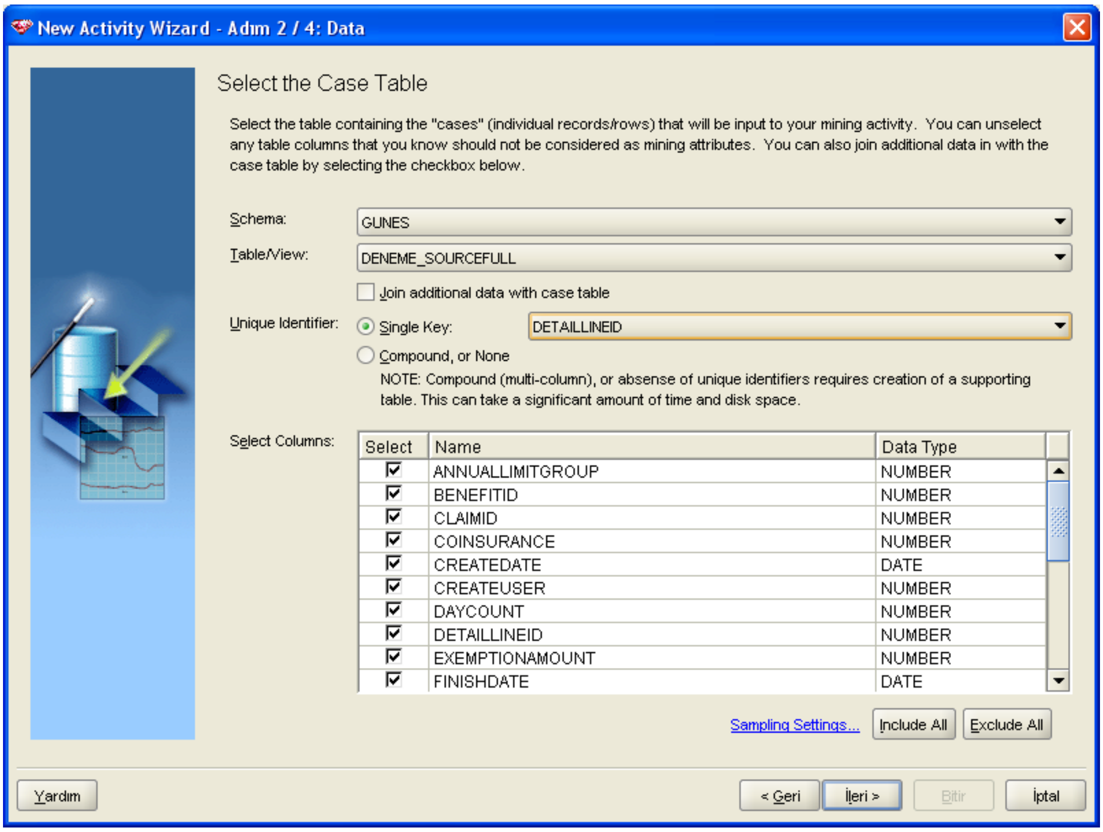
One-Class Support Vector Machine algorithm:

- Maximum prediction accuracy that avoids overfit
- Supports sparse transactional data
- Supports text data

Usage:

Standard binary supervised classification algorithms require the presence of both positive and negative examples (counterexamples) of a target class. Anomaly Detection requires only the presence of examples of a single target class. In outlier detection, typical examples in a distribution are separated from the atypical (outlier) examples.

Figure III.14 Activity Creating Wizard



Select the Case Table

Select the table containing the "cases" (individual records/rows) that will be input to your mining activity. You can unselect any table columns that you know should not be considered as mining attributes. You can also join additional data in with the case table by selecting the checkbox below.

Schema: GUNES

Table/View: DENEME_SOURCEFULL

□ Join additional data with case table

Unique Identifier: ● Single Key: DETAILLINEID

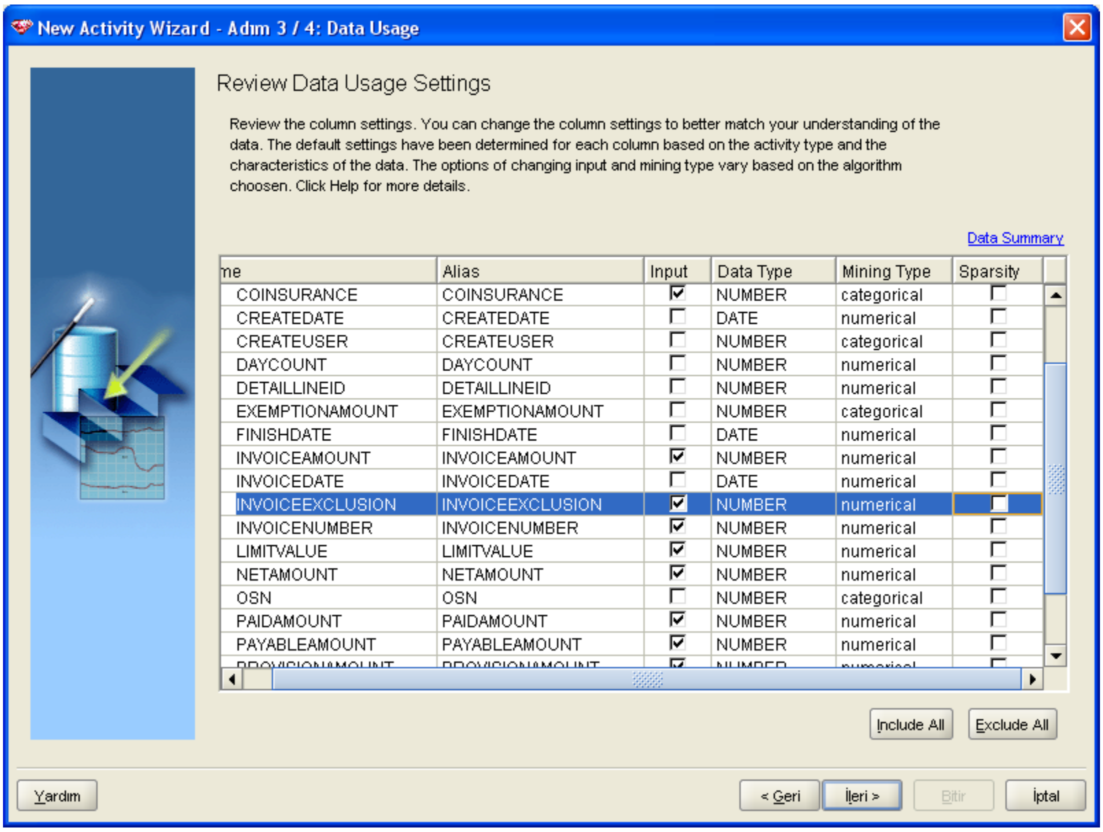
○ Compound, or None

NOTE: Compound (multi-column), or absense of unique identifiers requires creation of a supporting table. This can take a significant amount of time and disk space.

Select Columns:

Select	Name	Data Type
☑	ANNUALLIMITGROUP	NUMBER
☑	BENEFITID	NUMBER
☑	CLAIMID	NUMBER
☑	COINSURANCE	NUMBER
☑	CREATEDATE	DATE
☑	CREATEUSER	NUMBER
☑	DAYCOUNT	NUMBER
☑	DETAILLINEID	NUMBER
☑	EXEMPTIONAMOUNT	NUMBER
☑	FINISHDATE	DATE

Figure III.15 Selecting Case Table

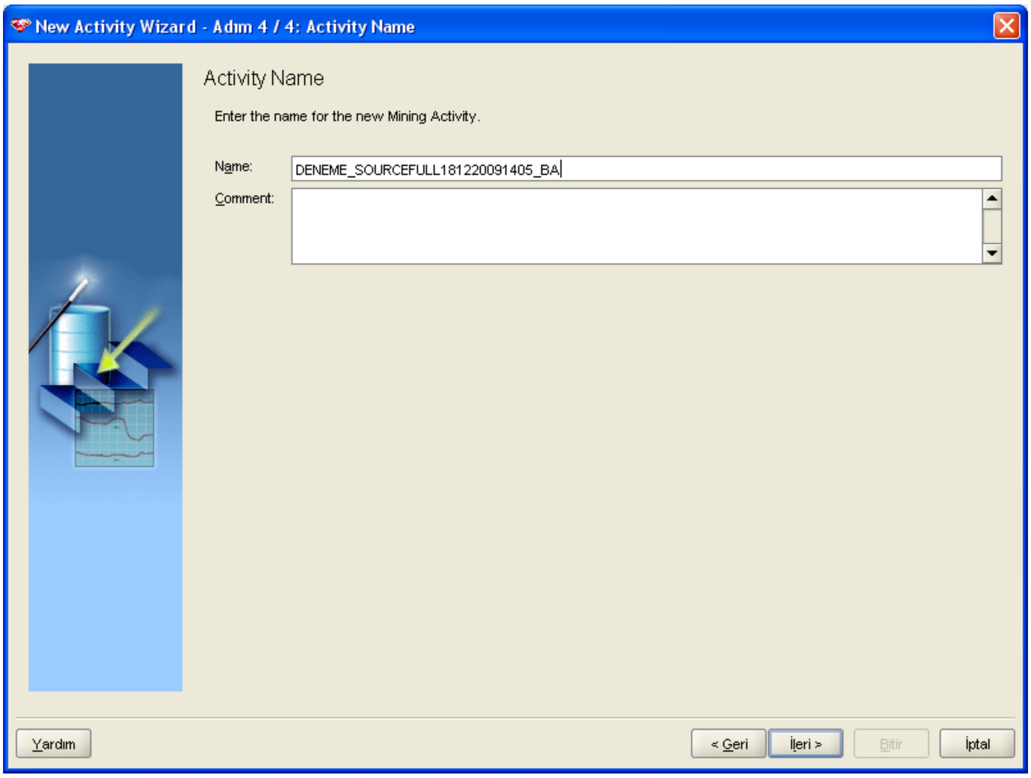


Review Data Usage Settings

Review the column settings. You can change the column settings to better match your understanding of the data. The default settings have been determined for each column based on the activity type and the characteristics of the data. The options of changing input and mining type vary based on the algorithm choosen. Click Help for more details.

me	Alias	Input	Data Type	Mining Type	Sparsity
COINSURANCE	COINSURANCE	☑	NUMBER	categorical	□
CREATEDATE	CREATEDATE	□	DATE	numerical	□
CREATEUSER	CREATEUSER	□	NUMBER	categorical	□
DAYCOUNT	DAYCOUNT	□	NUMBER	numerical	□
DETAILLINEID	DETAILLINEID	□	NUMBER	numerical	□
EXEMPTIONAMOUNT	EXEMPTIONAMOUNT	□	NUMBER	categorical	□
FINISHDATE	FINISHDATE	□	DATE	numerical	□
INVOICEAMOUNT	INVOICEAMOUNT	☑	NUMBER	numerical	□
INVOICEDATE	INVOICEDATE	□	DATE	numerical	□
INVOICEEXCLUSION	INVOICEEXCLUSION	☑	NUMBER	numerical	□
INVOICENUMBER	INVOICENUMBER	☑	NUMBER	numerical	□
LIMITVALUE	LIMITVALUE	☑	NUMBER	numerical	□
NETAMOUNT	NETAMOUNT	☑	NUMBER	numerical	□
OSN	OSN	□	NUMBER	categorical	□
PAIDAMOUNT	PAIDAMOUNT	☑	NUMBER	numerical	□
PAYABLEAMOUNT	PAYABLEAMOUNT	☑	NUMBER	numerical	□
PROVISIONAMOUNT	PROVISIONAMOUNT	☑	NUMBER	numerical	□

Figure III.16 Selecting Source Table Columns



New Activity Wizard - Adım 4 / 4: Activity Name

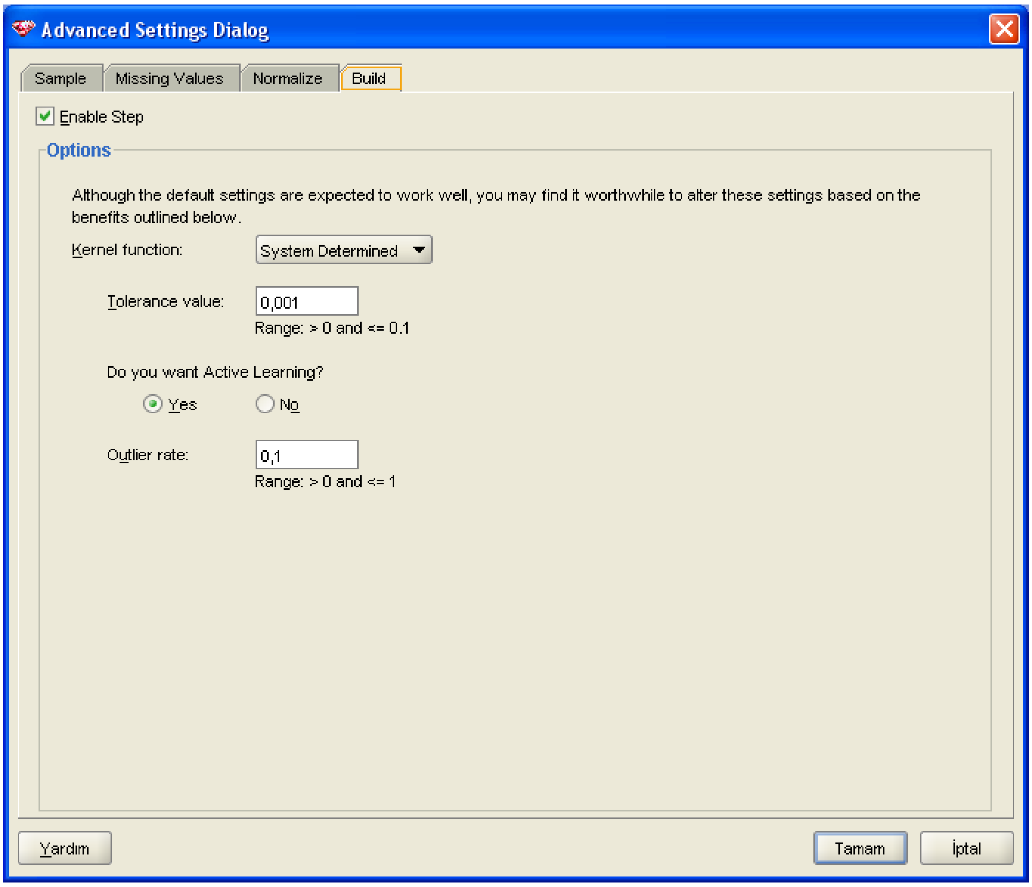
Activity Name

Enter the name for the new Mining Activity.

Name: DENEME_SOURCEFULL181220091405_BA

Comment:

Figure III.17 Activity Name window



Advanced Settings Dialog

☑ Enable Step

Options

Although the default settings are expected to work well, you may find it worthwhile to alter these settings based on the benefits outlined below.

Kernel function: System Determined

Tolerance value: 0,001

Range: > 0 and <= 0.1

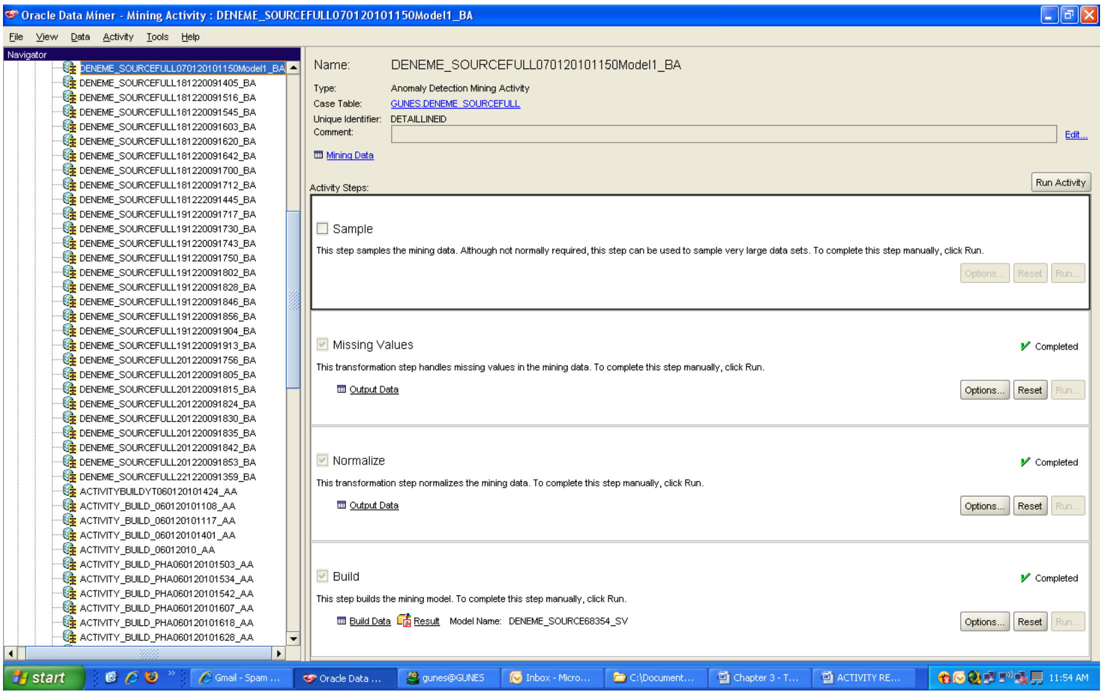
Do you want Active Learning?

