RISK ESTIMATION FOR INTRAUTERINE GROWTH RESTRICTION USING ULTRASOUND INDICES AND CLASSIFIERS IN EMERGENCY CASES

by

Zeynep Zengin

B.S., Information Technologies, Işık University, 2006

B.S., Computer Engineering, Işık University, 2006

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2008

RISK ESTIMATION FOR INTRAUTERINE GROWTH RESTRICTION USING ULTRASOUND INDICES AND CLASSIFIERS IN EMERGENCY CASES

APPROVED BY:

Prof. Fikret Gürgen ……..
(Thesis Supervisor)

Prof. Ethem Alpaydın ……..

Asst.Prof. N. Ziya Perdahçı ……..

DATE OF APPROVAL: 06. 08. 2008

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Fikret Gürgen, for many insightful conversations during the development of the ideas in this thesis, and for helpful comments on this research, as well as for the many discussions carried on.

I also thank Prof. Ethem Alpaydın, for the courses and the materials which were referenced in this thesis. I am also grateful to him for his many helpful suggestions and comments on our experimental results.

My sincere appreciation goes out to Asst.Prof. N. Ziya Perdahçı, my teacher and my committee member.

I am also grateful to Dr. Füsun Varol, for her supervising throughout my thesis and encouragements.

I am appreciated to my friend Aslı Uyar for her moral support and technical assistance when I need a hand to push me a little.

I also thank to Prof.Ahmet Kaşlı, head of CS department at Okan University and my colleagues Özlem Çağın, Pınar Sayan, Timuçin Aktan and Emre Çakmak for their moral support and being there whenever I needed.

I am also grateful to Özgür Erbaş. Throughout my thesis, he provided encouragement, advice, and lots of good ideas. I also thank him for his incredible patience and for his continued moral support. So, I thank him for all he has done for me.

Finally, I wish to thank my small and sweet niece, Doğa, for her innocent motivation. To her I dedicate this thesis.

ABSTRACT

RISK ESTIMATION FOR INTRAUTERINE GROWTH RESTRICTION USING ULTRASOUND INDICES AND CLASSIFIERS IN EMERGENCY CASES

The main objective of this study is to provide automatic recognition of IUGR in the early stages of pregnancy by using noninvasive method. The difficulty faced in interpretation of IUGR in the early stages forced researchers to study about automatic detection of growth restriction. We aim to make fast and effective classification of ultrasound readings that are collected from emergency deliveries. Using intelligent data analysis techniques, computer programs could easily interpret maternal, placental and fetal measurements, predict presence or absence of growth restriction and provide real-time analysis and diagnosis. In this study, several machine learning techniques have been applied to IUGR dataset for classification using PI (Pulsality Index), RI (Resistancy Index) of UA (Umblical Artery), MCA (Middle Cerebral Artery) and DV (Ductus Venosus), and AFI (Amniotic Fluid Index) measurements. These measurements are taken from ultrasound readings from the mothers at emergency room. After data acquisition and scaling processes of the data, we applied 13 different classification algorithms. These 13 classifiers that have been used in this study can be divided into three groups. First group consists of single classifiers such as Support Vector Machines, k-Nearest Neighbors and Logistic Regression. In the second group, we tried to reject low confident test instances to achieve higher classification accuracy with higher confidence. Third group uses hybrid classifiers in order to benefit from several classifiers. Among these groups, performance of second group outperformed the third and lowest performance obtained from the first group. Within second group, SVM classification with rejection of low confident test samples results are shown to outperform competing classification results.

ÖZET

ACİL VAKALARDAKİ ULTRASON VERİLERİNİN SINIFLANDIRILMASI SONUCU INRTAUTERİN BÜYÜME GERİLİĞİNİN RİSK ANALİZİ

Bu tezde yapılan çalışmanın temel amacı bebeklerdeki intrauterin büyüme kısıtlılığının hamileliğin erken safhalarında otomatik olarak tespit edilmesini sağlamaktır. Büyüme kısıtlılığın erken tespit edilmesinin zor olduğu kadar önemli olması araştırmacıları bu alana yöneltmiştir. Bu tez, acil servise gelen annelerin ultrasonlarından edinilen verilerin hızlı ve efektif bir şekilde sınıflandırılmaları sonucu intrauterin büyüme geriliği riskini düşürmektir. Günümüzde akıllı veri analizi sistemleri kullanımıyla anneye özgü, plasental ve cenine özgü verilerin edinilmesi doğrultusunda büyüme kısıtlılığınının olup olmadığı tespit edilerek gerçek zamanlı analiz ve teşhis imkanı sağlanmaktır. Bu çalışmada intrauterin büyüme kısıtlılığının önceden belirlenmesi için çeşitli makina öğrenmesi sitemleri kullanılmıştır. Önce, verinin boyutu manuel bir bir şekilde indirgenmiş ve sınıflandırma performansını artırdığı ispatlanan ölçeklendirme işlemi uygulanmıştır. Bu aşamadan sonra veriye onüç farklı sınıflandırma yöntemi uygulanarak sonuçlar karşılaştırılmıştır. Bu yöntemler temel olarak üç farklı grupta toplanabilirler. İlk grup Destek Vektör Makineleri (DVM), k-En Yakın Komşu (k- Nearest Negihbor – k-NN) ve Lojistik Regresyon yöntemleri gibi tekil sınıflandırma yönetemleridir. İkinci grup yöntemler DVM ve k-NN sınıflandırma yöntemlerinde güven analizi yaparak düşük güvenli test verilerinin reddedilmesi sonucu daha yüksek sınıflandırma doğruluğu elde edilmesini amaçlamaktadır. Üçüncü grup deneyler ise ilk gruptaki üç sınflandırma yöntemlerinin değişik kombinasyonlarda birleşerek beraber veya sıralı bir şekilde sınıflandırma yapmalarına dayanmaktadır. Bu üç grup deneyler içinde sınıflandırma performansı sırasıyla en yüksek ikinci olmak üzere, üçüncü ve birinci sınıfladır. İkinci grup içinde ise DVM sınflandırması kullanan ve sınıflandırma güvenilirliği düşük vakaların reddedilmesi sonucu elde edilen sınıflandırma doğruluğu bütün testlerdeki en yüksek performası elde etmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AFD	Acute Fetal Distress
AFI	Amniotic Fluid Index
ANN	Artificial Neural Networks
DT	Decision Trees
DV	Ductus Venosus
FN	False Negative
FP	False Positive
HMM	Hidden Markov Models
IUGR	Intrauterine Growth Restriction
k-NN	k-Nearest Neighbor
LR	Logistic Regression
MCA	Middle Cerebral Artery
MCC	Matthews Correlation Coefficient
NST	Non-Stress Test
PI	Pulsality Index
PPV	Positive Predictive Value
RI	Resistancy Index
SVM	Support Vector Machines
TN	True Negative
TP	True Positive
UA	Umbilical Artery

1. INTRODUCTION

To study medicine, bioinformatics integrates computational sciences and engineering principles. In recent years, with bioinformatics systems, computer engineers have developed approaches for the prevention, diagnosis and treatment of diseases, for patient rehabilitation and for improving health.

Due to the diversity of medical data and their significance of early prediction, intelligent data analysis methods in medicine have attracted many researchers. Many machine learning researches and applications in medical applications have been listed [1, 2]. Artificial Neural Networks, Fuzzy Systems, Statistical Approaches are three of many methods that have been used, and in recent years a newer method, Support Vector Machines, has been implemented in medical applications [3]. All of these researches present several advantages over analyzing the data manually and making interpretations in medical applications. Computer-based machine learning programs could easily interpret complex patient related data, predict future indicator values based on past data, provide automated real-time analysis and diagnosis and enables rapid identification and classification of input data. Intelligent machine learning methods are expected to be powerful tools to enhance current medical diagnostic techniques [1].

This study presents the results of application of several machine learning techniques like Support Vector Machines, k-Nearest Neighbors, Logistic Regression and their combinations for the classification of Intrauterine Growth Restriction (IUGR).

Prenatal monitoring and detection of intrauterine growth restriction (IUGR) have improved dramatically in recent decades by the usage of antenatal ultrasound assessment [8-12]. Our knowledge of the pathophysiology of IUGR has been greatly extended by studies with sophisticated epidemiologic analyses and by advanced ultrasound technology to find out the relationships between fetoplacental hemodynamic characteristics, fetal behavior, amniotic fluid production, and regulation of fetal heart rate. With from mild to severe placental diseases, fetal growth delay and adaptive organ responses become evident

in uterus. Exhaustion of the placental and fetal adaptive potential leads to decompensation, with variable progression and manifestations in the fetal organ systems. Adaptive responses that are intended to enhance fetal survival in a hypoxic environment may become destructive. Important portion of the long-term effect of growth restriction is unclear and severe disturbance of fetal growth is a challenge to many researchers.

The dataset used in this study has been obtained from the emergency service of Gynecology and Obstetrics Department of Trakya University. IUGR diagnosis are made to newborns with a birth weight and/or birth length below the 10th percentile for their gestational age and whose abdominal circumference is below the 2.5th percentile with pathologic restriction of fetal growth [4]. Although many IUGR cases have unknown causes, they are usually caused by maternal, fetal, or placental factors.

In this study, IUGR risk estimation by noninvasive ultrasound readings procedure involves data acquisition, classification and interpretation of results. Finally 13 different classification methods have been applied to obtain the predicted results. These 13 classifiers are Support Vector Machines, k-Nearest Neighbor and Logistic Regression algorithms and their combinations. Classifier combination is a newer technique that aims to achieve better classification performances by benefiting from more than one classification methods. The results of classification tasks have been presented and compared.

1.1. Motivation

Fetal growth restriction is generally caused by inadequate nutritional environment in uterus. It characterizes newborns that have not attained its growth potential and these infants are disadvantaged before they enter the world. This enforces researchers to study automatic detection of growth restriction of fetuses before they are born.

Estimating the expected risk of a medical case is very important to diagnose earlier. Hence, computer aided risk estimation systems that automatically detect possible diseases became very crucial for medicine.

Automatic IUGR analysis is critical for diagnosis and treatment of potentially restricted babies. Computer-assisted IUGR recognition enables more reliable management of fetus care and growth. Various techniques have been utilized to classify IUGR cases. The reliable detection of IUGR constitutes a challenge. Consequently, many researches have focused on the accurate diagnosis of growth restriction.

The objective of this study is to classify certain IUGR cases using classification techniques. Since it is very critical to classify medical data, resulting over 80% classification is required as good performance. We used several classification models and propose methods that meet those medical requirements. Those methods include SVM, k-NN, Logistic Regression and some classifier ensemble of those individual methods. Especially we used a relatively new intelligent analysis method, SVM, in several experiments and compared its performance with others. Most of the methods are expected to meet the necessities.

1.2. Outline

Chapter 1 is the introductory part of this thesis. Base information about the study, motivation of the research and outline is given.

In Chapter 2, Background chapter, detailed information about the Intrauterine Growth Restriction is given. The characteristics of ultrasound readings from emergency cases, which include basic measurements of fetuses that help to identify IUGR, are explained.

Chapter 3 (Experimental Study) presents what we have done in this study. The used IUGR dataset, applied dimensionality reduction techniques, various classification algorithms and ensembles for risk estimation in emergency cases, methods and early usage of those algorithms have been explained in this chapter.

Chapter 4 includes the experimental results of this study. The results of individual techniques are given in this part. Comparison of results and explanation of performance diversity of classifier systems have been discussed in this chapter. Moreover, different

performance measures of a classification have been explained and those measures of each 13 experiments are given.

Chapter 5, which is also the conclusion of the overall thesis, summarizes used models and compares them with the outperforming model within those 13 tried ones. Possible future research directions are also discussed in this chapter.

2. BACKGROUND

2.1. Medical Data Analysis

Dealing with real life data and classifying it is a very important problem. If it is the case of medical data it is even more important and crucial to analyze it correctly. Therefore, intelligent data analysis methods in medicine have attracted many researchers. There are many researches and applications of intelligent systems in medical applications [1, 5].

As in many areas, medical diagnosis and bioinformatics use data mining and machine learning techniques successfully. These techniques, such as artificial neural networks (ANN), Decision Trees (DT), Hidden Markov Models (HMM), k-Nearest Neighbor (k-NN), and support vector machines (SVM) etc. have been used to extract valuable knowledge from medical databases [6]. In addition, researchers also focus on determining best machine learning techniques for different data types like small or large populated data, high risk (like medical or military data) data, data that have few features or many features etc. SMV is one of the best techniques that deal with sensitivity, false negative decision rate, and large populated datasets. Hence, it is suitable for medical data analysis.

For medical data, False Negative (FN: deciding a case as negative while it is actually positive) decisions are highly riskier than False Positive (FP: deciding a case as positive while it is actually negative) decisions. For example, deciding that one person does not have cancer though s/he has is much worse than deciding s/he has cancer while s/he doesn't have.

Thus, the sensitivity of a classification is very crucial. SVM's biggest advantage is its ability to control the sensitivity of classification. This improves the usefulness of SVM based classification in medical data [7]. Thanks to these recent achievements, SVM has become one of the highly used techniques. In this study, we applied several classifiers and their combinations to IUGR dataset and compare their performance with SVM classifier.

2.2. Intrauterine Growth Restriction (IUGR) Data

The two main fetal growth disorders are intrauterine growth restriction (IUGR) and macrosomia, both of which are associated with increased prenatal mortality rates and short and long term morbidity rates [8]. Prenatal monitoring and detection of fetal growth disorders have improved dramatically in recent decades by the usage of antenatal ultrasound assessment [9-11]. Growth is a dynamic process and only the comparison of absolute measurements with gestational age reference ranges allows the detection of deviations between expected and actual growth. It is also known that the classification of fetuses by birth weight percentile has a significant advantage for the detection of IUGR.

Our knowledge of the pathophysiology of IUGR has come from animal experiments but has been greatly extended by human studies with sophisticated epidemiologic analyses and by advanced ultrasound technology [9-12] to find out the relationships among fetoplacental hemodynamic characteristics, fetal behavior, amniotic fluid production, and regulation of fetal heart rate. With placental diseases from mild to severe, fetal growth delay and adaptive organ responses become evident in uterus. Exhaustion of the placental and fetal adaptive potential leads to decompensation, with variable progression and manifestations in the fetal organ systems. Adaptive responses that are intended to enhance fetal survival in a hypoxic environment may become destructive such as acute ischemia-reperfusion. Important portion of the long-term effects of growth restriction are unclear and severe disturbances of fetal growth is a challenge to the many researchers.

