GAZİ ÜNİVERSİTESİ
Gazi
1926

BUZLANMA TAHMİNİ İÇİN BİR ALGORİTMA GELİŞTİRİLMESİ VE MOBİL UYGULAMASININ GERÇEKLEŞTİRİLMESİ

Hatice TIRAŞOĞLU

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2017

Hatice TIRAŞOĞLU tarafından hazırlanan "BUZLANMA TAHMİNİ İÇİN BİR ALGORİTMA GELİŞTİRİLMESİ VE MOBİL UYGULAMASININ GERÇEKLEŞTİRİLMESİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Yrd. Doç. Dr. Fecir DURAN

Bilgisayar Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ...……

Başkan: Prof. Dr. Ömer Faruk BAY

Elektrik-Elektronik Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ……...

Üye: Yrd. Doç. Dr. Mustafa YENİAD

Bilgisayar Mühendisliği, Yıldırım Beyazıt Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ……...

Tez Savunma Tarihi: 16/06/2017

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..……

Prof. Dr. Hadi GÖKÇEN

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Hatice TIRAŞOĞLU

16/06/2017

BUZLANMA TAHMİNİ İÇİN BİR ALGORİTMA GELİŞTİRİLMESİ VE MOBİL UYGULAMASININ GERÇEKLEŞTİRİLMESİ

(Yüksek Lisans Tezi)

Hatice TIRAŞOĞLU

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2017

ÖZET

Soğuk hava ve ağır kış şartları, yollarda buzlanmaya sebep olmakta ve bu nedenle her yıl birçok ölümlü, yaralanmalı ve maddi hasarlı kaza meydana gelmektedir. Bu çalışmada yollardaki buzlanmadan kaynaklı kazaların engellenmesine yönelik bir buzlanma tahmin algoritması ve mobil uygulaması geliştirilmiştir. Geliştirilen uygulama ile sürücülerin güzergâhları doğrultusunda buzlanma oluşumu ile ilgili ön bilgi verilmesi amaçlanmaktadır. Yol durum sensörü ve hava istasyonlarından alınan sıcaklık, çiğ noktası, hissedilen sıcaklık, rüzgâr şiddeti, rüzgâr yönü, bağıl nem, rüzgâr hızı ve buzlanma bilgisi geliştirilen tahmin algoritmasının eğitim veri kümesi olarak kullanılmıştır. Eğitim veri kümesi iki sınıfa sahip olup, buzlanmanın olup olmadığını ifade etmektedir. Geliştirilen mobil uygulama üzerinde sistemin eğitimi tamamlandıktan sonra meteorolojiden hava durumu özellikleri tahmin bilgisi alınarak gelecek 12 saat için buzlanma tahmini yapılmaktadır. Bu çalışmada ayrıca, veri madenciliği yöntemlerinden çok katmanlı algılayıcı (ÇKA) sinir ağı modeli, doğrusal ve doğrusal olmayan destek vektör makinaları (DVM) geliştirilen sistemin doğruluğunu ölçmek ve veri kümesini değerlendirmek için kullanılmıştır. Performans değerlendirmesi amacıyla çalışmada kullanılan algoritmaların sınıflandırma doğruluğuna bakıldığında, toplam doğru sınıflandırılan örnek sayısı temel alındığında ÇKA en iyi sonucu vermiştir. Sadece buzlanmanın olduğu sınıfın doğru sınıflandırılan örnek sayısına bakıldığında ise Doğrusal DVM üstünlük sağlamıştır. Geliştirilen tahmin algoritmasında sınıflandırma doğruluğu diğerlerine kıyasla daha az olmasına rağmen eğitimde kullanılan örnek sayısı arttıkça, buzlanma tahmin doğruluğunda doğru orantılı olarak iyileşme gözlenmiştir.

Bilim Kodu	:	92429
Anahtar Kelimeler	:	Gömülü Sistemler, Veri Madenciliği, Hava Tahmini, Buzlanma Tahmini, Android Uygulama Geliştirme
Sayfa Adedi	:	83
Danışman	:	Yrd. Doç. Dr. Fecir DURAN

DEVELOPMENT OF AN ALGORITHM FOR ICING FORECAST AND IMPLEMENTATION OF MOBILE APPLICATION

(M. Sc. Thesis)

Hatice TIRAŞOĞLU

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2017

ABSTRACT

Cold weather and heavy winter conditions cause ice on the roads, and this is why many deaths, injuries and property accidents happen every year. In this study, an icing forecast algorithm and a mobile application were developed to prevent accidents caused by ice on the roads. With the developed application, it is aimed to give information about the formation of icing in the direction of the routes of the drivers. Temperature, dew point, real feel temperature, wind intensity, wind direction, relative humidity, wind speed and ice information from the road condition sensor and weather stations were used as training data set of the developed estimation algorithm. The training data set has two classes and indicates whether or not there is icing. After the system training is completed on the developed mobile application, forecast information about the meteorology weather features is taken and icing estimation is done for the next 12 hours. In this study, multi-layer (MLP) neural network model, linear and non-linear support vector machines (SVM) from data mining methods are used to measure the accuracy of the developed system and evaluate the data set. When the classification accuracy of the algorithms used for the performance evaluation is considered, it is the best result of MLP based on the total number of correctly classified samples. When the class in which only icing occurs is based on the correct number of correctly classified samples, Linear SVM is superior. Despite the fact that the accuracy of classification in the proposed estimation algorithm is less than that of the others, as the number of samples used in education increases, the proportion of improvement in the prediction accuracy of ice has been observed

Science Code	:	92429
Key Words	:	Embedded System, Data Mining, Weather Prediction, Icıng Prediction, Android Application Programming
Page Number	:	83
Supervisor	:	Assist. Prof. Dr. Fecir DURAN

TEŞEKKÜR

Çalışmalarım boyunca yardımları ve katkılarıyla bana yol gösteren değerli danışman hocam Sayın Yrd. Doç. Dr. Fecir DURAN' a, destekleriyle Dr. Meral ÖZARSLAN YATAK 'a, eğitimim boyunca bana ilham kaynağı olan sayın hocalarıma, maddi ve manevi varlığıyla eğitimim boyunca bana destek olan sevgili babam Prof. Dr. Engin TIRAŞOĞLU 'a, beni hiçbir zaman yalnız bırakmayan anneme ve ağabeylerime teşekkürü bir borç bilirim. Bu çalışma 2210-D Sanayiye Yönelik Yurt içi Yüksek Lisans burs programı kapsamında (başvuru numarası 1649B021507233) TÜBİTAK tarafından desteklenmiştir.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ ix

ŞEKİLLERİN LİSTESİ xii

RESİMLERİN LİSTESİ xiv

SİMGELER VE KISALTMALAR xv

1. GİRİŞ 1

2. BUZLANMA TAHMİNİ İLE İLGİLİ YAPILAN ÇALIŞMALAR 5

3. MATERYAL VE METOT 11

3.1. Hava İstasyonu 11

3.2. Yol Durum Sensörleri 12

3.3. Android Platform 13

3.3.1. Android işletim sistemi 13

3.3.2. Android Studio geliştirme ortamı 17

3.3.3. React Native kütüphanesi 18

3.3.4. Android ile web servis kullanımı 20

3.3.5. Android kullanıcı arayüzü 24

3.4. WEKA Yazılımı 28

3.5. Verilerin Temini 29

3.6. Verilerin İşlenmesi 31

3.6.1. Doğrusal Regresyon analizi ile kayıp verilerin tamamlanması 34

Sayfa

3.7.Veri Madenciliği 36

3.7.1. Destek Vektör Makinaları 40

3.7.2. Çok Katmanlı Algılayıcı yapay sinir ağı 45

3.8. Problem Çözümünde Algoritmik Yaklaşım 50

4. BUZLANMA TAHMİNİ İÇİN BİR MODEL GELİŞTİRİLMESİ 53

4.1. Veri Analiz Yöntemleri 53

4.1.1. Liste boyunca silme yöntemi 53

4.1.2. Doğrusal regresyon yöntemi 54

4.1.3. Rastgele altküme üretilmesi yöntemi 57

4.2. Buzlanma Tahmini için Bir Algoritma Geliştirilmesi 58

4.2.1. Tasarlanan model 58

4.2.2. Model sonuçları ve doğruluk 61

5. BUZLANMA TAHMİNİ İÇİN MOBİL UYGULAMA GELİŞTİRİLMESİ VE DENEYSEL SONUÇLAR 63

5.1. Buzlanma Tahmini için Mobil Uygulama Geliştirilmesi 63

5.2. Veri Madenciliği Yöntem Uygulamaları 67

5.2.1. Destek Vektör Makinaları tahmin sonuçları 68

5.2.2. Çok Katmanlı Algılayıcı yapay sinir ağı tahmin sonuçları 69

5.2.3. Geliştirilen buzlanma tahmin algoritmasının sonuçları 71

5.3. Buzlanma Tahmin Yöntem Uygulama Sonuçlarının Karşılaştırılması 71

6. SONUÇ VE ÖNERİLER 73

KAYNAKLAR 75

ÖZGEÇMİŞ 83

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 3.1. Hava durumu eğitim veri seti..... 31

Çizelge 3.2. YSA terminolojisi..... 46

Çizelge 4.1. Tamamlanan eksik rüzgâr hızı ve yönü verileri 56

Çizelge 5.1. DVM Kernel fonksiyonlarının toplam doğruluk yüzdeleri 68

Çizelge 5.2. “Buzlanma Var” sınıfı doğru sınıflandırılan örnek yüzdesi……... 69

Çizelge 5.3. Düğüm sayısına göre doğru sınıflandırılan örnek yüzdesi……. 70

Çizelge 5.4. Toplam doğru sınıflandırılan örnek yüzdesi……..……. 71

Çizelge 5.5. “Buzlanma Var” sınıfı doğru sınıflandırılan örnek yüzdesi…...……. 71

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 3.1. Android işletim sistemi mimarisinin temel bileşenleri 14

Şekil 3.2. Android sisteminin aktivite yaşam döngüsü akış şeması 15

Şekil 3.3. Donanım cihazıyla birleşmiş Android ve Java akış şeması 16

Şekil 3.4. Terminal kurulum komutu 18

Şekil 3.5. Proje oluşturma komutu 18

Şekil 3.6. Paketlerin yüklenmesi 19

Şekil 3.7. Yazılımın emülatore yüklenmesi işlemi 19

Şekil 3.8. Web servis yaşam döngüsünün analizi 20

Şekil 3.9. Hava durumu verileri XML doküman örneği 22

Şekil 3.10. Hava durumu verileri JSON örneği 23

Şekil 3.11. Kullanıcı arabirimi düzenini tanımlayan bir görünüm hiyerarşisinin örneği 24

Şekil 3.12. Giriş kontrol ögeleri 25

Şekil 3.13. Düzen parametreleri ile birlikte gösterilen görünüm hiyerarşisi 26

Şekil 3.14. Bir XLM kullanıcı arayüz kodu örneği 27

Şekil 3.15. WEKA arayüzü 28

Şekil 3.16. Regresyon denklem grafiği 35

Şekil 3.17. Tek bağımsız değişkene sahip basit doğrusal regresyon örneği 35

Şekil 3.18. Veri madenciliğinin beraber çalıştığı disiplinler 37

Şekil 3.19. Veri madenciliğinin modelleme sıralaması 39

Şekil 3.20. DVM algoritmasında bir dizi hiperdüzlemler ile ayrım yapılma örneği 41

Şekil 3.21. Doğrusal DVM grafiği 42

Şekil 3.22. Radyal tabanlı fonksiyon grafiği 44

Şekil 3.23. Sinir sistemi elemanlarının Yapay Sinir Ağı modelindeki terminolojisi 45

Şekil **Sayfa**

Şekil 3.24. ÇKA sinir ağı modeli örneği..... 47

Şekil 4.1. Hava durumu veri setinde bir kısım eksik veri örneği (.arff dosyası) 54

Şekil 4.2. Rastgele altküme üretilmesi yönteminde maksimum yayılım oranı seçimi ... 57

Şekil 4.3. Önerilen algoritmanın akış şeması..... 59

Şekil 4.4. Sistemin eğitilmesi ve iyileşme oranları..... 61

Şekil 5.1. Geçmiş hava durumu verileri..... 64

Şekil 5.2. Meteorolojiden alınan veriler 65

Şekil 5.3. Android Studio düzenleme editörü..... 66

Şekil 5.4. Android Studio tasarım XML dosyası..... 66

Şekil 5.5. React Native arayüz tasarımı..... 67

Şekil 5.6. Çok katmanlı algılayıcı sinir ağı grafiksel kullanıcı arayüz 70

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim 3.1. Hava istasyonu..... 11

Resim 3.2. Yol durum sensörü..... 12

Resim 3.3. Buzlanma önleyici otomatik püskürtme sistemi……..... 30

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
km	kilometre
h	saat

Kısaltmalar	**Açıklamalar**
ART	Android Runtime
API	Application Programming Interface
APK	Android Package
AVD	Android Virtual Devices
CBS	Coğrafi Bilgi Sistemleri
CPU	Central Processing Unit
DVM	Support Vector Machines/Destek Vektör
GY	Back Propagation/Geri Yayılım
HTTP	Hyper Text Transfer Protocol
JDK	JAVA Development Kit
JSON	JavaScript Object Notation
KNN	K-Nearest Neighbors Algorithm
MGM	Meteoroloji Genel Müdürlüğü
PCA	Principal Componant Analysis
RPC	Remote Procedure Call
REST	Representational State Transfer
SMOTE	Synthetic Minority Over-sampling Technique
SDK	Software Development Kit
SQL	Structured Query Language
SOAP	Simple Object Access Protocol
SMTP	Simple Mail Transfer Protocol

Kısaltmalar	Açıklamalar
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
WEKA	Waikato Environment for Knowledge Analysis
WSDL	Web Services Description Language
XML	Extensible Markup Language
YRA	Yapısal Risk Azaltma
YSA	Artificial Neural Networks

1. GİRİŞ

Her yıl soğuk hava sartlarında meydana gelen buzlanma oluşumu can ve mal kayıplarına yol açan kazalara sebep olmaktadır. Buzlanma oluşan bölgelerde ulaşım olumsuz etkilenmekte ve araçların kaza yapma riski artmaktadır. Meteoroloji Genel Müdürlüğü kazalardan kaçınmak ve ulaşımdaki aksamaların giderilmesi adına, sıcaklığın -10 derece ve altında olduğu durumlarda mecbur kalınmadıkça seyahat edilmemesini tavsiye etmektedir. Ayrıca seyahat durumlarında, Hava Tahmin Raporunu öğrenmeyi ve Hava Tahmin Bültenlerinin sürekli izlenmesinin önemini vurgulamaktadır.

Belirli bir ülke, bölge veya merkezde, bir zaman dilimi içinde görülebilecek meteorolojik olayların gözlem ve analizlere dayanılarak subjektif veya objektif yöntemler kullanılarak önceden öngörülme çalışmaları hava tahmini olarak adlandırılır.Hava tahmin raporları hazırlanırken sırasıyla gözlem, analiz ve tahmin yapılmaktadır. Gözlemler yer, gemi, yüksek atmosfer, radar, uydu ve meteorolojik istasyonlardan gelen veriler kullanılarak yapılır. Toplanan veriler haritalara işlenir. Bu haritalarda basınç, sıcaklık, nem, rüzgâr yön ve hızı, yağış vb. birçok meteorolojik olayların bilgilerini içermektedir. Yapılan tüm analizler; atmosferin durumunu gösteren değişkenlerin (sıcaklık, rüzgâr, nem ve basınç) zamana ve yere bağlı değişimlerini ifade eden denklemlerin (hareket, termodinamik, süreklilik, hidrostatik eşitlik) matematik çözümleri yapılarak gelecekteki durumunu tahmin etme işlemine "Sayısal Hava Tahmini" denir.

Türkiye'de ve dünyada kullanılan birçok Sayısal Hava Tahmin Modelleri mevcuttur. Yapılan yer, uydu, radar ve sayısal tahmin analizleri değerlendirilir ve hava tahmin raporları oluşturulur. Böylece can ve mal kayıplarını azaltmak için Bölgesel veya Uluslararası Tahmin ve Erken Uyarı Merkezleri havanın sıcaklık, basınç, nem, yağış vb. verilerini de içeren günlük veya da haftalık hava tahminleri yapmaktadır [1].

Belirli bir alanda özelleşmiş çeşitli sistemler için farklı hava tahmin modelleri geliştirilmiştir. Kontrol sistemlerinde hava tahmin modellerinin geliştirilmesinde geleneksel yöntemlerin yanında Yapay Sinir Ağı, Bulanık Mantık, Destek Vektör Makineleri vb. farklı veri madenciliği yöntemlerinden yararlanılmaktadır [2-4]. Literatürde bu yöntemlere benzer, sayısal hava tahmin modellerinin dışında spesifik amaçları yerine getirmek için

tasarlanmış bir çok tahmin modeli bulunmaktadır. Tahmin modelleri fiziksel (Sayısal modeller), istatistiksel model (Yapay Sinir Ağı, Regresyon vb.) ve ikisinin beraber kullanıldığı karma modeller olmak üzere ele alınmaktadır [5].

Ülkemizde ve diğer ülkelerde çeşitli alanlarda hizmet etmek üzere birçok erken uyarı sistemleri geliştirilmiştir. López ve diğerleri sel riskinin değerlendirilmesi ve analizi için akıllı araçlara ve sinir ağlarına dayalı uygulamalar ve uyarıların gösterimi için bir mobil cihaza uyarıcı araçların kurulmasını önermişlerdir [6]. Vodafone, mobil teknolojiler yardımıyla orman yangınlarının önüne geçmek için "Ormanlarımız Yanmasın" erken uyarı telemetri sistemini geliştirmiştir [7]. Bu sistem sayesinde yangın, başladığı an algılanabilecek ve erken müdahale sayesinde pek çok yangının önüne geçilerek ekosistem korunabilecektir.

Yol buyunca meydana gelen buzlanma veya don olaylarının tahmin edilmesi ve sürücülerin en hızlı şekilde buzlanmanın oluşma ihtimalini öğrenebilmesini sağlamak için bir erken uyarı sistemine ihtiyaç duyulmaktadır. Günümüzde yaygın olarak kullanılan mobil telefonlar bu amacı gerçekleştirmeye yönelik en etkili ve güvenilir yollardan biridir. Hareket halindeki sürücülerin bilgiye en kısa sürede ulaşması ve yol durumu hakkında erkenden bilgilendirilmesi sağlanarak ulaşımdaki aksaklıkların zamanında giderilmesi sağlanılabilmektedir. Erken uyarı sistemlerinin mobil telefon üzerinde gerçekleştirilmesi bilgiye erişim açısından en iyi çözümdür.

Bu çalışma kapsamında, buzlanma tahmininde bulunan bir erken uyarı sisteminin mobil uygulaması gerçekleştirilmiştir. Bu hususta yer ve hava istasyonlarından sıcaklık, nem, rüzgar gib hava durumunu değerleri ve buzlanma olup olmadığına dair bilgiler alınmıştır. Toplanılan veriler, elverişli bir veri kümesi haline getirilerek sistemin eğitilmesi için kullanılmıştır. Bu eğitim veri kümesi ile kendi kendini eğitebilen bir makina öğrenme algoritması mobil uygulama için geliştirilmiştir. Geliştirilen mobil uygulama ile meteorolojiden (önümüzdeki 12 saatin hava durumu özellikleri) hava durumu tahmin değerleri alınarak buzlanma tahmini üretilmiştir. Geliştirilen uygulamanın performansının değerlendirilmesi için veri madenciliği yöntemlerinden destek vektör makinaları ve çok katmanlı algılayıcı sinir ağı yöntemleri ile karşılaştırılmıştır.

Bu çalışma beş bölümden oluşmaktadır. İkinci bölümde, hava olayları ve buzlanma oluşumu ile ilgili yapılan çalışmalar hakkında bilgiler ortaya konulmaktadır. Üçüncü bölümde kullanılan materyal ve metotlar verilmektedir. Seçilen veri madenciliği yöntemleri, hangi yazılım araçlarının kullanıldığı, mobil uygulamanın hangi platformda yazıldığı ve özellikleri hakkında kapsamlı bilgiler verilmiştir. Aynı zamanda, kullanılan verilerin nasıl temin edildiği ve alınan verilerin işlenmesi basamaklarından bahsedilmiştir. Çalışmanın dördüncü kısmında veri analiz yöntemleri ve buzlanma tahmini için geliştirilen algoritma hakkında bilgiler verilmiştir. Beşinci bölümde, buzlanma tahmini için geliştirilen mobil uygulama ve deneysel sonuçlar verilmiştir. Altıncı ve son kısımda sonuç ve önerilere yer verilmiştir.

2. BUZLANMA TAHMİNİ İLE İLGİLİ YAPILAN ÇALIŞMALAR

Hava tahmini birden fazla uzmanlık alanı içeren karmaşık bir prosedürdür. Hava tahmininde, mevcut atmosferin durumunun nasıl değişeceğinin tahmini gereklidir. Bilgisayarlar tarafından havanın tahmininin yapılması prosedürü "Sayısal Hava Tahmini" olarak bilinir. Sayısal anlamda havanın tahmini için meteorologlar atmosferik modeller geliştirmişlerdir. Bu modeller atmosferik sıcaklıkların, basınç ve nemin zamanla nasıl değişeceğini tanımlamak için matematiksel yöntemler kullanılarak çıkarılan yaklaşımlardır. Denklemler bilgisayarda programlanır ve şimdiki atmosfer verileri ile bilgisayar beslenir. Bilgisayar denklemler sayesinde gelecek zamanda nasıl atmosferin değişebileceğine dair tahminler üretebilmektedir [4].

Sayısal hava tahminlerinin yanında çeşitli sistemler için geliştirilmiş birçok hava tahmin modeli vardır. Drezga ve Rahman kaba tahminleri tekrar eğitime sokarak daha iyi tahmin elde edebilmek için Yapay Sinir Ağları ile K-en yakın komşu metodunu kullanmışlardır [8]. Taylor ve Buizza yaptıkları çalışma ile YSA ile yapılan tahminlerin kullanılmasının geleneksel yöntemlerden daha iyi sonuçlar ortaya koyduğunu gözlemlemişlerdir. Hava çok hızlı değişen bir yapıdadır. Bu yüzden tahminde yapılan küçük hatalar önlem alınamazsa hızla büyür ve tahmin edilebilirliği olumsuz etkiler. Numerik tahmin modelinde tahmin edilebilirlik, hava süreç simülasyonlarının gösterdiği model hatalarından dolayı kısıtlanmaktadır [9].