● Yes ○ No

Outlier rate: 0,1

Range: > 0 and <= 1

Figure III.18 Advanced Setting Dialog



DENEME_SOURCEFULL070120101150Model1_BA
DENEME_SOURCEFULL181220091405_BA
DENEME_SOURCEFULL181220091516_BA
DENEME_SOURCEFULL181220091545_BA
DENEME_SOURCEFULL181220091603_BA
DENEME_SOURCEFULL181220091620_BA
DENEME_SOURCEFULL181220091642_BA
DENEME_SOURCEFULL181220091700_BA
DENEME_SOURCEFULL181220091712_BA
DENEME_SOURCEFULL181222091445_BA
DENEME_SOURCEFULL191220091717_BA
DENEME_SOURCEFULL191220091730_BA
DENEME_SOURCEFULL191220091743_BA
DENEME_SOURCEFULL191220091750_BA
DENEME_SOURCEFULL191220091802_BA
DENEME_SOURCEFULL191220091828_BA
DENEME_SOURCEFULL191220091846_BA
DENEME_SOURCEFULL191220091856_BA
DENEME_SOURCEFULL191220091904_BA
DENEME_SOURCEFULL191220091913_BA
DENEME_SOURCEFULL201220091756_BA
DENEME_SOURCEFULL201220091805_BA
DENEME_SOURCEFULL201220091815_BA
DENEME_SOURCEFULL201220091824_BA
DENEME_SOURCEFULL201220091830_BA
DENEME_SOURCEFULL201220091835_BA
DENEME_SOURCEFULL201220091842_BA
DENEME_SOURCEFULL201220091853_BA
DENEME_SOURCEFULL221220091359_BA
ACTIVITYBUILDYT060120101424_AA
ACTIVITY_BUILD_060120101108_AA
ACTIVITY_BUILD_060120101117_AA
ACTIVITY_BUILD_060120101401_AA
ACTIVITY_BUILD_06012010_AA
ACTIVITY_BUILD_PHA060120101503_AA
ACTIVITY_BUILD_PHA060120101534_AA
ACTIVITY_BUILD_PHA060120101542_AA
ACTIVITY_BUILD_PHA060120101607_AA
ACTIVITY_BUILD_PHA060120101618_AA
ACTIVITY_BUILD_PHA060120101628_AA

Name: DENEME_SOURCEFULL070120101150Model1_BA
Type: Anomaly Detection Mining Activity
Case Table: GUNES.DENEME_SOURCEFULL
Unique Identifier: DETAILLINEID
Comment: Edit...

Mining Data

Activity Steps:

□ Sample

This step samples the mining data. Although not normally required, this step can be used to sample very large data sets. To complete this step manually, click Run.

☑ Missing Values ✓ Completed

This transformation step handles missing values in the mining data. To complete this step manually, click Run.

Output Data

☑ Normalize ✓ Completed

This transformation step normalizes the mining data. To complete this step manually, click Run.

Output Data

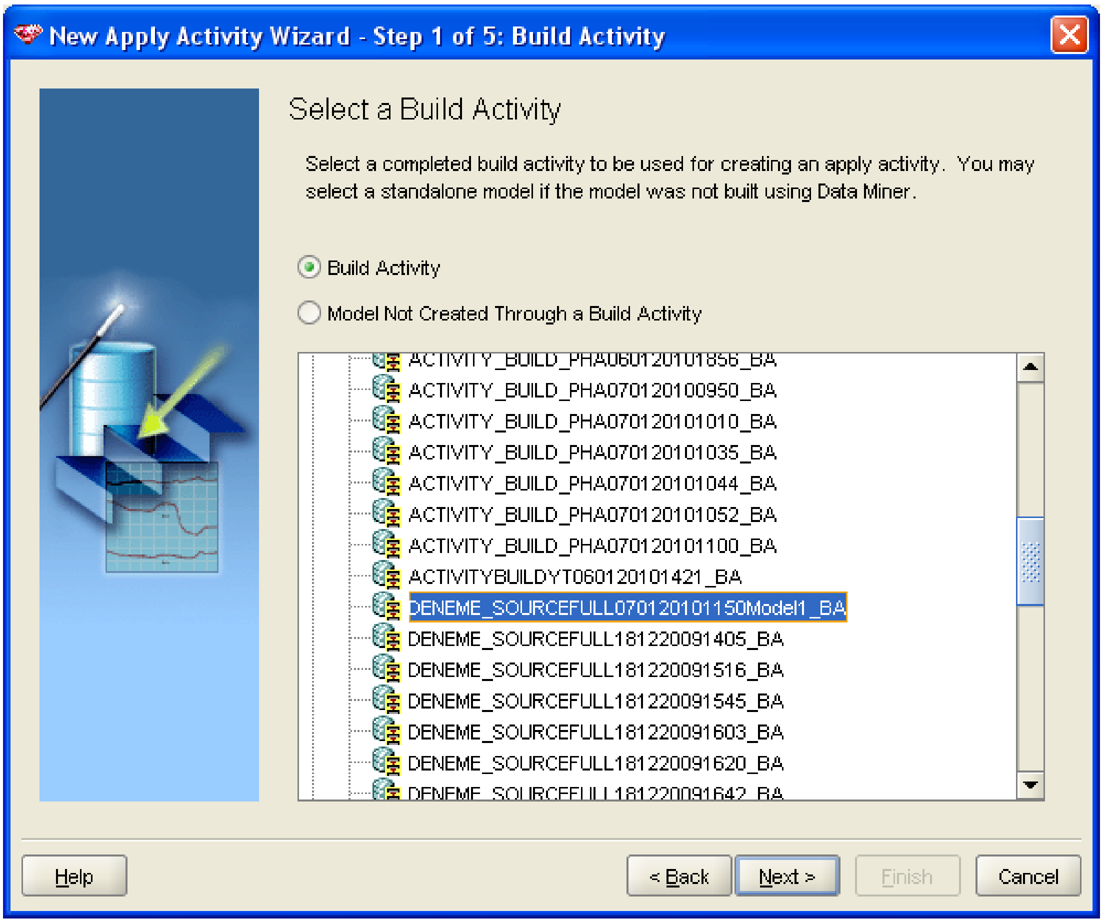
☑ Build ✓ Completed

This step builds the mining model. To complete this step manually, click Run.

Build Data Result Model Name: DENEME_SOURCE68354_SV

Figure III.19 Activity Build Solution Window

After building the activity, this activity was applied on whole data. First Apply was chosen from the Activity pull-down menu and the Anomaly Detection Build activity was highlighted. Completed build activity was selected to be used for creating an apply activity. **Figure III.20** shows selecting build activity window. Then the source table was selected, which contains normal and suspicious data, for applying the model. **Figure III.21** shows selecting apply data sources window. Then the supplemental columns were selected to include in the apply output table along with the standard prediction column. **Figure III.22** shows selecting supplemental columns window. Descriptive name was given for this apply activity and apply activity was finished. **Figure III.23** shows entering apply activity name window.



New Apply Activity Wizard - Step 1 of 5: Build Activity

Select a Build Activity

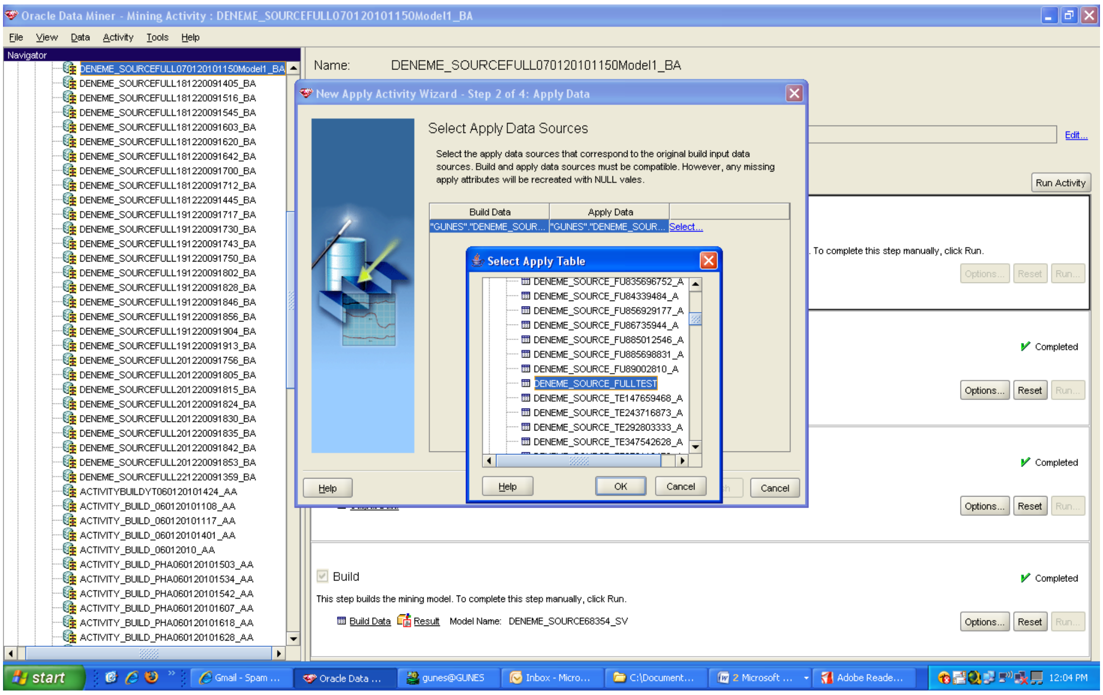
Select a completed build activity to be used for creating an apply activity. You may select a standalone model if the model was not built using Data Miner.

● Build Activity

○ Model Not Created Through a Build Activity

ACTIVITY_BUILD_PHA060120101856_BA
ACTIVITY_BUILD_PHA070120100950_BA
ACTIVITY_BUILD_PHA070120101010_BA
ACTIVITY_BUILD_PHA070120101035_BA
ACTIVITY_BUILD_PHA070120101044_BA
ACTIVITY_BUILD_PHA070120101052_BA
ACTIVITY_BUILD_PHA070120101100_BA
ACTIVITYBUILDYT060120101421_BA
DENEME_SOURCEFULL070120101150Model1_BA
DENEME_SOURCEFULL181220091405_BA
DENEME_SOURCEFULL181220091516_BA
DENEME_SOURCEFULL181220091545_BA
DENEME_SOURCEFULL181220091603_BA
DENEME_SOURCEFULL181220091620_BA
DENEME_SOURCEFULL181220091642_BA

Figure III.20 Selecting Build Activity Window



Oracle Data Miner - Mining Activity : DENEME_SOURCEFULL070120101150Model1_BA

DENEME_SOURCEFULL070120101150Model1_BA
DENEME_SOURCEFULL181220091405_BA
DENEME_SOURCEFULL181220091516_BA
DENEME_SOURCEFULL181220091545_BA
DENEME_SOURCEFULL181220091603_BA
DENEME_SOURCEFULL181220091620_BA
DENEME_SOURCEFULL181220091642_BA
DENEME_SOURCEFULL181220091700_BA
DENEME_SOURCEFULL181220091712_BA
DENEME_SOURCEFULL181222091445_BA
DENEME_SOURCEFULL191220091717_BA
DENEME_SOURCEFULL191220091730_BA
DENEME_SOURCEFULL191220091743_BA
DENEME_SOURCEFULL191220091750_BA
DENEME_SOURCEFULL191220091802_BA
DENEME_SOURCEFULL191220091828_BA
DENEME_SOURCEFULL191220091846_BA
DENEME_SOURCEFULL191220091856_BA
DENEME_SOURCEFULL191220091904_BA
DENEME_SOURCEFULL191220091913_BA
DENEME_SOURCEFULL201220091756_BA
DENEME_SOURCEFULL201220091805_BA
DENEME_SOURCEFULL201220091815_BA
DENEME_SOURCEFULL201220091824_BA
DENEME_SOURCEFULL201220091830_BA
DENEME_SOURCEFULL201220091835_BA
DENEME_SOURCEFULL201220091842_BA
DENEME_SOURCEFULL201220091853_BA
DENEME_SOURCEFULL221220091359_BA
ACTIVITYBUILDYT060120101424_AA
ACTIVITY_BUILD_060120101108_AA
ACTIVITY_BUILD_060120101117_AA
ACTIVITY_BUILD_060120101401_AA
ACTIVITY_BUILD_06012010_AA
ACTIVITY_BUILD_PHA060120101503_AA
ACTIVITY_BUILD_PHA060120101534_AA
ACTIVITY_BUILD_PHA060120101542_AA
ACTIVITY_BUILD_PHA060120101607_AA
ACTIVITY_BUILD_PHA060120101618_AA
ACTIVITY_BUILD_PHA060120101628_AA

Name: DENEME_SOURCEFULL070120101150Model1_BA

New Apply Activity Wizard - Step 2 of 4: Apply Data

Select Apply Data Sources

Select the apply data sources that correspond to the original build input data sources. Build and apply data sources must be compatible. However, any missing apply attributes will be recreated with NULL vales.

Build Data	Apply Data	
"GUNES"."DENEME_SOUR...	"GUNES"."DENEME_SOUR...	Select...

Select Apply Table

DENEME_SOURCE_FU835696752_A
DENEME_SOURCE_FU84339484_A
DENEME_SOURCE_FU856929177_A
DENEME_SOURCE_FU86735944_A
DENEME_SOURCE_FU885012546_A
DENEME_SOURCE_FU885698831_A
DENEME_SOURCE_FU89002810_A
DENEME_SOURCE_FULLTEST
DENEME_SOURCE_TE147659468_A
DENEME_SOURCE_TE243716873_A
DENEME_SOURCE_TE292803333_A
DENEME_SOURCE_TE347542628_A

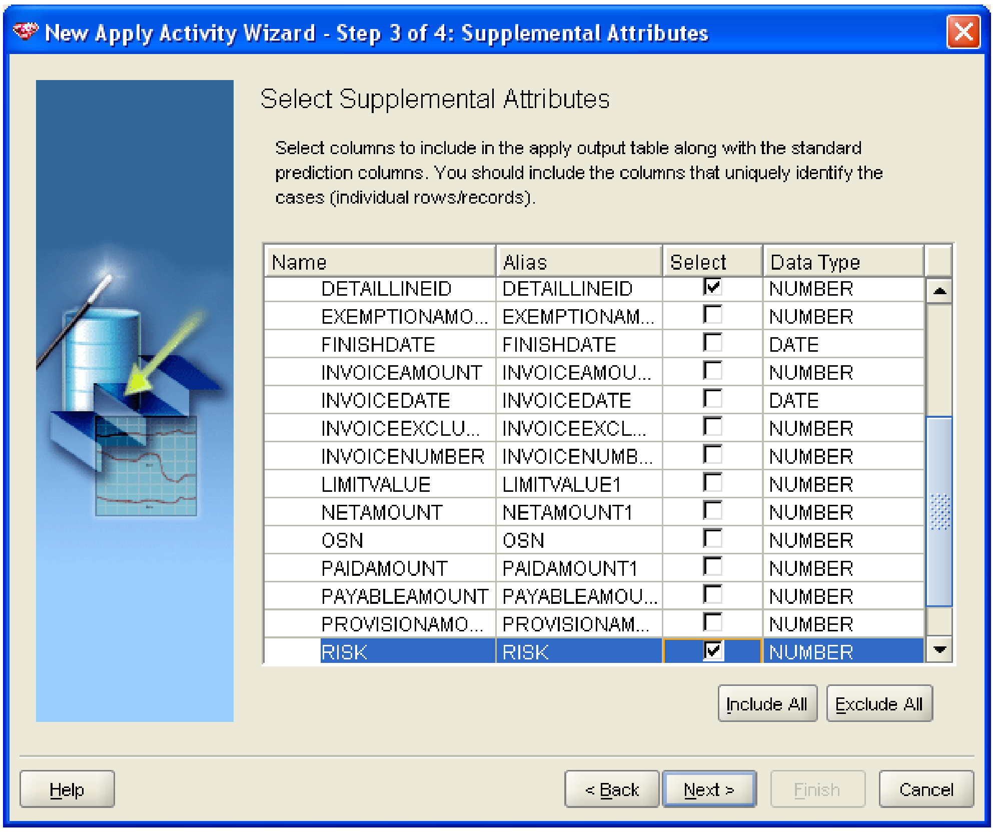
. To complete this step manually, click Run.