Screening for fetal growth restriction is charted during each antenatal visit and maternal uterine fundus is objectively measured. After 20 weeks' gestation, the normal symphyseal fundal height in centimeters approximates the number of weeks' gestation, after appropriate allowances for maternal height and fetal station. The reported sensitivity for the detection of IUGR ranges from 60% to 85%, and the positive predictive values (PPV) are 20% and 80%. Although measurement of the symphyseal fundal height is a poor screening tool for the detection of IUGR, the accuracy of the subsequent ultrasound prediction of IUGR is enhanced if there is suspicion of IUGR, based on lagging fundal height.

Fetuses with a small abdominal circumference (AC) percentile are at risk for IUGR [8]. A flattening growth curve on two consecutive examinations at least 14 days apart (in the third trimester, preferably 21 days apart) heightens diagnostic suspicion (Figure 2.1). Beyond 24 weeks, an elevated UA Doppler index is a strong supportive evidence for IUGR as a result of placental dysfunction. A false-positive diagnosis is likely in sonographically small fetus with normal findings on UA Doppler examination, and the risk of fetal stress in labor as a result of chronic hypoxia is low. After 34 weeks, the UA Doppler index may be within the normal range, and a decreased cerebroplacental ratio or MCA Doppler index may be the only supporting evidence of placental-based IUGR. After the completion of anatomic survey and assessment of amniotic fluid volume, the fetus is categorized as either likely or unlikely to have IUGR.

As a summary, the ultimate severe condition in IUGR is accepted as absent end diastolic flow or reverse flow in umbilical artery (UA) (high values of PI), decreased PI in fetal middle cerebral artery (MCA), appearance of retrograde flow pattern in ductus venosus (DV), severe oligohydramnios (AFI <5), non-reactivity in nonstress test (NST). These findings are supportive evidences of fetal growth disorder and may be employed as typical features to detect and monitor the fetuses with IUGR at risk.

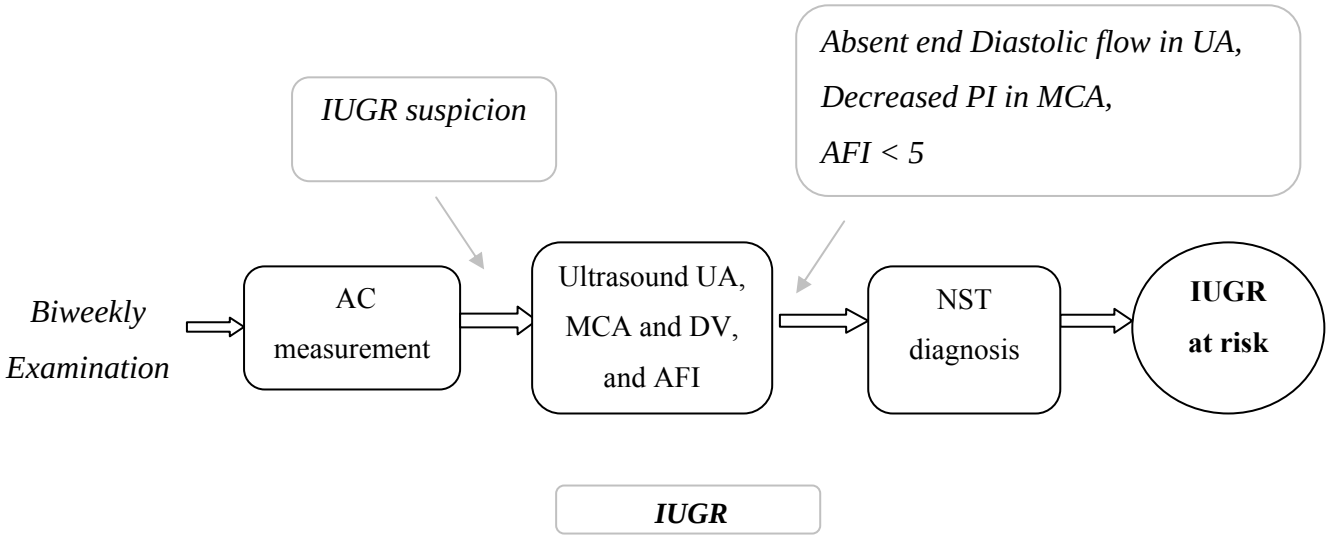

Figure 2.1. A flowchart for IUGR fetuses at risk

In a typical IUGR case, when AFI takes a value of less than 5 (80-90% of amniotic fluid), we most probably observe variations on the Doppler indices of UA. Nextly, we expect an orderly disturbance of the blood flow (redistribution of blood flow) in MCA and DV in the following days and the AFI worsens (more than 90%). The following events may be AFD, nonreactive NST with an increased risk of hypoxia (fetal compromised). But in many cases, one solution might be to conduct the delivery without waiting further development of hypoxic conditions.

2.3. Classification of IUGR Data

Support vector machines (SVM) classifier has been proven to be one of the most promising among the recently developed statistical decision techniques with attractive properties [13-15]. The SVM has already been successfully applied to optical character recognition, text classification, bankruptcy prediction and some other medical diagnostics areas. The SVM offers some advantages that make it preferable in the management of the IUGR fetuses for risk estimation: it gives a single solution characterized by the global minimum of the optimized functional and not multiple solutions associated with local minima as in the case of neural networks (NN). Moreover, it does not rely so heavily on heuristics, i.e. an arbitrary choice of the model, and has a more flexible structure. Hence, this flexible, globally optimized technique with higher generalization ability becomes suitable in the risk analysis of IUGR near various applications with high dimensional and complex data. Developed from the theory of structural risk minimization, SVM aims to perform classification with better generalization by upper bounding the expected risk (ER) on test error rather than minimizing the empirical risk on training error [13]. The minimum ER is obtained by the optimal separating hyper-planes in feature space that maximizes the distance margin between the closest samples of each class.

In addition to SVM, we applied k-NN, Logistic Regression and ensemble strategies that consist of these three classifiers and compared the performances. One of the classifier ensembles are consensual medical decision system based on SVM, k-NN and LR classifiers to predict IUGR cases in emergency from a small population using the Doppler indices of

placental fetal vessels at a minimum expected risk (Figure 2.2.). The noninvasive measurements of pulsatility index (PI), resistance index (RI) of UA, MCA and DV vessels and amniotic fluid index (AFI) are inputs of the system. The consensual system distinguishes reactive and nonreactive and/or AFD cases as an indication of the suspicion of placental dysfunction at the first clinical stage. In the second clinical stage, NST value is taken as a validation of the system's findings.

The proposed system has some advantageous properties for the small population IUGR application: first, it employs three classifiers: a globally optimized SVM, a local, minimum distance based k-NN and a maximum likelihood based LR with probabilistic output. Second, confidence measures are defined to combine the classifiers to describe the effectiveness of each classifier in the small population input space. Finally, a consensual decision is made for suspicious samples.

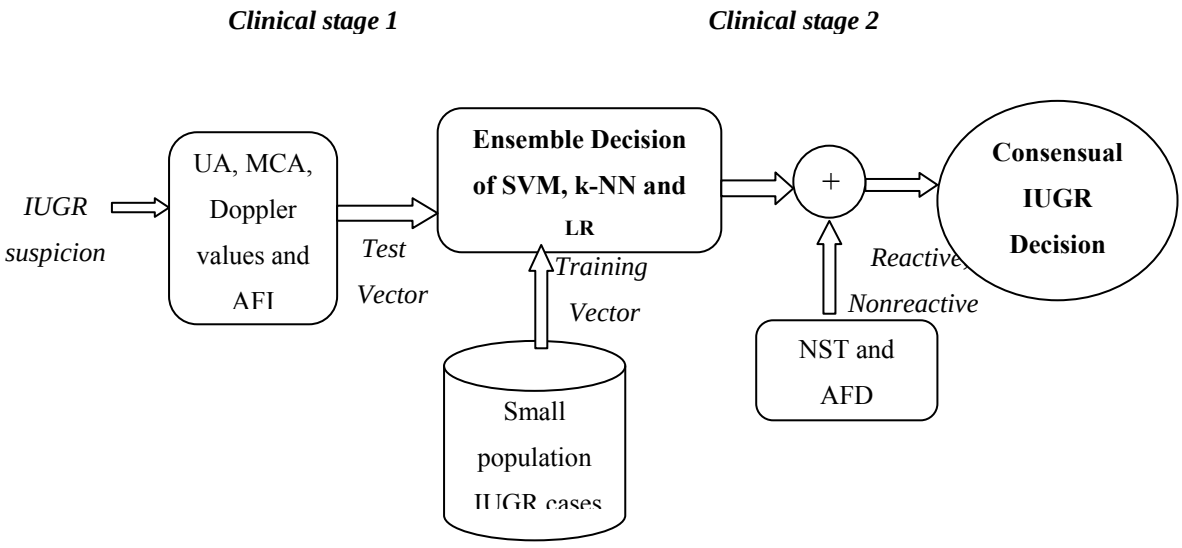

Figure 2.2. Consensual IUGR Decision

Generally, there is a necessity for immediate risk analysis or estimation in medical emergency cases. In a fetal growth disorder, it is suitable to employ a noninvasive method such as the measurement of PI, RI and the observation of non-metric AFI (values being <5 or >5) at the first clinical stage and to make a dependable decision based on the findings in a short time interval. Since the learning to know the actual risk may become very costly in emergency cases, an instant minimum expected risk estimation is also vital. In this aspect, a fast, noninvasively obtained feature set and a reliable, accurate, consensual decision system becomes a good choice for the IUGR risk application with a small population.

In the second stage, we combine the other testing modalities such as NST test, etc. to monitor fetuses with IUGR to support our results and thus consider the wide clinical spectrum and the variance in the relationship between testing and the outcome.

2.3.1. Data Statistics

The IUGR database that is used in this study has 44 high risk preterm pregnancies before 34 weeks. They were accepted to the Gynecology and Obstetrics Department of Trakya University for emergency care [12]. The cases were divided into two groups: group 1 includes eighteen pregnancies with IUGR (%40.1) and group 2 has twenty-six pregnancies that do not have IUGR (%59.9). Table 2.1 illustrates some statistics about the data. The IUGR dataset that is used in this study has 21 features. Some of the features are about mother, such as age and the pregnancy period; some are about fetus, for instance measurements from ultrasound images that describe his/her growth rate, and some data are about newly born baby, such as weight, and sex. The measurements about the baby after delivery have not been used in this study.

Table 2.1. Features of 44 high risk pregnancy cases

	GRUP 1 (n=18)	GRUP 2 (n=26)	overall (n=44)
Age	25.67 ± 4.3	27.35 ± 5.9	26.7 ±5.3
OHA (oligohidramnios)	8 (44%)	10 (38%)	18 (41%)
EMR (Early membrane rupture)	3 (17%)	14 (54%)	17(39%)
PE (preeklampsi)	10 (56%)	7 (27%)	17(39%)
PTL (preterm labor)	3 (17%)	18 (69%)	21 (48%)
UA – PI	1.41 ±0.08	1.08 ±0.08	1.60 ±0.35
UA – RI	0.77 ±0.35	0.70 ±0.28	0.80 ±0.08
MCA – PI	1.49 ±0.38	1.67 ±0.5	1.22 ±0.32
MCA – RI	0.78 ±0.34	0.82 ±0.35	0.73 ±0.12
DV – PI	0.77 ±0.13	0.63 ±0.12	0.69 ±0.24
DV – RI	0.58 ±0.28	0.50 ±0.17	0.53 ±0.14
AFI	4.56 ±0.76	6.62 ±0.51	5.77 ±3.8
reactive NST	8 (44%)	18 (69%)	26 (59%)
nonreactive NST	10 (56%)	8 (31%)	18 (41%)

The most important and early indicator of IUGR is non-stress test (NST). Reactive NST is a good sign of not seeing IUGR. For making early detection of IUGR, this study aims to predict reactivity possibilities of NST of a fetus. As seen from Table 1.1, 59% of overall cases do not have reactive NST while others have non-reactive NST and/or AFD. So 41% of the cases in this dataset are risky. Furthermore, as indicated in the table, the first group has been more nonreactive cases than reactive cases. In this study, the classification of IUGR has done with NST value, since it is a good and early indicator of growth restriction.

3. EXPERIMENTAL STUDY

In this study, SVM system is proposed for the risk estimation of IUGR. The process of risk estimation has two clinical stages: the first clinical stage is feature extraction for risk estimation. Noninvasive UA, MCA and DV Doppler indices PI and RI, and amniotic fluid index (AFI) are measured in actual emergency cases and are labeled as “reactive” and “nonreactive and/or acute fetal distress (AFD)” fetuses. “Reactive” class corresponds to fetuses in normal conditions and “nonreactive and/or AFD” class corresponds to fetuses at hypoxia and mortality risk. Then, these features are employed in the SVM system to obtain a two-class decision. In the second clinical stage, the fast risk decision is enhanced by a NST tool. As a result, the overall system consists of ultrasound readings, SVM based decision and NST tests at cascaded levels for screening the risk of IUGR.

3.1. System Overview

This study compares Support Vector Machines (SVM) with other single and compound classifiers. The used IUGR classification system with any of the classifiers has three main stages. First stage is “Data Acquisition” that consists of handling missing values, Scaling and Dimensionality Reduction. In this study, instead of automatic dimensionality reduction techniques, we used ultrasound readings that are collected from emergency cases, not the entire dataset.

The second part of the system is “Classification”. For classification, there are many techniques in the literature. K-Nearest Neighbor, Logistic Regression and some other joint classifiers that combine these two with SVM have been applied to same dataset in order to compare these techniques with SVM. Besides joint classifiers also rejecting low confident predictions after single classification applied to data have also been tried in this study. Finally, results of classifications have been evaluated and compared both in terms of machine learning and medical biostatistics.

3.2. Data Acquisition

The dataset that we used has 44 IUGR recordings (group one includes 18 pregnancies with IUGR and group two has 26 pregnancies that do not have IUGR) that have 20 features both. This database stores various data about mother, fetus and baby after born. These data can be processed to detect various kinds of abnormalities about fetus.

Because of the long time requirement of the data collection period and fewness of emergency room deliveries of pregnant cases there are only 44 instances in this dataset. 44 instance dataset is insufficient for data analysis systems to have good classification performance. On the other hand, this dataset includes only 5 missing values which are all pH values and there are no unclassified instances.

By using all 19 features and classifying the baby being healthy or exitus is not very beneficial and doesn't give high accuracy. Rather, with the help of medical experts, we decided to classify non-stress test (NST) values of the fetus. NST being reactive, nonreactive or Acute Fetal Distress (AFD) is very important implication of the action being taken by the doctors. So in this work, we classified NST being reactive or not.