Hava tahmininde YSA algoritmasının yanı sıra Destek Vektör Makinaları algoritması da yağış tahmini belirleme ve hava tahmini yapma gibi çevresel parametrelerin tahmini için kullanılabilmektedir [10]. Lu ve diğerleri kısa süreli yapılan hava tahminlerinin kalitesinın doğrusal olmayan bağlanım metotlarının kullanılmasına dayandığı için doğrusal olmayan DVM modelini önermişlerdir [11]. Rao ve diğerleri çok katmanlı algılayıcı (ÇKA) öğrenme algoritması ile DVM yöntemini hava tahmini için karşılaştırmış ve DVM 'in daha iyi olduğunu gözlemlemişlerdir [12]. Bunun yanısıra, Hava tahminlerinde birden fazla yöntemin kullanılmasının tahmin doğruluğunu iyileştirdiği düşünüldüğü için karma modeller tercih edilmektedir [13]. Nikhil ve diğerleri öz denetim nitelik haritalandırma veya nitelik seçimi ile ÇKA modelini birlikte kullanarak mevsimsel değişimlerle beraber bir karma sinir ağı modeli önermişlerdir. Uygun nitelik seçimi hem nitelik sayısını azaltılmasını

hem de tahmin doğruluk oranının artmasını sağlar. Hava tahmini, sıcaklık, yağış, nem, rüzgar hızı ve yönü, atmosfer basıncı vb. niteliklerin tahminini gerektirir. Coğrafi etkenler, civardaki yapılar ve çevre kirliliğinden dolayı doğru hava tahmininde bulunmak oldukça zordur. Tahmin sistemlerinin doğruluğu tüm bu faktörler göz önüne alınarak geliştirilebilir [14]. De Giorgi ve diğerleri güç üretim sistemi için YSA (Elman ağı ve Çok katmanlı Algılayıcı ağı) ve Numerik hava tahmin modellerini birlikte kullanarak bir model geliştirmişlerdir. Burada YSA yapısına dayalı bir karma model ile daha iyi sonuçların elde edildiği gözlemlenmiştir [5]. Chen ve diğerleri rüzgar gücü tahmini için numerik hava tahmini ile olasılık yöntemlerinin (gauss işlemi) birlikte kullanıldığı bir model geliştirilmiştir. Geliştirilen model YSA ve klasik model ile karşılaştırıldığında iyi sonuç vermiştir [15].

Hava tahmin sistemi ile buz oluşumu önceden tahmin edilip önlenebilmektedir. Yapay Sinir Sistemi (YSA) Modelleri ile sıcaklık, nem, rüzgâr hızı ve yönü gibi hava tahmin değerleri kullanılarak buz oluşumunun tahmini yapılabilmektedir. Costa ve Pasero, YSA yöntemi ile buz oluşumunun tahmin etme, tuz dağıtımının yapılması ve ince buz katmanının oluşmasını engellemek üzerine çalışmalar yapmışlardır. Çalışmada hazırlanan hava modeli farklı hava koşulları kullanılarak eğitilmiştir ve eğitilen sistem rüzgâr hızı, rüzgar yönü, sıcaklık ve nemi tahmin edebilmektedir. Tipik bir ince buz tabakası oluşumu tespit edildiğinde buz önleme sistemine bir alarm gönderilmektedir [16].

Hava tahminlerinde istasyonlardan ve meteorolojiden gelen hava tahmin verileri kullanılır. Bunlar; Sıcaklık, nem, rüzgâr ve yönü, minimum sıcaklık, maksimum sıcaklık gibi niteliklerdir. En iyi tahmin yapılırken bu niteliklerden bir kaçı veya hepsi kullanılabilmektedir. Doğru bir tahmin algoritmasının seçilmesinin yanı sıra hangi niteliklerin kullanılacağının seçilmesi de önemlidir. Hangi niteliğin daha az etkili olup hangisinin tahmin doğruluğunu daha fazla arttırdığının bulunmasında nitelik seçim yöntemlerinden yararlanılır. Bu bağlamda, tahmin yapılırken çoğunlukla veri madenciliği modelleri ile beraber yol buzlanma tahmininin doğruluğunu arttırmak için nitelik seçim yöntemleri kullanılır. Çünkü hava değerlerinin tümü buzlanmada etkin olmamaktadır. Jian çalışmasında nitelik seçimi için özellikle sürekli verilerin söz konusu olduğu problemlerde etkin sonuçlar verebilen, işleyiş ve operatörleri itibariyle genetik algoritmaya dayanan popülasyon temelli sezgisel optimizasyon tekniklerinden diferansiyel gelişim algoritmasını

kullanmıştır. Numerik yol buzlanma tahmin yaklaşımı yerel bir bölge için tasarlanmıştır. Hangi hava verisinin buzlanmayı önemli ölçüde etkilediğinin tespitine odaklanılmıştır [17].

Yol yüzeyinde buzlanma olması için [18];

- Sıcaklıklar sıfıra yakın veya sıfırın altında olmalıdır.
- Yol yüzeyinde bulunan su mevcudiyeti (nem) bulunmalıdır.
- Sıcaklık ve nemi etkileyen diğer meteorolojik özellikler dikkate alınmalıdır.

Mandale ve Jadhawar, hava tahmininde veri madenciliği teknikleri kullanımı analiz ederek performans karşılaştırılması gerçekleştirilmiştir. Çok fazla veri olması hava tahminini zorlaştırır. Hava tahmininde minimum ve maksimum sıcaklık, yağış, buharlaşma ve rüzgâr hızı hava değerleri kullanılmıştır [19]. Joshi ve diğerleri benzer şekilde veri madenciliği teknikleri ile hava tahmininde bulunmaya çalışmıştır. Çalışmada verinin anlaşılır olması ve tanımlayıcı olmasından dolayı karar ağaçları kullanılmıştır [4].

Buzlanmayı önlemek adına geliştirilen hava tahmin uzman sistemleri yol üzerindeki buzlanmayı oluşumunu önceden algılamak dışında çeşitli tahmin algoritmaları ile beraber farklı alanlar için de kullanılmıştır. Uçuş sırasında uçak üzerinde meydana gelen buzlanmalar genel havacılık kazalarına neden olan faktörlerden biridir [20]. Uçaklarda buzlanmayı önlemek için sıcaklık ve bağıl nemin yanı sıra uçuş seviyesinin yukarısı ve aşağısındaki nem verilerinin eklenmesinden dolayı Sinir Ağları yöntemi kullanılması buzlanma tahminin de doğru bir seçimdir [21]. Farklı bir çalışma olarak, nehir üzerinde ki buz oluşumu tahmininin yapılmasında genetik algoritması ve sinir ağının beraber kullanılmıştır. Karma buz tahmin sinir ağ modelleri yüksek performans göstermiştir [22]. Buzlanma tahmininin kullanıldığı bir diğer saha, enerji hat iletkenleri üzerinde buzlanma ve buz dökülmesi için kullanılmasıdır [23]. İletişim hatlarının buzla kaplanması çok büyük hasarlar meydana getirmektedir. Bu sebeple yerel bölgelerde sinir ağı modeline dayalı buzlanmayı izleme ve erken uyarı sistemleri geliştirilmiştir. Meteorolojik veriler kaydedilerek hava koşulları izlenmekte ve tahmin üretip hava koşullarının oluşturduğu tehlikeli buz oluşumu sonucu oluşabilecek kazalar için koruyucu tedbirler alınması amaçlanmaktadır [24]. İletişim hatları üzerinde ki buzlanmanın kalınlığını tahmin etmede bir diğer yöntem bulanık mantık modelleri kullanılmasıdır. Buzlanma kalınlığının tam olarak ne kadar olduğunu tahmin etmek olanaksızdır. Bu yüzden bazı araştırmacılar Bulanık Mantık ‘dan yararlanılarak buzlanma kalınlığı “ince”, “orta”, “ağır” olmak üzere üç kısımda

incelemişlerdir [25]. Daha sonraki çalışmalarda buzlanma kalınlığını ölçmede önerilen bulanık tahmin yöntemi olarak bulanık Markov Zincir modeli örnek verilebilir [26]. Buzun neden olduğu hasar güç şebekelerinde verimliliğin düşmesinde büyük rol oynamaktadır. Buz kalınlığını hesaplamada yerel bölgelerde kullanılmış bir diğer tahmin modeli DVM 'dir [27]. Modelin giriş değerleri olarak sıcaklığını, nemi, rüzgar hızını, rüzgar yönünü, güneşi ve hava basıncı kullanılmıştır. GY (Geri Yayılım) sinir ağ modeli ile DVM karşılaştırılmıştır ve DVM modelinin örnek sayısının az olduğu durumlarda buzlanma yük tahmininde Sinir Ağı modelinden daha uygun olduğu öngörülmüştür. DVM daha az sayıda öğretilmiş veri ile doğru tahmin yapabilir. Eğitilmiş veri ve tahmin verileri buzlanma sürecinde farklı olsa bile DVM uygulanabilir [28]. Bu nedenle, DVM bu çalışmada kullanılan veri madenciliği yöntemlerinden biri olarak seçilmiştir.

Yollar üzerindeki araçlar gibi sürekli akışın olduğu durumlara entegre edilen kontrol sistemlerinde hızlı müdahale hayati önem taşımaktadır. Özellikle verileri sürekli değişen ve ulaşılması zor bölgelere kurulan sistemlerin ana bilgisayardan uzaktan kontrolünün sağlanması gerekmektedir.

Mobil cihazlar herhangi bir rekabet olmadan hızla büyümektedir. Akıllı telefon numaralarındaki hızlı artış ve akıllı telefon uygulamalarının kolayca yüklenebildiği yerler olan pazarlar mobil geliştirmeyi büyük bir market haline dönüştürmüştür. Kullanıcı sayısı arttığı gibi, mobil uygulamalara alan talep de artmıştır. Bu da hızlı sonuçlar için hızlı geliştirme döngüsüne yol açmıştır. Android platform üzerinde uygulama geliştirmenin en iyi yöntem olmasının sebebi şu anda en hızlı büyüyen ve çok yaygın bir platform olmasıdır [29].

Android akıllı telefonlar, tablet ve daha birçok vb. şeylerde kullanılır. Android, Google üretimidir ve baskın olarak piyasada beklenenden daha fazla avantajlı hale gelmiş ve talep görmektedir. Mobil Telefon için geliştirilen çeşitli tahmin uygulaması bulunmaktadır [30-31]. AFAD Mobil Deprem uygulamalarını temel alarak Türkiye'de ve yakın çevresinde meydana gelen gerek güncel depremleri, gerekse geçmişte meydana gelen deprem bilgilerini akıllı telefonlar ve tablet bilgisayarlar aracılığı ile anında kullanıcılara sunmayı amaçlayan yerli bir mobil uygulama geliştirmiştir [32].

Günümüzde mobil telefonlarda kısa mesajlar yoluyla uyarı mekanizmalarının kullanılması büyük ölçüde artmıştır. Bu uygulamanın en iyi temsilcileri Amerika Birleşik Devletleri ve Japonya' dır. Erken uyarı sistemi şimdilerde en çok önerilen sistemlerden biridir [6].

3. MATERYAL VE METOT

3.1. Hava İstasyonu

Bu çalışmada verilerin bir kısmı Lufft firmasının WS-600 modeli hava istasyonu kullanılarak temin edilmiştir. Sıcaklık, bağıl nem, rüzgâr hızı, rüzgâr yönü, hava basıncı yağış tipi ve yağış miktarı gibi birçok veriyi ölçebilen bu istasyon kompakt bir yapıya sahiptir (Resim 3.1).

WS-600 hava istasyonu, düşük güç tüketimi, ısıtma, emilen radyasyon kalkanı, bakım gerektirmeyen çalışma ve açık iletişim protokolü gibi özelliklere sahiptir. Rüzgar hızı, yağış yoğunluğu duyarlılığı sırasıyla 0.1 m/s ve 0.1 mm/h değerindedir. -50 ile 60 °C arasındaki sıcaklıkları ±0.2 °C doğruluk ile ölçebilmektedir. 0 ile100 % RH arasında bağıl nemi ±2 % RH doğruluk ile ölçebilmektedir. 300 ile 1200 hPa arasındaki hava basıncını ±0.5 hPa doğruluk ile ölçebilmektedir [33].

Resim 3.1. Hava istasyonu

3.2. Yol Durum Sensörleri

Bu çalışmada kullanılan verilerin bir kısmı Luft firmasının ürettiği ARS31-UMB ve IRS31-UMB yol durum sensörleri modellerinden alınmaktadır. IRS31-UMB modeli yol yüzeyinin sıcaklığı, yol altı sıcaklığı, yol yüzeyi sıvı miktarı, sıvının donma noktası ve bunun gibi birçok veriyi ölçebilir. Bu sensör herhangi bir işlem yapmadan sadece ölçüm yaptığı için pasif bir sensördür. İki parçalı gözde tasarımı bakım ve yeniden kalibrasyonun yapılmasının kolaylaştırır. ARS31-UMB modeli, aktif yol sensörü olarak geçmektedir. Özellikle yol yüzeyinde bulunan sıvının donma noktasını belirlemek için kullanılmaktadır (Resim 3.2).

Etkin ARS31-UMB sensörü, yol yüzeyine yerleştirilir ve sensör yüzeyinin aktif soğutulması ve ısıtılması yoluyla dondurucu sıcaklığı hesaplar. Donma sıcaklığı ölçümü karışımdan bağımsızdır. İki bölmeli gövde tasarımı, birleştirilmiş sensör/elektronik ünitesinin bakım amacıyla herhangi bir zamanda birkaç dakika içinde çıkarılmasını sağlar. Arabirim dönüştürücü 8160.UISO ile birlikte, sensör yeni ve mevcut UMB ağlarına entegre edilebilir. Pasif sensör IRS31-UMB ve aktif sensör ARS31-UMB zorlanmadan birleştirilebilir. Sensörler adreslenebilir ve ağa bağlanabilir. Yol durum sensörlerinin ölçüm duyarlılığı 0.1 değerindedir [34].

Resim 3.2. Yol durum sensörü

3.3. Android Platform

Android günümüzde en hızlı büyüyen mobil platformdur [29]. Google ve Open Handset Alliance tarafından, mobil cihazlar için geliştirilmekte olan "Java" programlama dili ile yazılmış Linux Tabanlı bir işletim sistemidir.

Android, derlenmiş Java kodunu çalıştırmak için dinamik çevirmeli Android Runtime (ART) kullanır ve cihazların fonksiyonelliğini artıran uygulamaların geliştirilmesi için çalışan geniş bir programcı-geliştirici çevresine sahiptir [35]. Herkese açık geliştirici özelliği ve pazarda yayınlama kolaylığı ile uygulama geliştirmek için yaygın olarak kullanılmaktadır.

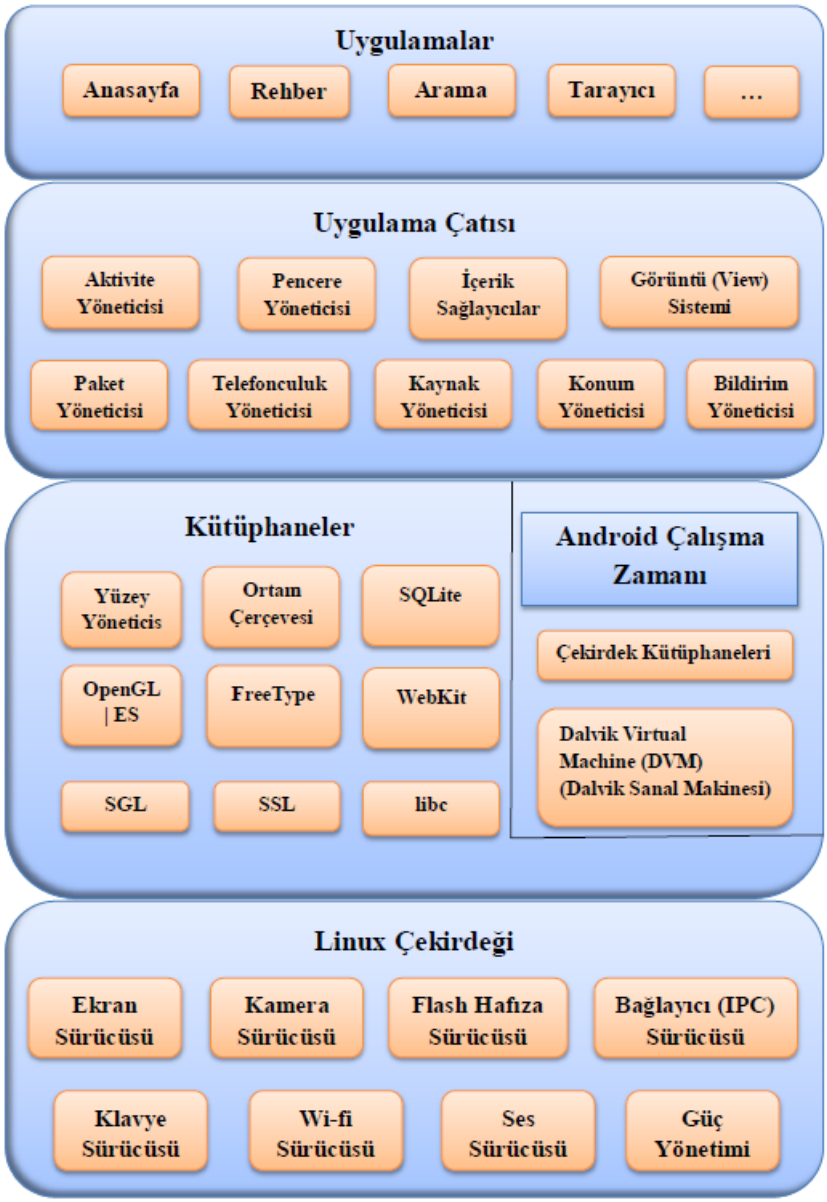
Uygulamalar, Google Play Store uygulama ortamına yüklenerek tüm Android kullanıcıların hizmetine sunulmaktadır. Geliştirilen uygulamaların mobil telefona yüklemek için kullanılan uzantı ".apk"'dır. Java programlama dilinde uygulama geliştiren, Windows kullanıcıları SDK, JDK (Java Geliştirici kiti) yükleyerek, Eclipse, Android Studio veya NetBeans yazılım geliştirme ortamlarında uygulama geliştirilebilmektedir.

Bu bölümde, geliştirilen mobil uygulamanın yazıldığı platformun sistem özellikleri, kullanılan yazılım araç gereçleri ve web servis hizmetleri hakkında kapsamlı bilgiler yer almaktadır.

3.3.1. Android işletim sistemi

Android işletim sistemi açık kaynak kodlu olup, ücretsiz ve kullanımı kolay sade bir arayüze sahiptir. Bu işletim sistemi açık kaynak kodlu olsa da, kodlarının bir kısmı Google tarafından kapalı tutulmaktadır. Google tarafından ücretsiz olmasının sebebi, sistemin daha hızlı ve çabuk gelişmesini sağlamaktır. Sistem ara katman yazılımı, kütüphaneler ve API C diliyle yazılmıştır. Uygulama yazılımları ise, Apache harmony üzerine kurulu Java-uyumlu kütüphaneleri içine alan uygulama iskeleti üzerinden çalışmaktadır.

Android işletim sisteminin yazılım geliştirme kiti (SDK), kullanıcıların kendi uygulamalarını geliştirebilmeleri için gereken araç ve uygulama programlama arayüzlerini (API) sağlamaktadır.



Şekil 3.1. Android işletim sistemi mimarisinin temel bileşenleri [36]

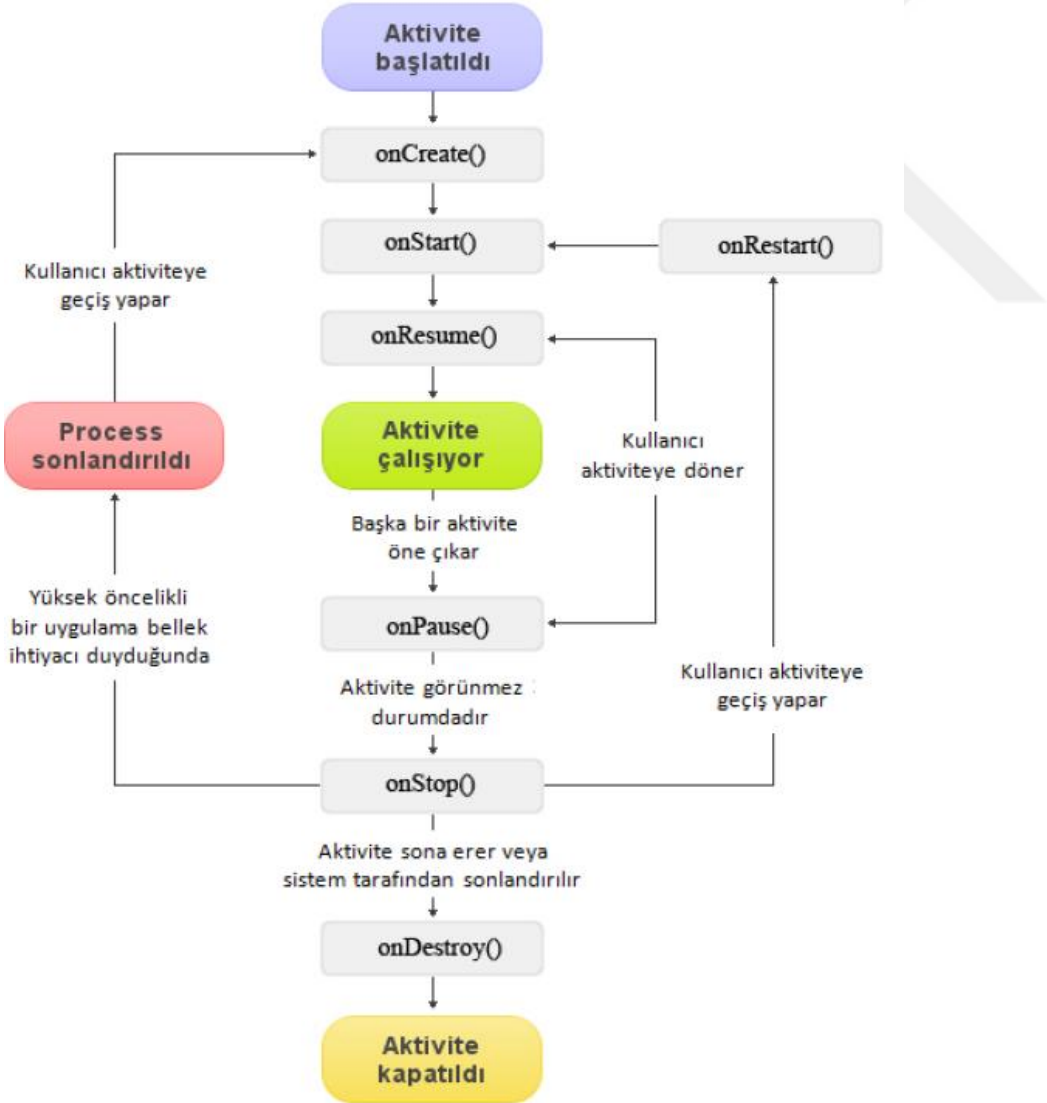
Android işletim sistemi Linux Kernel, kütüphaneler ve Android çalışma zamanı, uygulama çatısı, uygulamalar olmak üzere 4 ana katmandan oluşmaktadır (Şekil 3.1).