☑ Build

This step builds the mining model. To complete this step manually, click Run.

Build Data Result Model Name: DENEME_SOURCE68354_SV

Figure III.21 Selecting Apply Data Source Window



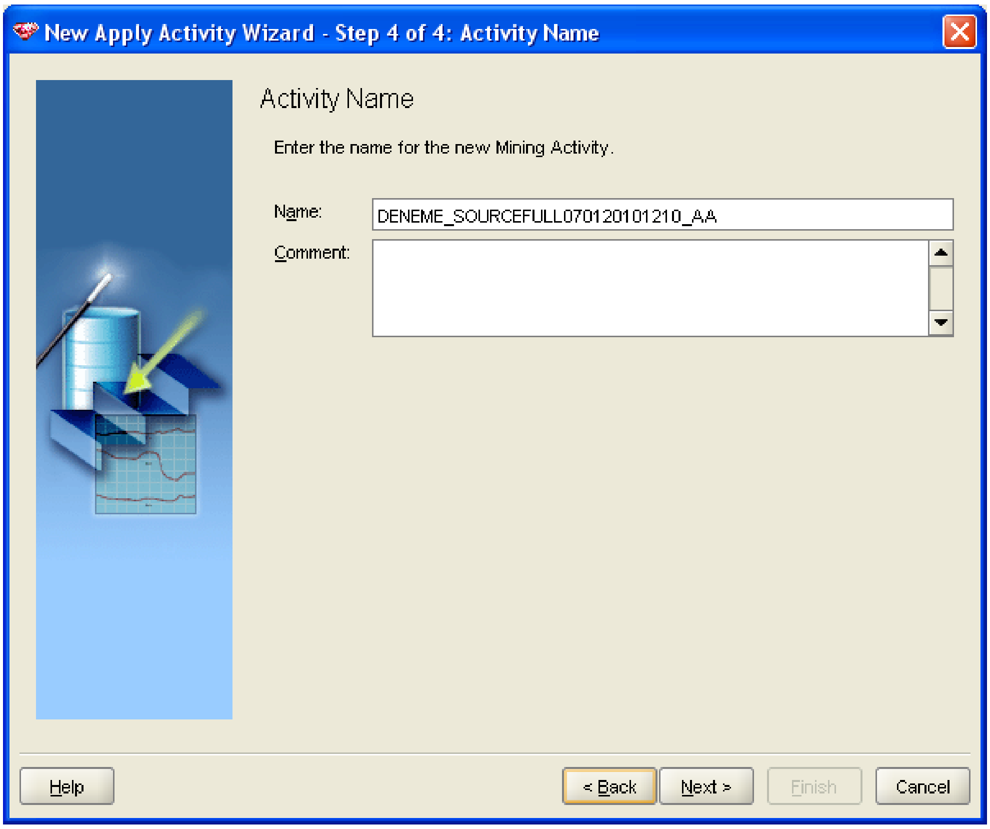
New Apply Activity Wizard - Step 3 of 4: Supplemental Attributes

Select Supplemental Attributes

Select columns to include in the apply output table along with the standard prediction columns. You should include the columns that uniquely identify the cases (individual rows/records).

Name	Alias	Select	Data Type
DETAILLINEID	DETAILLINEID	☑	NUMBER
EXEMPTIONAMO...	EXEMPTIONAM...	□	NUMBER
FINISHDATE	FINISHDATE	□	DATE
INVOICEAMOUNT	INVOICEAMOU...	□	NUMBER
INVOICEDATE	INVOICEDATE	□	DATE
INVOICEEXCLU...	INVOICEEXCL...	□	NUMBER
INVOICENUMBER	INVOICENUMB...	□	NUMBER
LIMITVALUE	LIMITVALUE1	□	NUMBER
NETAMOUNT	NETAMOUNT1	□	NUMBER
OSN	OSN	□	NUMBER
PAIDAMOUNT	PAIDAMOUNT1	□	NUMBER
PAYABLEAMOUNT	PAYABLEAMOU...	□	NUMBER
PROVISIONAMO...	PROVISIONAM...	□	NUMBER
RISK	RISK	☑	NUMBER

Figure III.22 Selecting Supplemental Columns Window



New Apply Activity Wizard - Step 4 of 4: Activity Name

Activity Name

Enter the name for the new Mining Activity.

Name: DENEME_SOURCEFULL070120101210_AA

Comment:

Figure III.23 Entering Apply Activity Name Window

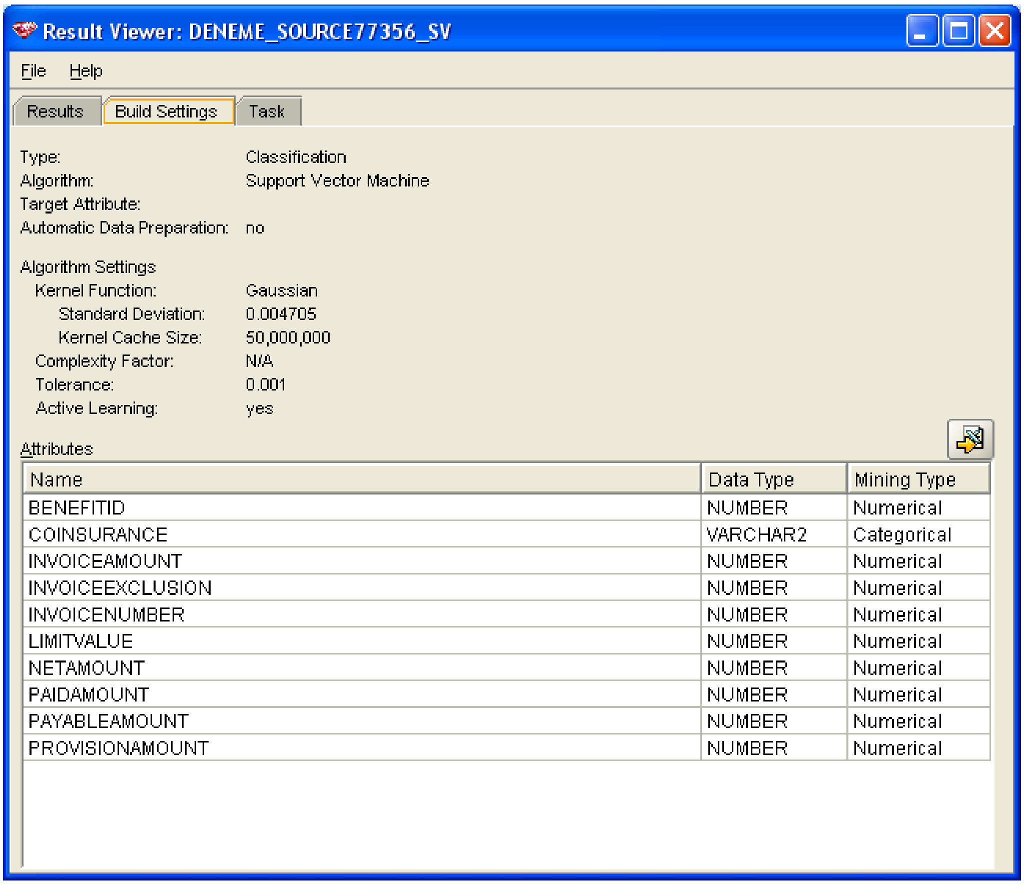
Apply activity creates a new table. It adds all columns of apply source table and two new columns whose names are prediction and probability. The Prediction column in the Anomaly Detection output table always has value 1 for a case determined to be “normal” and 0 for a “suspicious” case. Since the RISK column in the input data had value 0 for Low Risk and 1 for High Risk, the correctly predicted cases are those in which RISK_1 =1 and PREDICTION = 0. Apply activity created DENEME_SOURCE_FU486870458_A table. First of all, SQL statement which counts the suspicious data in this table was executed. **Figure III.24** shows this SQL statement that was used to count the suspicious data. There are 849 suspicious data. Then SQL statement which counts the correctly predicted cases was executed. **Figure III.25** shows the SQL statement that was used to count the correctly predicted cases. ODM sets 847 rows’ prediction column to 0. It means this model could eliminate 2 suspicious transactions. ODM selected Gaussian kernel function for this model. The selected kernel function and tolerance value can be seen at result viewer screen. **Figure III.26** shows result viewer screen. Support Vector machine is a kernel-based algorithm. A kernel is a function that transforms the input data to a high-dimensional space where the problem is solved. Kernel functions can be linear or nonlinear. Oracle Data Mining supports a linear kernel and a Gaussian (nonlinear) kernel. The linear kernel function reduces to a linear equation on the original attributes in the training data. A linear kernel works well when there are many attributes in the training data. The Gaussian kernel function transforms each case in the training data to a point in an n-dimensional space, where n is the number of attributes. The algorithm attempts to separate the points into subsets with homogeneous target values. The Gaussian kernel uses nonlinear separators, but within the kernel space it constructs a linear equation. The default kernel function is determined by the algorithm based on the number of attributes in the training data. When there are many attributes, the algorithm uses a linear kernel function; otherwise it uses a nonlinear (Gaussian) kernel. So oracle data miner controlled the number of attributes in build activity and determined to used the Gaussian kernel function.

```
SELECT COUNT (*) FROM DENEME_SOURCE_FU486870458_A TA WHERE
TA.RISK = 1
```

Figure III.24 SQL statement for counting suspicious data

```
SELECT COUNT (*) FROM DENEME_SOURCE_FU486870458_A TA WHERE
TA.RISK = 1 AND TA.PREDICTION = 0
```

Figure III.25 SQL statement for counting correctly predicted case



Type: Classification
Algorithm: Support Vector Machine
Target Attribute:
Automatic Data Preparation: no

Algorithm Settings
Kernel Function: Gaussian
Standard Deviation: 0.004705
Kernel Cache Size: 50,000,000
Complexity Factor: N/A
Tolerance: 0.001
Active Learning: yes

Attributes

Name	Data Type	Mining Type
BENEFITID	NUMBER	Numerical
COINSURANCE	VARCHAR2	Categorical
INVOICEAMOUNT	NUMBER	Numerical
INVOICEEXCLUSION	NUMBER	Numerical
INVOICENUMBER	NUMBER	Numerical
LIMITVALUE	NUMBER	Numerical
NETAMOUNT	NUMBER	Numerical
PAIDAMOUNT	NUMBER	Numerical
PAYABLEAMOUNT	NUMBER	Numerical
PROVISIONAMOUNT	NUMBER	Numerical

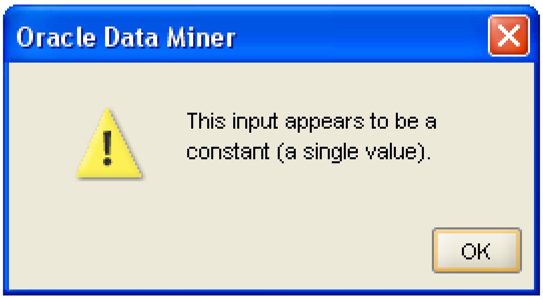
Figure III.26 Result Viewer Screen

As the result is not adequate, new activity models were created with different advanced settings. Outlier rate is the percentage of “suspicious” cases in system population. The outlier rate of this fraud type was learned from health insurance experts. Experts think that the outlier rate of this fraud type is approximately 0.1 %.

First of all a model whose kernel function is linear kernel function, outlier rate is 0.001 and tolerance value is 0.1 was created. When this model was applied on apply source dataset, DENEME_SOURCE_FU324478734_A table was created and model eliminated 64 suspicious transactions. In order to get better results, new detection models that have same outlier rate but different tolerance value were created. A model whose kernel function is linear, outlier rate is 0.001 and tolerance value is 0.01 was created and applied on apply activity source dataset. DENEME_SOURCE_FU219815835_A table was created. This model eliminated 22 suspicious transactions. Creating new models with different tolerance value was continued. A model whose kernel function is linear, outlier rate is 0.001 and tolerance value is 0.001 created DENEME_SOURCE_FU669344133_A table and it eliminated 102 suspicious transactions. A model whose kernel function is linear, outlier rate is 0.001 and tolerance value is 0.0001 created DENEME_SOURCE_FU756954109_A table and it eliminated 22 suspicious transactions. A model whose kernel function is linear, outlier rate is 0.001 and tolerance value is 0.00001 created DENEME_SOURCE_FU557357407_A table and it eliminated 28 suspicious transactions. As learning that the best tolerance value is 0.001 when outlier rate is 0.001 and kernel function is linear, new models whose kernel function is linear, tolerance value is 0.001 was created. Outlier rate was modified to find better models. A model whose kernel function is linear, tolerance value is 0.001, and outlier rate is 0.1 was created and applied on activity apply dataset. It created DENEME_SOURCE_FU98415740_A table and eliminated 15 suspicious transactions. Then creating the models whose outlier rate is smaller than this model was started. A model whose kernel function is linear, tolerance value is 0.001, and outlier rate is 0.01 was created and applied on activity apply dataset. It created DENEME_SOURCE_FU835696752_A table and gave the same result as before. Then a model whose kernel function is linear, tolerance value is 0.001, and outlier rate is 0.0001 was created and applied on activity apply dataset. It created DENEME_SOURCE_FU976256714_A table and eliminated only 2 suspicious transactions. After this model two models whose outlier rate is 0.00001 and 0.000001 respectively were created. They created DENEME_SOURCE_FU5099568_A table and DENEME_SOURCE_FU656227081_A tables. They eliminated 9 and 2 suspicious transactions respectively. At the end of these experiments, it was seen that a model whose tolerance value is 0.001 and outlier rate is 0.001 is the best model of the linear kernel function based models. After these models, Gaussian kernel function based models were created. First of all a model whose kernel function is Gaussian kernel function, outlier rate is 0.001 and tolerance value is 0.001 was created. When this model was applied on apply source dataset, DENEME_SOURCE_FU992713407_A table was created and model could not eliminate any suspicious transaction. Then a model whose kernel function is Gaussian kernel function, outlier rate is 0.001 and tolerance value is 0.1 was created. When this model was applied on apply source dataset, DENEME_SOURCE_FU205718232_A table was created and it could not eliminate any suspicious transaction too. As a result, new Gaussian kernel function based models whose tolerance values were 0.001 were created. A model whose kernel function is Gaussian kernel function, tolerance value is 0.001, and outlier rate is 0.1 created DENEME_SOURCE_FU486870458_A table and eliminated 2 suspicious transactions. A model whose kernel function is Gaussian kernel function, tolerance value is 0.001, and outlier rate is 0.0001 created DENEME_SOURCE_FU296875536_A table and eliminated 1 suspicious transaction. A model whose kernel function is Gaussian kernel function, tolerance value is 0.001, and outlier rate is 0.000001 created DENEME_SOURCE_FU102322062_A table and it could not eliminate any suspicious transaction. As these models could eliminate very few suspicious transactions, two models whose tolerance value and outlier rate were determined randomly. Gaussian kernel function based model whose tolerance value is 0.00001 and outlier rate is 0.00001 created DENEME_SOURCE_FU82351919_A table and eliminated 5 suspicious transactions. Other Gaussian kernel function based model whose tolerance value is 0.000001 and outlier rate is 0.00001 created DENEME_SOURCE_FU84339484_A table and eliminated only 1 suspicious transaction.