There are 26 reactive, 13 non-reactive and 5 AFD instances. Because of the unfair class distribution, multi-class classification would make unreliable predictions, so the dataset is divided into two groups labeled as Reactive and Nonreactive. Hence, there are 26 cases in the reactive group and 18 cases in the positive group. In all of the classification and data preprocessing tasks, this two-class dataset have been used.

3.3. Support Vector Machines (SVM)

SVM is considered as a very effective machine learning tool in recent years. SVM can be used as a fast classification algorithm for huge datasets that is used to extract valuable information from them.

SVMs map data points to a high dimensional feature space where a separating hyperplane can be found. SVM transforms input space into a high dimensional feature space and identifies a separating hyperplane between classes. This separating hyperplane is computed by maximizing the distance of the closest patterns, i.e. margin maximization.

Risk minimization is one purpose of SVM. It handles risk by minimizing true error. Especially in medical cases minimizing False Negatives (saying someone healthy while she is not) is very crucial. SVMs generate black box models which lack the explanation capability on how to reach a decision [6].

Pattern Recognition, Regression, Multimedia, Bio-informatics, Artificial Intelligence are some areas that SVMs have been applied to make classification in many real world problems [3]. Besides SVM, there are also many other techniques, such as decision trees, neural networks, genetic algorithms, etc. that are also been used in these areas. However, SVM is different from them in terms of its solid mathematical foundation which is based on the statistical learning theory.

SVM minimizes the structural risk rather than minimizing the empirical risk (training error). This expresses an upper bound on the generalization error, i.e., the probability of an erroneous classification on unseen examples.

In most classification tasks, SVM outperforms other methods, at least their performances match [20]. Also, SVM classifier can deal with high dimensional data. Since medical datasets are large especially in number of attributes, SVM method is very valuable for those datasets. Several studies about applications of SVMs to medical decision support have been listed in recent years.

In this study, it is shown that the results of standard SVM classifier outperform other methods like Logistic Regression, and K-Nearest Neighbor. Those three classifiers are baseline classifiers of this study. Secondly, rejection strategies were used. Rejection of predicting low confident test instances have been tried on SVM and k-NN and again SVM outperformed k-NN. Moreover, we applied several classifier combining techniques with different classifier combinations. Those ensembles were used to form either confidence for

all samples or confidence for low-confidence samples. Within those, SVM happened to improve performance of classifier ensembles. Following parts will discuss types of SVMs in detail since it is the base classifier that we used

3.3.1. Types of SVMs

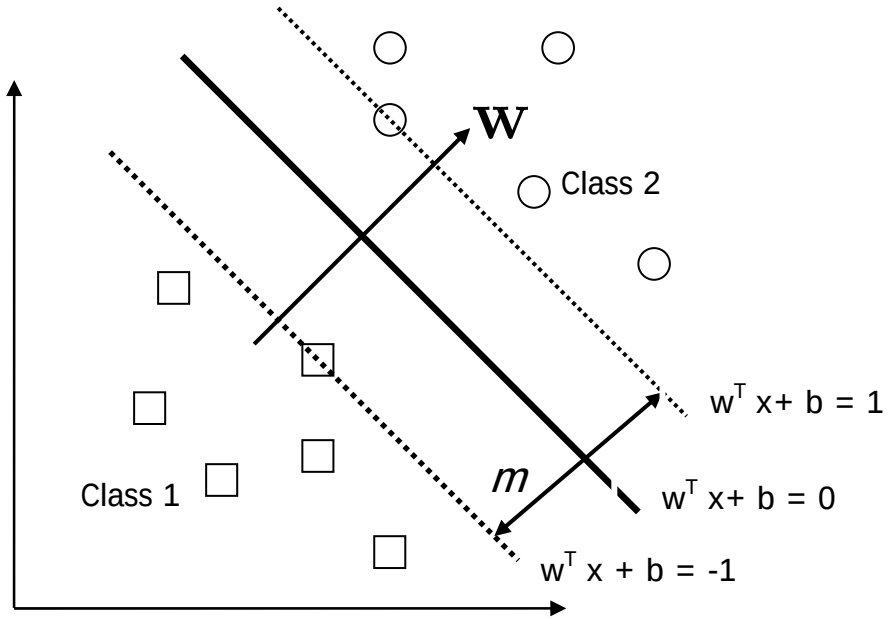

Figure 3.1. Decision boundary for two classes

First type of SVMs is Linear. Let (x_1, ..., x_n) be our data set and y_i $\in$ (1,-1) be the class label of x_i. Assuming all data are at least distance 1 from the decision boundary, the following two constraints follow for a training set (x_i ,y_i):

$$w^T x_i + b \geq 1 \quad \text{if } y_i = 1 \tag{3.1}$$

$$w^T x_i + b \leq -1 \text{ if } y_i = -1 \tag{3.2}$$

From equations (3.1) and (3.2) it can be written:

$$y_i (w^T x_i + b) \geq 1 \tag{3.3}$$

As from the Equation (3.3) we can see that a better generalization can be done with more distance between instances and the hyperplane [18]. Instances of each class should be as far away from the decision boundary as possible. Thus, we should maximize the margin (**m**), which is the distance between the hyperplane and the instances closest to it.

In order to maximize the margin we should minimize ||w||. So we can rewrite the problem as

$$\text{Minimize } \frac{1}{2} \| w \|^2 \tag{3.4}$$

$$\text{Subject to } \quad y_i \, (\mathbf{w^T x_i} + b) \geq 1 \quad \forall i \tag{3.5}$$

This is a standard quadratic optimization problem, and the solution involves constructing a dual problem where a Lagrange multiplier α_i is associated with every constraint in the primary problem.

$$\max \ \mathrm{W}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{3.6}$$

$$\text{subject to } \alpha_i \geq 0, \ \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.7}$$

So, w can be written as

$$\mathrm{w} = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{3.8}$$

The size of the dual depends on sample size, N, and not on the input dimension, d, [18].

The solution suggests that many of the α_i s are zero. x_i s with non-zero α_i are called support vectors (SV). Meaning; examples that are closest to the hyperplane are support vectors. And the decision boundary is determined only by the support vectors.

For testing phase, rather than using margin, we can calculate $g(x) = w^T x + b$ and choose according to the sign of $g(x)$ [18]. We choose C_1 (class 1) if $g(x) > 0$ and C_2 (class 2) otherwise.

Another SVM type is Soft Margin SVMs. Generally two classes are not linearly separable. In those cases, we look for the separating hyperplane that causes least error. We allow error ξ_i "slack variable" in classification which is the deviation from the margin.

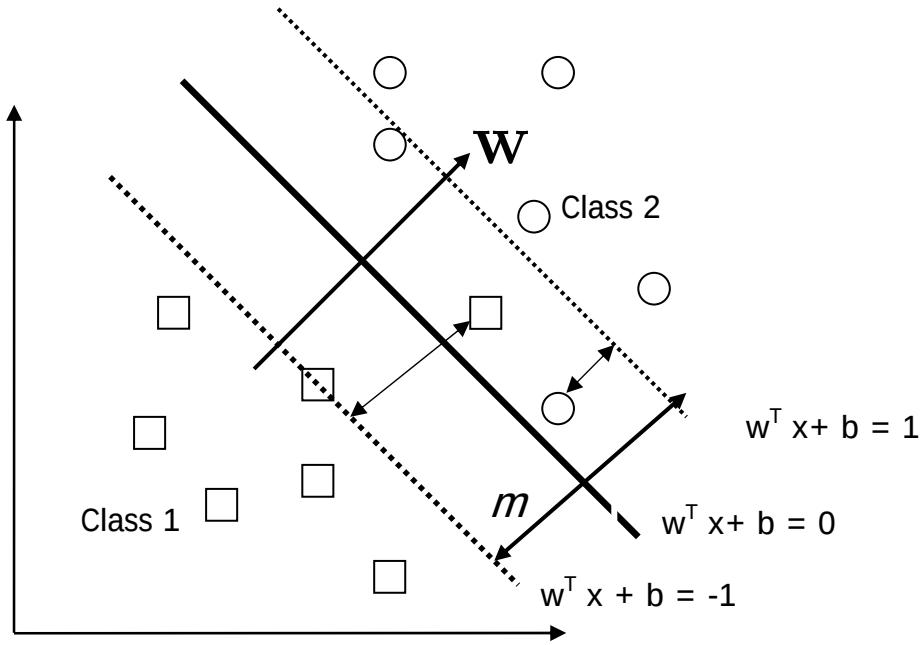

Figure 3.2. Soft-Margin decision boundary for two classes

Deviations can be divided into two: An instance may be in the wrong side of the hyperplane and be misclassified, or, it may be on the right side but may lie in the margin that is not adequately away from hyperplane. Then equation (3.5) becomes:

$$y_i \, (\mathbf{w^T x_i} + b) \geq 1 - \xi_i \qquad (3.9)$$

For $\xi_i = 0$, there is no problem. For $0 < \xi_i < 1$, x_i is correctly classified but it is inside of the margin. If $\xi_i > 1$, x_i is misclassified. Last case is the situation that we are trying to handle. In soft margin scenario we want to minimize:

$$\frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i \tag{3.10}$$

C is the cost parameter that determines the tradeoff parameter between error and margin and can be defined as the penalty factor for classification. That is the tradeoff between the complexity (number of support vectors) and the misclassified data [18]. So, again the quadratic optimization problem is solved.

Third SVM type is Non-Linear SVMs. Those are formed based on a kernel function. When the two classes are not linearly separable, which is a real life situation, there is a second option rather than fitting a nonlinear function. That option is, mapping the data to a higher dimensional space.

The main idea with the non-linear SVMs is mapping the original input space to some higher dimensional feature space where the training set is separable. In such a case we are interested in a method whose complexity does not depend on the input dimensionality but depends on the number of training instances [18].

We use soft margin hyperplane because the problem may not be linearly separable in the new feature space. It is critical here to choose a proper C, the penalty factor to eliminate overfitting and underfitting. If C is too large, we have high penalty for non-separable points and we may store many support vectors and overfit, if it is too small we may have underfit [18].

The key idea of non-linear SVMs is kernel functions that are used for mapping data to a higher dimensional space. The most used kernel functions are:

Linear: $$K(x_i, x_j) = x_i^T x_j$$

Polynomial of degree p: $K(x_i, x_j) = (1 + x_i^T x_j)^p$

Radial Basis Function: $K(x_i, x_j) = \exp\left[-\frac{\|x_i - x_j\|^2}{\sigma^2} \right]$

Sigmoidal Function: $K(x_i, x_j) = \tanh(2x_i^T x_j + 1)$

3.3.2. Multi-Class SVMs

Generally SVM is used for two class classification problems. We can make Multiclass classification by reformulating the problem as N (number of classes) binary classifications. More commonly, the dataset is divided into two parts "intelligently" in different ways and a separate SVM is trained for each way of division. Multiclass classification is done by combining the output of all SVM classifiers.

3.3.3. Combining SVMs with Various Techniques

SVM is a very powerful classification technique and it has been studied extensively in the area of machine learning. On the other hand, in the case of imbalanced datasets, instances of one class outnumbers the instances of the other class, the success of SVM is very limited [21].

Application areas such as gene profiling, medical diagnosis and credit card fraud detection not only have highly skewed datasets with a very small number of positive instances which are hard to classify correctly, but also very crucial to classify correctly [21]. Classifiers generally perform worse on imbalanced data than on balanced datasets because they tend to generalize from sample data and output the simplest hypothesis that fits best to data [21].

It is important to design the classifier sensitive to noise and more prone to learn erroneously. Making the classifier too specific may work opposite to this purpose. There are techniques that modify the behavior of existing algorithms like IM3 algorithm for k-NN, pruning of Decision Trees, or soft margin SVMs. These techniques make classifiers more immune to noisy instances. On the other hand, some positive instances can be treated as noises and ignored completely while trying to eliminate noise by the classifier [21].

There are two popular approaches to solve this problem. First one is in order to pay more attention to the positive instances biasing the classifier. This can be done, for instance, by increasing the penalty associated with misclassifying the positive class relative to the negative class. Second approach is to preprocess the data by over sampling the majority class or under sampling the minority class, in order to create a balanced dataset [21].

SVMs often suffer from a large number of features. As mentioned earlier, there are methods to reduce feature count like feature selection and extraction. As a supporting fact there are many studies that combined feature selection strategies with SVM have reported increased test accuracy [21]. These methods select important features first and then SVM is applied for classification. By selecting most important features, SVM can build more reliable model by reducing redundant data.

3.4. Logistic Regression

Logistic regression is a probabilistic machine learning model. The main idea of Logistic Regression is fitting the data to a logistic curve (Figure 3.4) and predicting probability of belonging class of test samples. Logistic regression is generally used for binary classification [23-24].

The difference between Logistic Regression and Linear Regression is Logistic Regression fits the data to a curve while Linear Regression fits the data to a linear line. A basic logistic curve is shown in Figure 3.3.

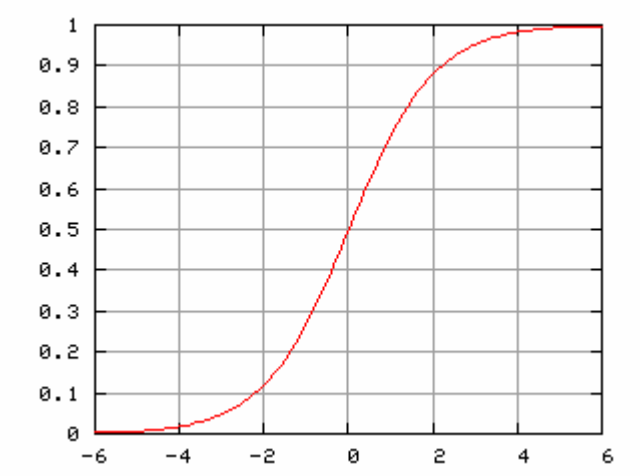

Figure 3.3. Logistic Curve

If 1 and 0 are the two outcomes of dependent variable, logistic model can be written like equation (3.11).

$$Logit(Y=1) = \ln(\frac{P(Y=1)}{P(Y=0)}) = \alpha + \beta X \tag{3.11}$$

In this equation, Y is the dependent variable, 1 is the desired outcome and X is the independent variable vector. Moreover, α and β are parameters that are going to be identified by maximum likelihood method using the training data.