Linux Kernel katmanında, donanımsal bilgiler ve uygulamaların çalışabilmesi için gerekli sürücüler yer almaktadır. Kütüphaneler katmanında,C++ ve C dilleri ile yazılmış kütüphaneleri içerir. Bu katmanda sistem kütüphaneleri, mp3, mpeg4, jpg gibi çoklu ortam bileşenleri için medya kütüphaneleri ve 2D/3D grafikler için OpenGL/SGL içeren kütüphaneler bulunmaktadır. Android İşletim Sistemi kendi ilişkisel verilerini tutabildiği bir SQLite isimli bir veritabanına sahiptir. Kütüphaneler katmanında veri tabanı için SQLite kütüphaneleri gibi temel kütüphaneler de yer alır. Uygulama çatısı katmanı incelenecek

olursa Android, yazılım geliştiriciler için zengin bir platform sunmaktadır. Android uygulamalarının yazılımını oluştururken yazılımcıya uygulama çatısı sağlayan uygulama servisleri aşağıdaki gibidir [37];

Aktivite yöneticisi

Aktivitelerin yaşam döngüsünü kontrol eder. Aktivite yığınının yönetimini içerir ve bir uygulamanın nasıl çalıştığını anlamada yardımcı. bir grafiksel kullanıcı arayüzü bileşenidir. Uygulamalar ile grafiksel kullanıcı arayüzü arasındaki haberleşme aktiviteler tarafından gerçekleştirilmektedir. Her aktivitenin kendi yaşam döngüsü ve durumları Şekil 3.2. de gösterilmiştir.



Şekil 3.2. Android sisteminin aktivite yaşam döngüsü akış şeması [36]

Görüntü Sistemi

Görünümler, Aktiviteler için kullanıcı ara yüzü yapılmasında kullanılır.

Bildirim yöneticisi

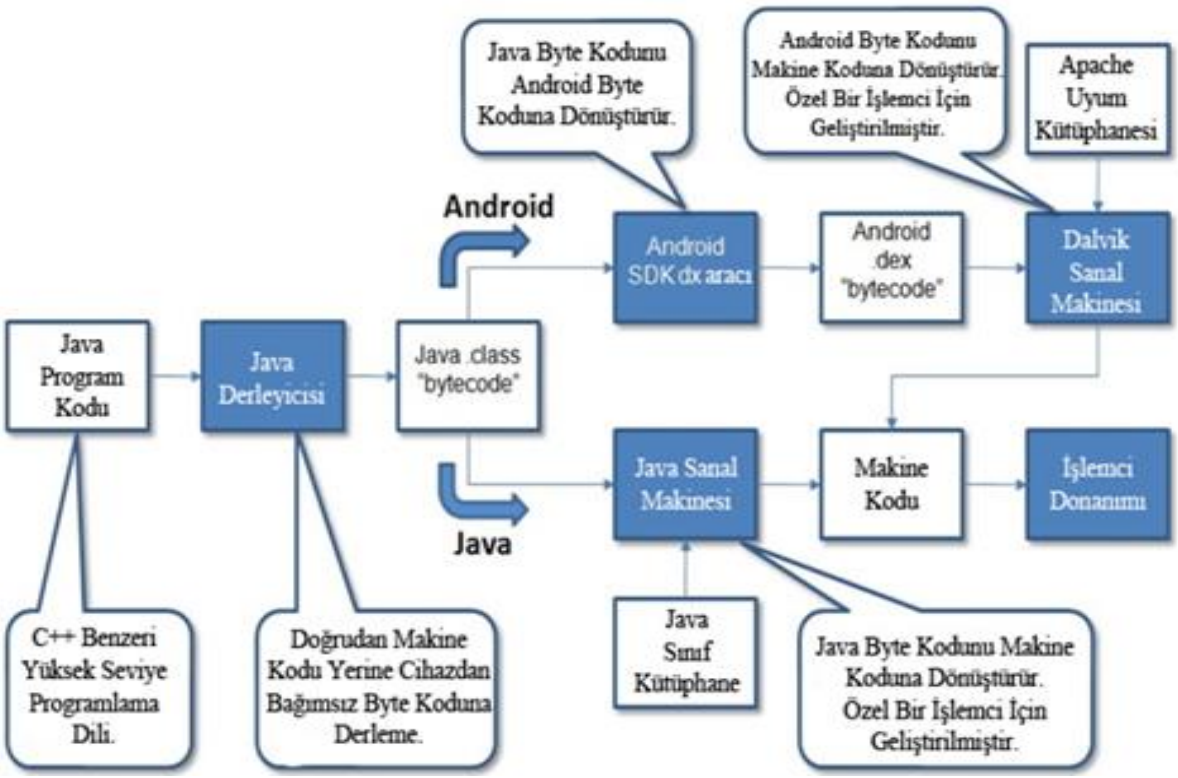
Kullanıcılara yapılan bildirimler ve uyarılar için uyumlu ve tutarlı işlev sağlar.

İçerik sağlayıcılar

Uygulamanın veri paylaşımını sağlar. Telefon rehberi, resim, müzik vb. verilerin uygulamalarca erişimini sağlayan arabirimlerdir. SQL benzeri erişim ara yüzüne sahiptirler.

Kaynak yöneticisi

Dışarıda tutulmak üzere diziler ve grafikler gibi kodsuz kaynakları destekler.



Şekil 3.3. Donanım cihazıyla birleşmiş Android ve Java akış şeması [38]

Şekil 3.3'de donanım cihazıyla birleşmiş bir Android ile Java akış şemasında söz konusu bir yazılım cihazında kullanılabilir bir program üretmek için Java ve Android işletim sisteminde, çeşitli parçaların bir arada çalışma mantığı gösterilmiştir.

3.3.2. Android Studio geliştirme ortamı

Android Studio, Mobil uygulama geliştirmek için 2013 yılında özel olarak tasarlanmış, Apache lisansı ile lisanslanmış ücretsiz bir ortamdır. Android uygulamalarının geliştirildiği, üst seviye özelliklere sahip ve Google tarafından da önerilen resmi programlama aracıdır.

Intellij IDEA'ya dayanan Android Studio, kodlama ve çalışan iş akışında hız sağlamaktadır. Android Studio geliştirme ortamında zengin kod düzenleme, hata ayıklama, test etme ve profil oluşturma araçları da dahil olmak üzere Android geliştiricileri için özel olarak hazırlanmış araçlar sunulmaktadır.

Android Studio'un temel özellikleri;

- Esnek gradle tabanlı proje inşa sistem yapısı,
- Bilgisayar üzerinde çalışma imkânı sağlayan Emulatör,
- Farklı özellik ve sürümlere göre çalışma imkânı sunan birden fazla APK çıktısı,
- Temel proje şablonlarıyla hızlı ve kolay proje üretimi,
- Ekran tasarımlarını kolaylaştıran sürükle-bırak özellikli zengin editör,
- Uygulamanın performansı, kullanılabilirliği, farklı sürümlerde çalışabilirliğinin kontrol edilebileceği test araçları,
- Kolay ve güvenli APK imzalanması,
- Ek uğraşa gerek kalmadan Google hizmetlerini uygulamaya ekleyebilme.
- Uygulama tasarımı, testi, debug yapılmasını sağlayan yazılım geliştirme kiti (SDK)

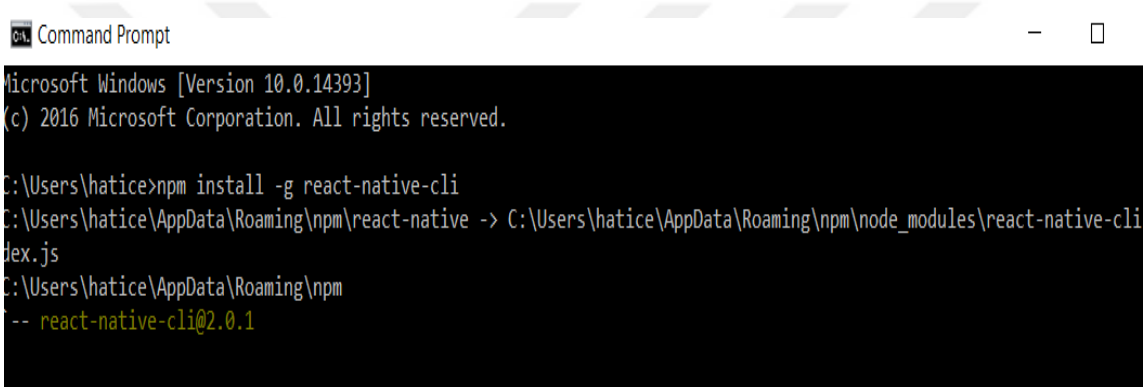
Android Studio Java geliştirme için oldukça kullanışlı, SDK yönetimi rahat, kütüphane erişimi, debug gibi işlemler için kullanım kolaylığı sağlayan bir yapıdadır [39].

Android Emülatör ile uygulamalar gerçek cihaza gereksinin duymadan sanal cihazlar kullanılarak birden fazla farklı API üzerinde çalıştırılabilmektedir. Android cihaz yapılandırmalarında prototip olarak çeşitli telefonlar, tabletler, Android Wear ve Android

TV cihazları. bulunmaktadır. Ayrıca, GPS konumu, ağ gecikmesi, hareket sensörleri ve çoklu dokunmatik girdi gibi çeşitli donanım özelliklerini simüle edebilmektedir [40].

3.3.3. React Native kütüphanesi

React Native, Facebook tarafından geliştirilen açık kaynak kodlu bir kütüphanedir. Javascript kullanarak Native mobil uygulama geliştirilmesini sağlamaktadır. React bir uygulama geliştirilirken sırasıyla Android Studio ve gerekli yazılım geliştirme araçları kurulur. Android Sanal Cihazlar (AVD) ile çeşitli API'de uygulamada kullanılmak üzere emülatör oluşturulur.



Şekil 3.4. Terminal kurulum komutu

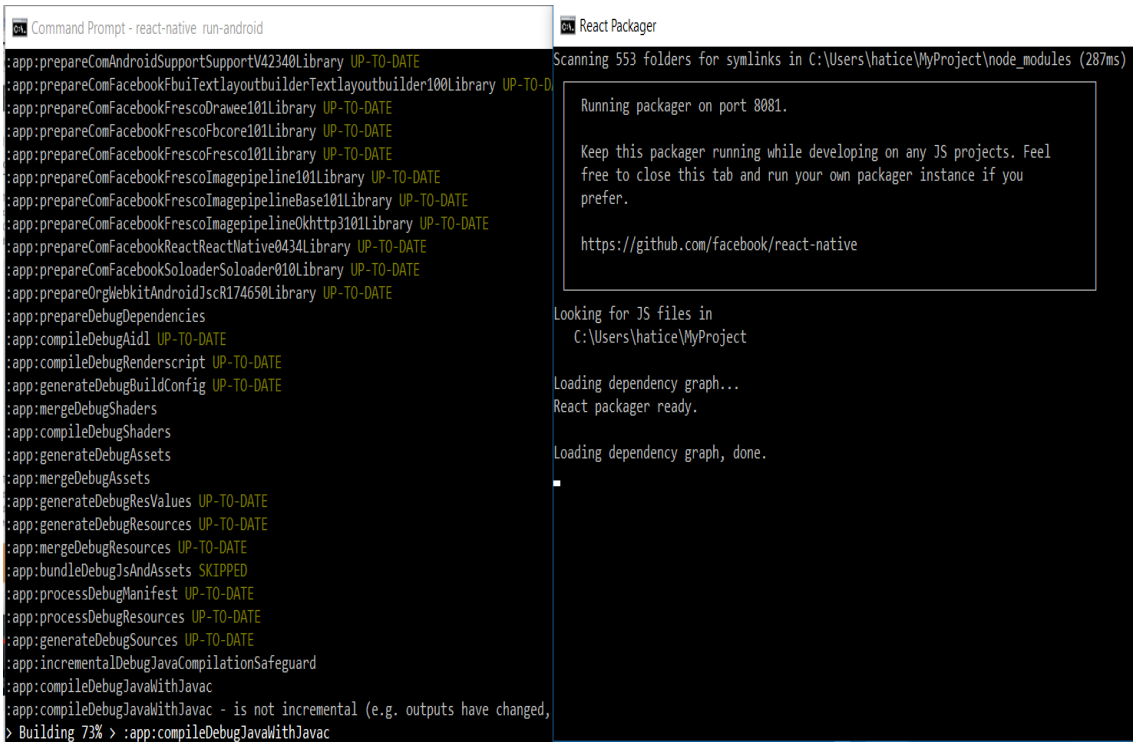
Şekil 3.4. de görüldüğü gibi –npm install –g react-native-cli yazılarak React Native npm ile yükleme tamamlanmaktadır.

```
react-native init AwesomeProject
cd AwesomeProject
react-native start
```

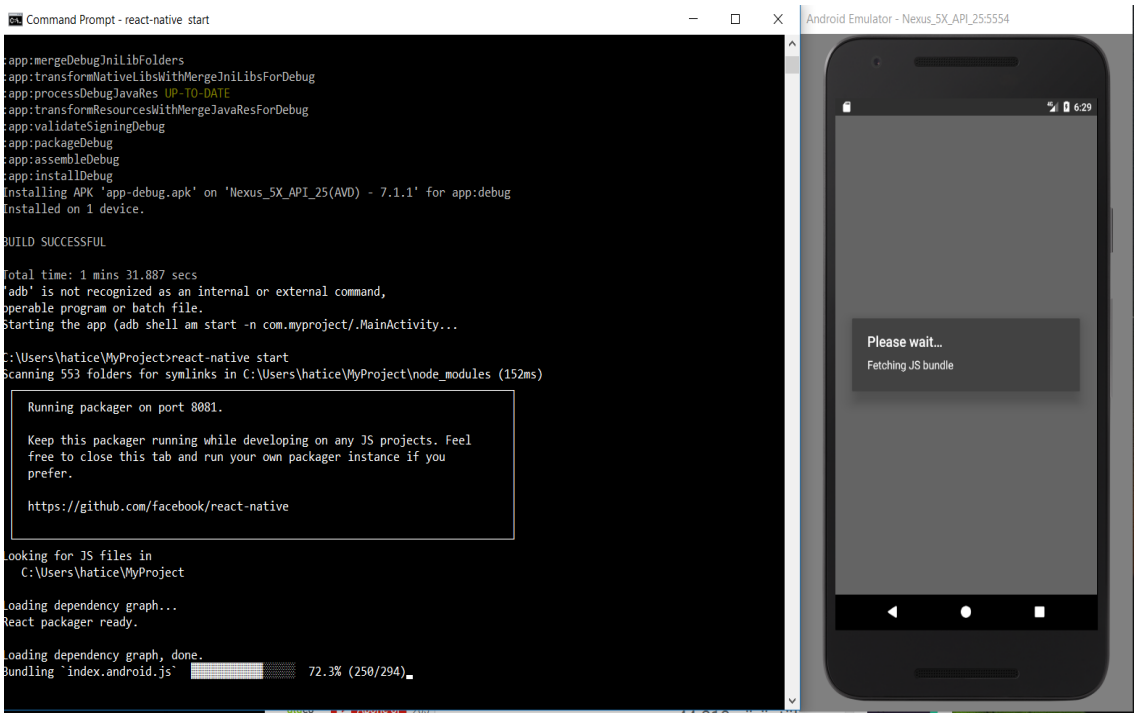
Şekil 3.5. Proje oluşturma komutu [41]

Şekil 3.5'de görüldüğü gibi $react-native init ProjeAdı yazılarak bir proje oluşturulmaktadır ve "start" komutu ile çalıştırılmaktadır.

Komut satırından dosyanın bulunduğu konuma $react-native run-android komutu yazılarak kütüphane dosyaları yüklenmektedir (Şekil 3.6) ve Native kısmı tamamlandıktan sonra (Şekil 3.7) proje yüklenerek açılmaktadır.



Şekil 3.6. Paketlerin yüklenmesi



Şekil 3.7. Yazılımın emülatore yüklenmesi işlemi

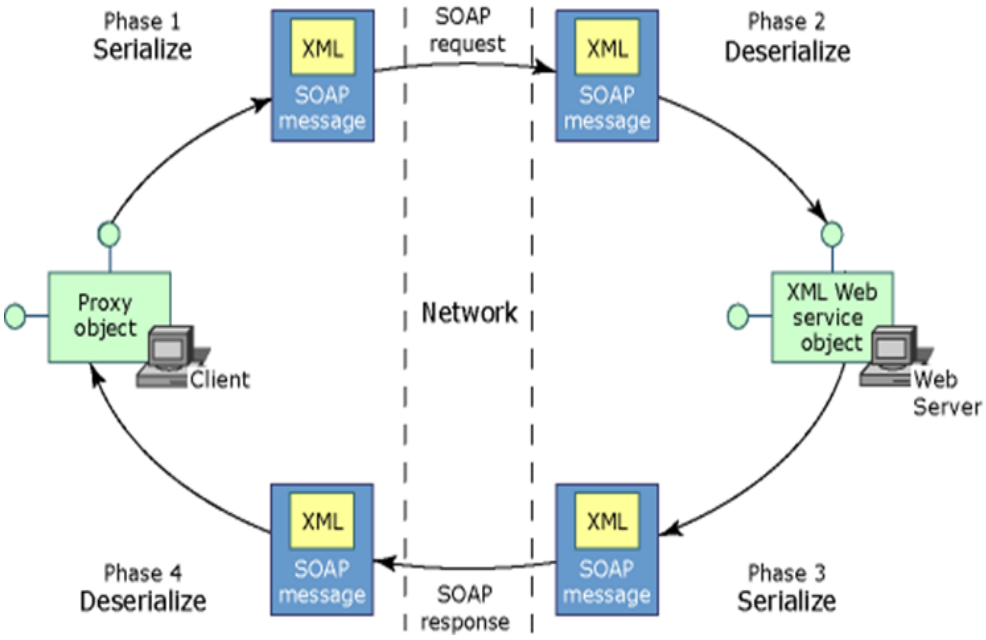
3.3.4. Android ile web servis kullanımı (SOAP & REST)

Mobil cihazlar üzerinde haberleşmek için gerekli teknoloji olarak web servisler ortaya çıkmıştır [29]. Web tabanlı uygulamaların çoğu başka sistemlerle veri alış verişinde bulunmaktadır ve HTTP Protokolü üzerinde çalışmaktadırlar. Yaygın bir şekilde kullanılan web hizmetleri:

- SOAP Tabanlı Web Hizmetleri (Google, Blogger vb.)
- REST Tabanlı Web Hizmetleri (Yahoo, Amazon, eBay vb.)

SOAP tabanlı web hizmetleri

Basit Nesne Erişim Protokolü kısaltması ile SOAP, dağıtık sistem olan internet üzerinden veri alış verişi için tasarlanmış RPC (Remote Procedure Call) modelini kullanan bir haberleşme protokolüdür. SOAP sisteminde mesajlaşma, XML formatında (Şekil 3.8) yapılmaktadır ve mesaj iletimi HTTP, SMTP [42] gibi uygulama protokolleri kullanılarak gönderilir. SOAP sisteminde mesajlaşma daima XML formatında olacağı için yazılan uygulamanın hangi dilde yazıldığının bir önemi yoktur ve bu özelliği ile platform bağımsız bir yapıdadır.



Şekil 3.8.Web servis yaşam döngüsünün analizi [42]

REST tabanlı web hizmetleri

REST teriminin açılımı Temsilî Durum Aktarımı olan dağıtık sistemdir. REST; kaynak, kaynak tanımlayıcı, temsil, temsili veriler, kaynak verileri ve kontrol verileri olarak 6 öge içerir. İlk öğe kaynaktır. İkinci öğe kaynak tanımlayıcısı URL ‘dur [29]. URL üzerinden bir kimlik bilgisine benzer mantıkla çalışan rest servisler istemci sunucu mimarisine dayalıdır.

HTTP protokolü bir istemci sunucu haberleşme protokolüdür. REST mimarisinde haberleşme HTTP üzerinden bu protokole ait metotlar (delete, get, post, put) kullanılarak yapılabilmektedir. REST, veri alış verisinde esnek bir yapıya sahiptir ve JSON, XML vb.yapılar kullanılarak haberleşme sağlanmaktadır. Mobil uygulama ile bir masaüstü web uygulaması gibi birbirinden bağımsız yapılar ortak bir arayüz kullanılarak haberleşebilmektedir.

Günümüz uygulamaların birçoğu web tabanlı ve iş yükü dağılımı kullanıcı tarafında daha fazladır. REST, SOAP web servisine göre haberleşmede WSDL gibi tanımlama diline gerek olmaksızın direk HTTP metotlarını kullandığı için daha hızlıdır.

Patel ve diğerlerinin istemci ve mobil kullanıcı arasında bilginin en hızlı ve en hafif şekilde değişiminin yolunu bulma amacıyla yaptığı araştırmalara göre SOAP web servisi kaynak kısıtlı mobil cihazlar üzerinde beklenmeyen olumsuz performanslar gösterdiği anlaşılmaktadır. SOAP tabanlı web servislerinde mesajların ayrıştırılması maliyetli ve işlemci gücü, bellek, pil ve bant genişliği gibi yüksek kaynaklara ihtiyaç duymaktadır. REST mimarisi URI (Tekdüzen Kaynak Konum Belirleyicisi) kullanan web servisleri arasında direk olarak haberleşme sağlayabilmektedir ve mesajların parçalanmasında uygulama performansını bozacak faktörlerden kaçınmaktadır. REST mimarisi SOAP ile karşılaştırıldığına daha esnektir ve mobil uygulamalarda daha iyi ve uygun olduğu için daha çok tercih edilmektedir [29].