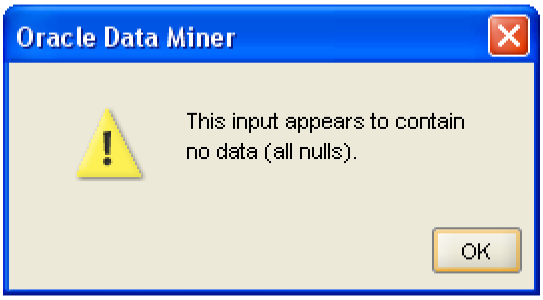
In all of these models, odminer get benefitId, coinsurance, createUser, invoiceAmount, invoiceExclusion, invoiceNumber, limitValue, netAmount, paidAmount, payableAmount, provisionAmount columns as the attributes of the build activity. It analyzed all columns and determined that vatrate, exemptionAmount and annualLimitGroup columns have constant values and lots of them have null value, and also dayCount, finishDate, and startDate columns have only null values. So it did not use these columns. To see the affects of selecting the columns, new models whose build columns are different were created. First of all same of the model which was created before and whose kernel function is linear, tolerance value is 0.001 and outlier rate is 0.001 was created. When building the model, all columns of the table were selected as attributes of this model. When vatrate, exemptionAmount and annualLimitGroup columns were selected to use odminer gave warning which is “This input appears to be a constant (a single value)”. **Figure III.27** shows this warning message. Also when dayCount, finishDate, and startDate columns were selected it gave warning which is “This input appears to contain no data (all nulls)”. **Figure III.28** shows this warning message. In spite of these warning, these columns were selected and model was created. It created DENEME_SOURCE_FU378992985_A table and it eliminated 3 suspicious transactions. Then a new model whose kernel function is linear, tolerance value is 0.001 and outlier rate is 0.001 was created. This time only claimId, createDate and invoiceDate columns was added to other attributes that were selected by ODM. This model created DENEME_SOURCE_FU870927537_A table and it eliminated 183 suspicious transactions. Then a model whose kernel function is linear, outlier rate is 0.001 and tolerance value is 0.01 was created. ClaimId, createDate and invoiceDate columns was added to other attributes. It applied on apply activity source dataset. DENEME_SOURCE_FU134535650_A table was created. This model eliminated 30 suspicious transactions. As the last linear kernel function based model whose outlier rate is 0.001 and tolerance value is 0.1 was created. ClaimId, createDate and invoiceDate columns was added to other attributes too. It created DENEME_SOURCE_FU510940182_A table and eliminated 91 suspicious transactions. Because the changing selected column affects the result, new Gaussian kernel function based models were created. First a model whose tolerance value is 0.001 and outlier rate is 0.1 was created. All columns were selected as attributes of model. This model created DENEME_SOURCE_FU909447088_A table and eliminated only 2 suspicious transactions. After this model, a model that has same settings was created. But in this time only ClaimId, createDate and invoiceDate columns were added to other attributes of model. This model created

DENEME_SOURCE_FU560473391_A table and it eliminated 2 suspicious transactions too. Then a Gaussian kernel function based model whose tolerance value is 0.001 and outlier rate is 0.001 was created. All columns were selected when model was building. I created DENEME_SOURCE_FU863870708_A table, and eliminated 2 suspicious transactions. Then a model that has same settings was created. Only ClaimId, createDate and invoiceDate columns were added to other attributes of the model. It was applied on apply activity source dataset and it created the DENEME_SOURCE_FU508913344_A table and eliminated 2 suspicious transactions. **Table III.9** shows the results of created models.



This input appears to be a constant (a single value).

Figure III.27 Constant Input Warning Message



This input appears to contain no data (all nulls).

Figure III.28 Null Input Warning Message

Table III.9 Models Results for Fraud Type 1

Model Number	Function Name	Tolerance Value	Outlier Rate	Selected Column	Correctly Predicted	Table Name
1	Linear	0.001	0.1	Default	834	DENEME_SOURCE_FU98415740_A
2	Linear	0.001	0.01	Default	834	DENEME_SOURCE_FU835696752_A
3	Linear	0.001	0.001	Default	747	DENEME_SOURCE_FU669344133_A
4	Linear	0.001	0.0001	Default	847	DENEME_SOURCE_FU976256714_A
5	Linear	0.001	0.00001	Default	840	DENEME_SOURCE_FU5099568_A
6	Linear	0.001	0.000001	Default	847	DENEME_SOURCE_FU656227081_A
7	Linear	0.01	0.001	Default	827	DENEME_SOURCE_FU219815835_A
8	Linear	0.1	0.001	Default	785	DENEME_SOURCE_FU324478734_A
9	Linear	0.0001	0.001	Default	827	DENEME_SOURCE_FU756954109_A
10	Linear	0.00001	0.001	Default	821	DENEME_SOURCE_FU557357407_A
11	Linear	0.001	0.001	All	846	DENEME_SOURCE_FU378992985_A
12	Linear	0.001	0.001	Special*	666	DENEME_SOURCE_FU870927537_A
13	Linear	0.01	0.001	Special*	819	DENEME_SOURCE_FU134535650_A
14	Linear	0.1	0.001	Special*	758	DENEME_SOURCE_FU510940182_A
15	Gaussian	0.001	0.1	Default	847	DENEME_SOURCE_FU486870458_A
16	Gaussian	0.001	0.001	Default	849	DENEME_SOURCE_FU992713407_A
17	Gaussian	0.001	0.000001	Default	849	DENEME_SOURCE_FU102322062_A
18	Gaussian	0.1	0.001	Default	849	DENEME_SOURCE_FU205718232_A
19	Gaussian	0.00001	0.00001	Default	844	DENEME_SOURCE_FU82351919_A
20	Gaussian	0.000001	0.00001	Default	848	DENEME_SOURCE_FU84339484_A
21	Gaussian	0.001	0.0001	Default	848	DENEME_SOURCE_FU296875536_A
22	Gaussian	0.001	0.1	All	847	DENEME_SOURCE_FU909447088_A
23	Gaussian	0.001	0.1	Special*	847	DENEME_SOURCE_FU560473391_A
24	Gaussian	0.001	0.001	All	847	DENEME_SOURCE_FU863870708_A
25	Gaussian	0.001	0.001	Special*	847	DENEME_SOURCE_FU508913344_A

***** Default columns and claimed, createDate and invoiceDate columns.

III.7.2 FRAUD TYPE 2

The second selected fraud type is "transactions which involve medicine taking without doctor inspection or surgical operation within four days". Generally patient goes to the doctor, the doctor diagnoses his illness and administers drug. Patient then goes to buy these medicines. Health insurance experts think that if a patient buys medicine without doctor inspection or surgical operation, this can be a fraud. This type includes buying medicine without prescription, selling medicine to another person without the knowledge of insured, buying medicine for insured's friends and acquaintances, creating fake invoice by employee in insurance company for himself or another crime organization. First of all, medicine claims, whose cost is bigger than zero, must be listed. Detection model can be adversely affected by missing value in building data or worthless claim. Benefit table contains assurances of the insured like medicine, doctor, surgical operation. This table is linked to claimDetail table with benefitId column. Benefit id of medicine is 105, benefit id of doctor is 104 and benefit id of surgical operation is 126. PaidAmount column in claimDetail table contains the total cost of claim. So the paidAmount of the claim must be bigger than 0 and benefitType of the claim must be equal to 105. Also the building activity source claims must have transactions whose benefit types were 104 or 126 in past four days.

Before building a data mining model, source view has to be created. So source table were created from claim and claim detail table. This table includes only medicine claims that have doctor diagnosis or surgical operation in 4 days. Also oracle data mining tool requires RISK column which represents suspicious cases. The data has been transformed so that the build data source consists only of records with RISK = 0, representing no risk. Activity_build_pharm view was created as build source. In health insurance company's system there are 700 medicine taking transactions without doctor inspection or surgical operation in four days. But ten of these transactions have equal value in paidAmount column and sixteen of them have null value in paidAmount column. These 26 transactions are inconsistent data and they were eliminated in creating the model build source. The number of rows in activity build source is 674. **Figure III.29** shows the SQL statement for creating the build source view. LastVisitDay column contains day count since the same insured consulted a doctor or underwent an operation.

```
create or replace view activity_build_pharm as
SELECT
       c.customerid,
       c.healthcenterid,
       c.createuser claimuser,
       cd.*,
       ROUND(NVL((c.eventdate - (SELECT Max(C2.EventDate)
                       FROM   Claim C2,
                              ClaimDetail Cd2
                       WHERE  C2.ClaimID = Cd2.ClaimID AND
                              (Cd2.BenefitID = 104 OR Cd2.BenefitID = 126) AND
                              C2.CUSTOMERID = c.CustomerID AND
                              C2.EventDate <= c.Eventdate)),999)) LastVisitDay
, 0 RISK
FROM   claim c,
       claimdetail cd
WHERE  c.ClaimID = cd.ClaimID AND
       cd.benefitid = 105 AND
       cd.paidamount > 0 AND
       ROUND(NVL((c.eventdate - (SELECT Max(C2.EventDate)
                       FROM   Claim C2,
                              ClaimDetail Cd2
                       WHERE  C2.ClaimID = Cd2.ClaimID AND
                              (Cd2.BenefitID = 104 OR Cd2.BenefitID = 126) AND
                              C2.CUSTOMERID = c.CustomerID AND
                              C2.EventDate <= c.Eventdate)),999)) < 4
```

Figure III.29 Activity Build Source View SQL Statement for Fraud Type 2

Then activity apply data source was created. Oracle data mining tool requires a test source that has rarely suspicious data. So the test data source table has a few records with RISK = 1, representing the unusual cases that the Anomaly Detection model will try to find. Activity_apply_pharm was created as activity source. **Figure III.30** shows the SQL statement for creating the activity apply source view. The number of rows in the Activity_apply_pharm is 1387. As the number of rows in activity_build_pharm, source table of build activity, is 674, 713 new rows was added to source table of apply activity. These are the suspicious transactions.

```
       create or replace view activity_apply_pharm as
SELECT
       c.customerid,
       c.healthcenterid,
       c.createuser claimuser,
       cd.*,
       ROUND(NVL((c.eventdate - (SELECT Max(C2.EventDate)
                      FROM   Claim C2,
                             ClaimDetail Cd2
                      WHERE  C2.ClaimID = Cd2.ClaimID AND
                             (Cd2.BenefitID = 104 OR Cd2.BenefitID = 126) AND
                             C2.CUSTOMERID = c.CustomerID AND
                             C2.EventDate <= c.Eventdate)),999)) LastVisitDay ,
                             case when
   ROUND(NVL((c.eventdate - (SELECT Max(C2.EventDate)
                      FROM   Claim C2,
                             ClaimDetail Cd2
                      WHERE  C2.ClaimID = Cd2.ClaimID AND
                             (Cd2.BenefitID = 104 OR Cd2.BenefitID = 126) AND
                             C2.CUSTOMERID = c.CustomerID AND
                             C2.EventDate <= c.Eventdate)),999)) < 4 then
     0
   else
     1
 end risk
FROM   claim c,claimdetail cd
WHERE  c.ClaimID = cd.ClaimID AND
       cd.benefitid = 105 AND cd.paidamount > 0
```

Figure III.30 Activity Apply Source View SQL Statement for Fraud Type 2

After creating source views for data mining activity, build activity step was started. As in the first fraud type, anomaly detection model was created and One-Class Support Vector Machine algorithm was selected. The modified source data Activity_build_pharm view is selected, and the attribute detailLineId column is designated as the unique identifier. Default settings were selected. As it was explained before, in default settings kernel function was left to odminer's option. It can select linear or Gaussian kernel functions. Also tolerance value was set to 0.001, outlier rate was set to 0.1 and active learning option was set to enable. Then the build activity was finished.

After building activity, this activity was applied on whole data. Completed build activity was selected to be used for creating an apply activity. Then the source view whose name is activity_apply_pharm was selected, which contains normal and suspicious data, for applying the model. Then the supplemental columns were selected to include in the apply output table along with the standard prediction column. Apply activity creates a new table. Apply activity created ACTIVITY_APPLY_P598259584_A table. There are 713 suspicious data. Activity determined 713 correctly predicted data. As a linear kernel works well when there are many attributes in the training data, the default kernel function must be determined by the algorithm based on the number of attributes in the training data. Oracle data miner controlled the number of attributes in build activity and determined to use Gaussian kernel function. Also it determines tolerance value 0.001 and outlier rate 0.1.

As the result is not adequate, new activity models were created with different advanced settings. The outlier rate of this fraud type was learned from health insurance experts. Experts think that the outlier rate of this fraud type is approximately 0.1 %.

First of all linear function based models whose tolerance values were fixed to 0.01 and outlier rates were modified were created. A linear function based model whose tolerance value is 0.1 and outlier rate is 0.01 was created. It was applied on activity apply data source. It created ACTIVITY_APPLY_P720662032_A table and eliminated 260 suspicious transactions. After this model a linear function based model whose tolerance value is 0.1 and outlier rate is 0.001 was created. It created ACTIVITY_APPLY_P699546311_A table and eliminated 633 suspicious transactions. Then decreasing the outlier rate was continuing. A linear function based model whose tolerance value is 0.1 and outlier rate is 0.0001 was created. It created ACTIVITY_APPLY_P213912614_A table and eliminated 633 suspicious transactions too. Also a new model whose kernel function is linear, tolerance value is 0.1 and outlier rate is 0.00001 was created. It created ACTIVITY_APPLY_P895635615_A table and eliminated 633 suspicious transactions too. According to these experiments decreasing the outlier to smaller than 0.001 is useless. So new linear function based models whose outlier rate was fixed to 0.01 were created. A linear function based model whose tolerance value is 0.01 and outlier rate is 0.01 was created. It created ACTIVITY_APPLY_P403952277_A table and eliminated 380 suspicious transactions. After this model a new model whose tolerance value is 0.01 and outlier value is 0.001 was created. It created ACTIVITY_APPLY_P586786320_A table and eliminated 640 suspicious transactions. Then two linear functional based and their tolerance value is 0.01 models created. First of them whose outlier rate is 0.001 created ACTIVITY_APPLY_P868726764_A table, the other one whose outlier rate is 0.0001 created ACTIVITY_APPLY_P315405580_A table. Both of them eliminated 640 suspicious transactions as the previous one. After all these models, two linear function based models whose tolerance values were fixed to 0.001 were created to eliminate more suspicious transactions. First of them whose outlier rate 0.01 created ACTIVITY_APPLY_P120630591_A table and eliminated 331 transactions and the other one whose outlier rate is 0.001 created ACTIVITY_APPLY_P735314841_A table and eliminated 620 transactions. In conclude, linear function based model whose tolerance value is 0.01 and outlier rate is 0.011 gave the best result.

After these experiments Gaussian function based models were created. Like working with linear function based models, first of all tolerance value was fixed to 0.1. A Gaussian function based model whose tolerance value is 0.1 and outlier rate is 0.01 was created. It was applied on activity apply data source. It created ACTIVITY_APPLY_P268062625_A table and did not eliminate any suspicious transactions. Then a new model whose tolerance value is 0.1 and outlier value is 0.001 was created. It created ACTIVITY_APPLY_P598259584_A table and eliminated only 4 suspicious transactions. Then a Gaussian function based model whose tolerance value is 0.1 and outlier rate is 0.00001was created. It created ACTIVITY_APPLY_P928515129_A and eliminated 7 suspicious transactions. So new models whose tolerance values are smaller than 0.1 were created. First a Gaussian function based model whose tolerance value is 0.001 and outlier rate is 0.001 was created. It was applied on activity apply source dataset. It created ACTIVITY_APPLY_P253500003_A table and eliminated 2 suspicious transactions. Then a model whose tolerance value is 0.00001 and outlier rate is 0.00001 was created. It created ACTIVITY_APPLY_P549597236_A table and could not eliminate any suspicious transactions.