When logistic regression applied to the data; first, the unknown parameters α and β are estimated from the training data. The probability of test case can be calculated using Equation (3.12) and then it can be compared with a predefined threshold value in order to make an estimation of test cases belonging class.

$$P(Y=1) = \frac{1}{1+e^{-(\alpha+\beta X)}} \tag{3.12}$$

Since there are two classes, the probability of the class 0 is as Equation (3.13).

$$p(y=0|x) = 1 - p(y=1|x) \tag{3.13}$$

SVM and LR methods are closely related; both of them construct a separating hyperplane in the feature space depending on the weighted linear combination of training input vectors.

In addition to applying standalone techniques to a dataset, applying a combination of classifiers techniques are being used recently. For instance, [25-26] proposes a hybrid SVM-LR method that they claim this technique improves the test accuracy.

3.5. k-Nearest Neighbor Classifier

K-Nearest Neighbor (k-NN) classifier tries to predict a test instance's class by looking at its neighbors. It is a distance based classification algorithm. Based on the minimum distance from the test instances to the first k nearest training samples, it tries to estimate the correct class [27]. The class of the test instance is predicted according to majority of these nearest neighbors. Simply, we sort the distances of all training samples to the test instance and determine the k-th minimum distance.

For the special case of the algorithm, if k equals to 1 it is called Nearest Neighbor Algorithm (NN). Meaning, decision of a test instance's belonging class is made by looking its nearest training instance. Class decision of the test instance is same as the nearest training neighbor of it. k-NN means the class of a test instance is decided to be the same as the class appearing most frequently among k-neighborhood of the test data. It is reasonable to say that close instances is more likely to belong to the same class and make assumptions accordingly. For large datasets, choosing k more than one would give better results by majority voting of nearest k neighbors instead of single nearest neighbor. The number k should be large enough to minimize the probability of misclassification, and small enough to give accurate prediction with the samples optimally form a group.

To determine the nearest neighbor of one instance we need a distance measure. Generally used distance measures are Euclidean Distance, Manhattan Distance, Hamming Distance, etc. In this study we used Euclidean Distance for k-NN classification.

The generalization of above equation to N dimensional space is (p_i and q_i are the coordinates in dimension i):

$$d(p,q) = \sqrt{\sum_{i=1}^{N} (p_i - q_i)^2} \tag{3.14}$$

The steps for k-NN classification task can be summarized as follows:

- Determine parameter k: number of neighbors.
- Calculate the distance between test instance and all the training samples
- Sort the distances and determine nearest neighbors based on the k-th minimum distance
- Use simple majority of the class of nearest neighbors as the prediction value of the test instance

The only challenges of the algorithm is determining best k value and choosing the distance type to calculate distances of samples. Computation cost is quite high, because we need to compute distance of each test sample to all training samples. Other than those, k-NN algorithm is simple to implement, robust to noisy training data and effective if the training data are large.

In terms of hybrid classifiers, combining k-NN and SVM is also a technique that has been applied earlier [28]. Generally compound classifiers that consist of more than one single classification algorithm outperform the single best classifier that has been applied to a case [29]. The main questions are how to combine classifiers and which classifier combination can achieve better performance. In this study, some classifier combinations have been tried like k-NN-SVM, SVM-than-k-NN, k-NN-SVM-LR. Results will be pointed out in next section.

While k-NN and SVM algorithms both work on distance measures, this study shows that SVM outperforms k-NN on classification tasks [29]. Classification performances of several classifiers will be compared and discussed in experimental results part of this study.

3.6. Confidence and Rejection

In machine learning, besides test accuracy, confidence is also a very important performance measure [29, 21, 32]. While giving the accuracy percentage of a classification process, indicating how confident you are with this classification is more meaningful.

Sometimes, rather than making less confident classifications, rejecting classification of risky instances might be better. Especially for medical data, it is very important to reduce false classifications. Instead of accepting false decisions, it might be better not to diagnose. On the other hand, some true decisions can be eliminated within those rejected cases. There is a huge trade-off between confidence and rejection. In our study we have applied several classification techniques, some gives more importance to confidence and reject unconfident data, and some try to make predictions of entire cases in the dataset without considering confidence.

Measuring the confidence is somehow a difficult process [30]. Some classifiers give estimation of confidence together with their classification accuracy. For instance, Logistic Regression test accuracy is also a good indicator of confidence [30]. This is because it calculates posterior probabilities and decides according to those probabilities. For the SVM classifier, whose outputs are distances of test cases to the separating hyperplane, also confidence can be measured by those distances. So, classification accuracy of SVM is somehow about confidence and rejection can be made depending on probability distance measures to improve confidence [30]. For k-NN classifier, because the prediction of a test instance is done by class information of k nearest training instances, there is no correlation between classification accuracy and confidence.

For measuring the confidence of k-NN classifier, earlier proposed method by Roy and Madhvanath [31] was taken as base model. After finding the optimum k value for the

dataset, we first found the most commonly occurred class within each test instances' k nearest neighbor which is the standard procedure for k-NN classification. Test sample is labeled as the most common class label within those nearest neighbors. For example, in Figure 3.4, most common class within five nearest neighbor of Test Sample (T_1) is the class that is shown with circle (assuming k=5). After finding the most commonly occurring class, we found the classification confidence of that sample by dividing the count of most common class to the k value. In the case of Figure 3.4, predicted class of T_1 would be circle class and confidence of this prediction is 60%. If there were more circles within k neighbor, confidence of classifying that specific test instance would be higher. After applying these procedures for every test instance, overall classification confidence is calculated as that average of individual confidence measures of tests samples. This confidence value is an important performance measure for k-NN besides its classification accuracy.

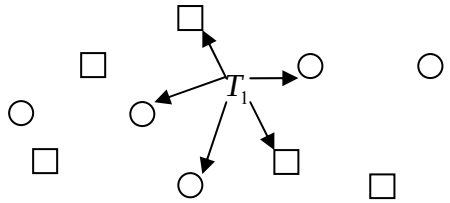

Figure 3.4. Confidence of k-NN

3.7. Classifier Ensembles

Combining different classifiers to a single decision making process has some different names like; classifier ensembles, hybrid methods, sensor fusion and decision combinations [32-34]. As mentioned earlier, using hybrid classifiers can improve test accuracy radically. There are several ways to combine different classifiers for a single classification. They can be combined during a single decision making process or one classifier can get the rejected instances of other classifier and contribute in predicting class of those unconfident cases. However, for combining classifiers they must be independent, negative dependent or complementary. It would not make sense if we combine identical classifiers [35].

In this research several classifier combination techniques were used. First of all, low confident decisions that SVM classifier made have been rejected. Classification of those rejected instances has been made by k-NN classifier. No rejections occur in this model. Second combination model distinguishes from the first one after applying k-NN to rejected instances of SVM. In this method, after applying k-NN to those rejected dataset, the decision of SVM and k-NN has been compared on those rejected instances. If they predict same class for the same case it has counted as confident, if not those stayed unconfident. Figure 3.5 simply expresses the classification

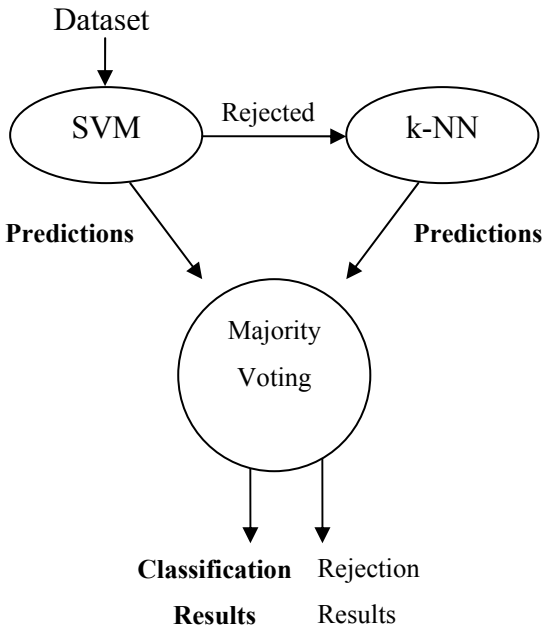

Figure 3.5. Used SVM - k-NN then consensus classifier ensemble model

Second classifier ensemble has been done with *majority voting* method. All predictions that have been made by three classifiers, SVM, k-NN and LR, were compared and more confident decision has been made according to those predictions. This method is well working and recently attracted method because its simplicity and good performance [35]. Also majority voting method has been applied with two combinations of SVM with other classifiers. Thus, 2 more classifications have been tried with majority voting method combining, SVM and k-NN, SVM and LR. There are several ways to apply majority voting method depending on error rate of classifiers, dependence of classifiers, etc. Those

variations form types of majority voting. Figure 3.6 shows a model of majority voting ensemble technique with two classifiers.

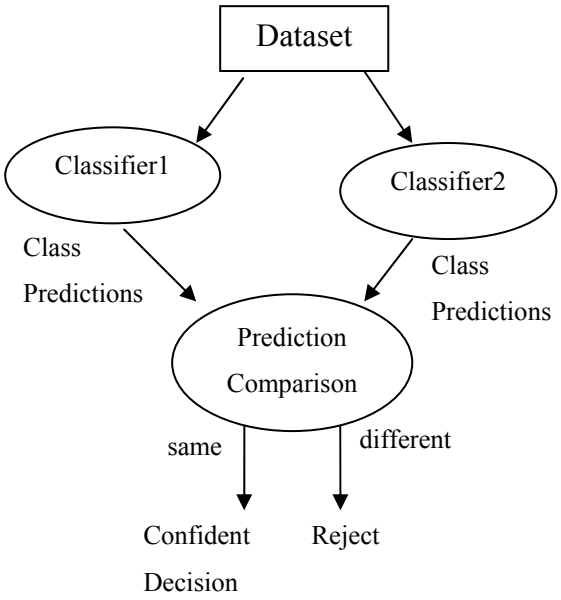

Figure 3.6. Majority voting model that combines two classifiers.

To combine three classifiers with majority voting, prediction is made by the majority vote of those classifiers. Thus, there would be no rejection. Besides, majority voting method can be made more confident by forcing the decision making process to predicting only when entire classifiers predict in the same class, and rejecting otherwise. For example, if each of three classifiers predicts in the same way that an instance is positive or negative, decision of majority voting can be made. Otherwise, majority voting rejects the case labeling it as unconfident. Both methods have been tried and compared in this work.

Another classifier ensemble has done with the variation of majority voting method which is *weighted majority voting*. For this method every classifiers have weights that contribute to decision making. In majority voting those weights were counted as equal. In Figure 3.7., only *class predictions* goes into the *Prediction Comparison* fusion, if it was the case of weighted majority voting, weights also would go in. For weights of weighted majority voting, confidence values of classifiers were used in this work. SVM, k-NN and LR were combined for weighted majority voting method.

.

4. EXPERIMENTAL RESULTS

For this study, several classification algorithms are used to classify IUGR data, and then their performances are compared. In this section, the results of conducted experiments will be given. First, SVM classifier results will be given and followed by and k-NN, LR and classifier ensembles.

4.1. Support Vector Machines

The LIBSVM software has been used for SVM classification tasks [36]. Proposed procedure at "A Practical Guide to Support Vector Classification" [37] has six steps as below:

- Transform data to the format of an SVM software.
- Conduct simple scaling on the data
- Consider the RBF kernel
- Use cross validation to find the best parameter C and γ
- Use the best parameter C and γ to train the whole training set
- Test

This proposed procedure was followed for SVM classification on IUGR dataset. Higher performance was acquired by including manual parameter selection. Due to the size of IUGR dataset 10 fold cross validation is applied that reduces negative effects of small datasets. Moreover, it is shown that scaling the data improves performance of classification.

MATLAB has a function that is used for scaling the data. This function is *prestd* function. The *prestd* function normalizes the dataset so that they have means of zero and standard deviations of one.

Scaling is very important; because it eliminates the possibility of great numeric values dominate other values that are respectively very small. Another advantage is to avoid numerical difficulties during the calculation. This is because kernel values usually depend on the inner products of feature vectors and large attribute values may cause numerical problems [37].

As mentioned earlier, 10 fold cross validation is used at each and every experiment. 10 fold cross validation means that at each fold, classifier uses different instances for training end test sets. This eliminates the chance of overfitting and underfitting. While dividing data into training and test sets, data can be coincidently selected that result as overfitting. Also, underfitting can occur in some other cases. By training and testing the model with different sets of data more than once we obtain a more robust classifier. After training and testing the model 10 times, I calculate the mean of those ten accuracy result to obtain the final average accuracy.

Another very important thing that has to be considered during dividing data into train and test sets is *stratification.* Stratification means that the correlation of positive and negative instances at the dataset must be maintained at training and test sets. To give an example, if positive class is 40% of entire dataset, after dividing it into training and test sets their positive class instance count also have to be 40% or close. Thus, each of 10 folds of IUGR dataset was stratified. At each experiment that has been done with different classifiers or approaches, same 10 stratified folds have been used. So, the tests are done with the same datasets that can be compared fairly.

4.1.1. Results After PCA and Manual Dimensionality Reduction

Most common dimensionality reduction technique, Principle Component Analysis (PCA) is tried, before the manual parameter selection. PCA is a preprocess stage of that is commonly used in machine learning. It is used for choosing a reduced set of original features. Dimensionality reduction techniques are suggested to increase efficiency and improve performance such predictive accuracy.

Because of IUGR dataset has 21 features, but 4 of them are measurements of baby after born which is not valuable information for classifying NST (non-stress test) of fetus. So there are only 16 features plus NST. 17 feature datasets are not counted as a high dimensional dataset and generally small datasets does not require parameter selection. Although it is not recommended, we applied PCA to compare the results.

There is a MATLAB function that applies PCA to the dataset automatically. That is called *prepca.* Prepca applies principle component analysis method that transforms the input data so that the elements of the input vectors will be uncorrelated. Thus, can be trained only with those uncorrelated input vectors results that in dimensionality reduction.

Table 4.1 shows some results that have obtained after applying automatic PCA tool of MATLAB to IUGR dataset. To compare results, PCA have applied with different minimum fraction values that results different number of Principle Components. Last row of the table is performance of SVM within entire dataset. 10 fold cross validation also have been used in these experiments to make comparisons meaningful. Scaling also applied to all datasets.

Table 4.1. Results of principal component analysis

# of Principal Components	SVM Test Accuracy (%)
1	69
2	54
3	63
4	60
5	55
6	66
7	58
8	67
9	70
10	64
11	60
12	65
13	61
14	62
15	60
16	60

As seen from Table 4.1, initial performance of IUGR dataset with whole features is 60% and it can be increased up to 70% with PCA. 70% test accuracy is obtained with 0.96 correlation and 9 principle components. However, 70% accuracy is not a satisfactory result in machine learning, especially for medical data.