Veri alış verişinde en sık kullanılan veri alış veriş metotları XML ve JSON’dır. Verini nasıl iletildiği çoğunlukla verinin iletim hızını ve performansını etkilemektedir.

XML

Ağaç veri yapısına sahip XML' in açılımı Genişletilebilir İşaretleme Dili olarak ifade edilir. XML, verilerin nasıl sunulacağını belirler ve veri içeriğini anlaşılır hale getirerek dokümanlar hazırlamaya yarayan bir işaretleme dilidir.

```
<?xml version="1.0" encoding="UTF-8" ?>
	<0>
		<DateTime>2017-02-11T16:00:00+03:00</DateTime>
		<EpochDateTime>1486818000</EpochDateTime>
		<WeatherIcon>4</WeatherIcon>
		<IconPhrase>Intermittent clouds</IconPhrase>
		<IsDaylight>true</IsDaylight>
		<Temperature>
			<Value>5.4</Value>
			<Unit>C</Unit>
			<UnitType>17</UnitType>
		</Temperature>
		<RealFeelTemperature>
			<Value>4.7</Value>
			<Unit>C</Unit>
			<UnitType>17</UnitType>
		</RealFeelTemperature>
		<WetBulbTemperature>
			<Value>2.2</Value>
```

Şekil 3.9. Hava durumu verileri XML doküman örneği

Şekil 3.9 'da XML kolay anlaşılır ve kullanımı rahat bir dildir. Web üzerinden direk kullanılabilir. Bir XML örneği gösterilmektedir.

JSON (JavaScript Object Notation)

Küçük boyutlarda veri alış verişlerinde kullanılabilir. JSON verilerinde Anahtar ve Değer ikilisi yer alır. String veri yapısında olan anahtar gösterilerek ilgili verilere erişilebilir. Değer veri türü olarak Integer, String, Boolean vb. olabilir. Verinin parçalanması ve kullanımı kolaydır. Şekil 3.10. da görüldüğü gibi bir meteoroloji sitesinden alınan veriler gösterilmiştir.

```
"DateTime":"2017-02-11T16:00:00+03:00",
"EpochDateTime":1486818000,
"WeatherIcon":4,
"IconPhrase":"Intermittent clouds",
"IsDaylight":true,
"Temperature":{
   "Value":5.4,
   "Unit":"C",
   "UnitType":17
},
"RealFeelTemperature":{
   "Value":4.7,
   "Unit":"C",
   "UnitType":17
},
"WetBulbTemperature":{
   "Value":2.2,
   "Unit":"C",
   "UnitType":17
},
"DewPoint":{
```


Şekil 3.10. Hava durumu verileri JSON örneği

Patel ve diğerlerine göre, JSON metodu XML metodundan daha hızlı çalışmaktadır. JSON aynı zamanda daha az kaynak tüketir, bunlara CPU kullanımı, bellek kullanımı, ayrıştırma çalışma zamanı ve toplam bant genişliği de dahildir.

JSON kullanılarak [29]:

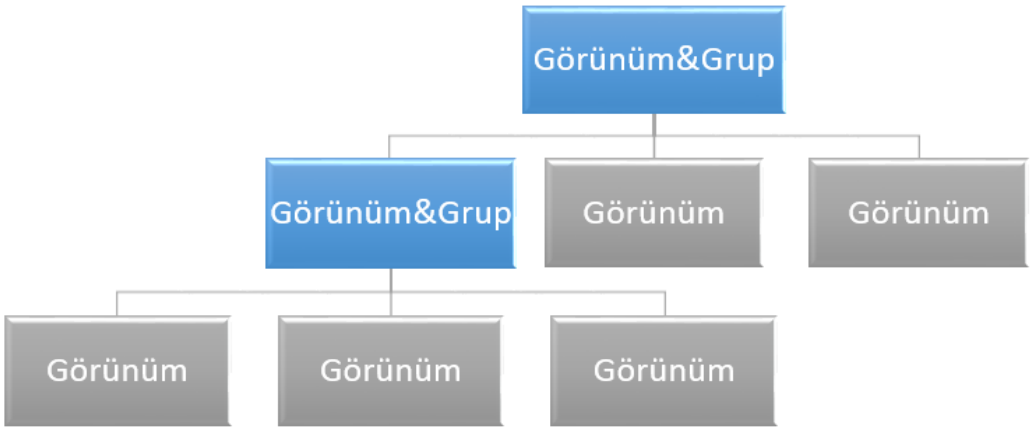
- Android platformda sınırlı veri depolama problemi çözülür.
- Android platform için veri tabanı istemcisini bağlamak daha kolaydır.
- Nesne yönelimli programlama mekanizması kullanarak, Android istemci üzerindeki işlemleri ve sınıflar arasında denetim üzerinden veri modeli tanımlanabilmektedir.

Web tabanlı ve mobil entegre uygulamalar geliştirmede REST kullanımının avantajı daha fazladır ve veri iletiminde JSON XML' e kıyasla daha hızlıdır, daha az yer kaplar ve daha anlaşılırdır. Bu yüzden, geliştirilen mobil uygulamada web üzerinden gerçek zamanlı olarak verilerin çekilmesi için REST tabanlı web hizmeti ile JSON metodu kullanılmıştır.

3.3.5. Android grafiksel kullanıcı arayüzü

Bir Android uygulamasında grafiksel kullanıcı arayüzü; Görünüm ve Görünüm Grup nesnelerinin birlikte kullanılması ile oluşturulmaktadır. Görünüm, kullanıcının etkileşime girebileceği bir şeyi ekrana çizer ve genellikle buton, metin alanı gibi kullanıcı arayüzü araçlarıdır. Görünüm Grup, arayüzün düzenini tanımlamak için diğer Görünüm ve Görünüm Grup öğelerini tutan görünmez nesnelerdir.

Her bileşen için kullanıcı arayüzü, Görünüm ve Görünüm Grup nesneleri hiyerarşik bir yapıda bir araya getirilerek tanımlanmaktadır (Şekil 3.11). Her bir Görünüm grubu, alt Görünümleri kontrol eden görünmez bir alandır. Bu alt görünümler (Child View) giriş kontrolleri veya kullanıcı arayüzünü bazı kısımlarını çizen diğer widget' lar olabilir. Bu hiyerarşi ağacı, yapmak istenilen uygulamaya göre basit veya karmaşık yapıda olabilir, fakat performans açısından değerlendirildiğinde bu gibi yapılarda sade tasarım en iyi ve hızlı sonuçları vermektedir.



Şekil 3.11. Kullanıcı arabirimi düzenini tanımlayan bir görünüm hiyerarşisinin örneği

Bir uygulamanın kullanıcı arayüzü tasarımı için Android Görünüm ve Görünüm Grup alt nesneleri, giriş kontrolleri (butonlar, metin alanları vb.) ve düzen (doğrusal düzen, göreceli düzen vb.) modelleri sağlamaktadırlar.

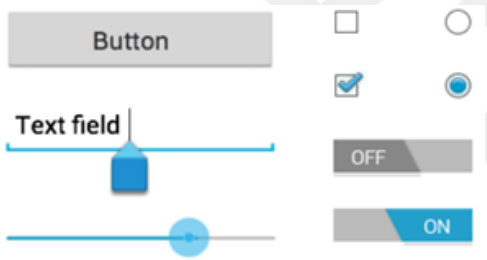
Bazen bir uygulamanın içerisinde bir satın alma yapılması veya reklama tıklanması gibi kullanıcının tam olarak bilgilendirilmesinin ve izninin gerektiği olaylar gelişebilir. Kötü niyetli bir uygulama, amacını gizleyerek bu tür durumların kullanıcının bilgisi dahili dışında

gerçekleşmesine neden olabilmektedir. Android çözüm olarak, Arayüz hassas işlevlere erişmeyi sağlayan görünümlerin güvenliğini artırmak için kullanılabilen bir dokunma filtreleme mekanizması sunmaktadır [43].

Giriş kontrolleri

Uygulamanın kullanıcı arayüzündeki etkileşimli bileşenlerdir. Android arayüzünde, butonlar, metin alanları, arama çubukları (text fields), onay kutuları (checkboxes), yakınlaştırma butonları, geçiş butonları (toggle) gibi birçok çeşitli kontrol yapıları kullanım için sağlanmaktadır.

Şekil 3.12'de giriş kontrollerinden bazıları gösterilmiştir. Uygulama geliştirme sırasında palet altında yer alan Widget'lar sürükle bırak yöntemi ile kullanılabilmektedir.



Şekil 3.12. Giriş kontrol ögeleri

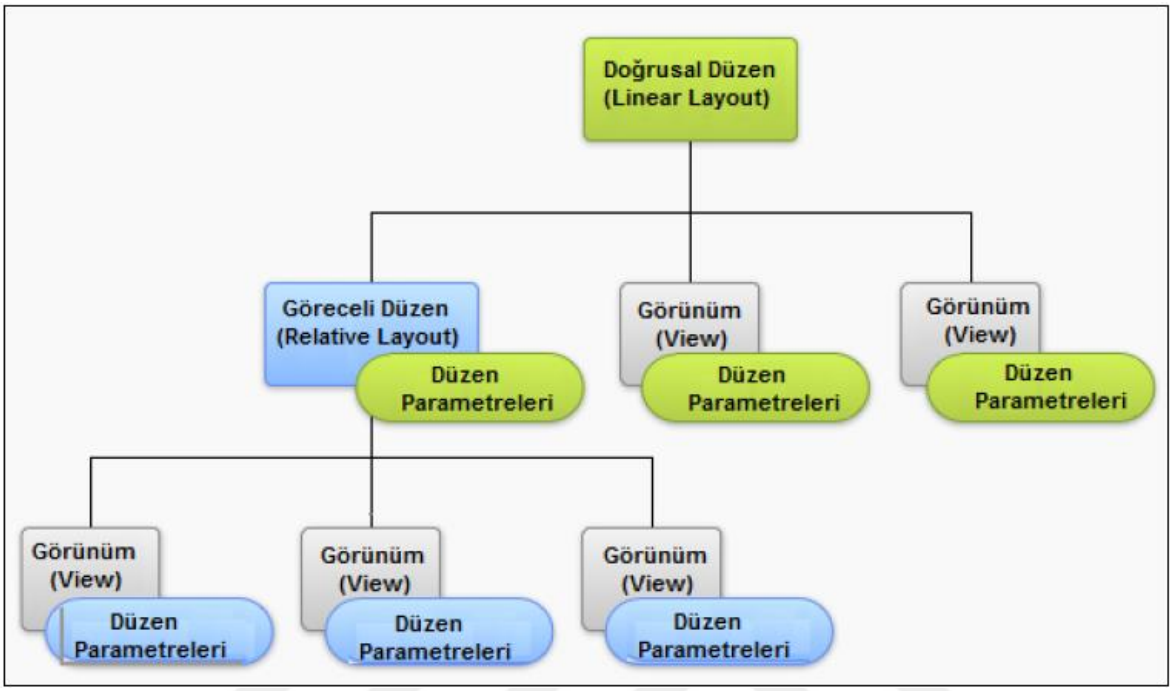
Düzen

Düzen, uygulamaların kullanıcı arayüzünü oluşturan elemanları barındıran ve bir yapı içerisinde görüntülenmesini sağlayan en alt katmandır. Aktivite veya uygulama gibi kullanıcı arayüzü için görsel yapısını tanımlar.

Görünüm objeleri kod içerisinde tanımlanarak bir ağaç oluşturulabilir, ancak Düzen tanımlamanın en kolay ve en etkili yolu bir XML dosyası kullanılmasıdır. XML, HTML 'ye benzer şekilde kullanıcı tarafından okunabilir bir yapı sunmaktadır.

Düzen parametrelerinde iç içe geçmiş bir sınıf uygulanmaktadır. Bir alt sınıf, Düzen grubu için uygun olan her alt Düzen için boyutu ve konumu tanımlayan özellik türlerini

içermektedir. Şekil 3.13'de görüldüğü gibi, üst Düzen grubu her alt Düzen için (alt görünüm grubu da dâhil olmak üzere) düzen parametrelerini tanımlar.



Şekil 3.13. Düzen parametreleri ile birlikte gösterilen görünüm hiyerarşisi [44]

Düzen oluşturmak için kullanılan yöntemler:

- Kullanıcı arayüzü elemanlarının XML olarak tanımlanması
- Kullanıcı arayüzü elemanlarının çalışma zamanında oluşturulması

Kullanıcı arayüzü elemanlarının XML olarak tanımlanması

Android düzen sınıflarına ve alt sınıflara açık ve anlaşılır çok geniş bir XML yapısı sunmaktadır. Her biri farklı xml etiketi ve özellikleri ile tanımlanan çok çeşitli arayüz elemanı ve bileşeni, Android uygulamalarının görsel olarak geliştirilebilmesine ve kullanıcıya yönelik özelliklerinin artırılabilmesine olanak sağlamaktadır

Bir XML öğesinin adı, temsil ettiği Android sınıfıyla ilgilidir. Yani bir <TextView> elemanı kullanıcı arayüzünde bir metin görünüm widget'ı oluşturur ve bir <LinearLayout> elemanı bir doğrusal düzen görünüm grubu oluşturur. Şekil 3.14'de mobil Android projelerinde oluşturulan düzen (layout) XML yapısı gösterilmektedir.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:layout_width="match_parent"
              android:layout_height="match_parent"
              android:orientation="vertical" >
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

Şekil 3.14. Bir XML kullanıcı arayüz kodu örneği [45]

Kullanıcı arayüzü elemanlarının çalışma zamanında oluşturulması

Uygulamalarda View ve ViewGroup nesneleri kolaylıkla kod tarafından oluşturulup, özellikleri yönetebilmektedir.

Kullanıcı Arayüzüne bir Düzen kaynağı yüklenildiği zaman, Android Düzen' in her düğümünü, ek davranışları tanımlamak, nesne durumunu sorgulamak veya Düzeni değiştirmek için kullanılabilecek bir çalışma zamanı nesnesine başlatır [46].

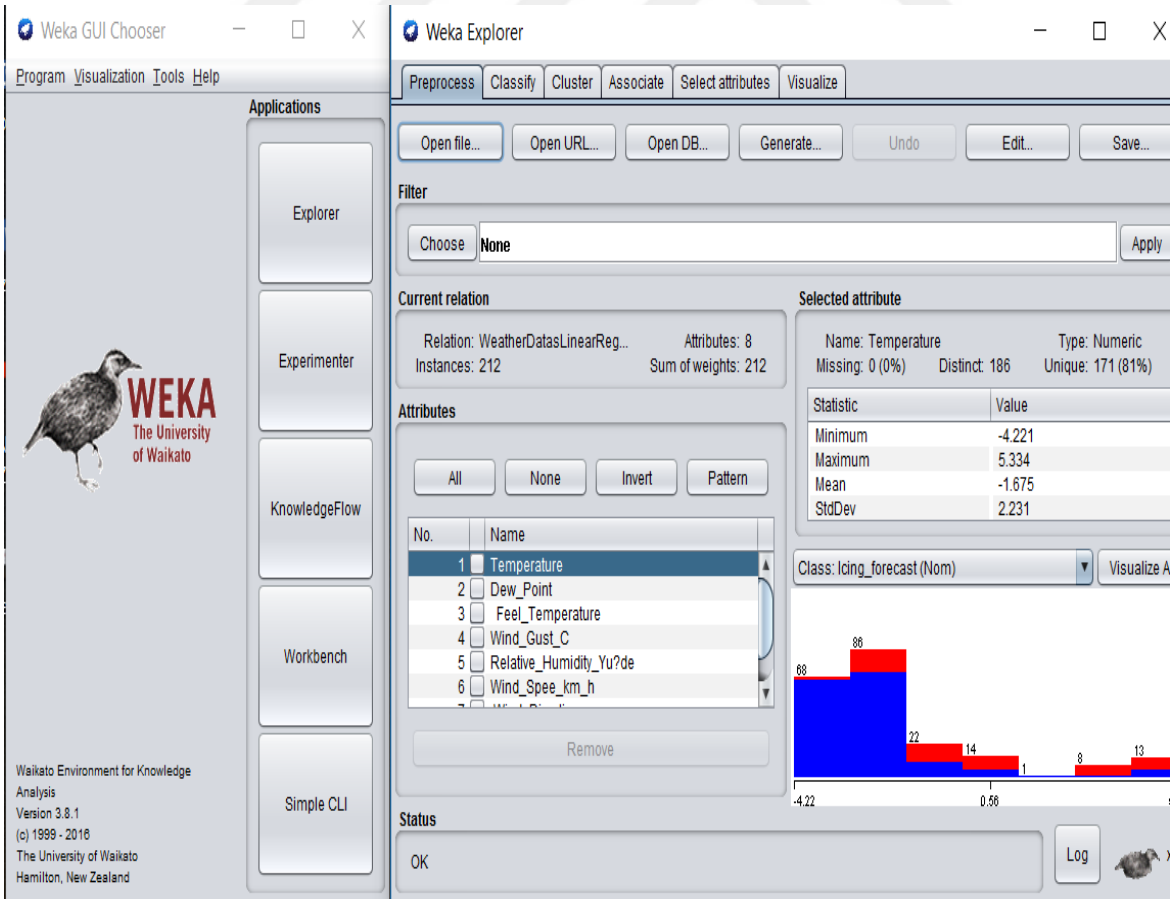
Mobil uygulama için kullanıcı arayüzü oluşturmada ve yönetmede bu yöntemlerden herhangi birini veya her ikisini birden kullanmak Android çerçevesine esneklik kazandırmıştır. Bir uygulamanın varsayılan düzenlerini, içinde görünecek ekran öğeleri ve özellikleri de dâhil olmak üzere XML olarak bildirebilir ve ardından çalışma zamanında XML 'de bildirilenler de dâhil olmak üzere ekran nesnelerinin durumunu değiştirecek kod eklenebilmektedir.

Mobil uygulamada arayüz geliştirmede kullanılan pek çok tasarım bileşeni bulunmaktadır. Geliştirilen arayüzün basit ve sade tutulması, uygulamanın her zaman kullanışlı ve anlaşılır olmasını sağlamaktadır.

3.4. WEKA Yazılımı

WEKA Yeni Zelanda'nın Waikato Üniversitesinde geliştirilmiş, Java'da uygulanan makine öğrenme algoritmalarını ve metotlarını içeren bir yazılımdır. Modüler bir tasarımda olup veri madenciliği ve veri analizi gibi birçok alanda kullanılmaktadır. Sınıflandırma, kümeleme ve ilişkilendirme gibi veri madenciliği işlemleri WEKA' da yapılabilmektedir [47].

WEKA sınıflandırma ve regresyon sayısal tahmini için kullanılan karar ağaçları, destek vektör makineleri, örnek tabanlı sınıflandırıcılar, Bayes karar şemaları, sinir ağları gibi çok sayıda algoritmayı içerisinde barındıran bir veri madenciliği çalışma sahasıdır. Aynı zamanda bu birçok farklı makine öğrenme algoritmasını karşılaştırmaya olanak tanımaktadır. Bu yazılım aracında sınıflandırma sonuçları bir karışıklık matrisi oluşturularak özetlenmektedir [48-49]. Şekil 3.15'da görüldüğü gibi WEKA anlaşılır arayüzü ile veri filtreleme, sınıflandırma, kümeleme ve ilişkilendirme gibi işlemler kolaylıkla uygulanabilmektedir.



Şekil 3.15. WEKA arayüzü

WEKA yazılımında bulunan öğrenme algoritmaları herhangi bir veri setine kolayca uygulanabilmektedir. Ayrıklaştırma ve örnekleme algoritmaları gibi veri kümelerini dönüştürmek için çeşitli araçlar da içermektedir. Bu yazılım aracı, hiç bir program kodu yazmadan bir veri kümesini ön işleme koyabilir, bir öğrenme şemasına besleyebilir ve ortaya çıkan sınıflandırıcıyı ve performansını analiz edebilir.

Regresyon, sınıflandırma, kümeleme, ilişki kuralı incelemesi ve öznitelik seçimi gibi 4 ana veri madenciliği problemi için yöntemler içermektedir. Aynı zamanda birçok veri görselleştirme olanağı ve veri ön işleme aracı sağlanmaktadır. Tüm algoritmalar, girdilerini, bir dosyadan okunabilen veya bir veritabanı sorgusu tarafından üretilebilen tek bir ilişkisel tablo şeklinde almaktadırlar.

WEKA'nın kullanılma amaçları şu şekilde özetlenebilir:

- Bir veri kümesine bir öğrenme yöntemi uygulamak ve veriyle ilgili daha fazla bilgi edinmek için çıktısını analiz etmek
- Yeni örneklerde öngörüler üretmek için öğrenilen modelleri kullanmak
- Birkaç farklı öğreniciyi uygulamak ve tahmin için birini seçmek için performanslarını karşılaştırmak

Etkileşimli WEKA arayüzünde bir menüden herhangi bir öğrenme yöntemi seçilebilmektedir. Bu yöntemlerin çoğunda, özellik sayfası veya nesne düzenleyicisi aracılığıyla erişilebilen ayarlanabilir parametreler bulunmaktadır [50].

3.5. Verilerin Temini

Çalışmada kullanılan veriler Ankara ili Çorum caddesine kurulu olan buzlanma önleyici otomatik püskürtme sisteminin merkezi endüstriyel bilgisayarından web üzerinden gerçek zamanlı olarak alınmaktadır (Resim 3.3).

Veriler yol durum sensörleri ve hava istasyonları kullanılarak elde edilmektedir. Elde edilen veriler işlenerek eğitim veri kümesi olarak kullanılmaktadır.

Resim 3.3. Buzlanma önleyici otomatik püskürtme sistemi

Eğitim veri kümesi 7 (sıcaklık, çiğ noktası, hissedilen sıcaklık, rüzgar şiddeti, rüzgar yönü, bağıl nem ve rüzgar hızı) tane özniteliğe sahip olup, iki sınıfa ayrılır. Sınıflardan biri buzlanmanın olduğunu ifade ederken, diğer sınıf buzlanma olmayan durumları ifade etmektedir. Çalışma içerisinde sınıflar “Buzlanma Var” ve “Buzlanma Yok” şeklinde ifade edilmiştir (Çizelge 3.1).