In all of these models, odminer get claimUser, createUser, customerId, healthCenterId, invoiceAmount, lastVisitDay, and paidAmount columns as attributes of the build activity. It analyzed all columns and determined that benefitId column has constant values. So it did not use this column. To see the effects of selecting the columns, new models whose build columns are different were created. First of all same of the model which was created before and whose kernel function is linear, tolerance value is 0.01 and outlier rate is 0.001 was created. When building the model, all columns of the table were selected as attributes of this model. When benefitId column was selected to use odminer gave warning which is "This input appears to be a constant (a single value)", but it was selected. Then model was applied on activity apply source dataset. It created ACTIVITY_APPLY_P582025455_A table and it eliminated 218 suspicious transactions. Then another linear function based model that when it was being created all columns were selected was created. Its tolerance value was set to 0.1 and outlier rate was set to 0.001. It created ACTIVITY_APPLY_P823831879_A table and eliminated 69 suspicious transactions. As the changing selected column affects the result of models, new Gaussian function based detection models were created to get better results. First of all new model whose tolerance value is 0.001 and outlier rate is 0.001 was created. It was applied on activity apply source dataset and it created ACTIVITY_APPLY_P433934817_A table. But it could not eliminate any suspicious transactions. Then a new Gaussian function based model created. Its tolerance value was set to 0.1 and outlier rate was set to 0.001. This model created ACTIVITY_APPLY_P181009808_A table but also it did not eliminate any suspicious transactions. **Table III.10** shows the results of created models.

Table III.10 Models Results for Fraud Type 2

Model Number	Function Name	Tolerance Value	Outlier Rate	Selected Column	Correctly Predicted	Table Name
1	Linear	0.1	0.01	Default	453	ACTIVITY_APPLY_P720662032_A
2	Linear	0.1	0.001	Default	80	ACTIVITY_APPLY_P699546311_A
3	Linear	0.1	0.0001	Default	80	ACTIVITY_APPLY_P213912614_A
4	Linear	0.1	0.00001	Default	80	ACTIVITY_APPLY_P895635615_A
5	Linear	0.01	0.01	Default	333	ACTIVITY_APPLY_P403952277_A
6	Linear	0.01	0.001	Default	73	ACTIVITY_APPLY_P586786320_A
7	Linear	0.01	0.0001	Default	73	ACTIVITY_APPLY_P868726764_A
8	Linear	0.01	0.00001	Default	73	ACTIVITY_APPLY_P315405580_A
9	Linear	0.001	0.01	Default	382	ACTIVITY_APPLY_P120630591_A
10	Linear	0.001	0.001	Default	93	ACTIVITY_APPLY_P735314841_A
11	Linear	0.01	0.001	All	495	ACTIVITY_APPLY_P582025455_A
12	Linear	0.1	0.001	All	644	ACTIVITY_APPLY_P823831879_A
13	Gaussian	0.1	0.01	Default	713	ACTIVITY_APPLY_P268062625_A
14	Gaussian	0.1	0.001	Default	709	ACTIVITY_APPLY_P598259584_A
15	Gaussian	0.1	0.00001	Default	706	ACTIVITY_APPLY_P928515129_A
16	Gaussian	0.001	0.001	Default	711	ACTIVITY_APPLY_P253500003_A
17	Gaussian	0.00001	0.00001	Default	713	ACTIVITY_APPLY_P549597236_A
18	Gaussian	0.001	0.001	All	713	ACTIVITY_APPLY_P433934817_A
19	Gaussian	0.1	0.001	All	713	ACTIVITY_APPLY_P181009808_A

III.7.3 FRAUD TYPE 3

The third selected fraud type is “health centers whose average incident cost bigger than the average incident cost of all health centers and rate of the payment to directly insured is bigger than the average payment to directly insured rate of all health centers”. Normally, the insured goes to the hospital or pharmacy and then consults to the insurance company to claim his money that he spent in health center back. Sometimes the insured doesn’t give money to health center; he shows his insurance document to the health center and the health center claims this money from insurance company directly. Health insurance experts think that if the average incident cost of a health center is bigger than overall average incident cost of all health centers and the health centers’ rate of the payment directly to insured is bigger than the average of the payment directly to insured of all health centers, that health center must be examined. That health center can make fraud with the collaboration of the insured.

Before starting the build models, statistical information of hospitals and pharmacies was collected. **Figure III.31** shows the SQL statement to take statistic information of hospitals and pharmacies.

```
SELECT  hct.typeid
        hct.typedesc,
        SUM(cd.payableamount),
        AVG(cd.payableamount),
        MIN(cd.payableamount),
        MAX(cd.payableamount),
        COUNT(DISTINCT cd.benefitid),
        COUNT(c.claimid),
        COUNT(cd.detaillineid),
        COUNT(DISTINCT c.customerid),
        COUNT(DISTINCT c.healthcenterid),
        SUM(c.wheretopayid-1),
        sum(decode(c.wheretopayid,1,cd.payableamount,0)),
        sum(decode(c.wheretopayid,2,cd.payableamount,0))
FROM    HealthCenter hc,
        HealthCenterType hct,
        Claim c,
        ClaimDetail cd
WHERE   c.ClaimID          = cd.ClaimID AND
        c.HealthCenterID = hc.HealthCenterID AND
        hc.TypeID          = hct.TypeID
GROUP BY hct.typedesc
```

Figure III.31 SQL Statement for Taking Health Centers' Statistics

Statistical work gives the average values and other valuable information. **Table III.11** shows the statistic information of health centers.

Table III.11 Statistical Information of Health Centers

ID	NAME	Share (%)	Paid Amount	Average Paid Amount (%)	Claim Count	Customer Count	Health Center Count	Payment Directly Customer (PDC)	PDC Rate (%)
1	Hospital	71	125.299.391	370	338.629	52.601	740	33.318	10
2	Pharmacy	7	11.971.964	40	300.775	43507	8.474	55.721	19

In this type of fraud, two datasets must be prepared. One of them is for hospitals and the other is for pharmacies. Because the average incident cost and rate of payment directly to insured are different for both of them. Also two types of models have to be created. Health center table contains information of health centers in insurance company context. TypeId column in this column keeps the health centers' type. Hospital's type is 1 and pharmacy's type is 2. To whom the insurance company paid the claim is kept on whereToPay column in claim table. Paying the claim to health center's whereToPay id is 1 and paying the claim to insured's whereToPay id 2. CustomerId column keeps the customer of this claim. Also statusId column keeps the claims status. If status id is 2, it means insurance company rejected that claim and did not pay the cost of it. PayableAmount column keeps the paid amount of the claim. As the detection model can be adversely affected by missing value in source data or worthless claim, data whose payable amount value equal to zero must be eliminated. Also rejected claims' transactions must be eliminated. First of all a table that contains all hospitals and their informative feature was created. There are 740 hospitals in insurance company system. But 10 of them have inconsistent data. These were eliminated when source datasets were being created. Consequently source dataset has 730 rows. **Figure III.32** shows the SQL statement to create source table for hospitals. Risk column was set to -1 for future update.

```
CREATE TABLE KurumAnalizHastane AS
SELECT  hc.healthcenterid KurumNo,
        hc.DESCRIPTION KurumADI,
        hc.typeid KurumTipi,
        SUM(cd.payableamount) ToplamTutar,
        Round(AVG(cd.payableamount),2) OrtalamaTutar,
        MIN(cd.payableamount)AsgariTutar,
        MAX(cd.payableamount) AzamiTutar,
        COUNT(DISTINCT cd.benefitid) TeminatAdet,
        COUNT(cd.detaillineid) IslemAdet,
        COUNT(DISTINCT c.customerid) MusteriAdet,
        sum(decode(c.wheretopayid,1,cd.payableamount,0)) Kuruma,
        sum(decode(c.wheretopayid,2,cd.payableamount,0)) Sigortaliya,
        -1 Risk
FROM    HealthCenter hc, Claim c, ClaimDetail cd, benefit b
WHERE   c.ClaimID        = cd.ClaimID AND
        c.HealthCenterID = hc.HealthCenterID AND
        cd.benefitid     = b.benefitid AND
        hc.typeid        = 1 AND
        cd.payableamount > 0 AND  c.statusid       != 2
GROUP BY hc.healthcenterid, hc.DESCRIPTION, hc.Typeid
```

Figure III.32 SQL Statement for Creating Hospital Source Table

According to the results of statistical information about health center, health insurance fraud experts designated that hospital whose average incident cost bigger than 740 or the rate of the payment directly to insured is bigger than 0.35 is have to be examined. So Risk column was set to 0 for health centers whose average incident cost smaller than or equal to 740 and the rate of the payment directly to insured is smaller than or equal to 0.35. **Figure III.33** shows the SQL statement to update hospital kurumAnalizHastane table for set Risk column of normal row to 0.

```
update kurumanalizHastane ka
 set ka.risk = 0
 where ka.kurumtipi = 1 and
       ka.ortalamatutar <= 740 and
       (ka.sigortaliya / ka.toplamtutar ) <= 0.35
```

Figure III.33 SQL Statement for Updating Hospital Source Table

After setting the normal rows’ risk column 0, remaining rows’ risk column was set to 1 as suspicious. **Figure III.34** shows the SQL statement to update remaining rows’ as suspicious.

```
update kurumanalizHastane ka

  set ka.risk = 1

  where ka.risk=-1
```

Figure III.34 SQL Statement for Updating Hospital Source Table as Suspicious

KurumAnalizHastane table will be used as source table of activity apply step. For activity build step, a source must be created. As activity build step is required a source that has only normal data, it was created from kurumAnalizHastane where risk column equals 0. Build activity source dataset has 254 rows. **Figure III.35** shows the SQL statement to create hospital activity build source.

```
create view kurumanalizHastane_buildsource as
select * from kurumanalizHastane where risk=0
```

Figure III.35 SQL Statement for Creating Hospital Activity Build Source for Fraud Type 3

After creating sources for data mining activity, build activity step was started. As in before fraud types, anomaly detection model was created and One-Class Support Vector Machine algorithm was selected. KurumAnalizHastane_buildSource view is selected, and the attribute kurumNo column is designated as the unique identifier. Default settings were selected. As it was explained before, in default settings kernel function was left to odminer's option. It can select linear or Gaussian kernel functions. Also tolerance value was set to 0.001, outlier rate was set to 0.1 and active learning option was set to enable. Then the build activity was finished.

After building activity, this activity was applied on whole data. Completed build activity was selected to be used for creating an apply activity. Then the source table whose name is kurumAnalizHastane was selected, which contains normal and suspicious data, for applying the model. Then the supplemental columns were selected to include in the apply output table along with the standard prediction column. Apply activity creates a new table. Apply activity created KURUMANALIZHASTA891781945_A table. There are 476 suspicious data. Activity determined 376 correctly predicted data. As a linear kernel works well when there are many attributes in the training data, the default kernel function must be determined by the algorithm based on the number of attributes in the training data. Oracle data miner controlled the number of attributes in build activity and determined to use Gaussian kernel function. Also it determines tolerance value 0.001 and outlier rate 0.1.

As the result is not adequate, new activity models were created with different advanced settings. The outlier rate of this fraud type was learned from health insurance experts. Experts think that the outlier rate of this fraud type is approximately 0.1 %.

First of all linear function based models whose tolerance values were fixed to 0.1 and outlier rates were modified were created. A linear function based model whose tolerance value is 0.1 and outlier value is 0.01 was created. It applied on activity apply source dataset. It created KURUMANALIZHASTA194325056_A table and eliminated 262 suspicious hospitals. After this model, four new models whose tolerance values are 0.1 and outlier rates are 0.001, 0.0001, 0.00001, and 0.000001 respectively. They created KURUMANALIZHASTA593400274_A, KURUMANALIZHASTA818823531_A, KURUMANALIZHASTA556394574_A, KURUMANALIZHASTA353955365_A table respectively. But both of them eliminated 262 suspicious hospitals like previous model. Then a linear function based model whose tolerance value is 0.001 and outlier rate is 0.001 was created. It created KURUMANALIZHASTA163621979_A table and eliminated 224 suspicious hospitals. After this model two linear function based models were created. Their tolerance values were 0.0001, 0.00001 respectively and outlier rates were 0.0001. First of them created KURUMANALIZHASTA218614571_A table, the other one created KURUMANALIZHASTA838507496_A table. Both of them eliminated 224 hospitals too. After these experiments, Gaussian function based models were created. First of all five models were created with 0.1 tolerance value. A Gaussian function based model whose tolerance value is 0.1 and outlier rate is 0.01 was created. It created KURUMANALIZHASTA629772554_A table and eliminated 260 suspicious hospitals. To increase the eliminated hospital number outlier rate was decreased. A model whose tolerance value is 0.1 and outlier value is 0.001 created KURUMANALIZHASTA919110957_A table and eliminated 282 suspicious hospitals. The other models that were created with 0.0001, 0.00001, 0.00001 outlier rate eliminated 282 hospitals too. Then a Gaussian function based model whose tolerance value is 0.001 and outlier rate is 0.001 was created. It created KURUMANALIZHASTA856400540_A table and eliminated 278 hospitals. Then a new model with 0.0001 tolerance value and 0.0001 outlier rate was created. It eliminated 277 suspicious hospitals. A model whose tolerance value is 0.00001 and outlier rate is 0.0001 eliminated same numbers of suspicious hospitals.

In all of these models, odminer get asgariTutar, azamiTutar, islemAdet, kuruma, kurumAdi, musteriAdet, ortalamaTutar, sigortliya, teminatAdet and toplamTutar columns as attributes of the build activity. It analyzed all columns and determined that kurumTipi column has constant values, and kurumAdi column’s type is varchar. So it did not use these columns. To see the affects of selecting the columns, new models whose build columns are different were created. First of all same of the model which was created before and whose kernel function is linear, tolerance value is 0.1 and outlier rate is 0.0001 was created. When building the model, all columns of the table were selected as attributes of this model. When kurumTipi column was selected to use odminer gave warning which is “This input appears to be a constant (a single value)”, but it was selected. Then model was applied on activity apply source dataset. It created KURUMANALIZHASTA691471007_A table and it eliminated 205 suspicious hospitals. Then a new linear function based model was created with 0.001 tolerance value and 0.001 outlier rate. It created KURUMANALIZHASTA789463079_A table and eliminated 124 hospitals. As the last of the linear function based model, a model was created with 0.0001 tolerance value and 0.0001 outlier rate. It created KURUMANALIZHASTA872574897_A table and eliminated 123 hospitals. After these models three Gaussian function based models were created with selecting all columns as attributes of them. First a model whose tolerance value is 0.1 and outlier rate is 0.001 was created. It eliminated 203 suspicious hospitals. Then two Gaussian based models were created with 0.001, 0.0001 tolerance values and 0.001, 0.0001 outlier rates respectively. First of them eliminate 124 suspicious hospitals and the other one eliminated 123 suspicious hospitals. **Table III.12** shows the results of created models.

Table III.12 Hospital Models Results for Fraud Type 3

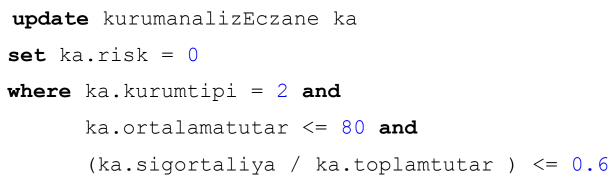
Model Number	**Function Name**	**Tolerance Value**	**Outlier Rate**	**Selected Column**	**Correctly Predicted**	**Table Name**
1	Gaussian	0.001	0.1	Default	376	KURUMANALIZHASTA891781945_A
2	Gaussian	0.1	0.001	Default	194	KURUMANALIZHASTA919110957_A
3	Gaussian	0.1	0.0001	Default	194	KURUMANALIZHASTA605731790_A
4	Gaussian	0.1	0.01	Default	216	KURUMANALIZHASTA629772554_A
5	Gaussian	0.1	0.00001	Default	194	KURUMANALIZHASTA226187905_A
6	Gaussian	0.1	0.000001	Default	194	KURUMANALIZHASTA467311318_A
7	Gaussian	0.00001	0.0001	Default	199	KURUMANALIZHASTA684926161_A
8	Gaussian	0.0001	0.0001	Default	199	KURUMANALIZHASTA342256137_A
9	Gaussian	0.001	0.001	Default	198	KURUMANALIZHASTA856400540_A
10	Gaussian	0.1	0.001	All	273	KURUMANALIZHASTA768662810_A
11	Gaussian	0.001	0.001	All	352	KURUMANALIZHASTA567067999_A
12	Gaussian	0.0001	0.0001	All	353	KURUMANALIZHASTA975917401_A
13	Linear	0.1	0.001	Default	214	KURUMANALIZHASTA593400274_A
14	Linear	0.1	0.0001	Default	214	KURUMANALIZHASTA818823531_A
15	Linear	0.1	0.01	Default	214	KURUMANALIZHASTA194325056_A
16	Linear	0.1	0.00001	Default	214	KURUMANALIZHASTA556394574_A
17	Linear	0.1	0.000001	Default	214	KURUMANALIZHASTA353955365_A
18	Linear	0.00001	0.0001	Default	252	KURUMANALIZHASTA838507496_A
19	Linear	0.0001	0.0001	Default	252	KURUMANALIZHASTA218614571_A
20	Linear	0.001	0.001	Default	252	KURUMANALIZHASTA163621979_A
21	Linear	0.1	0.0001	All	271	KURUMANALIZHASTA691471007_A
22	Linear	0.001	0.001	All	352	KURUMANALIZHASTA789463079_A
23	Linear	0.0001	0.0001	All	353	KURUMANALIZHASTA872574897_A

After working on this fraud type for hospital, a table that contains all pharmacy and their informative feature was created. To get only pharmacies from insurance company, typeId column in healthCenter column was set to 2. As the detection model can be adversely affected by missing values in source data or worthless claim, data whose payable amount value is equal to zero must be eliminated. Also rejected claims' transactions must be eliminated. There are 8474 pharmacies in insurance company system. But 138 of them have inconsistent data. These were eliminated when source datasets were being created. Consequently source dataset has 8336 rows. Figure **III.36** shows the SQL statement to create source table for hospitals. Risk column was set to -1 for future update.