To obtain better classification results, we thought to reduce dimension manually. As medical experts and [12] suggested, we used most important features that are better indicators of NST. Those are Pulsality Index (PI) and RI measurements of UA, MCA and DV values and Amniotic Fluid Index (AFI) value. Earliest measurement that can be obtained during pregnancy is UA measures. Then, MCA and DV measurements can be obtained respectively. We used several different combinations of these values and trained SVM. Table 4.2 shows some results of these tests.

Table 4.2. Test performances of SVM classifier with different features

Experiment		1	2	3	**4**	5	6	7	8	9	10	11	12
UA	PI	y	y	y	**y**	y	y	y	y	y	y	y	y
	RI	y	y	y	**y**	y	y	n	n	n	n	n	n
MCA	PI	y	y	y	**y**	n	n	y	y	y	y	n	n
	RI	y	y	y	**y**	n	n	n	n	n	n	n	n
DV	PI	y	y	n	**n**	n	n	y	y	n	n	n	n
	RI	y	y	n	**n**	n	n	n	n	n	n	n	n
AFI		y	n	y	**n**	y	n	y	n	y	n	y	n
Accuracy (%)		64	73	65	**81**	59	69	61	77	60	75	66	74

13 experiments conducted with different combination of seven features that are best indicators of NST. “y” in a cell means that, the corresponding experiment of that cell contains corresponding feature and “n” means that that feature is not included in that experiment. For instance, in experiment 4, PI and RI measurements of UA and MCA are included in classifier and in experiment 8 only PI measurements of UA, MCA and DV have been used.

As seen from the Table 4.2, experiment 4 gives the best classification performance with 81% which outperforms dimensionality reduction with PCA. In all of these 13 experiments, 10 fold cross validation was used. In each fold dataset was divided into training and test sets randomly. 34 of the entire dataset have been used for training and 10 for testing. The NST feature was used for class information. *Reactive* instances are treated as negative class and non-reactive and AFD (Acute Fetal Distress) instances are treated as positive class which indicates a possibility of Growth Restriction.

There are 18 non-reactive instances in the entire dataset, which is approximately 41% of the data. In order to keep data stratified, four of the test data have been selected from non-reactive and six of them have been selected from reactive cases in each fold. Then, average performances of these 10 folds have been selected as final performance. Table 4.3 shows the performances of each fold.

Table 4.3. Test performances of SVM classifier for each of 10 fold

Fold #	Test Accuracy (%)	# of Correctly Classified Samples
1	70	7/10
2	70	7/10
3	90	9/10
4	80	8/10
5	70	7/10
6	80	8/10
7	80	8/45
8	90	9/45
9	90	9/45
10	90	9/47
Average	**81**	**81/100**

Classification accuracy is calculated by dividing number of correctly classified samples to the number of overall samples. This approach is the most common approach to calculate classifier performance. As mentioned earlier kernel parameters C and γ are used

as 13000 and 0.03 respectively and both training and tests sets have been scaled so that they have means of zero and standard deviations of one.

While 81% of the data were classified correctly, 19% of them were misclassified. Some of those classifications are false positives and some of then are false negatives. Table 4.4 shows numbers of false and true classified instances for each fold. As illustrated in the table, within 19 misclassified samples False Positive count is very low than False Negative Count. As discussed earlier, FN occurrences are more severe than FP occurrences for medical data. But usually number of FNs is more than number of FPs. This is because negative instance counts are generally outnumbers positive instance counts. Thus, classifiers learn negative instances better, and classify a test instance as negative at critical points more frequently. Although is seems that SVM is not very effective because of having many FNs, it will be shown that other classifiers have more FNs.

Table 4.4. Performance parameters of SVM classifier for each of 10 fold

Fold #	TP	TN	FP	FN	Total
1	1	6	0	3	**10**
2	2	5	1	2	**10**
3	3	6	0	1	**10**
4	3	5	1	1	**10**
5	2	5	1	2	**10**
6	2	6	0	2	**10**
7	3	5	1	1	**10**
8	3	6	0	1	**10**
9	3	6	0	1	**10**
10	3	6	0	1	**10**
total	**25**	**56**	**4**	**15**	**100**
average	**2.5**	**5.6**	**0.4**	**1.5**	**10**

To illustrate the SVM with a two dimensional chart a data that gives 80% accuracy with SVM and has two features classifier has been arranges. This data has not been used in any experiments and it is processed once in order to visualize the training, test instances and support vectors that SVM models. Figure 4.1 shows UA-PI vs MCA-PI values that are two features that are also used in all classifications. In this figure, stars show positive instances and circles show negative instances. Stars and circles that are filled with dark color are test instances and others are training instances. Training instances that are covered

by a square are support vectors that SVM model fits. Those support vector forms the hyperplane.

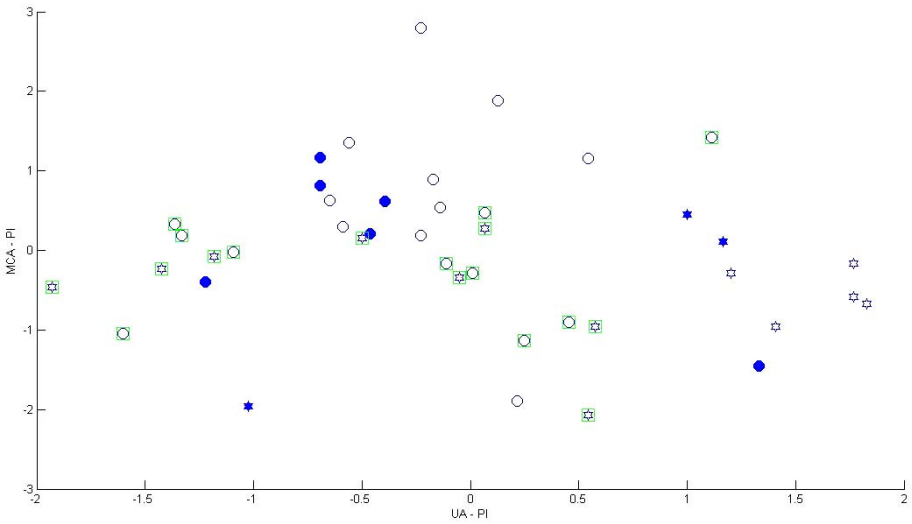

Figure 4.1. UA-PI vs MCA-PI values

4.1.2. Parameter Selection

For SVM classifier there are four common kernels called linear, polynomial, RBF and sigmoid which are mentioned in Section 3.5.3. As the guide [37] suggests that RBF kernel is a reasonable first choice, RBF kernel is used for the beginning. It writes that "The RBF kernel non-linearly maps samples into a higher dimensional space so it can handle the case when the relation between class labels and attributes are non-linear." Every kernel has hyperparameters which influence the complexity of the model. The RBF kernel has only one hyperparameter called gamma (γ)

Another important kernel parameter is cost (C) for all kernel functions. Cost is proportional to complexity of model. So, when cost increases, performance increases up to a point with the complexity. As I mentioned earlier, very high values of C can cause to overfit because of the model being to complex. So, γ and C values would be different for every dataset which requires some kind of parameter search. Guide [37] recommends grid search approach. In this approach, some values of a proposed interval ($C = 2^{-5}, 2^{-3}, ...2^{15}, \gamma = 2^{-15}, 2^{-13}, ...2^{3}$) for both γ and C are tried one by one. Pairs that

give the best accuracy are picked. After finding the values, another finer grid search around those values with smaller distances is conducted. This second grid search is the last step of parameter selection concluding in the best values of γ and C for this specific dataset.

After grid search tests, best values of γ and C are found as 0.03 and 13000 respectively. Below are the performances of SVM classifiers (Figure 4.2 and 4.3) on IUGR dataset on changing C when γ is constant and changing γ when C is constant respectively.

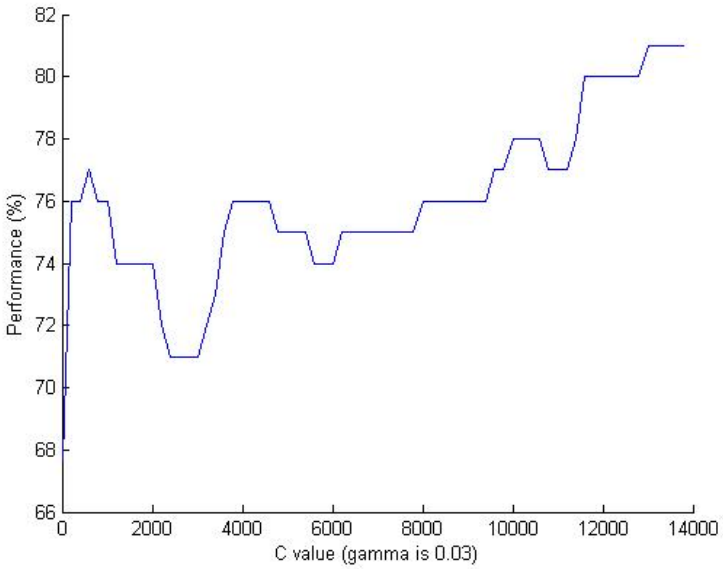

Figure 4.2. SVM performance depending on cost (C) with constant γ

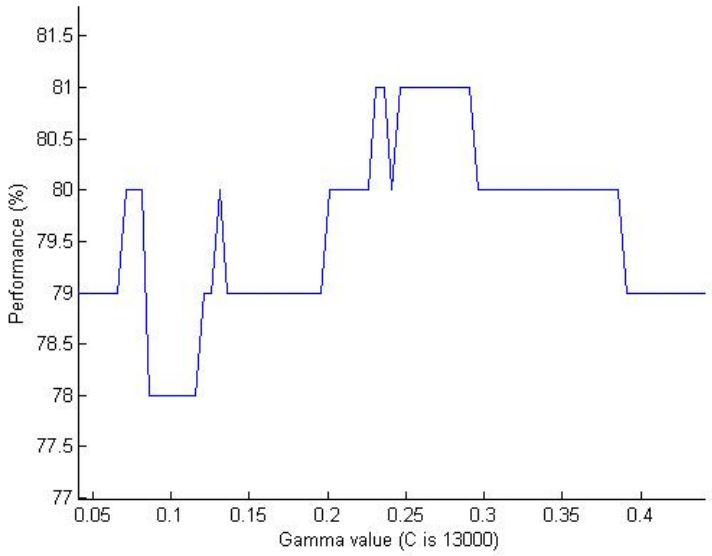

Figure 4.3. SVM test accuracy depending on γ with constant cost (C)

These parameters have been used in the rest of the experiments that have been done with SVM classifier.

After specifying best values of parameters, I applied different types of kernel functions to find out the best kernel type for IUGR dataset. The results are given in the below table.

Table 4.5. Performance of SVM classification with different kernel types applied

Kernel Type	**Test Accuracy**
Linear Kernel	0.72
Polynomial of degree 2	0.61
Polynomial of degree 3	0.73
RBF	0.81
Sigmoid	0.64

As seen from the table 4.5, RBF kernel gives best results for IUGR dataset, so RBF kernel have been used for following experiments.

Above tests are done with scaled datasets. As mentioned earlier, scaling data is very beneficial for most of the classification algorithms. Below are some default values that LIBSVM uses and Table 4.6 shows the results before and after scaling and/or parameter selection that have been done manually.

Default SVM classifier parameters:

Kernel Type: RBF

Cost: 1

Gamma: 0

Table 4.6. SVM classifier performance with/without scaling and/or parameter selection

	With default parameters	After parameter selection
Original dataset	59%	71%
Scaled dataset	77%	81%

Selected parameters: $C = 13000$ and $\gamma = 0.03$, RBF kernel.

4.2. K-Nearest Neighbor Classification

The only parameter that is used for classification with *k*-NN algorithm is *k*. K value is the number of neighbors which are active in decision. Usually k is an odd number because decision of test sample's class is based on the class type of majority of the k nearest training samples. Choosing k as odd avoids classifier to have same number of classes in the training sample within k nearest neighbor of a test instance. Figure 4.4 illustrates performance changes depending on different k values. K value is changed from 1 to 31 by increasing by 2.

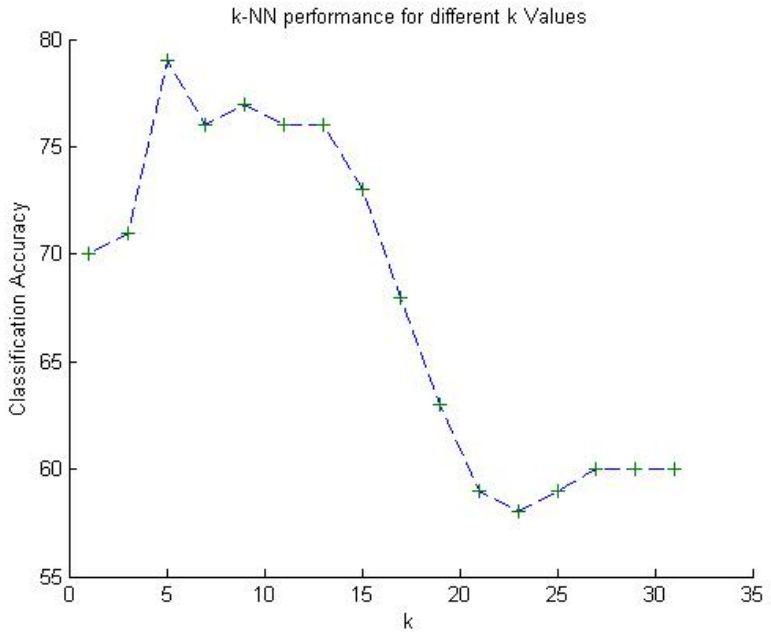

Figure 4.4. k-NN test accuracy depending on k

As seen from the figure, k-NN classifier performs its best when k equals to 5 obtaining 79% test accuracy.

10-fold-cross-validation method applied to the IUGR dataset that's dimensionality has been reduced manually like previous experiments. Also, each fold uses same samples as the SVM classifier used is previous experiments. Table 4.7 illustrates classification accuracy for each fold. Scaling also applied for k-NN classifier like SVM. However, it didn't improve performance of the k-NN classifier. So, IUGR data has not been scaled in these experiments.