Çizelge 3.1. Hava durumu eğitim veri seti

Sınıflar :	Buzlanma Var
	Buzlanma Yok
Nitelikler :	Sıcaklık
	Çiğ Noktası
	Hissedilen Sıcaklık
	Rüzgâr Şiddeti
	Bağıl Nem
	Rüzgâr Hızı
	Rüzgâr Yönü

“Buzlanma Var” sınıfına ait toplam örnek sayısı 53 tanedir. “Buzlanma Yok” sınıfına ait toplam örnek sayısı 4797 tanedir. Yaklaşık 5 bine yakın ham eğitim verisi toplanılmıştır. Elde edilen bu veriler dikkate alındığında, bu veri seti “Buzlanma Yok” sınıfındaki örnek sayısı “Buzlanma Var” sınıfından gelen örnek sayısından çok fazla olduğu için dengesiz bir veri setidir. Bir sınıfa ait çok sayıda örnek aynı zamanda yüksek boyutluluk problemini de beraberinde getirmektedir.

Veri madenciliği yöntemleri uygulanmadan önce veri setinin dengeli hale getirilmesi olması uygulanan yöntemlerin doğruluğunun artmasını sağlamaktadır. Bu yüzden, veri keşfetme sırasında veri karmaşıklığının daha aza indirgenmesi amacıyla özellik seçimi, özellik çıkarımı, regresyon analizi gibi aşamalardan geçirilmektedir.

Buzlanma tahmininde eğitim veri kümesi olarak kullanılmak üzere temiz ve dengeli bir veri kümesi oluşturmak için karmaşık ve temiz olmayan veriler ön işlemlerden geçirilerek filtrelenmektedir.

3.6. Verilerin İşlenmesi

Bir sistemde kullanılmak üzere toplanılan verilerde yagın olarak karşılaşılan problemler vardır. Bu problemler:

Yüksek boyutluluk

Veri kümesinde çok sayıda nitelik olması durumudur. Çok sayıda nitelik ve örnek sayısı, tahmin algoritmalarının kısa sürede doğru sonuçlara ulaşmasını engelliyor olabilir.

Dengesiz veriler

Bir sınıftaki örnek sayısı diğer sınıftan gelen örnek sayısından çok fazla olduğunda tahmin algoritmaları, daha çok sayıda örnek içeren sınıflardan öğrenmeye odaklanma eğilimindedir [51].

İki sınıfa ait verilerden bir sınıfın örnek sayısı diğer sınıftan çok fazla ise, tahminde örnek sayısı fazla olan sınıfa doğru sapma gözlemlenir. Bu yüzden, performans değerlendirmesi sağlıklı yapılamaz. Verilerin değerlendirmeye uygun dengeli hale getirilmesini sağlayan pek çok yöntem bulunmaktadır.

Dengesiz veri setini dengeli hale getirmek için;

- Daha fazla veri toplama
- Performans metrikleri değiştirilmesi
- Tekrar Örneklendirme
- Sentetik Veri Üretme
- Farklı Algoritmalar denemek (Doğrusal Regresyon, temel bileşenler analizi vb.)
- Cezalı Modeller
- Farklı yönlerden yeniden değerlendirme (Anormallik tespiti, Outlier algılama vb.)

işlemleri yapılabilmektedir. Yüksek boyutluluk probleminden kurtulmak için veri setinin işlenmesinde yaygın olarak kullanılan yöntemler veri kümesinin yeniden örneklendirilmesi, eksik\kayıp verilerin temizlenmesi ve tamamlanması işlemleridir.

Tekrar örneklendirme

Verilerin tekrar örneklendirilmesi işlemine denir. Veri setinden örnek alınırken iki sınıftan da dengeli bir şekilde örnek alınması gerekmektedir. Örnek verilecek olursa, nadir bulunan herhangi bir bitki türünün sınıflandırıldığını düşünelim. Nadir bitki türüne ait mevcut 300

ve üzerinde veri olsun. Sınıflandırılan diğer bitki türünden ise mevcut 10000 ve üzerinde veri bulunsun. Bu tarz bir veri setinde diğer bitki türü örnek sayısını nadir bitki türü örnek sayısına yakın tutacak şekilde yeniden örneklendirilmesi iki tür arasında doğru bir değerlendirme yapabilmek için gerekmektedir. Bu kapsamlar söz konu olduğunda, WEKA yazılım aracında veri setini önişlemden geçirmek amacıyla kullanılan bazı yeniden örneklendirme ve sentetik veri üretme yöntemleri kullanılmaktadır. Bunlar:

Alt örnek örneği (SpreadSubsample)**:** Veri setinden dengeli bir şekilde örnek alınmasını sağlar. Bu ön işlem altında maksimum örnek alma sayısı ve veri dağılım oranı seçilebilmektedir.

SMOTE: Başarılı yeni örnekler üretilebilmesi için eksik verilere yeni değer ataması en yakın komşulara bakılarak yapılmaktadır

Veri tamamlama bir ya da çoklu doldurma metotları ile alt bölümlere ayrılabilen birkaç farklı algoritma kullanarak uygulanmaktadır. Tekli veri doldurma metotlarında kayıp değer tek bir değer ile doldurulmaktadır. Çoklu veri doldurma metotlarında ise kayıp değeri doldurmada olasılık hesapları ile değerler hesaplanır ve "en iyi" değer seçilir [52]. Belirli durumlarda veriden bir değişken eksikse, eğer mümkünse, bunu sezgisel verilerle doldurmak çok önemlidir. Bu değişken için uygun bir verinin makul bir tahminini eklemek boş bırakmaktan daha iyidir. Verilerdeki bu boşlukları doldurma işlemine veri doldurulması denir. Aynı zamanda çok fazla örnek içeren kümede eksik değere sahip örnekler tamamlanmak yerine silinmesi daha makuldur.

Liste-boyunca (ya da durum-boyunca) silme

Liste boyunca silme yöntemi analizden öngörülen tüm kayıtların silinmesi işlemidir. Bu teknik genellikle birçok istatistik ve otomatik öğrenme algoritmaları tarafından kullanılan varsayılan metottur. Bu tekniğin birkaç avantajı vardır;

- Herhangi bir veri madenciliği analizinde kullanılabilir
- Başarmak için herhangi özel bir istatistiksel metoda ihtiyaç duymaz
- Değişkenlerin tamamen bağımsız olduğu veriler için iyidir

- Doğrusal regresyon, Lojistik ve Poisson regresyonu gibi yöntemler ile beraber kullanılabilmektedir.

İkili silme

İkili silme işlemi, bir değişkenin değerleri ile bu değişkenin kovaryansının hesaplanmasında kullanılmasını sağlamaktadır. Bu yaklaşımın avantajı bir doğrusal regresyonun sadece örnek ortalaması ve kovaryans matrisi kullanılarak tahmin elde edilmesidir.

Uygun bir değer atfetme

Kayıp olmayan durumların ortalaması ile kayıp değerlerin atfedilmesinde sık sık ortalama ikamesi olan uygun bir değer verilir. Eğer kayıp değerin özel bir değerini uygulayan bir karar kuralını güvenli bir şekilde uygulayabilirseniz, o zaman ortalama ikamesinden bile doğru bir değere yaklaşmış olursunuz [53].

Bu çalışmada "Buzlanma yok" sınıfına ait eksik verilere sahip olan örnek satırları liste boyunca silme yöntemi kullanılarak temizlenmiştir ve "Buzlanma var" sınıfına ait örnek satırlarında bulunan eksik veriler Doğrusal Regresyon yöntemi kullanılarak tamamlanmıştır.

3.6.1. Doğrusal Regresyon analizi ile kayıp verilerin tamamlanması

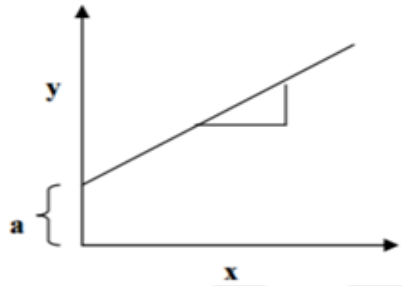
İki ya da daha çok değişken arasında ilişkiyi ölçmek için, regresyon analizi yöntemi kullanılmaktadır. Regresyon analizi ile değişkenler arasındaki ilişkinin varlığı ve eğer ilişki var ise bunun gücü hakkında bilgi edinilebilir [54]. Literatürde regresyon analizini kullanan pek çok araştırma vardır [55-58]. Bu çalışmada "buzlanma var" örnek satırlarında bulunan eksik veriler, regresyon analizi ile istasyondan alınan hava değişkenlerinin birbirleriyle ilişkileri gözlemlenerek tamamlanmıştır.

Doğrusal regresyonda amaç, matematiksel ve istatistiksel olarak modellenmesi zor olan gerçek karmaşık verilerin matematiksel bir fonksiyon kullanarak daha ilkel bir modellemesini yapmaktır.

Eş. 3.1'de regresyon denklemi genel ifadesi verilmektedir.

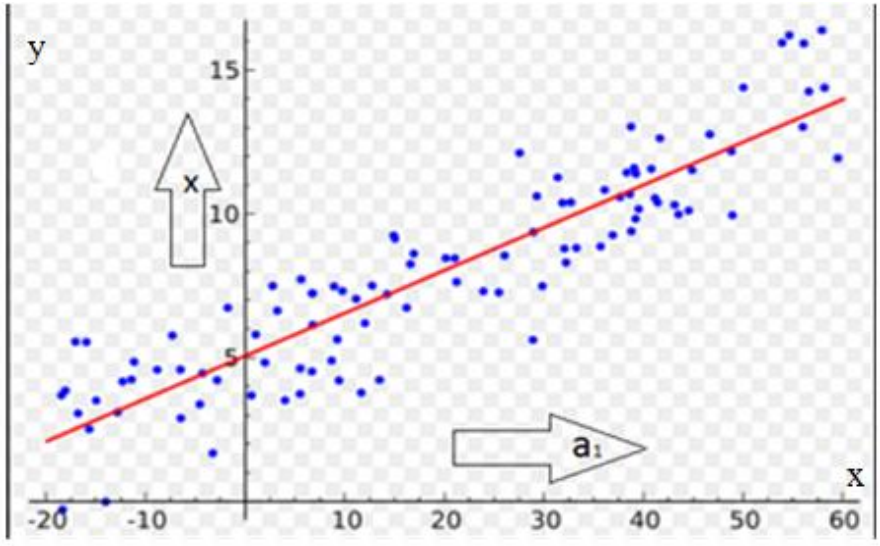
$$Y` = a + bX \quad (3.1)$$

Şekil 3.16' de verilen regresyon denklem grafiğinde X ifadesi seçilen bağımsız değişkenin değerini, Y` ifadesi seçilmiş X değerine için tahmin edilen Y değerini, a ifadesi doğrunun Y ekseni kestiği noktanın değerini, b ifadesi doğrunun eğimini ve a, b ifadeleri regresyonun katsayılarını ifade etmektedir.



Şekil 3.16. Regresyon denklem grafiği [59]`

Formülde yer alan X seçilen bağımsız değişkenin değeri, Y seçilmiş X değeri için tahmin edilen Y değeri, a doğrunun Y eksenini kestiği noktanın değeri, b doğrunun eğimi, a ve b ise regresyon katsayılarıdır. Basit doğrusal regresyon analizinde bir bağımlı ve bir bağımsız değişken vardır [59].



Şekil 3.17. Tek bağımsız değişkene sahip basit doğrusal regresyon örneği

Şekil 3.17' de doğrusal regresyon örneğinde görüldüğü gibi, grafik üzerinde yerleştirilen tüm verilere en yakın geçen doğru çizilmiştir. Regresyon modeli eksik verilerin tamamlanması, kestirim veya tahmin gibi veri madenciliği problemlerinde kullanılabilmektedir. Eş. 3.2'de formül verilmektedir.

$$X = W_0 + W_1 a_1 + W_2 a_2 + \cdots + W_K a_K \tag{3.2}$$

a_1 ... a_K = Özellikler ve W_0 ... W_K = Ağırlıkları ifade etmektedir. Eğitim verisinde ağırlıklar hesaplanır. Eş. 3.3' de ağırlıklar hesaplandıktan sonra, ilk eğitim verisi $a^{(1)}$ değeri için hesaplanan tahmin değeri ifade edilmektedir.

$$W_0 a_0^{(1)} + W_1 a_1^{(1)} + W_2 a_2^{(1)} + \cdots + W_K a_K^{(1)} = \sum_{j=0}^{K} W_j a_j^{(1)} \tag{3.3}$$

Burada ağırlıklar gerekli şekilde ayarlandığı zaman hata oranı en aza indirilebilmektedir. K tane özellik formüle edilecek ise, en az K+1 tane örnek gerekmektedir. Her daim özellik sayısından daha fazla örnek sayısı gereklidir.

Eğitim verisinde kareli hatayı en aza indirmek için ağırlıkları seçilir. Eş. 3.4' de hata miktarını minimize eden formül;

$$\sum_{i=0}^{n} (x^{(i)} - \sum_{j=0}^{k} w_j a_j^{(i)})^2 \tag{3.4}$$

x_i, gerçek değerdir. $w_j a_j^i$, ise, i. özellik ile özeliğin ağırlığının çarpımından elde edilen tahmin değeridir. Her bir nitelik için bu ikisi birbirinden çıkarılıp kareleri alınır. Örneğin i=1 için x_1 ifadesi ilk özellik değeridir. $w_j a_j^1$, ifadesinde eğitim verisinde hata minimize edilmesi için ağırlıklar seçilir. Bu ifade tahmin edilen değerdir. Gerçek değer ile tahmin değer toplamı arasındaki farkın karesi alınarak hesaplanır. Elde edilen sonuçta toplam hata en aza indirgenmiş olur.

3.7. Veri Madenciliği

Veri Madenciliğini, büyük ve karmaşık yapıdaki veri kümelerinin belirli yöntemler ve algoritmalar kullanılarak veriyi anlamlandırma ve gelecek ile ilgili tahminde bulunma süreci olarak tanımlanmaktadır. Veri madenciliği ve veri bilimi uzaktan algılama, Biyoinformatik, Simülasyon gibi bilim dünyasında birçok alanda kullanılmaktadır. Veriyi iki sınıfta incelemek mümkündür;

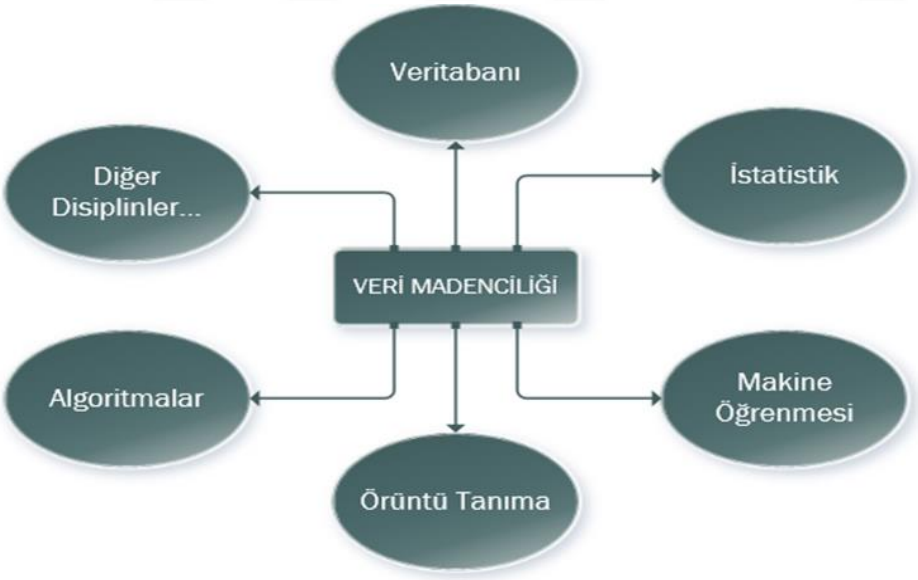
Enformasyonel veri: Bu veri kişiye yönelik, bütünleşik, zaman içerisinde oluşabilen, birleştirilmiş verilerden oluşur.

Operasyonel veri: Bu veri daha çok uygulamaya yönelik, dağınık, kısa zamanda oluşan, tekrarlayabilen verilerden oluşmaktadır.

Bu çalışmada uygulama için hazırlanmış gerçek-zamanlı elde edilen Operasyonel veri kullanılmıştır. Saklanan verinin boyutu arttıkça, bu veriler içerisinde anlamlı bilgilerin değerlendirilmesi o kadar önem kazanmıştır. Veri madenciliği sürecinde izlenen yol;

- Problemin tanımlanması
- Veri kaynaklarından alınan verilerin hazırlanması
- Veri ambarları ile modelin kurulması
- Modelin izlenmesi
- Testler ve raporlar ile tanımlanan probleme göre modelin değerlendirilmesi

Veriden bilgi keşfi sürecinde, veri tabanlarından alınan veri temizlenir, seçimi yapılır ve veri madenciliği süreci başlatılır.



Şekil 3.18. Veri Madenciliğinin beraber çalıştığı disiplinler

Veri madenciliği veri tabanları, istatistik, makine öğrenmesi, örüntü tanıma, algoritmalar (Şekil 3.18) gibi birçok disiplinle beraber çalışır. Veri Madenciliği ile keşfedilen bilgi, çevrimiçi veri takibi, iş yönetimi, ürün kontrol sistemleri, pazarlama, tıp, finans, kimya, coğrafi bilgi sistemleri (CBS), mühendislik, görüntü tanıma ve robot görüş sistemleri, uzay

bilimleri, sosyal bilimler, davranış bilimleri, meteoroloji gibi alanlarda yaygın bir şekilde kullanılmaya başlanmıştır.

Veri Madenciliği Bilgi Keşfi ilk defa Piatetsky-Shapiro tarafından 1989 yılında kullanılmıştır. Veri madenciliği yöntemleri zamanla gelişmeye devam etmiştir. Agrawal, nicel birliktelik kurallarının madenciliği için hızlı bir algoritma olan Apriori'yi önermişlerdir. Fayyad, veri tabanlarında bilgi keşfinin süreçlerine ve veri madenciliğinin bu süreçteki yerine yönelik bir akış sunmuşlardır.

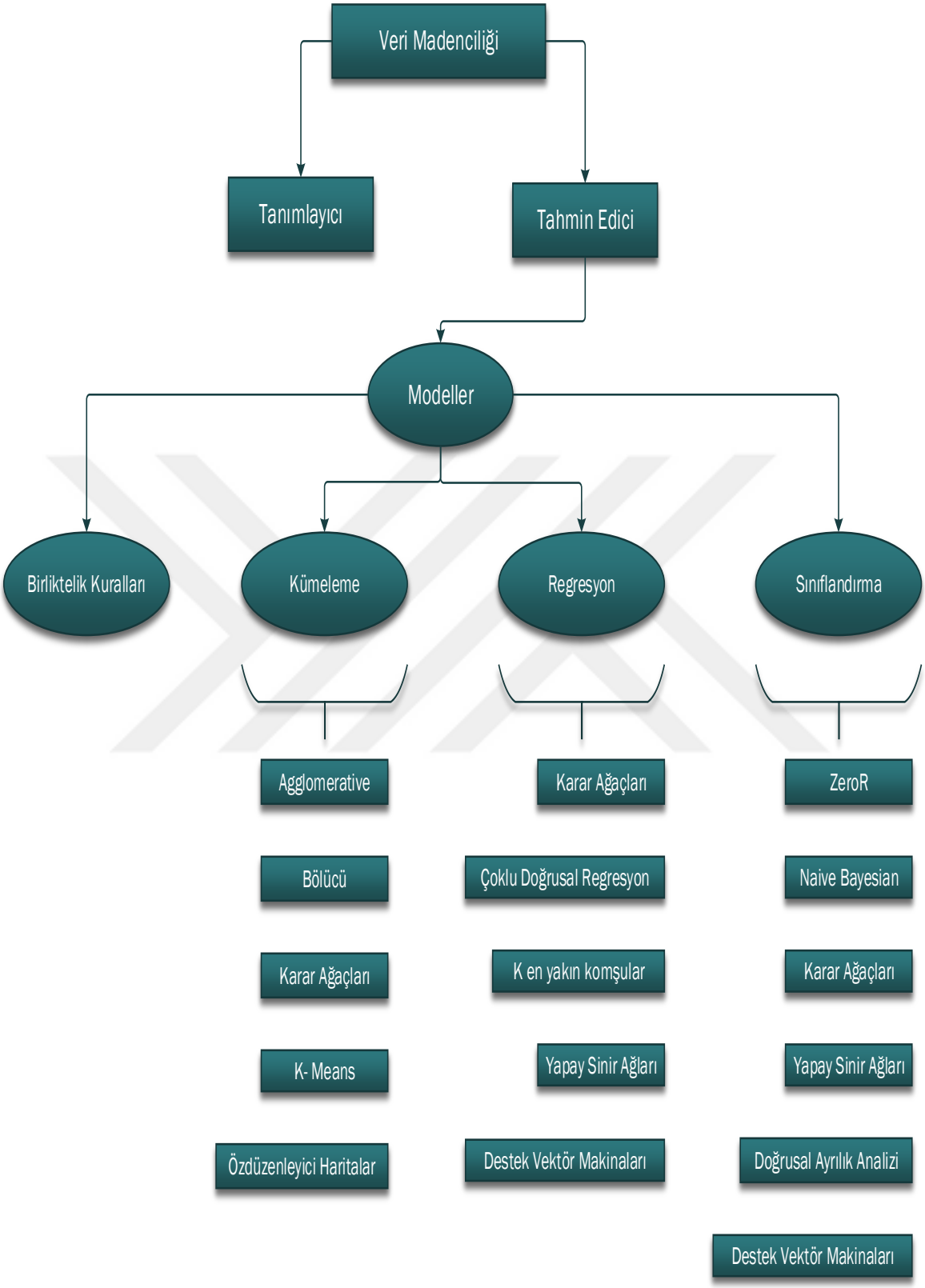
Veri madenciliğinin uygulama yaygınlığı kazanmasıyla birlikte uygulamaya yönelik çalışmalar ağırlık kazanmaya başlamıştır. Genellikle bilişim ve istatistik olmak üzere iki bakış altında incelenmektedir [60]. Veri madenciliği genellikle sırasıyla; problem ve verinin tanımlanması, veri hazırlama, veri keşfi, modelleme, değerlendirme ve uygulama aşamalarından geçmektedir.

Veri madenciliği yöntemleri, veriye ve verinin kullanılacağı alana göre değişmektedir. İki ana alanda ayrıştırıldığında;
Tanımlayıcı Model: Bu modelde eldeki veriler değerlendirilmesi, geçmiş verilerin açıklanması ve analizi, veri ilişkilerini incelemeyi ve modelleri seçmeyi amaçlamaktadır.

Tahmin Edici Model: Mevcut verilerin analizi ile gelecek durumlar hakkında analiz ve yorumlanması, elde olmayan verinin tahmini yapılmaktadır.