According to results of statistical information about health center, health insurance fraud experts designated that pharmacy whose average incident cost bigger than 80 or the rate of the payment directly to insured is bigger than 0.6 is have to be examined. So Risk column was set to 0 for health centers whose average incident cost smaller than or equal to 80 and the rate of the payment directly to insured is smaller than or equal to 0.6. **Figure III.37** shows the SQL statement to update pharmacy kurumAnalizEczane table for set Risk column of normal row to 0.

```
CREATE TABLE KurumAnalizEczane AS
SELECT  hc.healthcenterid KurumNo,
        hc.DESCRIPTION KurumADI,
        hc.typeid KurumTipi,
        SUM(cd.payableamount) ToplamTutar,
        Round(AVG(cd.payableamount),2) OrtalamaTutar,
        MIN(cd.payableamount)AsgariTutar,
        MAX(cd.payableamount) AzamiTutar,
        COUNT(DISTINCT cd.benefitid) TeminatAdet,
        COUNT(cd.detaillineid) IslemAdet,
        COUNT(DISTINCT c.customerid) MusteriAdet,
        sum(decode(c.wheretopayid,1,cd.payableamount,0)) Kuruma,
        sum(decode(c.wheretopayid,2,cd.payableamount,0)) Sigortaliya,
        -1 Risk
FROM    HealthCenter hc,
        Claim c,
        ClaimDetail cd,
        benefit b
WHERE   c.ClaimID          = cd.ClaimID AND
        c.HealthCenterID = hc.HealthCenterID AND
        cd.benefitid       = b.benefitid AND
        hc.typeid          = 2 AND
        cd.payableamount > 0 AND
        c.statusid        != 2
GROUP BY hc.healthcenterid,
        hc.DESCRIPTION,
        hc.Typeid
```

Figure III.36 SQL Statement for Creating Pharmacy Source Table



```
update kurumanalizEczane ka
set ka.risk = 0
where ka.kurumtipi = 2 and
      ka.ortalamatutar <= 80 and
      (ka.sigortaliya / ka.toplamtutar ) <= 0.6
```

Figure III.37 SQL Statement for Updating Pharmacy Source Table

After setting the normal rows' Risk column 0, remaining rows' risk column was set to 1 as suspicious. **Figure III.38** shows the SQL statement to update remaining rows' as suspicious.

```
update kurumanalizEczane ka
  set ka.risk = 1
  where ka.risk=-1
```

Figure III.38 SQL Statement for Updating Pharmacy Source Table as Suspicious

KurumAnalizEczane table was going to be used as source table of activity apply step. For activity build step a source must be created. As activity build step is required a source that has only normal data, it was created from kurumAnalizEczane where Risk column equals 0. Activity apply source dataset has 733 rows. **Figure III.39** shows the SQL statement to create pharmacy activity build source.

```
create view kurumanalizEczane_buildsource as
select * from kurumanalizEczane where risk=0
```

Figure III.39 SQL Statement for Creating Pharmacy Activity Build Source for Fraud Type 3

After creating sources for data mining activity, build activity step was started. As in before fraud types, anomaly detection model was created and One-Class Support Vector Machine algorithm was selected. KurumAnalizEczane_buildSource view is selected, and the attribute kurumNo column is designated as the unique identifier. Default settings were selected. As it was explained before, in default settings kernel function was left to odminer's option. It can select linear or Gaussian kernel functions. Also tolerance value was set to 0.001, outlier rate was set to 0.1 and active learning option was set to enable. Then the build activity was finished.

After building activity, this activity was applied on whole data. Completed build activity was selected to be used for creating an apply activity. Then the source table whose name is kurumAnalizEczane was selected, which contains normal and suspicious data, for applying the model. Then the supplemental columns were selected to include in the apply output table along with the standard prediction column. Apply activity creates a new table. Apply activity created KURUMANALIZECZAN76715533_Atable. There are 7603 suspicious data. Activity set 601 rows' prediction column to 0 that means these pharmacies have to be determined. Their transactions can be fraudulent. As a linear kernel works well when there are many attributes in the training data, the default kernel function must be determined by the algorithm based on the number of attributes in the training data. Oracle data miner controlled the number of attributes in build activity and determined to use Gaussian kernel function. Also it determines tolerance value 0.001 and outlier rate 0.1.

As the result is not adequate, new activity models were created with different advanced settings. The outlier rate of this fraud type was learned from health insurance experts. Experts think that the outlier rate of this fraud type is approximately 0.1 %.

First of all linear function based models whose tolerance values were fixed to 0.1 and outlier rates were modified were created. A linear function based model whose tolerance value is 0.1 and outlier value is 0.01 was created. It applied on activity apply source dataset. It created KURUMANALIZECZAN216190019_A table and eliminated 6111 suspicious pharmacies. After this model, four new models whose tolerance values are 0.1 and outlier rates are 0.001, 0.0001, 0.00001, and 0.000001 respectively. They created KURUMANALIZECZAN548838062_A, KURUMANALIZECZAN47085897_A, KURUMANALIZECZAN99074880_A, KURUMANALIZECZAN27940039_A table respectively. But both of them eliminated 6111 suspicious pharmacies like previous model. Then tolerance value was fixed to 0.01 and four models were created with different outlier rate. A linear function based model whose tolerance value is 0.01 and outlier rate is 0.01 was created. It created KURUMANALIZECZAN223412803_A table and eliminated 5932 suspicious pharmacies. After this model three linear function based models whose tolerance values are 0.01 were created. Their outlier rates were 0.001, 0.0001, and 0.00001 respectively. All of them eliminated 5932 suspicious pharmacies. Then a linear function based model whose tolerance value is 0.001 and outlier rate is 0.1 was created. It was applied on activity apply dataset. It created

KURUMANALIZECZAN732687684_A table and eliminated 4622 suspicious pharmacies. After this model a model whose tolerance value is 0.001 and outlier rate is 0.001 was created. It created KURUMANALIZECZAN848141643_A table and eliminated 5874 suspicious pharmacies. As the last linear function based model, a model whose tolerance value is 0.0001 and outlier rate is 0.001 was created. It created KURUMANALIZECZAN892723864_A table and eliminated 5867 suspicious pharmacies. After these experiments, Gaussian function based models were created. First of all five models were created with 0.1 tolerance value. A Gaussian function based model whose tolerance value is 0.1 and outlier rate is 0.01 was created. It created KURUMANALIZECZAN650457884_A table and eliminated 5797 suspicious pharmacies. To increase the eliminated pharmacies number outlier rate was decreased. A Gaussian function based model whose tolerance value is 0.1 and outlier rate is 0.001 was created. It created KURUMANALIZECZAN205597873_A table and eliminated 7084 suspicious pharmacies. Then three Gaussian function based models with 0.1 tolerance value was created. Their outlier rates were 0.0001, 0.00001, and 0.000001 respectively. All of them eliminated 7084 suspicious pharmacies. To get better results, new Gaussian function based models were created with smaller tolerance value. Four models whose tolerance values are fixed to 0.01 and outlier rates are different were created. First of them whose outlier rate is 0.01 created KURUMANALIZECZAN998006237_A table and eliminated 7029 suspicious pharmacies. The other three models whose outlier rates are 0.001, 0.0001, and 0.00001 respectively eliminated 7064 suspicious pharmacies. At last two models whose tolerance values are 0.001 and 0.0001 respectively and outlier rates are 0.001 were created. First one created KURUMANALIZECZAN852697408_A table. The other one created KURUMANALIZECZAN63769779_A table. Both of them eliminated 7062 suspicious pharmacies.

In all of these models, odminer gets asgariTutar, azamiTutar, islemAdet, kuruma, kurumAdi, musteriAdet, ortalamaTutar, sigortliya, teminatAdet and toplamTutar columns as attributes of the build activity. It analyzed all columns and determined that kurumTipi column has constant values, and kurumAdi column’s type is varchar. So it did not use these columns. To see the affects of selecting the columns, new models whose build columns are different were created. First of all same of the model which was created before and whose kernel function is linear, tolerance value is 0.1 and outlier rate is 0.001 was created. When building the model, all columns of the table were selected as attributes of this model. When kurumTipi column was selected to use odminer gave warning which is “This input appears to be a constant (a single value)”, but it was selected. Then model was applied on activity apply source dataset. It created KURUMANALIZECZAN1839730_A table and it eliminated 6205 suspicious pharmacies. Then a new linear function based model was created with 0.01 tolerance value and 0.001 outlier rate. It created KURUMANALIZECZAN859867518_A table and eliminated 4546 pharmacies. As the last of the linear function based model, a model was created with 0.001 tolerance value and 0.001 outlier rate. It created KURUMANALIZECZAN862394184_A table and eliminated 5197 pharmacies. After these models three Gaussian function based models were created with selecting all columns as attributes of them. First a model whose tolerance value is 0.1 and outlier rate is 0.001 was created. It eliminated 5773 suspicious pharmacies. Then two Gaussian based models were created with 0.01, 0.001 tolerance values respectively and 0.001 outlier rates. First of them eliminate 4546 suspicious pharmacies and the other one eliminated 4317 suspicious pharmacies. **Table III.13** shows the results of created models.

Table III.13 Pharmacy Models Results for Fraud Type 3

Model	**Function**	**Tolerance**	**Outlier**	**Selected**	**Correctly**	**Table Name**
Number	**Name**	**Value**	**Rate**	**Column**	**Predicted**	
1	Gaussian	0.001	0.1	Default	601	KURUMANALIZECZAN76715533_A
2	Gaussian	0.1	0.001	Default	519	KURUMANALIZECZAN205597873_A
3	Gaussian	0.1	0.01	Default	1806	KURUMANALIZECZAN650457884_A
4	Gaussian	0.1	0.0001	Default	519	KURUMANALIZECZAN599648301_A
5	Gaussian	0.1	0.00001	Default	519	KURUMANALIZECZAN174432342_A
6	Gaussian	0.1	0.000001	Default	519	KURUMANALIZECZAN681034554_A
7	Gaussian	0.01	0.001	Default	539	KURUMANALIZECZAN604556256_A
8	Gaussian	0.001	0.001	Default	541	KURUMANALIZECZAN852697408_A
9	Gaussian	0.0001	0.001	Default	544	KURUMANALIZECZAN330546300_A
10	Gaussian	0.01	0.01	Default	574	KURUMANALIZECZAN998006237_A
11	Gaussian	0.01	0.0001	Default	539	KURUMANALIZECZAN954360164_A
12	Gaussian	0.01	0.00001	Default	539	KURUMANALIZECZAN63769779_A
13	Gaussian	0.1	0.001	All	1830	KURUMANALIZECZAN325539355_A
14	Gaussian	0.01	0.001	All	3057	KURUMANALIZECZAN720032268_A
15	Gaussian	0.001	0.001	All	3286	KURUMANALIZECZAN794808268_A
16	Linear	0.001	0.1	Default	2981	KURUMANALIZECZAN732687684_A
17	Linear	0.1	0.001	Default	1492	KURUMANALIZECZAN548838062_A
18	Linear	0.1	0.01	Default	1492	KURUMANALIZECZAN216190019_A
19	Linear	0.1	0.0001	Default	1492	KURUMANALIZECZAN47085897_A
20	Linear	0.1	0.00001	Default	1492	KURUMANALIZECZAN99074880_A
21	Linear	0.1	0.000001	Default	1492	KURUMANALIZECZAN27940039_A
22	Linear	0.01	0.001	Default	1671	KURUMANALIZECZAN535933532_A
23	Linear	0.001	0.001	Default	1729	KURUMANALIZECZAN848141643_A
24	Linear	0.0001	0.001	Default	1736	KURUMANALIZECZAN892723864_A
25	Linear	0.01	0.01	Default	1671	KURUMANALIZECZAN223412803_A
26	Linear	0.01	0.0001	Default	1671	KURUMANALIZECZAN549745207_A
27	Linear	0.01	0.00001	Default	1671	KURUMANALIZECZAN1568861_A
28	Linear	0.1	0.001	All	1398	KURUMANALIZECZAN1839730_A
29	Linear	0.01	0.001	All	3057	KURUMANALIZECZAN859867518_A
30	Linear	0.001	0.001	All	2406	KURUMANALIZECZAN862394184_A

CHAPTER IV

RESULTS AND DISCUSSIONS

In this “Master of Science Thesis”, a data mining based fraud detection system is designed for insurance companies. The main purpose of this thesis is to support the health insurance sector by developing a robust system for fraud detection and prevention.

There are different structural types of fraud detection system. In order to determine the most suitable type of fraud detection system, types of fraud detection system were examined. Some of the very well known methods for anomaly detection are time series analysis based [19, 20], one class support vector machine based [21], distance based [23, 24, 25, 30], clustering based [26, 27, 45, 28, 29], artificial intelligence based [31, 34, 44], decision trees based [37], classification based [39], and clinical pathway based [40]. There are several databases and data mining tools available in the market. After examining some of them Oracle 11G database and Oracle data mining tool were selected to use. A new database was created for fraud detection system. A huge data provided by some insurance companies was imported into the new database. Three health insurance fraud types were determined to investigation. These are the claims whose paid amount is greater than its invoice amount that insurance company will pay, transactions which is medicine taking without doctor inspection or surgical operation within four days, and health centers whose average incident cost bigger than the average incident cost of all health centers and rate of the payment to directly insured is bigger than the average payment to directly insured rate of all health centers. In the last fraud type, health centers were separated into two types as hospital and pharmacy, and models were created for both of them.

In the first health insurance fraud type, twenty five models were created. Fourteen of them are created with linear kernel function, and eleven of them are created with Gaussian kernel function. Fraud detection models that created with

Gaussian kernel function didn’t give adequate results. They couldn’t eliminate sufficient suspicious rows. Maximum five suspicious rows were eliminated. After these results, fraud detection model created with linear kernel functions were focused. As the health insurance experts said this type of fraud rate is 0.1 %, first of all five models were created with 0.001 outlier rate and different tolerance value. One of these five models whose tolerance value is 0.001 gave the best result. Then five models whose tolerance values were 0.001 were created. They could not give better results. In these models, columns were determined by ODM as attributes of model. To see the affects of changing selected columns in building activity on results, four models were created. As the model whose outlier rate is 0.001 and tolerance value 0.001 gave best result, two linear kernel function based models whose outlier rate are 0.001 and tolerance values are 0.001 were created. When first of them was being created, all columns were selected as attributes of the model, and when the other model was being created, only ClaimId, createDate and invoiceDate were added to default columns as attributes of the model. Like these activities, two new models whose attribute columns were changed were created too. Their outlier rates are 0.001 and tolerance values are 0.1 and 0.01 respectively. As the effect of changing selected columns on the result was seen, four new Gaussian kernel function based models were created. However, in Gaussian based models changing columns did not affect the results.