Table 4.7. 10-Fold-Cross-Validation results for k-NN for best k value (k=5)

Fold #	TP	TN	FP	FN	Accuracy (%)	Total
1	1	6	0	3	70	**10**
2	2	6	0	2	80	**10**
3	2	6	0	2	80	**10**
4	1	5	1	3	60	**10**
5	2	5	1	2	70	**10**
6	3	5	1	1	80	**10**
7	4	6	0	0	100	**10**
8	3	6	0	1	90	**10**
9	2	6	0	2	80	**10**
10	2	6	0	2	80	**10**
total	**22**	**57**	**3**	**18**		**100**
average	**2.2**	**5.7**	**0.3**	**1.8**	**79**	**10**

The results show that, in k-NN classification methodology, smaller k values work better for IUGR dataset. Although, both k-NN and SVM are distance based classification techniques, SVM works better with this dataset. SVM deals with FNs better than k-NN so that it outperforms k-NN. Below table shows the false and true positive and negative counts of each fold that has been used for k-NN classification. To compare with SVM, although k-NN has one less FP decision it has three more FN decisions which are more critical that FPs.

4.3. Logistic Regression Classification

For logistic regression classifier, same dataset is divided into same 10 folds for cross validation and scaled in the same way as SVM classification. Table 4.8 shows the TP, TF, FP, FN counts and test accuracy in each fold.

Table 4.8. 10-Fold-Cross-Validation results for Logistic Regression classifier

Fold #	TP	TN	FP	FN	Accuracy (%)	Total
1	1	4	2	3	50	**10**
2	2	5	1	2	70	**10**
3	2	5	1	2	70	**10**
4	3	5	1	1	80	**10**
5	1	5	1	3	60	**10**
6	2	5	1	2	70	**10**
7	3	5	1	1	80	**10**
8	3	5	1	1	80	**10**
9	3	5	1	1	80	**10**
10	3	5	1	1	80	**10**
Total	**23**	**49**	**11**	**17**		**100**
Average	**2.3**	**4.9**	**1.1**	**1.7**	**72**	**10**

As seen form the table, test accuracy of Logistic Regression classifier has been varied from 60% to 80% whereas k-NN and SVM has achieved 100% and 90% test accuracy respectively in some folds. Although LR classifier concludes in less FNs than k-NN classifier, it has many more FPs which highly reduces test accuracy. Average test accuracy of 10-fold-cross-validation of Logistic Regression classification is 72%, which is lower than both k-NN and SVM classifiers.

LR achieves lower accuracy than SVM because its mathematical foundation is weaker than SVM. So, LR models the data in a simpler way which does not give better performances in a data like IUGR because of its smallness and complexity. The reason that k-NN performs better than LR is similar. Because k-NN is independent of size of training instances, size of dataset is not very important aspect of k-NN which is important for LR to form a model.

Figure 4.5 demonstrates Logistic Regression Classification results. LR calculates the probability outputs of each test input and decision is made on these probability outputs. The smooth line indicates the probability outputs of each test sample and the pointed line shows the decision boundary. Outputs that are below 0.5 (decision boundary) are predicted as positive classes (non-reactive NST) and others are predicted as negative (reactive NST). The circles on the smooth line show the misclassified samples. Circles below the decision

boundary are the samples that are predicted as positive but are negative in real life (FPs) and circles above the decision boundary shows FNs.

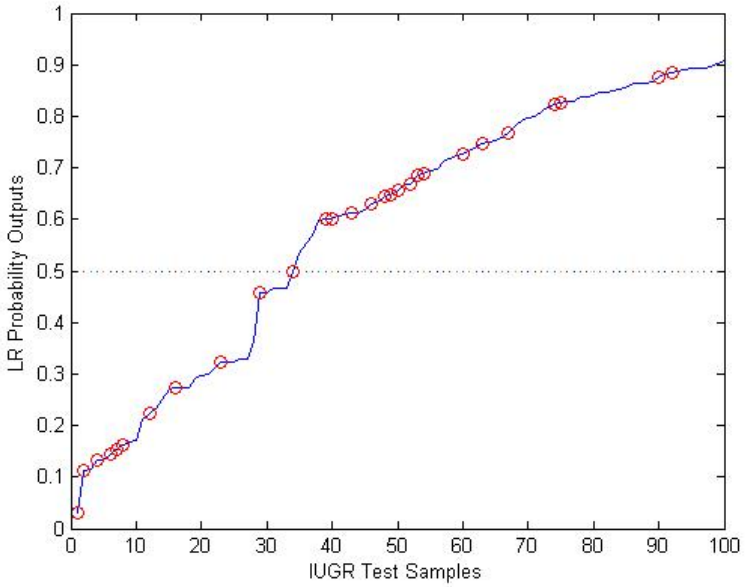

Figure 4.5. LR probability output, decision boundary and misclassified samples

4.4. Confidence Measurements and Rejection

As mentioned earlier, even though the numeric scores from the outputs of SVM and LR are predictive of confidence, this is not the case for k-NN because of not being probabilistic classifier in strict sense. The numeric scores coming from the output of k-NN are not well correlated with classification confidence. There are also more indirect ways of conveying confidence to the user than by a numeric score such as presenting explanation cases or highlighting the features that contribute negatively or positively to the classification. Here we focus on the numeric scores to express the confidence of the classification.

Here, we study a case-based small population IUGR application that would significantly benefit from a feature to attach confidence predictions to positive classifications. We propose ensemble solutions of three classifiers that aggregates a collection of confidence metrics and observe that this offers effective solution in the small population IUGR problem.

4.4.1. Confidence Improvement of k-NN Classifier

Confidence of k-NN has been calculated as explained in Section 3.8. After specifying optimum k value, dominating class within those k values are found for each test instance. After that, confidence of each test instance has been calculated as the percentage of dominating class within k nearest neighbors.

As stated in previous section, best k value for IUGR dataset has been found as five. The confidence values in each ten fold has been calculated as the average of ten test data's confidence. Table 4.9 illustrates the confidence values for each ten fold. It gives confidence values for each test data of each fold. For example, 4/5 in the cell that T1 and Fold 1 intersects means that, four of five nearest neighbor of test instance 1 (T1) are in the same class with the predicted class of T1 at Fold 1. Thus, T1 at Fold one is classified 80% confidently. As the summary of the k-NN 10 fold classification gives average of 83.2% confidence with IUGR data.

Table 4.9. Confidence findings for each fold with k-NN classifier

Fold	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**Average Confidence**
1	4/5	4/5	5/5	4/5	4/5	3/5	4/5	5/5	5/5	3/5	82%
2	3/5	5/5	4/5	3/5	4/5	4/5	5/5	5/5	3/5	4/5	80%
3	5/5	4/5	3/5	4/5	4/5	5/5	5/5	3/5	4/5	5/5	84%
4	3/5	4/5	3/5	4/5	5/5	5/5	3/5	4/5	5/5	3/5	78%
5	4/5	3/5	4/5	5/5	5/5	3/5	4/5	5/5	3/5	5/5	82%
6	3/5	3/5	5/5	5/5	3/5	4/5	5/5	3/5	5/5	5/5	82%
7	3/5	4/5	5/5	5/5	4/5	5/5	3/5	5/5	4/5	5/5	86%
8	4/5	5/5	5/5	3/5	5/5	4/5	5/5	5/5	5/5	3/5	88%
9	5/5	5/5	3/5	3/5	4/5	5/5	5/5	5/5	3/5	5/5	86%
10	5/5	3/5	3/5	5/5	5/5	5/5	5/5	3/5	5/5	3/5	84%
Average Classification Confidence of 10 Folds											**83.2%**

To improve the confidence of the k-NN classifier we suggest refusing to classify test instances that have 60% (3/5) confident decisions. 60% confidence means that only 3 of 5

nearest neighbors are in the same class with the prediction, which is the smallest confidence value that a case can have with these parameters. After rejecting the test instances with 60% confidence, Table 4.9 becomes Table 4.10. Blank entries indicate that that test instance have been rejected because of low confident prediction in that fold. *Average Confidence* values in the table indicate confidence in that fold after rejections made. Rejected instances are varying from two to four for each fold, giving an average of 29% overall rejection. As indicated in Table 4.10, average confidence of k-NN classification improves from 83.2% to 92.9% after confidence improvement process.

Table 4.10. Confidence findings for each fold with k-NN classifier after rejection

Fold	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**Reject Count**	**Average Confidence**
1	4/5	4/5	5/5	4/5	4/5	-	4/5	5/5	5/5	-	2/10	87.5%
2	-	5/5	4/5	-	4/5	4/5	5/5	5/5	-	4/5	3/10	88.6%
3	5/5	4/5	-	4/5	4/5	5/5	5/5	-	4/5	5/5	2/10	90%
4	-	4/5	-	4/5	5/5	5/5	-	4/5	5/5	-	4/10	90%
5	4/5	-	4/5	5/5	5/5	-	4/5	5/5	-	5/5	3/10	91.4%
6	-	-	5/5	5/5	-	4/5	5/5	-	5/5	5/5	4/10	96.7%
7	-	4/5	5/5	5/5	4/5	5/5	-	5/5	4/5	5/5	2/10	92.5%
8	4/5	5/5	5/5	-	5/5	4/5	5/5	5/5	5/5	-	2/10	95%
9	5/5	5/5	-	-	4/5	5/5	5/5	5/5	-	5/5	3/10	97.1%
10	5/5	-	-	5/5	5/5	5/5	5/5	-	5/5	-	4/10	100%
Overall Average for 10 Folds											**29%**	**92.9%**

After rejecting test instances that are classified unconfidently, other confident classification performance of IUGR dataset is improved by 8.4%. New test accuracy is 87.38% (Table 4.11) with 29% rejection. *Total Confident Classifications* of each row indicates confidently classified test instances of each fold. In addition, it steps out from the table that there is no more FPs after confidence improvement process and FNs decisions are decreased than regular k-NN classification

Table 4.11. Results for k-NN classifier with k=5, after confidence improvement

Fold #1	TP	TN	FP	FN	Accuracy (%)	Total Confident Classifications
1	1	4	0	3	62.50	**8**
2	1	5	0	1	85.71	**7**
3	2	5	0	1	87.50	**8**
4	0	4	0	2	66.67	**6**
5	1	4	0	2	71.43	**7**
6	2	4	0	0	100.00	**6**
7	3	5	0	0	100.00	**8**
8	3	5	0	0	100.00	**8**
9	2	5	0	0	100.00	**7**
10	2	4	0	0	100.00	**6**
total	**17**	**45**	**0**	**9**		**71**
average	**1.7**	**4.5**	**0**	**0.9**	**87.38**	**7.1**

Deciding between 79% accuracy versus 87% accuracy with 29% rejection depends on risk of classification. Since IUGR data is medical, false decisions risk lives. Moreover, we are predicting reactivity of NST which is an early indicator of IUGR, hence rejected instances can be classified more confidently with later measurements. Thus, obtaining better classification accuracy is more important and rejections can be accepted to achieve it.

4.4.2. Confidence Improvement of SVM Classifier

Since SVM finds a separating hyperplane between two classes and the output of the SVM classifier is the distance of the test instances to that hyperplane, confidence of the decision depends on this distance value. As mentioned earlier SVM classification accuracy is an indicator of its confidence. So, for 10 fold IUGR classification the confidence can be set as 81%, same as its accuracy. To improve these confidence and accuracy values instances that are very close to hyperplane can be counted as unconfident and rejected [28].

In this study we rejected to classify test instances gives output below 1 and over -1. Table 4.12 gives distance measures of each test instance in each fold.

Table 4.12. SVM outputs

Fold #	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Miss-class. Count
1	**-4.43**	**-2.80**	16.56	**-13.99**	-0.44	-2.00	-4.64	-8.90	-7.09	-3.38	3
2	**-1.84**	10.80	**-10.56**	1.18	-2.04	-3.55	-5.05	-5.29	**2.49**	-2.03	3
3	11.01	**-9.15**	1.34	1.07	-3.76	-2.91	-3.28	-12.38	-1.61	-5.39	1
4	**-6.90**	0.29	2.15	0.29	-2.89	-4.40	-12.54	-1.45	-5.60	**0.65**	2
5	**-2.39**	0.03	**-2.39**	12.74	-3.71	-25.19	**0.24**	-6.47	-1.38	-7.21	3
6	**-0.36**	**-1.26**	9.62	8.48	-30.05	-0.92	-8.23	-2.24	-5.82	-5.89	2
7	**-1.94**	0.21	3.26	3.26	**0.76**	-6.60	-0.66	-6.02	-6.76	-4.17	2
8	8.62	7.33	7.33	**-3.13**	-3.33	-1.56	-4.19	-5.58	-4.92	-17.53	1
9	10.10	10.10	**-4.33**	4.91	-2.60	-2.66	-4.36	-5.59	-8.23	-4.18	1
10	8.71	**-1.76**	5.95	1.82	-2.85	-3.18	-3.85	-5.69	-5.03	-3.91	1
Actual Class Labels	POS	POS	POS	POS	NEG	NEG	NEG	NEG	NEG	NEG	

Highlighted values in the Table 4.12 are in the interval of [-1, 1] and will be rejected afterwards. Bold values indicate misclassifications. Negative distance values indicate that, predicted class of that test instance in corresponding fold is negative and positives indicate positive predictions. Last row gives the actual class labels of test instances. As seen from the table first four test instances in each fold are actually positive and the rest are negative. Bold values in a row indicate misclassifications for that fold.

As seen from the table 4.12, mostly rejected instances are not actually misclassified. Only 4 of the 11 rejected instances are misclassified and rejection only caught one of the FNs. Table 4.13 gives performance parameters of classification of SVM classifier after rejection of unconfident data (shown with grey highlighted values). As illustrated in the table average of 1.1 instances are rejected for their low confidence in each fold. Moreover classification accuracy is 83% which was 81% before rejection. The decision to choose either rejecting or not rejecting is more difficult here.

Table 4.13. Test performances of SVM classifier for each of 10 fold after rejections

Fold #1	TP	TN	FP	FN	Accuracy (%)	Total Confident Classifications
1	1	5	0	3	66.67	**9**
2	2	5	1	2	70.00	**10**
3	3	6	0	1	90.00	**10**
4	1	5	0	1	85.71	**7**
5	1	5	0	2	75.00	**8**
6	2	5	0	1	87.50	**8**
7	2	4	0	1	85.71	**7**
8	3	6	0	1	90.00	**10**
9	3	6	0	1	90.00	**10**
10	3	6	0	1	90.00	**10**
total	**21**	**53**	**1**	**14**		**89**
average	**2.1**	**5.3**	**0.1**	**1.4**	**83.06**	**8.9**

Above experiment rejected test instances that gave outputs below 1 and over -1. If we increase this interval, it is expected to have increased, classification accuracy and rejection count. So, we tried an increased interval whose borders are -2.5 and 2.5. As expected, accuracy was 87.14%, and rejection count was 31%. So, this last experiment outperforms all other experiments in terms of classification accuracy.