Bu çalışmada buzlanmanın tahmini ve analizi için ÇKA ve DVM veri madenciliği yöntemleri kullanılmaktadır.

Şekil 3.19 da görüldüğü gibi, veri madenciliği modelleri sıralamasına göre tahmin edici modeller sınıflandırma, regresyon, kümeleme ve birliktelik kuralları olmak üzere 4 kısımda incelenebilmektedir. Bu modellerin altında kendi alanlarını temsil eden algoritmalar yer almaktadır.



Şekil 3.19. Veri madenciliği modelleri sıralaması

3.7.1. Destek Vektör Makinaları

Destek Vektör Makinaları (DVM), Vapnik tarafından 1995 yılında iki sınıfı birbirinden ayırma problemine kısmen çözüm getiren yeni bir öğrenme yaklaşımı olarak doğmuştur. DVM, güçlü bir veri düzenleme özelliğine sahiptir ve yüksek boyutlu özellik alanlarını kolayca işleyebilmektedir. Aynı zamanda, genelleme kabiliyetlerini arttırarak en düşük hatayı sağlayabilecek kenar boşluklarını en üst düzeye çıkararak en uygun hiperdüzlem bulma işleminde Yapısal Risk Azaltma (YRA) ilkesine dayanmaktadır [61,62].

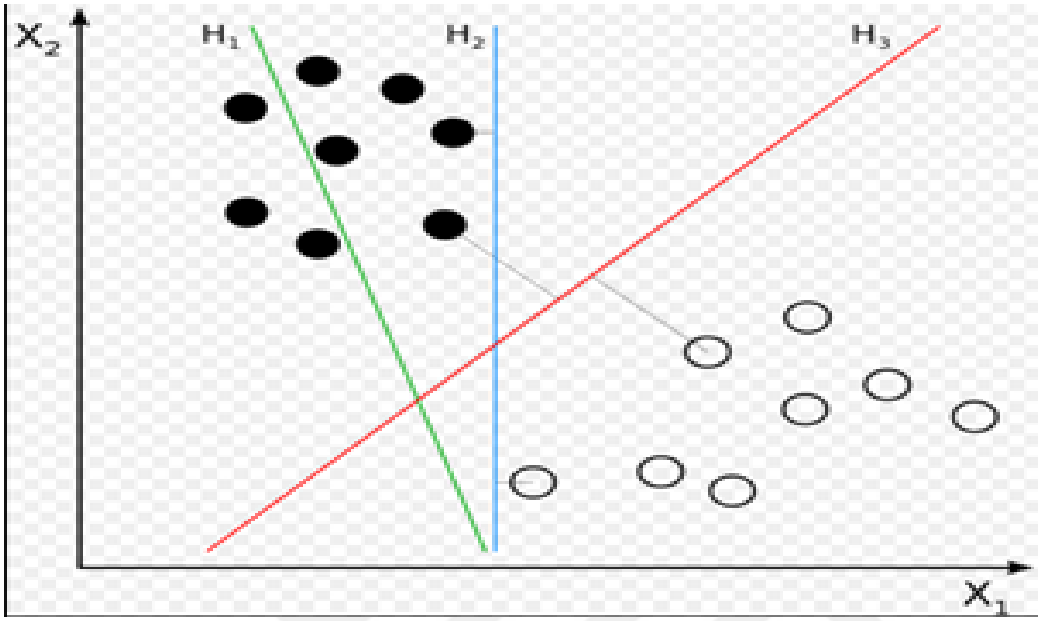
Makine öğrenmesinde DVM, sınıflandırma ve regresyon için verileri analiz etmede kullanılan öğrenme algoritmaları ile ilişkili bir denetimli öğrenme modelidir. Her biri, iki kategoriden birine ya da diğerine ait olacak şekilde belirtilmiş bir dizi eğitim örneği ele alındığında, DVM eğitim algoritması, verilen iki kategoriden birine ya da diğerine olacak şekilde örnekler atayan bir model oluşturur ve bu onu olasılıksal olmayan ikili doğrusal sınıflandırıcı yapar.

DVM modeli, örneklerin uzayda noktalar gibi temsil edilmesi, haritalandırılmasıdır. Bu yüzden, farklı kategorilerdeki örnekler mümkün olduğunca geniş bir şekilde temiz bir aralık, çizgi ile ayrılmalıdır. Daha sonra, yani örnekler aynı uzayda haritalandırılır ve aralığın (çizginin) hangi tarafına düşüyorsa o kategoriye ait olduğu tahmini yapılır. Doğrusal sınıflandırma gerçekleştirmesinin yanı sıra, DVM doğrusal olmayan sınıflandırma üzerinde de etkilidirler. “Kernel trick” adı verilen bir sistem kullanılarak, dolaylı olarak yüksek boyutlu özellik uzayında girişler haritalandırılır.

Sınıflandırmada veriler için etiketli ve etiketli olmayan olarak iki durum söz konusudur. DVM algoritmasında veri etiketli değilse, denetimli öğrenme yapılamaz. Bu gibi durumlarda denetimsiz öğrenme yaklaşımı gereklidir ve bu da verilerin gruplara doğal olarak kümelenmesi, daha sonra bu gruplara yeni veriler eşleştirilmesi girişimidir.

En kapsamlı tanım olarak, DVM sınıflandırma, regresyon veya diğer görevler için kullanılabilen yüksek veya sonsuz boyutlu bir uzayda bir hiperdüzlem veya sonsuz boyutlu uzayda bir dizi hiperdüzlemler (Şekil 3.20) oluşturur. Sezgisel olarak, herhangi bir sınıfın en yakın eğitim veri noktasına (işlevi payı) en uzak mesafedeki hiperdüzlem ile iyi bir ayrım

elde edilir, çünkü genel olarak payın daha büyük olması sınıflandırıcının genelleme hatasını düşürür [63].



Şekil 3.20. DVM algoritmasında bir dizi hiperdüzlemler ile ayrım yapılma örneği

Problemin kendisi sonlu boyutsal uzayda belirtilebiliyorken, bu uzayda kümeler sıklıkla doğrusal olarak ayırt edilemeyebilir. Bu sebeple, esas sonlu boyutsal uzayın çok daha büyük boyutsal bir uzaya haritalandırılması önerilmiştir. Bu sayede bu uzayda örnekleri ayırma işlemi çok daha kolay olacağı varsayılmıştır. Hesaplama yükünü mantıklı tutmak amacıyla, DVM şemaları tarafından kullanılan haritalandırılmalar, probleme uygun bir şekilde seçilen Kernel fonksiyon $k(x, y)$ cinsinden, gerçek uzaydaki değişkenlerin kolaylıkla hesaplanabilmesini garantilemek için tasarlanmıştır [64].

DVM, yüksek boyutlu verilerin ikili sınıflandırması için en gelişmiş model olarak yaygın şekilde kullanılmaktadır. DVM, doğru sınıflandırma payını en üst düzeye çıkarmak için eğitilir ve ortaya çıkan karar sınırları, özellik vektörlerinin hafif dalgalanmalarına karşı dayanıklıdır. DVM' nın üstün genelleme yetenekleri hem teorik çalışmalar hem de deneysel başarılarla teyit edilmiştir. Burada karar kuralı, hangi eğitici örneklerin karar sınırına yakın olduğunu gösteren iki özellik vektörü ve negatif olmayan katsayı $\{a_i\}_{i=1}^{n}$ arasındaki benzerliği hesaplayan bir çekirdek fonksiyonu K (x, x ') ile ifade edilir. DVM karar sınırına olan (işaretli) mesafesini hesaplayarak yeni örnekleri sınıflandırmaktadır. Eş. 3.5’ de sabit bir değere kadar bu mesafe hesabı verilmektedir.

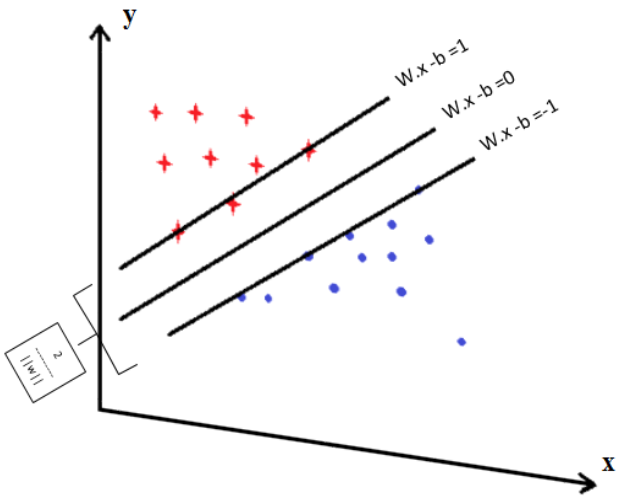
$$h(x) = \sum_{i=1}^{n} a_i(2y_i - 1)K(x_i, x) \quad (3.5)$$

Örneğin, toplamın tüm eğitim örnekleri üzerinde olduğu durumlarda, bu mesafenin işareti, üzerinde yatan karar sınırının tarafını belirtir. Uygulamada, h (x) değeri öznitelik vektörü x için bir ikili etiket tahmin etmek için eşik değerindedir. DVM' algoritmasında önce bir çekirdek fonksiyonu K (x, x ') belirtilir ve daha sonra eğitim kümesindeki doğru sınıflandırmanın payını en yükseğe çıkaran a_i katsayıları hesaplanarak eğitilirler [65].

Doğrusal Destek Vektör Makinaları

Elimizde 1 ... m ye kadar bir eğitim veri seti olsun:
$(\overrightarrow{x_1}, y_1), \ldots, (\overrightarrow{x_m}, y_m)$.
Burada y_m ifadesinin değeri ya -1 ya da 1 'dir. Aynı zamanda y değeri, her $\overrightarrow{x_m}$destek vektörlerinin ait olduğu sınıfı ifade eder. Doğrusal DVM' de amaç, sınıflandırılan iki gruba en uzak ayırımı doğrusal bir şekilde ifade edebilmektir (Şekil 3.21). Eş. 3.6' deki gibi bir hiperdüzlem olarak düşünüldüğünde ifade edilebilmektedir.

$$w.\vec{x} - b = 0 \quad (3.6)$$



Şekil 3.21. Doğrusal DVM grafiği

Burada W ifadesi ağırlık (hiper düzleme dik olan normal vektör) ve b değeri kayma noktasıdır. Eş. 3.7' de iki grup arasında mesafe farkı ifade edilmektedir.

$$\frac{b}{||w||} \quad (3.7)$$

Eğitim verileri doğrusal olarak ayrılabilirse, iki veri sınıfını birbirinden ayıran iki paralel hiperdüzlem seçilebilir, böylece aralarındaki mesafe ayırımın olabildiğince geniş olur. Bu iki hiperdüzlemle sınırlanan bölge "kenar" olarak adlandırılır ve maksimum paylı hiperdüzlem ikisinin ortasında olan hiperdüzlemdir. Farklı örnekleri ayırmak için kullanılan denklemlere bakıldığında ($w.\vec{x} - b = 1\ ve\ w.\vec{x} - b = -1$), doğrular arası mesafe b=2 olarak bulunmaktadır.

İki sınıfı ayırmak söz konusu olduğunda sonsuz sayıda doğru ile ayırım yapılabilir. Destek Vektör Makinalarında dikkat edilmesi gereken iki grup arasında en fazla aralığa (marja) sahip olanın seçilmesidir [66].

Destek Vektör Makinalarının avantajı, problemi dönüştürmek için belirli çekirdeklerden yararlanabilmeleri, böylece doğrusal olmayan sınıflandırma yöntemlerini lineer olmayan verilere uygulayabilmeleridir.

Doğrusal olmayan sınıflandırma

Esas maksimum paylı hiperdüzlem algoritması, Vapnik tarafından 1963 yılında bir doğrusal sınıflandırıcı olarak oluşturulmuştur. Bununla birlikte, Bernhard E. Boser, Isabelle M. Guyon ve Vladimir V. Vapnik, 1992 yılında çekirdek hilesini (başlangıçta Aizerman ve diğerlerinin önerdiği) maksimum paylı hiperdüzlemler uygulayarak doğrusal olmayan sınıflandırıcılar oluşturmanın bir yolunu önermiştir. Ortaya çıkan algoritma doğrusal DVM ile temelde benzerdir, sadece her nokta örneği, doğrusal olmayan Kernel fonksiyon tarafından değiştirilmiştir. Bu durum algoritmanın dönüştürülmüş özellik uzayı içinde maksimum paylı hiperdüzleme uymasını sağlamaktadır. Dönüşüm doğrusal olmayabilir ve dönüştürülmüş uzay yüksek boyutlu olabilir; Sınıflandırıcı dönüştürülmüş özellik alanındaki bir hiperdüzlem olmasına rağmen, orijinal giriş uzayı doğrusal olmayabilir. Doğrusal olmayan destek vektör makinalarında haritalama fonksiyonun kesin olarak ne olduğunu bilmeye gerek yoktur.

Yeterli örnek verildiğinde algoritmanın performansı iyi olmasına rağmen, yüksek boyutlu bir özellik alanında çalışması destek vektör makinalarının genel hata oranını arttırdığına dikkat çekmek gerekir [63].

Eş. 3.8’ de çekirdek fonksiyonunun genel ifadesi ifade edilmektedir.

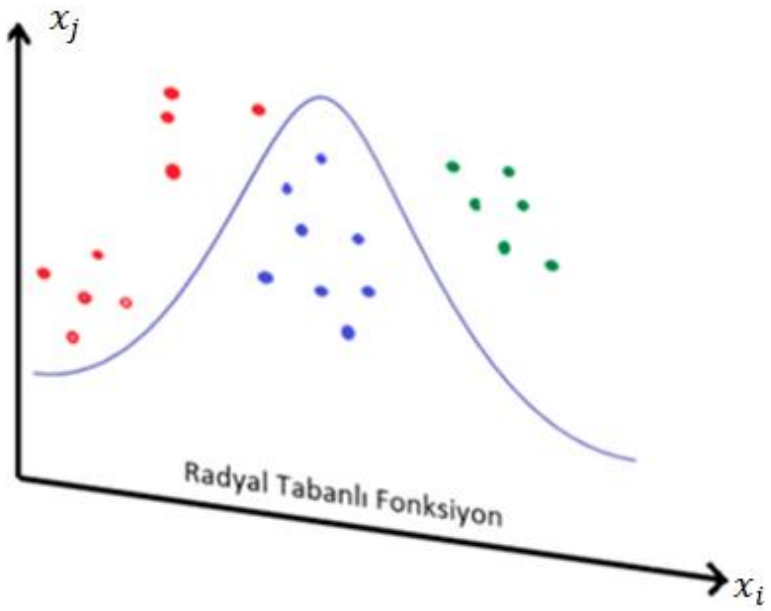
$$K(u,v) = \emptyset(u)\emptyset(v) = (uv+1)^2 \quad (3.8)$$

Çekirdek denklemlerinin uygulanması, veri örneklerini çok boyutlu uzayda ve bir veri örneğini diğerinden ayıran bir hiperdüzlemde düzenler. Çekirdek denklemleri, bir alandaki doğrusal olmayan ayrılabilir veriyi, örneklerin lineer olarak ayrılabilir hale geldiği başka bir alandan dönüştüren herhangi bir işlev olabilir. Çekirdek denklemleri doğrusal, kuadratik, Gauss veya bu özel amaca ulaşan başka herhangi bir şey olabilir. Verileri iki ayrı kategoriye böldükten sonra, amacımız iki küme örneklerini ayırmak için en iyi hiperdüzlemi elde etmektir [66].

Eş. 3.9, 3.10 ve 3.11’ de bazı yaygın çekirdekler (Kernel) verilmektedir [47].

- Polinomiyal Fonksiyon : $K(x_i, x_j) = (x_i x_j)^d$ (3.9)
- Sigmoid Fonksiyon : $K(x_i, x_i) = tahn\ (kx_i x_j - \delta)$ (3.10)
- Radyal Tabanlı Fonksiyon **:** $K(x_i, x_j) = \exp\left(-\gamma \left|\left|x_i - x_j\right|\right|^2\right), \gamma > 0$ (3.11)

Şekil 3.22’ de Radyal tabanlı fonksiyon grafiği gösterilmektedir.



Şekil 3.22. Radyal tabanlı fonksiyon grafiği

3.7.2 Çok Katmanlı Algılayıcı yapay sinir ağları

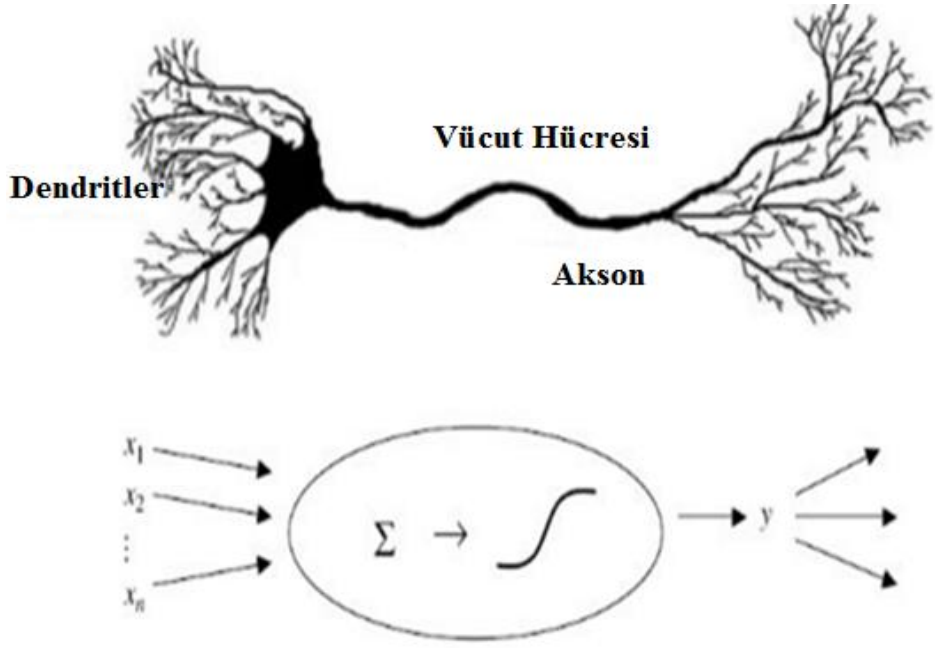
Yapay Sinir Ağı (YSA), insan beyninin yapısı ve davranışları örnek alınarak hazırlanmış ve insan beynine benzer yapıda bir modeldir. YSA, bilgileri işlemek için birbirine bağlı basit işlem birimlerinden oluşur. Bu modelin bilgisi, yapay nöronlar arasındaki bağlantıların her birinde bulunan ağırlıkta saklanmaktadır [67].

Yapay sinir ağları, karmaşık doğrusal olmayan işlevleri uygunlaştırma yetenekleri nedeniyle birçok alanda yaygın olarak kullanılmaktadır. Bu modeller;

- Genelleme,
- Adapte olma
- Örneklerden öğrenme yeteneği

gibi birçok avantaja sahiptir. Yıllar geçtikçe, bu alanda birçok çalışma ortaya çıkmış ve farklı YSA topolojileri oluşturulmuştur [67].

YSA, günümüzde kullanılan birçok algoritmanın çözümsüz kaldığı noktaya devreye girerek verilerin analiz edilmesini sağlamaktadır. Bu sistem, insan anatomisinin taklit edilmesi ile ortaya çıkan bu algoritma sayesinde otomatik olarak yeni bilgiler üretebilme yeteneğine sahiptir (Şekil 3.23).



Şekil 3.23. Sinir sistemi elemanlarının Yapay Sinir Ağı modelindeki terminolojisi [68]

İnsan anatomisine bakılarak benzetilen Yapay Sinir Ağı modelindeki her bir yapının işlevi bulunmaktadır. Bu modelin terminolojisi aşağıdaki (Çizelge 3.2) tabloda belirtilmiştir.

Çizelge 3.2. YSA terminolojisi

Sinir Sistemi	Yapay Sinir Ağı
Nöron	İşlem Elemanı
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon Fonksiyonu
Akson	Eleman Çıkışı
Sinaps	Ağırlıklar

YSA ile regresyon, sınıflandırma, optimizasyon, karar verme, doğrusal olmayan sistemlerin modellenmesi ve tahmin yürütme gibi işlemler yapılabilmektedir. Bu çalışmada yaygın olarak kullanılan yapay sinir ağı modellerinden çok katmanlı algılayıcı ağı kullanılmıştır.

Çok Katmanlı Algılayıcı (ÇKA) modeli, giriş verilerini uygun çıkışlara haritalandıran ileri beslemeli bir yapay sinir ağı modelidir. Bu model ile diğer sinir ağ modelleri gibi sınıflandırma, örüntü tanıma, tahmin, regresyon, optimizasyon vb. birçok uygulama alanında çalışılmaktadır.

Bir ÇKA sinir yapısında bulunan ara katmanlar en büyük üstünlüğüdür. Bu özelliği ile bu sinir ağı en az ara katmana sahip olduğu sürece herhangi bir sürekli fonksiyona yaklaşabileceğini veya ara katman sayısının birden fazla olması durumunda herhangi bir matematik fonksiyonuna uygunluk gösterebileceğini garanti eder [67]. Aynı zamanda etkili öğrenme algoritması, hata toleransı, genelleme kapasitesi ve paralel uygulamaları destekleme kabiliyetleri nedeniyle ÇKA sinir ağı yaygın şekilde kullanılmaktadır [69].

YSA veri odaklı modellerdir. Bir ÇKA sınıflandırıcısı YSA ’ı eğitmek için kullanılan bilgi kadar iyidir. Ancak, çoğu vakada YSA eğitilirken, kullanıcılar veri kalitesini analiz etmemekte veya özellikleri seçmemektedir [70].