Consequently fraud detection model, whose kernel function is linear kernel function, tolerance value is 0.001, outlier rate is 0.001, and which ClaimId, createDate and invoiceDate columns were added to its default attributes gave sufficient results. 849 suspicious rows were in source table and this model eliminated 183 rows. When the models were created with smaller outlier rate, the number of eliminated rows was decreased. Also when the models were created with bigger tolerance value, the number of eliminated rows was decreased. After this elimination, ten rows that have biggest probability were listed. **Figure IV.1** shows the SQL statement that lists the ten rows.

```
SELECT *
FROM
(SELECT  T.DETAILLINEID , T.RISK , T.PREDICTION ,
T.PROBABILITY ,
F.INVOICEAMOUNT , F.COINSURANCE , F.PAIDAMOUNT ,
F.PAIDAMOUNT - (F.INVOICEAMOUNT * ((100-F.COINSURANCE)/100)) AS
DIFFERENCE
FROM     DENEME_SOURCE_FU870927537_A T ,
         DENEME_SOURCE_FULLTEST F
WHERE    T.RISK = 1 AND
         T.PREDICTION = 0  AND
         T.DETAILLINEID = F.DETAILLINEID
ORDER BY T.PROBABILITY DESC )
WHERE ROWNUM < 11
```

Figure IV.1 SQL Statement for Getting Top Most Probability Claims for Fraud Type 1

Result of this SQL statement is shown in **Table IV.1**. Health insurance expert investigated these claims and made a decision that the last three claims are not fraudulent transactions. Differences between paid amount and multiplication of invoice amount and the ratio of health insurance responsibility were arisen because of rounding. First three claims are fraudulent claims. Experts found more than one another transaction of same insured like that. Fourth claim is not fraudulent. Experts examined all other relations of this transaction in all database tables. They controlled the payment of this claim, talked with the accountant of the company and learned that insurance company paid this amount for one of the director’s friend. This transaction is very important for this study. Although it is not a fraudulent transaction, it shows this model’s success on finding the anomalous transactions. The other three of them have to be examined in more detailed. As a conclusion, applicable kernel function of this fraud type is linear kernel function.

Table IV.1 Top Ten Claims for Fraud Type 1

DetailLineId	Risk	Prediction	Probability	Invoice Amount	Coinsurance	Paid Amount	Difference
165896	1	0	0.92827200	32013.660	0.00	60000.0	27986.34
834084	1	0	0.92213886	345.60000	20.00	320.000	43.52
831073	1	0	0.92200963	45.490000	20.00	45.4900	9.098
829139	1	0	0.92199427	73.170000	20.00	65.3000	161.464
828868	1	0	0.92194721	78.650000	20.00	67.9500	5.03
828434	1	0	0.92184732	1144.9800	20.00	916.020	0.036
829011	1	0	0.92086698	779.76000	20.00	623.820	0.012
828447	1	0	0.92050449	411.46000	20.00	329.180	0.012
833138	1	0	0.91328795	27.450000	30.00	19.2200	0.005
833775	1	0	0.91268032	77.470000	30.00	54.2300	0.001

In second health insurance fraud type, nineteen models were created. Twelve of them were created with linear kernel function, and seven of them were created with Gaussian kernel function. Fraud detection models that were created with Gaussian kernel function didn't give adequate results. They couldn't eliminate sufficient suspicious rows. Maximum seven suspicious rows were eliminated. After these results, fraud detection model created with linear kernel functions were focused. First of all tolerance values were fixed to 0.1 and outlier rates were modified to get best result. These models' whose tolerance value was set to 0.1, results were fixed when decreasing the outlier rate. No matter how decrease the outlier rate, models could not eliminate more than 633 transactions. Then new models whose tolerance values were fixed to 0.01 were created. Like the previous experiment, a model whose outlier rate is 0.001 eliminated 640 suspicious transactions and the other models whose outlier rates were smaller than 0.001 gave the same results. Then two models whose tolerance values were fixed to 0.001 were created but they gave worse results. In these models, columns were determined by ODM as attributes of model. To see the effects of changing selected columns in building activity on results, four models were created. Two models from linear function based models and two models from Gaussian function based models were fallowed. All columns were selected as attributes of models. But the new four models where all columns were selected gave worse results.

Consequently fraud detection model created with linear kernel function, 0.01 as tolerance value and 0.001 as outlier rate gave sufficient results. 713 suspicious rows were in source table and this model eliminated 640 rows. This result showed that health insurance experts are rightful because they have said that this type of fraud rate is 0.1 %. When the models were created with smaller outlier rate, the number of eliminated rows was decreased. Also when the models were created with bigger tolerance value, the number of eliminated rows was decreased.

After this elimination, ten rows that have biggest probability were listed. **Figure IV.2** shows the SQL statement that lists the ten rows.

```
SELECT *
FROM
(SELECT P.DETAILLINEID, T.RISK , T.PREDICTION, T.PROBABILITY,
P.HEALTHCENTERID, P.CLAIMUSER, P.LASTVISITDAY
FROM  ACTIVITY_APPLY_P586786320_A, ACTIVITY_APPLY_PHARM P
WHERE T.DETAILLINEID = P.DETAILLINEID AND
      T.RISK = 1 AND
      T.PREDICTION = 0
ORDER BY T.PROBABILITY DESC )
WHERE ROWNUM < 11
```

Figure IV.2 SQL Statement for Getting Top Most Probability Claims for Fraud Type 2

Result of this SQL statement is shown in **Table IV.2**. Health insurance expert investigated these claims and made a decision that second, third and fifth claims' fraudulent probability is very high, because they are originated from the same health center. These results throw suspicion on health center with health center id 32777. Because of the claim user of second, third, fourth, fifth, and eighth claims are the same; these transactions' probability of being fraudulent is very high. Likewise, first, seventh and ninth claims' fraudulent probability is high. Also the numbers of the same user's claims in ACTIVITY_APPLY_P586786320_A table that model determined as fraudulent were calculated. **Figure IV.3** shows the SQL statement that was used for this calculation.

Table IV.2 Top Ten Claims for Fraud Type 2

DetailLineId	Risk	Prediction	Probability	Health Center Id	Claim User	Last Visit Day
828483	1	0	0.72159561	30049	5103	22
828474	1	0	0.71027055	32777	5098	14
828476	1	0	0.70794895	32777	5098	14
828452	1	0	0.70621722	35434	5098	999
828475	1	0	0.70529115	32777	5098	14
828469	1	0	0.70258121	35157	5104	21
828466	1	0	0.69231544	38416	5103	999
828451	1	0	0.69049391	35434	5098	999
833508	1	0	0.65395804	446	5103	50
828388	1	0	0.65326590	2804	5100	999

Table IV.3 shows the numbers of same users' claims. With the help of this result, experts started to examine all transactions of these users. As a conclusion, applicable kernel function of this fraud type is linear kernel function.

```
SELECT P.CLAIMUSER , COUNT(P.CLAIMUSER)
FROM  ACTIVITY_APPLY_P586786320_A T ,
ACTIVITY_APPLY_PHARM P
WHERE T.DETAILLINEID = P.DETAILLINEID AND
      T.RISK = 1 AND
      T.PREDICTION = 0
group by p.claimuser
```

Figure IV.3 SQL Statement for Calculation of the Number of Claims of Same User

Table IV.3 Numbers of Claims of Same Users

CLAIM USER	COUNT OF CLAIMS
5100	10
5103	8
5097	8
5098	6
5094	6

In the third health insurance fraud type, two different model types were created. One of them was for hospital health centers and the other one for pharmacy. Source table of hospital model was kurumAnalizHastane and source table of pharmacy model was kuerumAnalizEczane. For the hospital model twenty three models were created. Eleven of them were created with linear kernel function, and twelve of them were created with Gaussian kernel function. First of all Linear function based models were created. Tolerance value was fixed to 0.1 and five models were created with different outlier rates. Their results are the same. They eliminated 262 suspicious hospitals. Then three linear function based models were created with different tolerance values and outlier rates. They all gave worse results. Then Gaussian function based models were created. Like previous work, first tolerance value was fixed to 0.1 and five models were created. A model whose outlier rate is 0.001 is determinative model. Models whose outlier rate is smaller than this model gave same result of it. Then three models with different outlier rate and tolerance value were created but no better result was obtained.

In these models, columns were determined by ODM as attributes of model. To see the effects of changing selected columns in building activity on results, six models were created. Three models from linear function based models and three models from Gaussian function based models were fallowed. All columns were selected as attributes of models. But new six models that all columns were selected gave worse results.

Consequently fraud detection model created with Gaussian kernel function, 0.1 tolerance value and 0.001 outlier rate gave sufficient results. 476 suspicious rows were in source table and this model eliminated 282 rows. Fraud detection models that created with linear kernel function didn’t give adequate results. They couldn’t eliminate suspicious hospitals as much as Gaussian function models eliminated. When the models were created with smaller outlier rate, the number of eliminated rows was decreased. Also when the models were created with bigger tolerance value, the number of eliminated rows was decreased. After this elimination, ten rows that have biggest probability were listed. **Figure IV.4** shows the SQL statement that lists the ten rows.

```
SELECT *
FROM
(SELECT H.KURUMNO, H.ORTALAMATUTAR,H.KURUMA ,
H.SIGORTALIYA, (H.sigortaliya / H.toplamtutar) ,
FROM KURUMANALIZHASTA919110957_A T , KURUMANALIZHASTANE
H
WHERE T.KURUMNO = H.KURUMNO AND
      T.RISK = 1 AND
      T.PREDICTION = 0
ORDER BY T.PROBABILITY DESC)
WHERE ROWNUM < 11
```

Figure IV.4 SQL Statement for Getting Top Most Probability Hospitals for Fraud Type 3

Result of this SQL statement is shown in **Table IV.4**. Health insurance expert investigated these hospitals and made a decision that first, second, six, seventh and eighth hospitals transactions could be fraudulent claims. Because their average cost are too high. Experts started to investigate all transactions of these hospitals. Third, fourth, fifth, ninth, and tenth hospitals transactions must be investigated too. Because insurance company pay all money to insured. Insurance company didn’t pay any money to these hospitals so insured could make fraud without hospitals knowledge.

Table IV.4 Top Ten Hospitals for Fraud Type 3

Healt CenterId	Average cost	To Health Center	To Insured	To Insured /Total Cost
26267	30141.7	0	693259.1	1
26397	23544.84	1935419.82	1384402.2	0.41701096
25540	50	0	50	1
25661	50	0	50	1
26654	50	0	50	1
25723	15057.09	0	301141.72	1
23264	819.07	17200.48	0	0
26389	12615.57	0	239695.92	1
26757	49.56	0	49.56	1
25482	48.6	0	48.6	1

For pharmacy detection type thirty models were created. Fifteen of them were created with linear kernel function, and fifteen of them were created with Gaussian kernel function. First of all Linear function based models were created. Tolerance value was fixed to 0.1 and five models were created with different outlier rate. Their results are the same. They eliminated 6111 suspicious pharmacies. Then four linear function based model whose tolerance values were fixed to 0.01 were created. They all eliminated 5932 suspicious pharmacies. To get better result, three models with different tolerance value and outlier rate were created but they gave worse results. After these experiments, Gaussian function based models were created. Like previous work, first tolerance value was fixed to 0.1 and five models were created. A model whose outlier rate is 0.001 is determinative model. Models whose outlier rates are smaller than this model's outlier rate gave same results. To eliminate more suspicious pharmacies four model with 0.01 tolerance values and the other three models with different tolerance value and outlier rate were created. But none of them gave better result.

In these models, columns were determined by ODM as attributes of model. To see the affects of changing selected columns in building activity on results, six models were created. Three models from linear function based models and three models from Gaussian function based models were fallowed. All columns were selected as attributes of models. But new six models that all columns were selected gave worse results.

Consequently fraud detection model created with Gaussian kernel function, 0.1 tolerance value and 0.001 outlier rate gave sufficient results. 7603 suspicious rows were in source table and this model eliminated 7084 rows. Fraud detection models that created with linear kernel function didn't give adequate results. They couldn't eliminate suspicious hospitals as much as Gaussian function models eliminated. When the models were created with smaller outlier rate, the number of eliminated rows was decreased. Also when the models were created with bigger tolerance value, the number of eliminated rows was decreased. After this elimination, ten rows that have biggest probability were listed. **Figure IV.5** shows the SQL statement that lists the ten rows.

```
SELECT *
FROM
(SELECT E.KURUMNO , E.ORTALAMATUTAR , E.KURUMA ,
E.SIGORTALIYA, E.SIGORTALIYA/E.TOPLAMTUTAR
FROM KURUMANALIZECZAN205597873_A T , KURUMANALIZECZANE E
WHERE T.RISK = 1 AND T.PREDICTION = 0 AND
      T.KURUMNO = E.KURUMNO
ORDER BY T.PROBABILITY  DESC )
WHERE ROWNUM < 11
```

Figure IV.5 SQL Statement for Getting Top Most Probability Pharmacy for Fraud Type 3

Result of this SQL statement is shown in **Table IV.5**. Health insurance expert investigated these pharmacies and made a decision that first, second, fifth, and eight pharmacies' transactions could be fraudulent claims. Because their average cost are too high. Experts started to investigate all transactions of these pharmacies. Third, fourth, fifth, ninth, and tenth pharmacies transactions must be investigate too. Because insurance company pay all money to insured. So insured could make fraud without pharmacies knowledge. As a conclusion, applicable kernel function of this fraud type is Gaussian kernel function.

Table IV.5 Top Ten Pharmacies for Fraud Type 3

Healt CenterId	Average cost	To Health Center	To Insured	To Insured /Total Cost
32017	909.77	0	14556.34	1
32615	623.32	0	5609.9	1
37009	57.82	0	9366.75	1
39020	57.07	0	3538.64	1
33486	574.67	0	4022.72	1
40163	16.6	0	13182.11	1
38177	27.34	0	7055	1
37978	182.31	0	9479.97	1
34473	28.01	0	5937.23	1
36436	32.24	0	1708.73	1

In these three fraud detection model one class support vector machine algorithm is used. Because one class support vector machine algorithm provides coefficients that are useful in understanding the relationship and patterns in the dataset. The SVM algorithms differ from the other algorithms by their adaptability for diverse types of data. One class support vector machine algorithm is simple and can be computed quickly. Data sets that include only normal rows were created and models were built on these source. Then these models are applied on datasets that includes normal and suspicious data. These tables have Risk column and were set 1 that represents suspicious cases. After applying models, they created anomaly detection output tables. These tables contain prediction column. This column has value 1 for a case determined to be “normal” and 0 for a “suspicious” case. Since the RISK column in the input data had value 0 for Low Risk and 1 for High Risk, the correctly predicted cases are those in which Risk equals 1 and prediction equals 0.

The models we constructed with the Gaussian kernel function did not show significantly stronger performance on first and second fraud types. Besides computational complexity of using Gaussian kernel model is higher than computational complexity of using linear kernel model. Models based on linear kernel function were suitable for these types. As a result of experiments in the first fraud type 0.001 tolerance value and 0.001 outlier rate, in second fraud type 0.01 tolerance value and 0.001 outlier rate were best choices for models based on linear kernel function.

In the third fraud type, two different detection model types were created and Gaussian kernel function based models whose tolerance value is 0.1 and outlier rate is 0.001 gave best result in both of them. Although the computational complexities of models based on Gaussian kernel function are higher than computational complexities of models based on linear kernel function, their performance is very satisfactory. All Gaussian kernel function based models demonstrate better performance in comparison with the linear kernel function based models on all third fraud type dataset.

When a model is being built, odminer analyzes all columns of activity build data source and determine which of them must be chosen for attributes of detection model. It does not take a column whose type is varchar. Also it controls the data of columns and if a column has a constant value or lots of null value, it does not select it as an attribute of the model. As a result of experiments, selecting all columns to create a detection model gives bad results. It decreases the sensitivity of detection model. Columns that are selected by odminer give sufficient results. But in some exceptional cases adding columns whose types are not varchar and have not much null value gives better results.

Tolerance value tells the algorithm when to stop building the model; increasing this value to a higher number will build the model faster but may be less accurate. If having some knowledge that the number of “suspicious” cases is a certain percentage of population, the outlier rate can be set to that percentage, and the model will identify approximately that many “rare” cases when applied to the general population. The default value that ODM gives is 10%, but this is too high for these fraud types.

The Complexity Factor prevents over-fitting by finding the best tradeoff between simplicity and complexity. The algorithm calculates and optimizes this value if a value is not specified. If the model skews its predictions in favor of one class, model may be rebuild with a manually-entered higher than the one calculated by the algorithm.