4.5. Classifier Ensembles Results

To improve the classification performance of IUGR dataset that have obtained from single classifiers, we tried several types of classifier ensembles. Below sections gives results and evaluations of those ensembles.

4.5.1. Applying K-NN to Rejections from SVM

As 11% of the test instances have been rejected because of having high possibility of unconfident classification, we classified the rejected ones from SVM by k-NN. We trained k-NN with rest of the data besides rejected ones and test with rejected instances for each fold. Below table shows the new class values of rejected instances.

Table 4.14. Prediction details after applying k-NN to SVM rejections

Fold #	Rejection Count	Actual Classes	k-NN Prediction	Decision Type
1	1	N	P	FP
2	0	-	-	-
3	0	-	-	-
4	3	P, P, N	N, N, N	FN, FN, TN
5	2	P, N	P, N	TP, TN
6	2	P, N	P, N	TP, TN
7	3	P, N, N	P, N, N	TP, TN, TN
8	0	-	-	-
9	0	-	-	-
10	0	-	-	-

If we combine these results with earlier results that obtained from SVM with rejection, overall performance of this classifier ensemble would be 82% (Table 4.15). Although there are more true classifications, there are two FNs which are very important misclassifications. Briefly, this method obtained a little higher classification accuracy than regular SVM but with more FNs, and it has obtained a little lower classification accuracy than SVM with rejections but there are no rejections. Again, this tradeoff depends on dataset type and FN count is very important for IUGR classification performance making the SVM with rejection method best of these three methods.

Table 4.15. Test performances of classifier that applies k-NN to rejections of SVM

Fold #1	**TP**	**TN**	**FP**	**FN**	**Accuracy (%)**	**Total**
1	1	5	1	3	60	**10**
2	2	5	1	2	70	**10**
3	3	6	0	1	90	**10**
4	1	6	0	3	70	**10**
5	2	6	0	2	80	**10**
6	3	6	0	1	90	**10**
7	3	6	0	1	90	**10**
8	3	6	0	1	90	**10**
9	3	6	0	1	90	**10**
10	3	6	0	1	90	**10**
Total	**24**	**58**	**2**	**16**		**100**
average	**2.4**	**5.8**	**0.2**	**1.6**	**82.00**	**10**

Second method that have been used to classify rejected instances from SVM is, after applying k-NN classification method to them, comparing the predictions of both classifiers for those rejected ones. K-NN was trained with the rest of the dataset besides the rejected test instances and predicted the class of rejected ones. After this process, the predictions from both SVM and k-NN were compared. If they predict in the same way, that prediction became confident and included in the classification, if not they remain unconfident and rejected.

Actual classes of those test instances, predictions that made by SVM and k-NN and final decisions for each fold is shown in the Table 4.16. In the table, second column gives the rejected instance count from SVM, and following three columns give actual, predicted from SVM and k-NN class information. If both classifiers predict in the same way, final decision is made in the same way also, otherwise rejection remains. For instance, for the only rejected case in fold one SVM predicted as Negative but k-NN predicted as positive, thus, that instance is considered as still unconfident and rejected. This method is called *Majority Voting.* Last column gives the rejection types in each fold. As seen from the table, 4 of 11 rejections were predicted again confidently and all off them were true predictions (2 TPs and 2 TNs). 7 of them stayed rejected.

Table 4.16. Predictions after applying k-NN then majority voting to SVM rejections

Fold #	Rejection Count	Actual Classes	SVM Prediction	k-NN Prediction	Final Decision	Decision Type
1	1	N	N	P	R	None
2	0	-	-	-	-	None
3	0	-	-	-	-	None
4	3	P, P, N	P, P, P	N, N, N	R, R, R	None
5	2	P, N	P, P	P, N	P,R	TP
6	2	P, N	N, N	P, N	R, N	TN
7	3	P, N, N	P, P, N	P, N, N	P, R, N	TP,TN
8	0	-	-	-	-	
9	0	-	-	-	-	
10	0	-	-	-	-	

After combining “SVM with rejections” results and “applying k-NN then majority voting to SVM rejections” results (Tables 4.13 and 4.16), classification performance is increased to 83.97% (Table 4.17). Regular SVM classification performance was 81% and classification performance of SVM after rejecting unconfident decisions was 83.06%. Even

though 83.06 and 83.97 seems very close, with reclassifying rejected decisions rejected case count reduces to 7% from 11% which is very important.

Table 4.17. Results of applying k-NN than majority voting to after SVM classifier

Fold #1	**TP**	**TN**	**FP**	**FN**	**Accuracy (%)**	**Total**
1	1	5	0	3	66.67	**9**
2	2	5	1	2	70.00	**10**
3	3	6	0	1	90.00	**10**
4	1	5	0	1	85.71	**7**
5	2	5	0	2	77.78	**9**
6	2	6	0	1	88.89	**9**
7	3	5	0	1	88.89	**9**
8	3	6	0	1	90.00	**10**
9	3	6	0	1	90.00	**10**
10	3	6	0	1	90.00	**10**
total	**23**	**55**	**1**	**14**		**93**
average	**2.3**	**5.5**	**0.1**	**1.4**	**83.79**	**9.3**

4.5.2. Majority Voting

Another way of combining classifiers is majority voting model. This approach has also been applied in previous section by combining k-NN decisions with the SVM decisions on the samples that have been rejected from SVM because of their low confidence.

This approach is very simple. Decision is made according to the majority of classifiers that predict in the same way. First we tried to combine k-NN and SVM then LR with SVM. To make decision with combining two classifiers, it was expected that both of them predict in the same way. Table 4.18 and 4.19 shows the k-NN – SVM and LR – SVM classifier ensemble results with Majority Voting method respectively.

Table 4.18. Majority voting model that combines k-NN and SVM

SVM – K-NN		
	With confidence	without confidence
correctly classified	73%	14%
incorrectly classified	**13%**	

As seen from Table 4.18, 73% of the data were classified correctly with confidence and 14% were rejected because of being unconfident. The critical data is 13% for this classification. They are critical because while we are saying that we have classified some samples confidently, they are actually misclassified. It is important to make this percentage closer to zero. The classification performance for k-NN – SVM classifier ensemble with majority voting method is 84.9%.

Table 4.19. Majority voting model that combines LR and SVM

SVM - LR		
	With confidence	without confidence
correctly classified	68%	17%
incorrectly classified	**15%**	

In the case of combining LR and SVM with Majority Voting method, *unconfident data* and *incorrectly classified data with confidence* percentages were negatively correlated to classification performance. The increase of those two values also resulted in decrease in correctly classified samples with confidence. The classification performance for LR – SVM classifier ensemble with majority voting method is 81.9%.

Finally majority voting method was applied in order to combine all three single classifiers. In this case, decisions are made depending on majority of the classes. For example, if at least two classifiers predict positive final decision is also positive. Thus, there are no unconfident decision is this method which results with no rejections. Table 4.20 gives information about this classifier ensemble method. This method obtains more correctly classified instances than other majority voting methods but incorrectly classified samples with confidence are also more than others. Thus, with 79%, this method achieves lower classification accuracy that previous two methods.

Table 4.20. Majority voting model that combines k-NN, LR and SVM

SVM - LR		
	With confidence	without confidence
correctly classified	79%	-
incorrectly classified	**19%**	

In order to achieve better performance with this combination, Majority Voting method made stricter and decisions are made only if all three classifiers predict in the same way, otherwise reject. Table 4.21 gives information about this final experiment. With this method we achieved better classification performance with 84% and 12% incorrectly classified instances with confidence. However, rejection count is higher than all other Majority Voting experiments. The decision between

Table 4.21. Stricter Majority Voting model that combines k-NN, LR and SVM

SVM - KNN - LR		
	With confidence	without confidence
correctly classified	63%	25%
incorrectly classified	**12%**	

4.5.3. Weighted Majority Voting

In this method, while combining individual classifiers with majority voting method, instead of giving all classifiers same importance, we tried to give them weights that contribute in decision making process. For weight values we used confidence measures over sum of all confidence measures. Confidence measures for SVM, k-NN and LR are 81%, 83% and 72% respectively. Thus, weights of those classifiers are 0.81/2.36, 0.83/2.36 and 0.72/2.36 for SVM, k-NN and LR respectively. We multiplied class information with weight (confidence) of classifier for each test data and check whether it is in the range that is considered as low confident. In our case, that range was -0.35 to 0.35. Table 4.22 gives classification information of Weighted Majority Voting model.

Table 4.22. Weighted Majority Voting model that combines k-NN, LR and SVM

SVM - KNN - LR		
	With confidence	without confidence
correctly classified	73	14
incorrectly classified	**13**	

4.6. More Performance Measures

Besides classification accuracy and confidence there are other performance indicators of classification. For example, we used FN count to compare two classifiers. Likewise TP, TN and FP counts are classification performance indicators. There are also other measures that are functions of these four parameters. These statistical performance measures will be explained in this section.

Sensitivity: It is the proportion of correctly identified positive samples over actual positive samples [38]. Equation of Sensitivity is as below. Sensitivity only tells about positive samples, it does not give information about negative samples. In medical data analysis, high Sensitivity values indicate that positive (sick) instances are well captured and there are low FN decisions.

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.1}$$

Specificity: Specificity measures how correctly negative instances are captured [38]. It is very similar with sensitivity on negative instances. For IUGR case Specificity gives us information about how correctly healthy babies are identified. 100% specificity indicates all negative instances are predicted correctly.

$$Specificity = \frac{TN}{TN + FP} \tag{4.2}$$

Positive Predictive Value (PPV): This measure gives us positive information about instances that are correctly classified [39]. This measure is also very important since in

medical cases it gives the proportion of patients with positive test results who are correctly diagnosed.

$$PPV = \frac{TP}{TP + FP} \tag{4.3}$$

Matthews Correlation Coefficient (MCC): This measure gives the quality of binary classification. MCC value can be between -1 and 1. 1 indicates perfect prediction, 0 means an average random prediction and -1 indicates an inverse prediction [40]. Instead of giving all four parameter that are used for indicating performance (TP, TN, FP, FN), MCC gives very good information of those measures. Formula to calculate Matthews Correlation Coefficient is given by Equation (4.4). A balance between sensitivity and specificity becomes an important issue in our case.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.4}$$

Probability Excess: This is also a performance measure that is independent of relative frequency of target class while Sensitivity, Specificity and MCC are highly affected by it [41]. Probability excess is simply equal to *Sensitivity*+*Specificity*-1. The Equation (4.5) gives the formula of Probability Excess.

$$\Pr ob.Excess = \frac{TP * TN - FP * FN}{(TP + FN)(TN + FP)} \tag{4.5}$$

In this study, we used 13 different classification methods to classify IUGR data. Some of them are single classifiers, some of them uses rejection, some or combined methods. These 13 techniques which are explained in earlier sections are:

1. Support Vector Machines with $C = 13000, \gamma = 0.03$ and RBF kernel
2. k-Nearest Neighbor with k=5
3. Logistic Regression

4. SVM with rejection of instances whose probability estimate is between the range -1 and 1
5. SVM with rejection of instances that's probability estimate is between -2.5 and 2.5
6. K-NN with rejection of unconfident decisions
7. K-NN, that has applied to rejected instances from SVM
8. K-NN that has applied to rejected instances from SVM. Than predictions of k-NN and SVM have been compared with majority voting method.
9. Majority Voting with k-NN and SVM
10. Majority Voting with LR and SVM
11. Majority Voting with LR, k-NN and SVM. (Needs full agreement for decision making)
12. Majority Voting with LR, k-NN and SVM (Majority agreement is enough for decision making)
13. Weighted Majority Voting with LR, k-NN and SVM.

Table 4.23 combines all of the performance measures of these 13 experiments in one table. First column indicates experiment numbers which can be referenced from above listing. Following four columns give TP, TN, FP, FN values that have been given earlier for each experiment separately. Sixth column gives accuracy percentage of each experiment. As seen from the table, fifth and sixth experiments achieved the best classification accuracy percentage. Those experiments also had best Sensitivity, Specificity, PPV, MCC, Probability Excess and Confidence values. However, 29% and 31% of the predictions were rejected because of their low confident estimations. In terms of FN count, experiment 5 is the best with 8 FNs.

Table 4.23. All performance measures for 13 experiments

EXP	tp	tn	fp	fn	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	MCC	Prob. Excess	Confidence (%)	Rejection %
1	25	56	4	15	81.00	62.50	93.33	86.21	0.60	0.56	81.00	-
2	22	57	3	18	79.00	55.00	95.00	88.00	0.57	0.50	83.20	-
3	23	49	11	17	72.00	57.50	81.67	67.65	0.41	0.39	72.00	-
4	21	53	1	14	83.15	60.00	98.15	95.45	0.66	0.58	83.14	11
5	16	45	0	8	88.41	66.67	100.00	100.00	0.75	0.67	87.14	31
6	17	45	0	9	87.32	65.38	100.00	100.00	0.74	0.65	93.88	29
7	24	58	2	16	82.00	60.00	96.67	92.31	0.63	0.57	78.70	-
8	23	55	1	14	83.87	62.16	98.21	95.83	0.68	0.60	83.79	7
9	20	53	0	13	84.88	60.61	100.00	100.00	0.70	0.61	100.00	14
10	22	46	1	14	81.93	61.11	97.87	95.65	0.65	0.59	100.00	17
11	17	46	0	12	84.00	58.62	100.00	100.00	0.68	0.59	100.00	25
12	25	56	4	15	81.00	62.50	93.33	86.21	0.60	0.56	90.00	-
13	20	53	0	13	84.88	60.61	100.00	100.00	0.70	0.61	91.14	14

The lowest performance achieved from every performance measure except Sensitivity was third experiment which is the experiment with Logistic Regression applied. Sensitivity of second experiment (k-NN) is lower than third because, count of FN instances of second experiment is higher. However, FP count of second experiment is much lower than third that causes the low sensitivity of second experiment. Sensitivity values are much higher than specificity values in every experiment. This is because count of FNs is much higher than FPs. The MCC measure, that gives better information about classification performance with using all of performance parameters (FP, FN, TP and TN), has the best value with experiment five and worst with the experiment three. This situation is also same with Probability Excess measure.