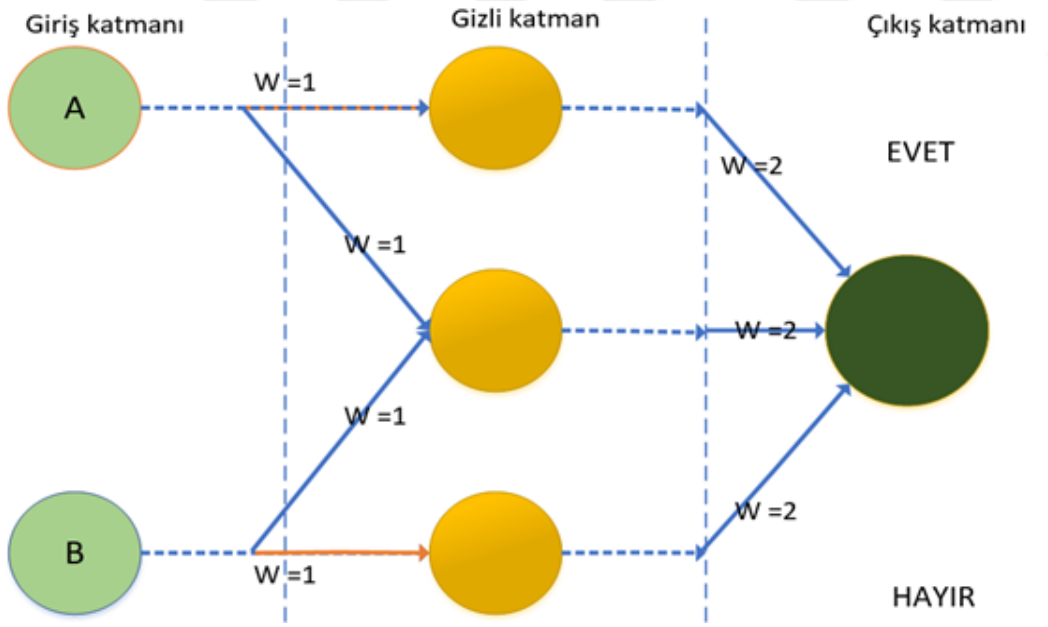
ÇKA, standart doğrusal algılayıcının bir modifikasyonudur ve doğrusal olarak ayrılamayan verileri ayırabilir [71].

Üç tabakadan meydana gelir;

- Giriş Katmanı
- Gizli Katman
- Çıkış Katmanı

Giriş katmanı, ağda sınıflandırılacak nesneden çıkarılan özellik vektörleridir. Çıkış katmanı, tipik olarak nesnenin sınıfını gösteren bir çıkışın yüksek iken tüm diğer çıkışların düşük olduğu bir blok kodudur. Analizi yapılacak olan problemin giriş değerleri giriş katmanında, çıkış değerleri çıkış katmanında verilir. Ara katman ise doğrusal olmayan bir problemin sınıflandırılmasında önemli rol oynar [72].

Çok katmanlı algılayıcı, yönlü grafikte birden fazla düğüm katmanından oluşur ve her bir tabaka bir diğerine tamamen bağlıdır. Giriş düğümleri haricinde, her düğüm doğrusal olmayan aktivasyon fonksiyonu ile beraber bir nöron (veya işleme ögesi) 'dir. Bu yöntemde, ağın eğitimi için geri yayılım olarak adlandırılan denetimli öğrenme tekniği kullanılır [73-74].



Şekil 3.24. ÇKA sinir ağı modeli örneği

Şekil 3.24' deki ÇKA modeline göre giriş katmanı, gizli katman ve çıkış katmanı olan çok katmanlı ileri beslemeli bir sinir ağ model örneği görülmektedir. 'W' ile gösterilenler ağın ağırlıklarıdır. Nöronlar arasındaki bağlantılar bir ağırlığa sahiptir.

Her katmanda düğümler (nöronlar) yer almaktadır. Her düğüm, ÇKA' nın yapay bir nöronunu temsil eder ve her yönlendirilmiş yaydaki etiketler, iki nöron arasındaki sinaptik bağlantının gücünü ve ÇKA' da sinyal akış yönünü belirtir.

Örüntü sınıflandırma için, ÇKA 'nın giriş katmanındaki nöronların sayısı, özellik alanındaki ilgili örüntüleri temsil etmek için, özelliklerin sayısına göre belirlenir. Çıktı katmanı, girdi verisinin ait olduğu sınıfların sayısı tarafından belirlenir. Giriş ve çıkış katmanlarındaki nöronlar, her nörona karşılık gelen bağlantıların giriş değerlerini ve ağırlık değerlerinin çarpımının toplamı üzerinde sigmoidal fonksiyonu hesaplar. ÇKA' nın eğitim sürecinde, sinaptik bağlantılarının güçlü yönlerini ayarlamayı ve böylece eğitim setinden alınan her girişi uygun bir şekilde yanıtlamayı içerir. Gizli katmandaki nöronların sayısı ve gizli katmanların sayısı eğitim sürecinde belirlenmelidir [75].

Düğümlere bağlı ağırlıklar, bir dizi özellik vektörü olan eğitim seti kullanılarak belirlenir [72]. Bir yapay sinir ağında bilgi ağırlıklara göre değerlendirilir. Ağırlıkların belirlenmesinde en uygun değer alınması için ağın eğitilmesi gereklidir. Bu sistemde en uygun ağırlık bulunana kadar ağırlıklar belirli bir şekilde dinamik olarak değiştirilerek ağ eğitilir. Sinir ağların farklı topolojilerinde ağırlıkların optimize edilmesinde bir eğitim algoritması kullanılmaktadır.

Bir sinir ağında, düğümler arasında bulunan ağırlıklar eğitime sokularak ayarlanır. Ağırlıkların ayarlanmasında çeşitli eğitim kuralları geliştirilmiştir. Bugüne kadarki en popüler kural, Paul Werbos tarafından geliştirilen, David Parker tarafından yeniden keşfedilen ve David Rumelhart tarafından yaygınlaştırılan geri yayılımdır [74]. Geri Yayılım modeli, teknik, tüm eğitim seti boyunca her çıkış düğümünde kareli hata toplamıyla oluşturulan bir hata yüzeyi üzerindeki ağırlık alanındaki eğim inişine dayanır. Ağırlıkların ayarlanması için bir başka kural da başlangıçta Singhal ve Wu tarafından önerilen genişletilmiş Kalman filtreleme algoritmasıdır. Kalman filtreleme yaklaşımı, ağdaki ağırlık değerlerinin bir tahmin problemi olarak bulunmasını göz önünde bulundurur. Kalman filtresinin amacı, zaman içinde alınan gözlemlere dayalı olarak bir sistemin durumunu tahmin etmektir.

Çok Katmanlı Algılayıcı hususunda, sistemin durumu ağın ağırlıklarına göre tahmin edilir. Gözlemler, eğitim vektörleri sunulduğunda ağın çıktılarıdır. Kalman filtresi, sistemin

doğrusal modeli ve sistemin durumu ile doğrusal olarak ilişkili gözlemlere dayanmaktadır. Çok katmanlı algılayıcı, doğrusal olmayan bir ağ olduğundan, genişletilmiş bir Kalman filtresi kullanılır. Genişletilmiş Kalman filtresi, doğrusal olmayan sistemin doğrusallaştırılmış bir modelini kullanır ve model her güncelleme sonrasında sistemin tahmini durumunda (yani, ağırlıklar güncellendiğinde) yeniden doğrusallaştırılmıştır. Bu algoritma, daha az sayıda eğitim vektörü kullanılarak ile iyi ağırlık değerleri bulabilir ancak geri yayılımla (Back Propagation) karşılaştırıldığında hesaplama yükü fazladır [72].

Giriş katman sayısı, analizi yapılmak istenen problemde verilen veri setinin özellik sayısına (sütun sayısı) eşittir. Her sinir ağ modelinde çıkış katmanında, tek çıkış vardır. Düğümlerin sayısı, seçilen modelin yapısına bağlı olarak farklılıklar gösterir.

Kullanılan katmanların sayısı yüksek oranda, sınıflandırma problemine göre belirlenir. Ancak, fonksiyonu ölçmek için en az bir gizli katman gereklidir. Gizli katman sayısını belirlemede birden fazla yöntem önerilmiştir. Cybenko, çalışmasında çoğunlukla benzer fonksiyonlar için en fazla bir gizli katmanın yeterli olduğunu sunmuştur. [71,76]; bununla birlikte, yapılan çalışmalara bakıldığında iki gizli katman ağının, bazı sınıflandırma problemlerinde bir gizli katmandan daha hızlı eğitilebileceği görülmüştür. Problemin kendisine bağlı olarak katmanlarda düğüm sayısı değiştiği gibi kullanılan katman sayısı da değişebilmektedir. Buradan anlaşılacağı üzere bir yapay sinir ağı modelinde gizli katmanlarda kaç düğüm kullanılacağı ya da katman sayısını belirlemede çalışma performansı en iyi olanın göz önünde bulundurulması gereklidir. Jeff Heaton, gizli katman için en uygun boyutun, genellikle giriş katmanı ile çıkış katmanı boyutu arasında olmasını önermiştir [77].

İleri beslemeli bir gizli katmana sahip ÇKA ağının, herhangi bir sürekli fonksiyonu istenilen doğrulukta yakınsayabileceği kanıtlanmıştır [71]. Gizli katmanın en uygun boyutu tek olarak [71,78] veya giriş ile çıkış katmanlarının boyutu genellikle arasında alınmaktadır. Gizli katmandaki düğümlerin sayısını giriş ile çıkış katmanları düğüm sayıları ortalaması [79] alınarak veya performans karşılaştırması [78] yapılarak belirlenmiştir.

ÇKA ağında, giriş ve çıkış katmanlarının beraberinde gizli birimler ve düğümler içeren bir veya daha fazla sayıdaki tabakalar, düğümlerdeki doğrusal olmayan özelliklerden kaynaklanmaktadır. Eğer düğümler lineer ögeler olsaydı, uygun seçilmiş ağırlıkları olan tek

katmanlı bir ağ, çok katmanlı herhangi bir diğer çok katmanlı ağ tarafından gerçekleştirilen hesaplamaları tam olarak tekrarlayabilirdi [80].

Bu çalışma kapsamında, ÇKA ve DVM veri madenciliği metotları ile sınıflandırma ve sonuçların analizi için WEKA yazılım aracı kullanılmaktadır. ÇKA, örnekleri sınıflandırmak için gradyen tabanlı algoritmalardan GY kullanmaktadır.

3.8. Problem Çözümünde Algoritmik Yaklaşım

Algoritma kelimesi başlangıcını Horazmi' nin "Cebir ve Mukayese kitabı" ndan almaktadır. Algoritma kavramının kendisi, Horazmi' den çok eskilere dayanmaktadır. Euclid ve Arşimet' in eserlerinde ardışık çözüm yolunu gösteren algoritmalara rastlanmaktadır. Algoritmalarla ilgili hem teorik hem de uygulamalara yönelik araştırmalar bilgisayarların gelişmesiyle hız kazanmıştır. Hesaplanabilir yöntemlerin yanı sıra, en etkin hesaplanabilir çözümlerin uygulanabilirliği gösterilmiştir. Günümüzde algoritma, matematik ve bilişimde önemli bir kavram olarak yerini almaktadır. Algoritmik işlemler dizisi bir veriye uygulandığında kesin bir sonuca, yani belli bir problemin çözümüne ulaşmaya imkân vermektedir. Dolayısıyla algoritma, bir sonucun bulunmasını kolaylaştıran bir yöntemdir [81].

İyi biçimlendirilmiş bir problemin çözümün doğruluğunu algoritmik yaklaşımla gösterilebileceği düşünülmektedir. Birçok problem aşağıdaki yaklaşımla çözülebilmektedir;

- Mümkün olabilecek durumlar arasında herhangi birisinin ele alınması
- Ele alınan duruma mümkün gidişler uygulanarak durumun değiştirilmesi
- Durumun değerlendirilmesi
- Faydasız durumların atılması
- Eğer sonuca ulaşılmışsa çözümün tamamlanması, aksi halde yeni değer ele alınarak işlemlerin tekrarlanması

Algoritmik yaklaşımda kesin bir durum ele alınarak faydasız durumlar elimine edilerek sonuca ulaşılmaktadır. Problem çözümüne yönelik çeşitli öneriler olsa da aşağıdaki soruların önceden bilinmesi gerekmektedir.

- Problem hangi sınırlar içerisinde belirlenmiştir

- Problemin başlangıç şartları nedir?
- Problemin çözümünde hangi sonuca erişilmesi amaçlanır?
- Sınırlamalar içerisinde hangi izinli hareketler ve davranışlar mümkün olmaktadır?

Descartes' e göre doğruluğu kesin olarak kabul edilmeyen hiç bir şey kabul edilmemelidir. Karşılaşılan her problem mümkün olduğunca parçalanmalıdır. Problemde kolay çözülebilecek sorunlardan zor ve karmaşık olanlara geçilmelidir.

Bir problemi çözmede izlenilecek adımlar:

- Problemin doğru analizi
- Probleme uygun çözüm yolunun tespiti
- Çözümün algoritmik olarak tanımlanması ve yazılması
- Test aşaması

Mühendislik, temel bilimler gibi çeşitli alanlarda yapılan çoğu bilimsel araştırmada veriler üzerinde yapılan yöntem ve analizlerin amacı geleceğe dair bir çıkarım yapmaktır. Teknoloji yaygınlaştıkça ihtiyacı karşılamak için veri depolama teknikleri de bir o kadar gelişmiştir. Depolanan veri yapıları büyüdükçe ve karmaşıklaştıkça, veriyi anlamlandırmak ve bu kadar çok veriden anlamlı çıkarımlarda bulunmak değer kazanmıştır. Tahmin oluşturmak için yaygın olarak kullanılan yöntemler makine öğrenmesi, istatistikler ve veri madenciliğidir [81].

Geçmiş verileri matematiksel ve istatistiksel yöntemler ile bilgisayara öğreterek gelecek veriler hakkında tahminlerde bulunulması işine makine öğrenmesi denir. Makine öğrenmesi Yapay Zekâ 'nın bir alt dalı olarak ortaya çıkmıştır. 1959'da Arthur Samuel, makine öğrenmesini bilgisayarın açıkça programlanmadan bilgisayara öğrenme kabiliyeti, bir işi öğrenme yeteneği kazandıran bir çalışma alanıdır [53].

4. BUZLANMA TAHMİNİ İÇİN BİR MODEL GELİŞTİRİLMESİ

Bu bölümde gerçekleştirilen çalışmada sırasıyla;

- Toplanılan verilerin kullanılmak üzere modelin doğruluk oranını arttırmak amacıyla temiz bir veri seti haline getirilmesi için hangi ön işlemlerden geçirildiği
- Buzlanma tahmini için geliştirilen algoritmanın çalışma prensibi

hakkında kapsamlı olarak bahsedilmektedir.

4.1. Veri Analiz Yöntemleri

Gözlemlenebilen/Ölçülebilen bütün sistemler de belirli oranlarda - az ya da çok - kayıp veri varlığı kaçınılmazdır. Veri kaybı belirlenen ya da belirlenemeyen pek çok nedenden kaynaklanabilir. Doğru ve güvenilir analizler için veri kümesinin eksiksiz olması oldukça önemlidir.

Kayıp verilerin değerlendirilmesindeki ilk yöntem kayıp veri olan satırı yok saymaktır. Ancak kayıp verinin çok olduğu yada az kayıta sahip testlerde bu çözüm sonucu yanlış değerlere saptırmaktadır. Bu yüzden kayıp verilerin yerine bir değer ataması yapılarak daha doğru sonuçlar elde edilebilmektedir [82]. Eğitimde kullanılmak üzere toplanan eğitim veri kümesinin doğru sonuçlar üretebilmesi için dengeli hale getirilmesi gereklidir

Bu çalışmada kayıp yada eksik verilerin değerlendirilmesi amacıyla;

- Liste Boyunca Silme Yöntemi
- Doğrusal Regresyon Yöntemi
- Rastgele Altküme Üretilmesi Yöntemi

kullanılarak eksik veriler kaldırılmış ya da veriler tamamlanmıştır.

4.1.1. Liste boyunca silme yöntemi

Bir sınıfın örnek sayısının diğer sınıftan fazla olduğu durumlarda genellikle az örnek sayısına sahip sınıf için tahmin veya sınıflandırma yapılmak istenir. Hava istasyonlarında alınan eğitim veri kümesinde koşulların sertleşmesi ve sistemlerdeki arızalardan dolayı bazı zamanlar verilerin alınmasında problemler yaşanabilmektedir. Bu gibi durumlarda eksik

(Şekil 4.1) veya alınamamış niteliklerin uygulamada kullanılması için tekrar değerlendirilmesi gereklidir. Bu nedenle, elimizde yer alan "Buzlanma Yok " sınıfına ait 4979 örnek içinden yaklaşık 600 tanesi liste boyunca silme yöntemi ile filtrelenmiştir.

```
@attribute 'Temperature'               numeric
@attribute 'Dew_Point'                 numeric
@attribute 'Feel_Temperature'          numeric
@attribute 'Wind_Gust_C'               numeric
@attribute 'Relative_Humidity_Yude'    numeric
@attribute 'Wind_Spee_km_h'            numeric
@attribute 'Wind_Direction'            numeric
@attribute 'Icing_forecast'  { 'Buzlanma_yok', 'Buzlanma_var'}


%  Temperature,Dew_Point,  Feel_Temperature,Wind_Gust_C,Relative_Humidity_Yude,Wind_Spee_km_h, Win


@data
4.253,-2.909,5.838,9.521,59.670,1.918,295.284,'Buzlanma_yok'
4.327,-2.830,6.030,9.615,59.713,1.157,61.758,'Buzlanma_yok'
4.327,-2.830,6.076,9.615,59.713,1.456,346.692,'Buzlanma_yok'
4.673,-2.507,5.871,9.615,59.700,1.684,255.653,'Buzlanma_yok'
4.816,-2.491,5.669,9.615,59.175,2.498,282.309,'Buzlanma_yok'
4.816,-2.491,5.669,9.628,59.175,2.105,335.233,'Buzlanma_yok'
4.926,-2.743,5.823,9.708,57.639,2.141,281.556,'Buzlanma_yok'
5.064,-2.381,5.951,9.748,58.641,1.765,267.212,'Buzlanma_yok'
5.064,-2.381,5.951,9.800,58.641,3.002,280.104,'Buzlanma_yok'
4.933,-2.561,5.773,9.800,58.392,3.119,85.460,'Buzlanma_yok'
4.770,-2.696,5.694,9.800,58.471,1.600,151.235,'Buzlanma_yok'
4.770,-2.696,5.535,9.800,58.471,1.316,44.736,'Buzlanma_yok'
4.856,-2.522,5.769,9.800,58.877,2.025,328.138,'Buzlanma_yok'
4.927,-2.478,5.849,9.800,58.780,0.000,0.000,'Buzlanma_yok'
4.927,-2.478,5.969,9.800,58.780,1.901,281.429,'Buzlanma_yok'
4.938,-2.537,5.905,9.800,58.477,4.269,36.337,'Buzlanma_yok'
4.781,-2.666,5.583,9.708,58.556,0.000,0.000,'Buzlanma_yok'
```

Şekil 4.1. Hava durumu veri setinde bir kısım eksik veri örneği (.arff dosyası)

4.1.2. Doğrusal Regresyon yöntemi

Hava olaylarından buzlanma olayı sadece kış aylarında belirli koşullar altında meydana gelmektedir. Bu yüzden elimizdeki "Buzlanma Var" sınıfına ait örnek sayısı azdır ve bu örnekler tahmin için önemlidir. Bu nedenle " Buzlanma Var" sınıfına ait 53 verinin içinde yer alan eksik veriler Doğrusal Regresyon Yöntemi kullanılarak doldurulmuştur. "Buzlanma Var" sınıfına ait 7 nitelik ve 53 örnekten, bazı örneklerin "Rüzgâr Yönü" ve "Rüzgâr Hızı" nitelikleri altındaki verileri eksiktir. Doğrusal Regresyon yöntemi bu eksik verilerin tamamlanmasında kullanılmıştır.

Doğrusal Regresyon kullanılarak, örneklere en yakın doğrunun çizilerek sistemin matematiksel olarak modellenmesi ve eksik verilerin tamamlanması sağlanmıştır. Elde edilen matematiksel yaklaşım_1 ve yaklaşım_2 (Eş. 4.1, Eş. 4.2) de verilmektedir.

Yaklaşım_1;

$$\begin{aligned}\text{Rüzgar_Yönü} = {} & (-17.5828 * \text{Sıcaklık}) \\ & + (-24.393 * \text{Hissedilen_Sıcaklık}) \\ & + (34.8641 * \text{Rüzgar_Şiddeti}) \\ & + (1.6018 * \text{Bağıl_Nem}) \\ & + 3.4677 \end{aligned} \quad (4.1)$$

Yaklaşım_2

$$\begin{aligned}\text{Rüzgar_Hızı} = {} & (-0.5543 * \text{Sıcaklık}) \\ & + (-0.5134 * \text{Hissedilen_Sıcaklık}) \\ & + (0.2713 * \text{Rüzgar_Şiddeti}) \\ & + (-0.1374 * \text{Bağıl_Nem}) \\ & + (0.82 * \text{Çiğ_Noktası}) \\ & + 15.6895 \end{aligned} \quad (4.2)$$

Yaklaşım_1 ve Yaklaşım_2 matematiksel modellemelerde nitelikler ile çarpılan sayısal değerler, o niteliklerin ağırlıklarıdır.