Active Learning is a methodology, internally implemented, that optimizes the selection of a subset of the support vectors which will maintain accuracy while enhancing the speed of the model. It is a method for controlling model growth and reducing model build time. It forces the support vector machine algorithm to restrict learning to the most informative training examples and not to attempt to use the entire body of data. In most cases, the resulting models have predictive accuracy comparable to that of a standard support vector machine model. Active Learning, in addition to increasing performance in the linear case, will reduce the size of the Gaussian model; this is an important consideration if memory and temporary disk space are issues.

In model based on Gaussian kernel function, activity learning, complexity factor, and the number of standard deviations are used to find the optimum medium between simplicity and complexity. A small value for sigma may cause over fitting, and a large value may cause excess complexity. The algorithm will calculate the ideal value internally.

The Gaussian kernel uses a large amount of memory in its calculations if Active Learning is not enabled. The default cache size is 50 Megabytes. Increasing the cache size may be required if the build operation seems very slow.

CHAPTER V

CONCLUDING REMARKS AND RECOMMENDATIONS

The result of the data mining based fraud detection system, which has been designed in this research, demonstrates that the whole system is applicable for health insurance companies. Health insurance companies try to detect frauds with the help of traditional method. In traditional method, medical experts take lots of medical transactions from insurance company's operational system. These transactions are chosen randomly. Medical experts investigate these transactions for detecting fraudulent activity. This method causes great waste of time and money. Also big portion of fraud events are omitted.

In the system that was created in this research, fraud detection models were created and they analyzed the normal data and then they were applied to dataset that contains normal and suspicious data. Finally they determined fraudulent transactions and members which have biggest fraudulent ratio. Medical experts take results of fraud detection system and control them. Experiments showed that, these models are successful for detecting fraudulent transactions and members.

The system that was created in this research is not a decision unit; it is a decision support system. Models were created to give an idea to health insurance fraud expert. With the help of these models, experts can focus on specific transactions and members in sector. Accordingly the ratio of detecting fraud events increases, and this fraud detection system causes big possession of time and money. Consequently this fraud detection system gives feasible and logical results and does its work properly.

In this research, support vector machine was used. The flexibility, scalability, and speed of SVM are better than the other methods. SVM performs well on data sets that have many attributes, even if there are very few cases on which to train the model. There is no upper limit on the number of attributes; the only constraints are those imposed by hardware. The other methods like traditional neural networks do not perform well under these circumstances.

Also Oracle data mining was used to create and apply models on source datasets. This system embeds data mining within the Oracle database. ODM algorithms operate natively on relational tables or views, thus eliminating the need to extract and transfer data into standalone tools or specialized analytic servers. Oracle Data Miner has its own proprietary implementation of SVM, which exploits the many benefits of the algorithm while compensating for some of the limitations inherent in the SVM framework. Oracle Data Mining SVM provides the scalability and usability that are needed in a production quality data mining system. Oracle Data Miner minimizes data preparation, tuning, and optimization. Oracle Data Mining SVM builds a model incrementally by optimizing small working sets towards a global solution.

There are four different important data mining tools which are IBM Intelligent Miner, SPSS Clementine, SAS Institute enterprise miner and Microsoft Business Intelligence Development Studio in the market. Each tool offers a wide range of functionalities to users. Researches on the performance and functionality of these tools in the market show that Clementine and SAS have stable, mature products that excel in nearly all aspects of data mining functionality. They have large market shares relative to the other players. In addition to market share, they are also distinguished by the breadth of their marketing programs, geographic coverage, technological investments, and commitment to quality implementation and support services. The other tools are primarily characterized by a slightly narrower scope of data mining functionality and less commitment to the industry compared to Clementine and SAS. The companies that created these tools are large software houses that offer various software solutions spanning multiple IT markets. Specializing in the data mining industry is not a priority for these organizations. So either Clementine or SAS can be selected and a new fraud detection system can be created with that tool.

As a future work, a graphical user interface based new software program can be written on this decision support system. This program can prepare all details of the fraudulent transactions and also can show all activities of fraudulent user or health center. It can help health insurance experts when they analyze and determine the suspicious transactions. This program can prepare graphical presentation of statistics of transactions or sector members' activity. Also future work can include experimenting with application of models that were created in this research to multi-classification problems and to other kernel methods.

This fraud detection system cannot designate fraud events when they occur. It analyzes the transactions and annotates the past processes. A rule based fraud detection systems, motif based fraud detection systems or decision tree algorithm based fraud detection systems must be used to meet the request of insurance companies that want to prevent fraud events while they occur. Rule based fraud detection systems perform early detection of fraudulent transactions by searching a database of health insurance companies for anomalous patterns. They characterize normal and abnormal behavior of users, transactions and patterns by a set of rules. The significance of each rule is carefully evaluated. Then when a new transaction happens, characteristics of this transaction are controlled if they obey the anomalous rules. Then transactions whose characteristics obey the fraudulent pattern rules are aborted.

For such a long time, health insurance sector's member can find new fraud types. Health insurance experts cannot anticipate these types. Therefore, unknown or modified patterns and situations will generally go undetected; hence, in practice, insurance sector can suffer from these fraud types. Artificial intelligence technology is the best solution of the problem of automatic detection of new candidate anomalous records in operation system. Learning algorithm based fraud detection models can help by finding these new fraud types. Naïve Bayes algorithm based fraud detection models and adaptive Bayes network algorithm based fraud detection models can also be used for such requirements. Also a software program can be written on these models. This program takes the experts' analyses about suspicious transactions that are determined by these models and gives these analyzes to the models as an input. Models can get these inputs and update their detection algorithm by increasing the weights of true positive anomalous fraud types and decreasing the weights of false positive anomalous fraud types.

In conclusion fraud detection system created in this research and the other algorithms based fraud detection systems can play an important role in the detecting fraud and abuse in health insurance sector because of their good performance in anomaly selection.

REFERENCES

[1] United States Department of Justice: “Health Care Fraud Report, Fiscal Year 1998”, Washington, DC: Department of Justice, **(1999)**.

[2] US General Accounting Office: “Health Insurance: Vulnerable Payers Lose Billions to Fraud and Abuse”, GAO-HRD-92-69, **(1992)**.

[3] The National Health Care Anti-Fraud Association: “Health Care Fraud – A Game Not Worth Playing”, 2007 Anti-Fraud Program Education Series, **(2007)**.

[4] Johnson, S.: “Nominal GDP list of countries”, International Monetary Fund, World Economic Outlook Database, **(2008)**.

[5] US Federal Bureau of Investigation: “Financial Crimes Report to the Public, Fiscal Year 2007”, Washington, DC: U.S. Department of Justice, **(2007)**.

[6] Han, J; Kamber, M.: “*Data Mining: Concepts and Techniques*”, 2nd Edition, Morgan Kaufman**, (2006).**

[7] Tan, P. N.; Steinbach, M.; Kumar, V.: “*Introduction to Data Mining*”, Addison-Wesley **(2005)**.

[8] Dunham, M. H.: “*Data Mining Introductory and Advanced Topics”*, Pearson Education, **(2003)**.

[9] Taft, M.; Krishnan, R.; Hornick, M.; Muhkin, D.; Tang, G.; Thomas, S.; Stengard, P.: “Oracle Data Mining Concepts”, Oracle, 3-2, **(2005).**

[10] Taft, M.; Krishnan, R.; Hornick, M.; Muhkin, D.; Tang, G.; Thomas, S.; Stengard, P.: “Oracle Data Mining Concepts”, Oracle, **(2005)**, 3-2.

[11] Chandola, V.: “Anomaly Detection for Symbolic Sequences and Time Series Data”, *PhD Thesis*, University of Minnesota, Minnesota, United States, **(2009)**.

[12] Edgeworth, F. Y.: “On Discordant Observations”, *Philosophical Magazine* 23, 5, **(1887)**, 364 – 375.

[13] Hawkins, D. M.: “*Identification of Outliers*”, Chapman and Hall, **(1980)**.

[14] Charran, E.: "Introduction to Data Mining with SQL Server", http://www.sql-server-performance.com/ec_data_mining.asp **(2006).**

[15] Lee, H. H.: "Data Preparation Tool for Exploration in Data Mining", *Masters Dissertations: Computer Science* , http://dspace.fsktm.um.edu.my/handle/1812/97 **(2007)**.

[16] Loureiro,A.; Torgo, L.; Soares, C.: "Outlier Detection Using Clustering Methods: a Data Cleaning Application, in Proceedings of KDNet Symposium on Knowledge-based Systems for the Public Sector", Bonn, Germany, **(2004)**.

[17] Niu, K.; Huang, C.; Zhang, S.; Chen, J.: "*ODDC: Outlier Detection Using Distance Distribution Clustering*", LNAI, 4819, **(2007),** 332-343.

[18] Zhang, J.;Wang, H.: "Detecting Outlying Subspaces for High-dimensional Data: the New Task, Algorithms, and Performance, Knowledge and Information Systems", 10(3): **(2006),** 333-355.

[19] Gueli, R.; Mongiovi, M.; Ferro, A; Giugno, R; Pulvirenti,A.; Marati, G.: "Time Series Data Mining: Techniques for Anomalies Detection in Water Supply Network Analysis", *7th Information Conference on Hydroinformatics*, HIC 2006, Nice, Fransa, **(2006).**

[20] Wei, L.; Kumar, N.; Lolla, V.; Keogh, E.; Lonardi,S.; Ratanamahatana, C.: "Assumption-Free Anomaly Detection in Time Series", Santa Barbara, CA, **(2005)**.

[21] Zhang, R.; Zhang, S.; Lan, Y.; Jiang, J.: "Network Anomaly Detection Using One Class Support Vector Machine", *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 1, **(2008)**.

[22] Deshmeh, G.; Rahmati, M.: "Distributed Anomaly Detection, Using Cooperative Learners and Association Rule Analysis", *Intelligent Data Analysis*, 12(4), **(2008)**, 339-357.

[23] Oh, J.; Gao, J.: "A kernel-based Approach for Detecting Outliers of High-dimensional Biological Data", *Proceeding of IEEE International Conference on Bioinformatics and Biomedicine*, Suppl 4, **(2008)**.

[24] Knorr, E.; Ng R.: "Algorithms for Mining Distance-based Outliers in Large Datasets", *Proc Int Conf Very Large Databases (VLDB 1998),* 392-403, **(1998)**.

[25] Angiulli, F.; Basta, S.; Pizzuti, C.: “Distance-based Detection and Prediction of Outliers”, *IEEE Trans on Knowledge and Data Engineering* .18:145-160, **(2006)**.

[26] Al- Zoubi, M.: “An Effective Clustering-Based Approach for Outlier Detection”, *European Journal of Scientific Research*, 28(2), **(2009)**, 310-316.

[27] Cutsem, B.; Gath, I.: “Detection of Outliers and Robust Estimation Using Fuzzy Clustering”, *Computational Statistics & Data Analyses*, 15-1, **(1993),** 47-61.

[28] Jiang, M.; Tseng, S.; Su, C.: “Two-phase Clustering Process for Outlier Detection”, *Pattern Recognition Letters*, 22: 691-700, **(2001)**.

[29] Hewahi, M.; Saad, M.: **“**Class Outliers Mining: Distance-Based Approach”, *International Journal of Intelligent Technology*, 2(1), **(2007)**, 55-68.

[30] Kumpulainen, P; Kylväjä M.; Hätönen, K.: *“*Importance of Scaling In Unsupervised Distance-Based Anomaly Detection”, *XIX IMEKO World Congress Fundamental and Applied Metrology*, Lisbon, **(2009)**.

[31] Cansado, A.; Soto, A.: “Unsupervised Anomaly Detection in Large Databases Using Bayesian Networks”, *Applied Artificial Intelligence*, 22(4), 309-330, **(2008)**.

[32] Yang WS: “A Process Pattern Mining Framework for the Detection of Health Care Fraud and Abuse”, *Ph.D. thesis*, National Sun Yat-Sen University, Taiwan, **(2003)**.

[33] National Health Care Anti-Fraud Association: “The Problem of Health Care Fraud: A serious and costly reality for all Americans”, *report of National Health Care Anti-Fraud Association (NHCAA)*, **(2005).**

[34] Viveros, M.; Nearhos, J.; Rothman, M.: “Applying Data Mining Techniques to a Health Insurance Information System”, *Proceeding of the 22th International Conference on Very Large Data bases*, **(1996)**, 286-294.

[35] Sokol, L.; Garcia, B.; West, M.; Rodriguez, J.; Johnson, K.: “Precursory steps to mining HCFA health care claims”, *In Proceedings of the 34th Hawaii International Conference on System Sciences*, **(2001).**

[36] Yamanishi, K.; Takeuchi, J.; Williams, G.; Milne, P.: “On-line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms”, *Data Mining and Knowledge Discovery,* 8, **(2004)**, 275–300.

[37] Williams, G.; Huang, Z.: “Mining the Knowledge Mine: The Hot Spots Methodology for Mining Large Real World Databases”, *Lecture Notes in Computer Science*, 1342, **(1997)**, 340–348.

[38] Major, J.A.; Riedinger, D.R.: “EFD: A Hybrid Knowledge/ statistical-based System for the Detection of Fraud”, *The Journal of Risk and Insurance,* 69(3), **(1992)**, 309–324.

[39] Liou, F.; Tang, Y.; Chen, J.: “Detecting Hospital Fraud and Claim Abuse through Diabetic Outpatient Services”, *Health Care Management Science*, 11(4), **(2008)**, 353-358.

[40] Yang, W.; Hwang, S.: “A process-mining Framework for the Detection of Healthcare Fraud and Abuse”, *Experts Systems with Application*, 31(1), **(2006)**, 56-68.

[41] Healy, W. L.; Ayers, M. E.; Iorio, R.; Patch, D. A.; Appleby, D.; Pfeifer, B. A.: “Impact of a Clinical Pathway and Implant Standardization on Total Hip Arthroplasty: A Clinical and Economic Study of Short-term Patient Outcome”, *The Journal of Arthroplasty*, 13(3), **(1998),** 266–276.

[42] Ireson, C. L.: “Critical Pathways: Effectiveness in Achieving Patient Outcomes”, *The Journal of Nursing Administration*, 27(6), **(1997),** 16–23.

[43] Hwang, S. Y.; Wei, C. P.; Yang, W. S.: “Process Mining: Discovery of Temporal Patterns from Process Instances”, *Computers in Industry,* 53(3), **(2004),** 345–364.

[44] Ortega, P.; Figueroa, C.; Ruz, G. A.: “Medical Claim Fraud/Abuse Detection System based on Data Mining: A Case Study in Chile”, *Proceedings of the 2006 International Conference on Data Mining*, Las Vegas, **(2006)**.

[45] Peng. Y.; Kou, G.; Sabatka, A.; Chen, Z.; Khazanchi, D.; Shi, Y.: “Application of Clustering Methods to Health Insurance Fraud Detection”, *The 3rd IEEE 2006 International Conference on Service Systems and Service Management*, **(2006)**.

[46] Scott, V.: "Extraction, Transformation, and Load Issues and Approaches", TDAN.com, 1 **(2000).**

[47] StatSoft Electronic Statistics TextBook: "*Support Vector Machines*", http://www.statsoft.com/textbook/support-vector-machines

[48] Heller, K; Svore, K.; Keromytis, A.; Stolfo, S.: "One Class Support Vector Machines for Detecting Anomalous Windows Registry Accesses", *In proc. of the workshop on Data Mining for Computer Security*, **(2003)**.

[49] Haberstroh, R.: "Oracle Data Mining Tutorial for Oracle Data Mining 10g Release 2, Oracle Data Mining 11g Release 1", http://www.oracle.com/technology/products/bi/odm/odminer.html **(20.01.2010).**

CURRICULUM VITAE

CÜNEYT AŞUK

Education	2006 –	Marmara Univ./ Computer Eng. Master, İstanbul
	2001 – 2006	Marmara Univ./ Computer Eng., İstanbul
	1998 – 2001	Yamanlar Fen Lisesi, İzmir
Experience	01.08.2006 - BT – Grup A.Ş., Kavacık, İstanbul • Software Engineer	
Knowledge	.Net, C#, Java, MS SQL, Oracle	
Personal Information	Date of Birth : 28/05/1983 Place of Birth : Nazilli	