Some of the confidence measures were explained earlier except voting experiments. For experiments that used majority voting method with two classifiers, we set confidence measure to 100%. This is because we made decision if both classifiers predict same class for an instance. This results in 100% confidence. This case is also same in experiment 11 since it needs full agreement in order to make classification. For experiment 12, if a prediction is made on having two classifier same prediction out of three classifiers, we assigned 60% confidence for that case, and if prediction is made on full agreement we

assigned 100% confidence, concluding in 90% overall confidence. For the last experiment since we used weight and multiplied them with class predictions of classifiers, we obtained values between 1 to -1. Values that are closer to 1 and -1 indicate more confident classifications and values between -0.35 and 0.35 have been rejected. Absolute values of those values were used as the confidence of those test instances. For every fold, confidence of that fold has been calculated as individual confidence values of test instances in the fold.

Since we are classifying IUGR data with respect to NST value, which is an early indicator of NST, classification performance is more important than rejection count, because rejected instances can be evaluated with later measurement during pregnancy. In addition, confidently classified decisions expedite the diagnosis stage. Thus, k-NN classification with rejecting and SVM classification with rejecting instances that's outputs are between -2.5 and 2.5 experiments outperforms all other classification methods that are used in this study. Between those two experiments, although experiment six has more rejections it is better than experiment five because of its lower FN count.

5. CONCLUSION

In this study, an intelligent IUGR risk estimation systems based on ultrasound readings PI, RI of UA, MCA and DV, and AFI is presented. A SVM, a k-NN and a LR classifier and various ensemble strategies are used. Voting and confidence based voting is applied. The prediction performance is evaluated by many performance measures such as; sensitivity, specificity, PPV, Matthews Correlation Coefficient, Probability Excess and Confidence. The outperformed system consists of data preprocessing, classification with SVM and rejection of low confident predictions. Data processing consists of scaling the data and dimensionality reduction. In this study it is shown that automated dimensionality systems like PCA has not affected classification performance positively. Instead, we reduced the dimension of IUGR dataset manually with the medical expertise. We used the ultrasound features such as PI and RI of UA, MCA and DV and AFI that are well correlated with NST (non-stress test). NST is an early indicator of IUGR. In this study we aim to predict NST measurement of a fetus being reactive or not.

It has observed that the classification performance of SVM and LR were improved by scaling the values in the data. Moreover, non-invasive ultrasound readings have increased accuracy of classification. The used confidence based methods provide uncertainty management and could be used to control diagnosis strategy and suppressing false alarms.

A balance between sensitivity and specificity is important. This is observed by values of sensitivity and specificity. Also MCC and probability excess are used as performance measures.

We used three different types of classifiers. First group consists of basic SVM, k-NN and LR classifiers. In this group SVM outperforms other techniques. We had two experiments that use SVM and rejecting unconfident test instances. The confidence intervals of rejection were different. Third rejection technique was using k-NN classifier that also rejects low confident test instances. In the third group we combined classifiers. Two of the experiments in this group were formed to classify rejected instances of another

classifier. Other five of the classifiers in the third group used several classifiers in order to form a consensus and make prediction on tests sets. Outperforming classifier was in the second group, one of the SVM based classifier that uses rejection.

Between those tried methods, experiments that use SVM seemed more consistent and outperformed others within its group. For overall evaluation, experiment that classifies IUGR data with SVM and rejecting test instances that's output is between -2.5 and 2.5 outperformed other twelve experiments. Since this study aimed to predict reactivity of NST measure to make early prediction of IUGR, high rejection of the outperforming experiment is less important than accuracy in medical basis. Rejected instances can be evaluated in later stages of pregnancy.

As a future work, the performance of SVM classifier with rejection on different medical datasets or on a larger IUGR dataset would be analyzed to generalize the performance of this technique on medical applications. Since, one of the most important issues in machine learning community is to find the optimal architecture of a classification system, another future direction of the study would be combining classifiers in different ways that may result more consistent predictions. Especially in the pattern recognition community, combinations of classifiers are proposed to improve the classification performance of single classifiers and widely used.

APPENDIX B: DOCUMENT FOR MATLAB INTERFACE OF LIBSVM

This tool provides a simple interface to LIBSVM, a library for support vector machines (http://www.csie.ntu.edu.tw/~cjlin/libsvm). It is very easy to use as the usage and the way of specifying parameters is the same as that of LIBSVM.

On Windows systems, pre-built 'svmtrain.dll' and 'svmpredict.dll' are included in this package, so no need to conduct installation.

Usage

matlab> model = svmtrain (training_label_vector, training_instance_matrix, [,'libsvm_options']);

-training_label_vector:
An m by 1 vector of training labels.
-training_instance_matrix:
An m by n matrix of m training instances with n features.
It can be dense or sparse.
-libsvm_options:
A string of training options in the same format as that of LIBSVM.

matlab> [predicted_label, accuracy, decision_values/prob_estimates] = svmpredict(testing_label_vector, testing_instance_matrix, model [,'libsvm_options']);

-testing_label_vector:
An m by 1 vector of prediction labels. If labels of test
data are unknown, simply use any random values.
-testing_instance_matrix:
An m by n matrix of m testing instances with n features.

It can be dense or sparse.

-model:

The output of svmtrain.

-libsvm_options:

A string of testing options in the same format as that of LIBSVM.

Returned Model Structure

The 'svmtrain' function returns a model which can be used for future prediction. It is a structure and is organized as [Parameters, nr_class, totalSV, rho, Label, ProbA, ProbB, nSV, sv_coef, SVs]:

-Parameters: parameters
-nr_class: number of classes; = 2 for regression/one-class svm
-totalSV: total #SV
-rho: -b of the decision function(s) wx+b
-Label: label of each class; empty for regression/one-class SVM
-ProbA: pairwise probability information; empty if -b 0 or in one-class SVM
-ProbB: pairwise probability information; empty if -b 0 or in one-class SVM
-nSV: number of SVs for each class; empty for regression/one-class SVM
-sv_coef: coefficients for SVs in decision functions
-SVs: support vectors

If you do not use the option '-b 1', ProbA and ProbB are empty matrices. If the '-v' option is specified, cross validation is conducted and the returned model is just a scalar: cross-validation accuracy for classification and mean-squared error for regression.

More details about this model can be found in LIBSVM FAQ (http://www.csie.ntu.edu.tw/~cjlin/libsvm/faq.html) and LIBSVM implementation document (http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf).

Result of Prediction

The function 'svmpredict' has three outputs. The first one, predictd_label, is a vector of predicted labels. The second output, accuracy, is a vector including accuracy (for classification), mean squared error, and squared correlation coefficient (for regression). The third is a matrix containing decision values or probability estimates (if '-b 1' is specified). If k is the number of classes, for decision values, each row includes results of predicting k(k-1/2) binary-class SVMs. For probabilities, each row contains k values indicating the probability that the testing instance is in each class. Note that the order of classes here is the same as 'Label' field in the model structure.

REFERENCES

1. Hussain, W. and W. Ishak, *The Potential of Neural Networks in Medical Applications*, http://www.generation5.org/content/2004/NNAppMed.asp, 2004.

2. Veropoulos, K., N. Cristianini and C. Campbell, *The Applications of Support Vector Machines to Medical Decision Support: A Case Study*, *ACAI99,* Chania, 1999.

3. Awad, M. and L. Khan, *Applications and Limitations of Support Vector Machines*, University of Texas at Dallas, 2004.

4. Valsamakis, G., C. Kanaka-Gantenbein, A. Malamitsi-Puchner and G. Mastorakos, "Causes of Intrauterine Growth Restriction and the Postnatal Development of the Metabolic Syndrome", *Women's Health and Disease: Gynecologic, Endocrine, and Reproductive Issues*, Volume 1092, 138–147, December 2006.

5. Page, D and D. Craven, "Biological Applications of Multi Relational Data Mining", *ACM SIGKDD Explorations Newsletter*, Vol. 5, Issue 1, pp 69-79, 2003.

6. Barakat, N., "Rule-Extraction from Support Vector Machines for Medical Diagnosis – Prediction and Explanation", *ITEE Seminar*, University of Queensland, Australia, 2005.

7. Guler, N. and F.S. Gurgen, "The Effects of Data Properties on Local, Piecewise, Global, Mixture of Experts and Boundary-Optimized Classifiers for Medical Decision Making", *International Symposium of Computer and Internet Sciences* (ISCIS2004), pp. 51-61, 2004.

8. James D.K., P.J. Steer, C.P. Weiner and B. Gonik, "High Risk Pregnancy Management Options", *Fetal Growth Disorders*, Chapter 12 by Ahmet Alexander Baschat, Third Edition, Saunders Elseiver, pp. 240-265, 2006.

9. Campbell S., S.L. Warsof, D. Little, et al., "Routine Ultrasound Screening for the prediction of Gestational Age", *Obstet Gynecol*, vol 65, pp. 613-620, 1985.

10. Gurgen, F., E. Onal and F. Varol, "IUGR Detection By Ultrasonographic Examinations Using Neural Networks", *IEEE Eng Medi Biol Mag*, Vol 16, No 3, pp. 55-58, May/june, 1997.

11. Gurgen, F., N. Guler and F. Varol, "Antenatal Fetal Risk Assessment Using a NeuroFuzzy Technique", *Engineering in Medicine and Biology Magazine, IEEE* Vol. 20, No 6, pp. 165-169, 2001.

12. Balik G. (Supervisor Fusun Varol), *Relationship Between Veous Doppler and Perinatal Outcomes in Fetal Growth Restriction*, Gynecology and Obstetrics Speciality Thesis, Trakya University, 2005.

13. Moguerza, J.M. and A. Munoz, "Support Vector Machines with Applications". *Statistical Science*. 21, pp. 322-226, 2006.

14. Vapnik V., "The Nature of Statistical Learning Theory", *Springer Verlag*, NewYork, 1995.

15. Bartlett, P.L., M.I. Jordan and J.D. McAuliffe, "Convexity, classification, and risk bounds". *Journal of the American Statistical Association* 101(473), 138–156, 2006.

16. Pelcksman, K., J. De Brabanter, J.A.K. Suykens and B. De Moor, "Handling Missing Values in Support Vector Machine Classifier", *Neural Networks*, Vol. 18, pp. 684-692, 2005.

17. Yu, L. and H. Liu, "Feature Selection for High Dimensional Data: A Fast Correlation-Based Filter Solution", *20th International Conference on Machine Learning,* Washington, 2003.

18. Alpaydin, E., *Introduction to Machine Learning*, MIT Press, 2005.

19. Vidal, R., Y. Ma and S. Sastry, "Generalized Principal Component Analysis (GPCA)", *IEEE Computer Society Conference,* 2003.

20. Burges, C.J.C., "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, Vol. 2, pp. 121-167, 1998.

21. Akbani, R., S. Kwek and N. Japkowicz, "Applying Support Vector Machines to Imbalanced Datasets", *XVth European Conference on Machine Learning (ECML04)*, 2004.

22. Chen W., and C. J. Lin. "Combining SVMs with various feature selection strategies". *Feature extraction, foundations and applications*. Springer-Verlag, Berlin, 2006.

23. Mojsilović, A. "A Logistic Regression Model for Small Sample Classification Problems with Hidden Variables and Non-Linear Relationships: An Application in Business Analytics", *ICASSP* 2005

24. Xu, L, M.C. Chow, and X.Z. Gao, "Comparisons of logistic regression and artificial neural network on power distribution systems fault cause identification"*, SMCia/05*, p. 128 – 131, 28-30 June 2005.

25. Uyar, A. and F. Gurgen, "Arrhythmia Classification Using Serial Fusion of Support Vector Machines and Logistic Regression"*, IDAACS 2007*, p 560 – 565.

26. Si, L. and T. Kanungo, "Thresholding Strategies for Text Classifiers: TREC-2005 Biomedical Triage Task Experiments", *Proceedings of the 2005 Text Retrieval Conference (TREC)*, 2005.

27. Tenomo, K., *Nearest Neighbor Tutorial*, http://people.reoledu.com/kard./tutorial/KNN, 2001.

28. Özkaya, A.U. anf F. Gürgen, "Arrhythmia Classification with Confidence-Driven Serial Fusion Based on Support Vector Machines, Learning and Intelligent Optimization" *LION'07,* Trento, Italy, 2007.

29. Kittler, J., M. Hatef, R.P.W. Duin, and J. Matas, "On Combining Classifiers"*, IEEE Transactions On Pattern Analysis And Machine Intelligence,* Vol. 20, No. 3, March 1998.

30. Delany S.J., P. Cunningham, D. Doyle and A. Zamolotskikh, "Generating Estimates of Classification Confidence for a Case-based Spam Filter", *Proceedings of the 6th International Conference on Case-based Reasoning (ICCBR '05),* LNAI 3620, p170-190, Springer Verlag, 2005.

31. Roy, V. and S. Madhvanath, "A Skew-tolerant Strategy and Confidence Measure for k-NN Classification of Online Handwritten Characters", HP Laboratories, *Submitted to International Conference on Frontiers on Hand-writing Recognition*, 2008.

32. Melnik, O., Y. Vardi and C. Zhang, "Mixed Group Ranks: Preference and Confidence in Classifier Combinations", *IEEE Transactions On Pattern Analysis and Machine Intelligence*, Vol. 26, No. 8, 2004.

33. Dimou, I.N., G.C. Manikis and M.E. Zervakis, "Classifier Fusion Approaches for Diagnostic Cancer Models", *EMBS Annual International Conference*, New York City, USA., 2006.

34. Kuncheva, L.I., J.C. Bezdek and R.P.W. Duin, "Decision templates for multiple classifier fusion: An experimental comparison", *Pattern Recognition*, 34, (2), 2001, 299-314., 2001.

35. Hong, P., L. Chengde, L. Linkai and Z. Qifeng, "Accuracy of Classifier Combining Based on Majority Voting", 2007 *IEEE International Conference on Control and Automation*, Guangzhou, CHINA, 2007.

36. Chang, C. C. and C. J. Lin, *LIBSVM: a Library for Support Vector Machines*, http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

37. Hsu, C. W., C. C. Chang and C. J. Lin, *A Practical Guide to Support Vector Classification*, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 2001.

38. Altman, D.G. and J.M. Bland, *Statistics Notes: Diagnostic tests 1: sensitivity and specificity*, *BMJ* 1994;308:1552.

39. Altman, D.G. and J.M. Bland, *Statistics Notes: Diagnostic tests 2: predictive values*, *BMJ* 1994;309:102.

40. Wikipedia, Free Encyclopedia, *Matthews Correlation Coefficient* http://en.wikipedia.org/wiki/Matthews_Correlation_Coefficient.

41. Su, C., C. Chen and Y. Ou, “Protein disorder prediction by condensed PSSM considering propensity for order or disorder”, *BMC Bioinformatics*, 2006.