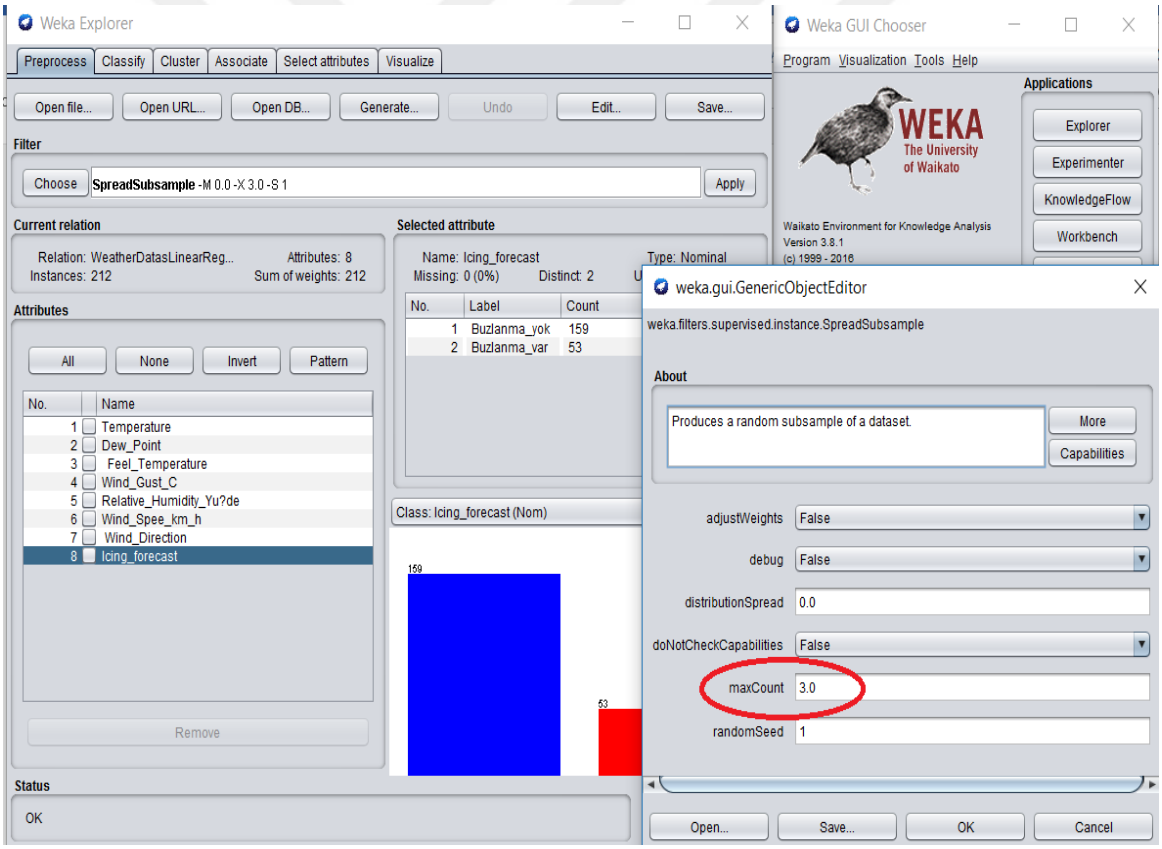
Bir niteliğin ağırlığı, sistemi ne kadar etkilediğini gösterir. Örneğin, Rüzgâr Yönü niteliği matematiksel modellemesinde Sıcaklık, Hissedilen Sıcaklık, Rüzgâr Şiddeti ve Bağıl Nem hesaplamada kullanılırken, Çiğ noktası niteliği etkisi (ağırlık) çok az olduğu için hesaba katılmamıştır. Çizelge 4.1 de hesaplanan veriler yer almaktadır.

Çizelge 4.1.Tamamlanan eksik rüzgâr hızı ve rüzgâr yönü verileri

Rüzgâr Hızı (km/sa)	Rüzgâr Yönü (°)
2,609	205,351
2,176	167,072
2,701	211,19
1,623	122,361
1,625	123,37
1,480	128,96
1,665	126,428
1,654	127,192
1,647	129,241
1,655	132,691
1,612	134,677
1,6	148,549
1,622	149,679
1,618	149,403
1,617	139,77
1,639	145,668
1,629	147,217
1,666	145,028
1,656	145,778
1,638	144,069
1,675	145,966
1,635	232,506
1,948	250,278
2,518	-

4.1.3. Rastgele altküme üretilmesi yöntemi

Hava durumu veri setinde eksik verilerin temizlenmesi ve tamamlanması için yapılan liste boyunca silme ve doğrusal regresyon yöntemlerinden sonra elimizde, "Buzlanma Yok" sınıfına ait yaklaşık 4300, "Buzlanma Var" sınıfına ait 53 örnek vardır. Bu veri seti veri madenciliği yöntemlerinin kullanılması için hala dengesizdir. Bu gibi durumlarda sınıflandırma süresi uzamaktadır ve yüzde tahmini sapmaktadır. Fazla örnek sayısına sahip sınıf doğruluk oranının yanlış değerlendirilmesine neden olmaktadır. Bu nedenle çalışmamızda dengesiz veri setinden dengeli veri seti üretmek amacıyla, WEKA filtreleme yöntemlerinde "SpreadSubsample" kullanılarak rastgele altküme üretilmiştir. Maksimum yayılım miktarı üç kat olarak (Şekil 4.2) belirlenmiştir.



Şekil 4.2. Rastgele altküme üretilmesi yönteminde maksimum yayılım oranı seçimi

Eksik veriler tamamlanarak veri seti dengeli bir hale getirildikten sonra, Sınıflandırılma ve tahmin için eğitim veri setinde kullanılmak üzere elimizde 53 "Buzlanma Var" sınıfına ait örnek ve 159 "Buzlanma Yok" sınıfına ait örnek bulunmaktadır.

4.2. Buzlanma Tahmini için Bir Algoritma Geliştirilmesi

Geçmiş buzlanma bilgisi ve meteorolojik nitelikleri ile eğitilen sistem gelecek nitelik bilgilerini esas alarak buzlanma olup olmayacağına dair bir tahmin öngörmektedir.

4.2.1 Tasarlanan model

Problemin çözümünde prensip olarak sırasıyla;

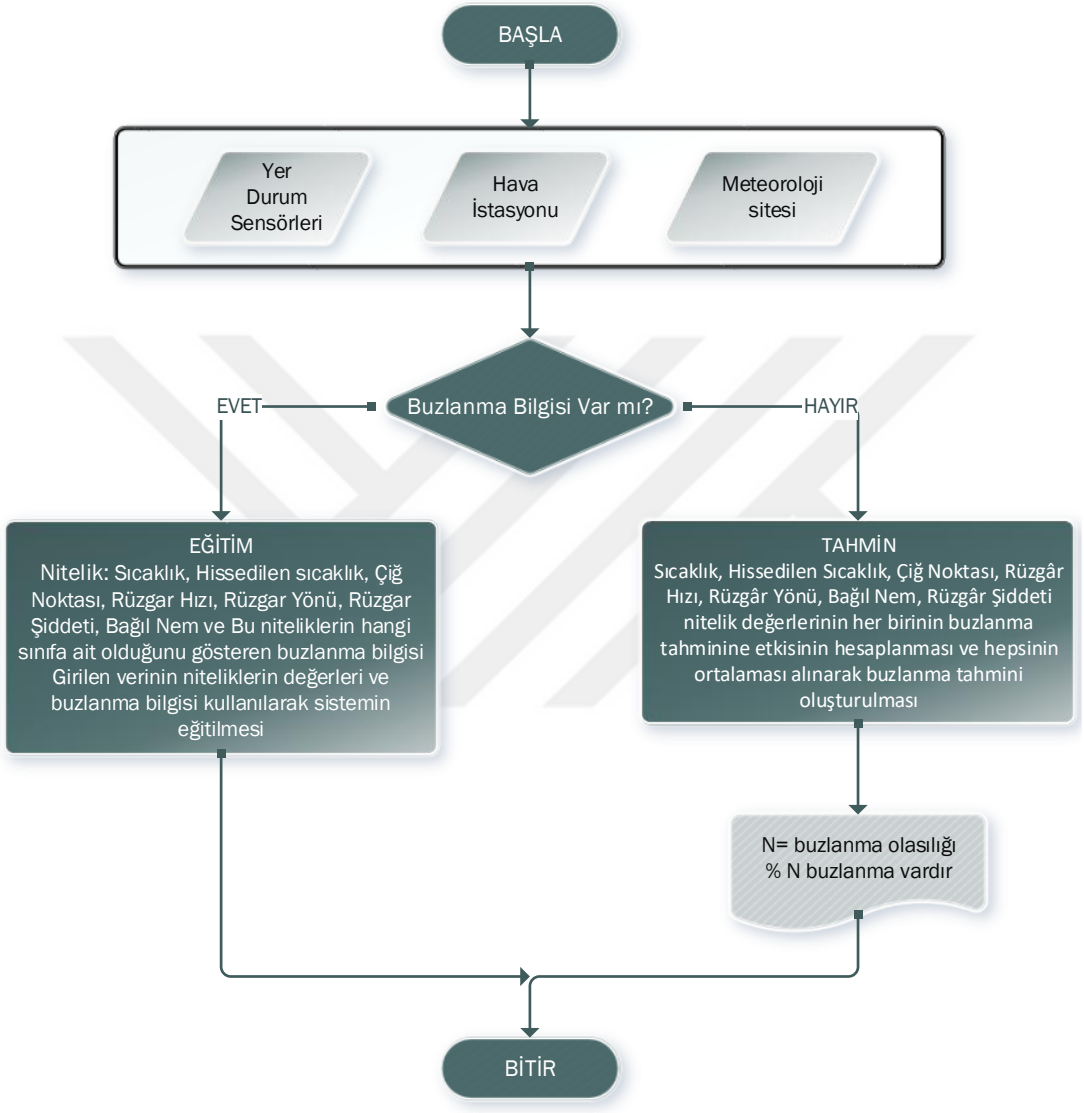
- Problem tanımı
- Problemim çözümüne yönelik veritabanının oluşturulması
- Sistemin önerilen algoritmik yaklaşıma göre eğitilmesi
- Son olarak eğitilen sistemin meteorolojiden alınan hava durumu nitelikleri kullanarak buzlanma tahmini yapılması adımları izlenmiştir.

Hava istasyonu, yer durum sensörleri ve meteoroloji web sitesinden elde edilen öznitelikler, geliştirilen algoritma için kullanılmaktadır. Her ne kadar aşağıdaki öznitelikler algoritmanın girişi olarak kullanılsada buzlanma bilgisi değeri ilk 7 özniteliğin çıktısıdır. Geçmiş, şimdiki ve gelecek zamana ait verilerden sisteme aktarılan bu öznitelikler şunlardır;

- Sıcaklık
- Hissedilen Sıcaklık
- Çiğ Noktası
- Rüzgâr Hızı
- Rüzgâr Yönü
- Bağıl Nem
- Rüzgâr Şiddeti
- Buzlanma Bilgisi (geleceğe ait bilgi bulunmamaktadır.)

Yukarıda verilen özniteliklere bakılarak gelecek 12 saat için birer saatlik periyotlarla buzlanma tahmini yapılmaktadır. Bu tahmin yüzde olarak 0 ile 100 değerleri arasında bir çıkış üretmektedir. Tahmin çıkış değeri %50 den büyük ise “Buzlanma var” değilse “Buzlanma yok” şeklinde sınıflandırılmaktadır.

Geliştirilen algoritmada gerçek zamanlı elde edilen veriler eğitim için kullanılmaktadır. Bu veriler hava durum değerleri ve yoldan alınan buzlanma bilgisini içermektedir. Geleceğe dair hava durumu tahmin verileriyle de yollar için buzlanma tahmini yapılmaktadır (Şekil 4.3).



Şekil 4.3. Önerilen algoritmanın akış diyagramı

Eğitim için buzlanma bilgisi olan veriler kullanılmaktadır.

$$N_{samples}^{attribute} = N_s^a = \{N_s^1 \dots N_s^7\} = \begin{cases} Sıcaklık \\ Hissedilen\ Sıcaklık \\ Çiğ\ Noktası \\ Rüzgar\ Hızı \\ Rüzgar\ Yönü \\ Bağıl\ Nem \\ Rüzgar\ Şiddeti \\ Buzlanma\ Bilgisi \end{cases} \quad (4.3)$$

Eş. 4.3'te 'attribute' nitelikleri 'samples' örnek sayısını belirtmektedir ve sırasıyla 'a' ve 's' olarak kısaltılarak gösterilmektedir. Eğitim veri kümesinde toplam 8 nitelik ve her bir niteliğin verileri bulunmaktadır. Örneğin N_{34}^{1} ={1.niteliğin (Sıcaklık) 34.verisinin } belirtmektedir.

$P_{sampling} = P_s$ Buzlanma bilgisi ifade edilmektedir. Örneğin P_{34} ={34. Örneğin buzlanma bilgisi} belirtmektedir. Örnekler "buzlanma var" ve "buzlanma yok" olarak sınıflandırılmaktadır. Buzlanma tahmini oluşturulurken bu nominal değerler numerik değerlere dönüştürülmektedir.

Tüm verilerin ağırlıkları w_s^a ile ifade edilmektedir. Örneğin w_{34}^{1} ifadesi 1. Niteliğin 34. verisinin ağırlığını ifade etmektedir. Eş. 4.4' te verilerin ağırlık hesaplaması ifade edilmektedir.

$$w_s^a = \frac{P_{s+1}^a - P_s^a}{N_{s+1}^a - N_{s+1}^a} \tag{4.4}$$

Ağırlıkların bulunması ve eğitim için kullanılması işlemi YSA gibi birçok sınıflandırma algoritmasında kullanılmaktadır. Geliştirilen algoritmada öznitelikler için ortalama ağırlıklar hesaplanmaktadır. Ortalama ağırlık hesaplanırken tüm verilerin ağırlıklarını içeren listeden faydalanılmaktadır $(1, w_1^a, w_2^a \ldots w_s^a)$. Bu listenin ilk değeri 1 olarak atanmaktadır. Eş. 4.5'te tüm öznitelikler için Ortalama Ağırlık hesabı ifade edilmektedir.

$$W^a = \frac{1 + \sum_{k=1}^{s} w_k^a}{samples\ (s)} \tag{4.5}$$

Eğitim ile hava durumu özniteliklerinin buzlanmayı nasıl etkilediği hesaplanmaktadır. Hesaplanan etki yeni buzlanma tahmini oluşturulması için kullanılmaktadır. Eş. 4.7 'de etki verilmektedir.

$$E = [\sum_{a=1}^{7} (\Delta N^a . W^a)] \tag{4.7}$$

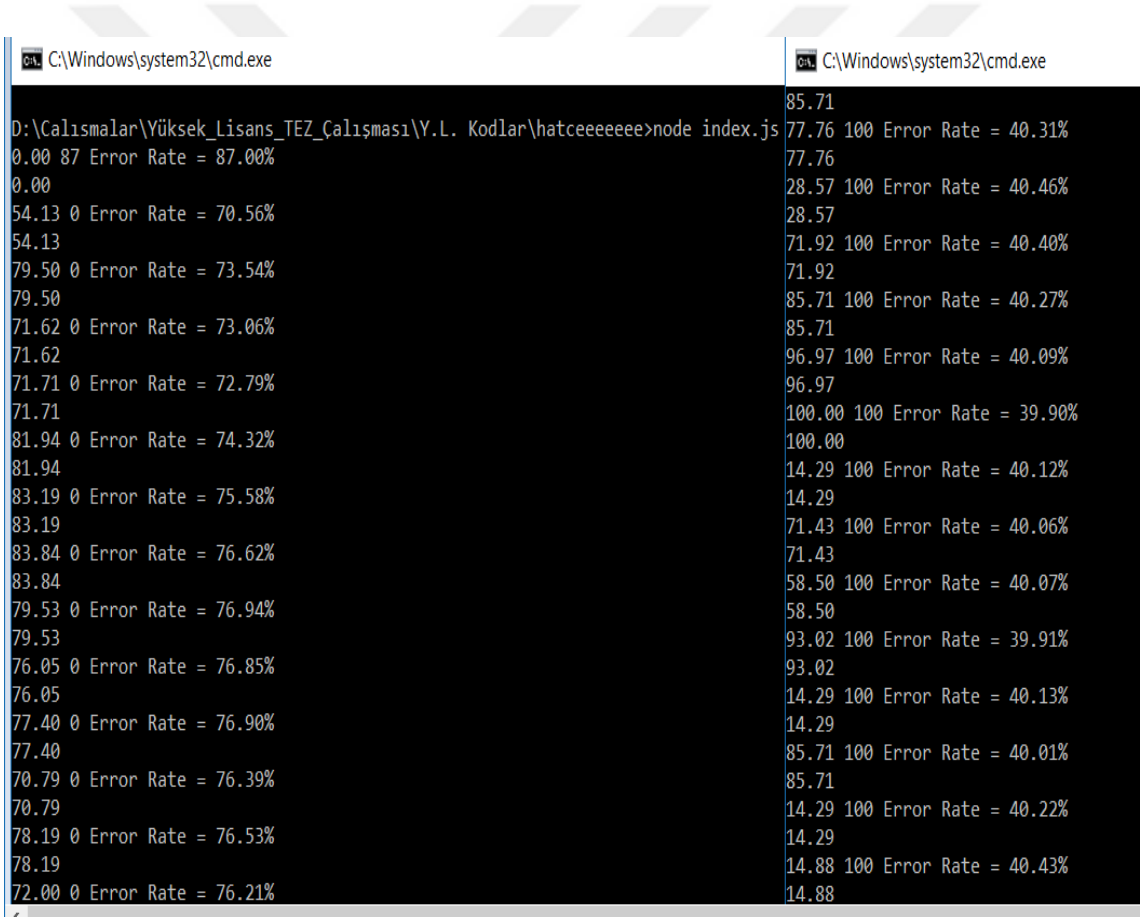
Buzlanma tahmini Q_s ile ifade edilmektedir (Eş. 4.7).

$$Q_s = P_{s-1} + \mathrm{E} \tag{4.7}$$

Buzlanmayı etkileyen her nitelik için hesaplamalar yapılmaktadır ve her niteliğin buzlanmaya etkisi dikkate alınmaktadır.

4.2.2 Model sonuçları ve doğruluk

Model eğitildikçe tahmin doğruluğunda iyileşme artmaktadır ve hata oranı azalmaktadır. Sistem ilk eğitildiğinde hata oranı %87 iken, eğitim sayısı arttıkça hata oranı %40'lara kadar gerilemektedir. Geliştirilen modelde hatanın nicel büyüklüğü her ne kadar büyük olsada hatadaki yönelim sürekli olarak sıfıra doğrudur. Sistemi eğitmek için kullanılan örnek sayısı ile tahmin doğruluğunun doğru orantılı olduğu görülmektedir (Şekil 4.4).



Şekil 4.4. Sistemin eğitilmesi ve iyileşme oranları

Bu çalışmada eğitim veri kümesinden alınan buzlama tahminleri oranları % olarak ifade edilmektedir. "Buzlanma Var" sınıfı %100 , "Buzlanma Yok" sınıfı %0 olarak atanmaktadır. %50 ve üzerindeki tahminler "Buzlanma var" olarak öngörülürken, %50 ve altı durumlarda buzlanma olma ihtimali düşük olduğu ifade edilmektedir ve "Buzlanma yok" olarak öngörülmektedir.

Geliştirilen algoritmada tüm verilere bakıldığında %78 oranında başarım sağlanmaktadır. “Buzlanma Var” sınıfına ait 53 veriden 32 tanesi %50’nin üzerindedir. Bu oranlara bakıldığında “Buzlanma Var” sınıfı doğru sınıflandırılan örnek yüzdesi %60.37 olmaktadır. Yapılan çalışmada doğru sınıflandırma oranı düşük olsa da, sistem daha fazla örnekle eğitildiğinde sistem kendini eniyilemektedir. Örneğin %100 buzlanma olma durumundaki bir örneği ilk eğitimde %58 buzlanma olabilir oranında tespit ederken, daha fazla sistem –iki kere- eğitildiğinde aynı örnek için %86 buzlanma olabilir oranına kadar çıkmaktadır.

5. BUZLANMA TAHMİNİ İÇİN BİR MOBİL UYGULAMA GELİŞTİRİLMESİ VE DENEYSEL SONUÇLAR

Bu bölümde sırasıyla;

- Buzlanma tahmini yapan bir mobil uygulama geliştirilmesi
- Veri madenciliği yöntemlerinden çok katmanlı algılayıcı sinir ağı modeli, doğrusal\ doğrusal olmayan destek vektör makinaları ve geliştirilen sistemin buzlanma tahmin yöntem uygulama sonuçlarının değerlendirmesi

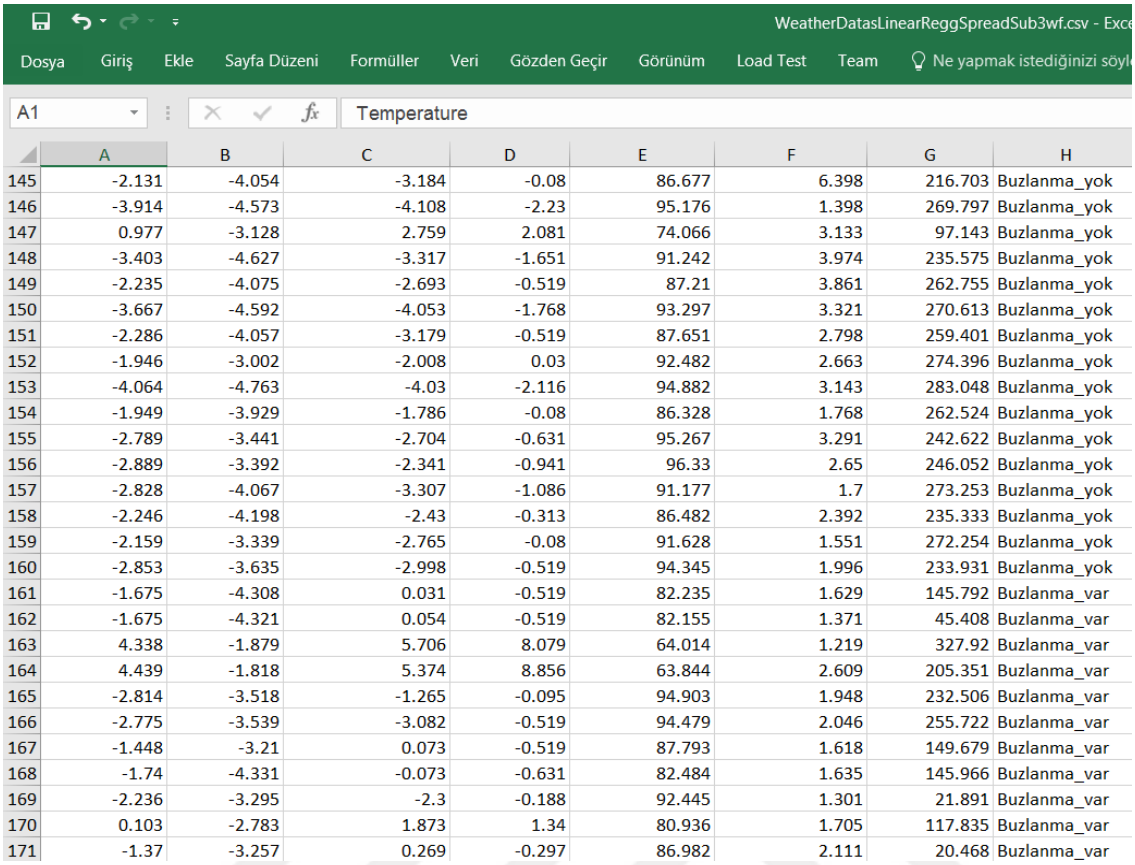
hakkında kapsamlı olarak bahsedilmektedir.

5.1. Buzlanma Tahmini için Mobil uygulama geliştirilmesi

Teknolojinin ilerlemesiyle beraber hayatımızın bir parçası haline gelen telefonlar, yaygın bir şekilde her alanda kullanılmaya başlanmıştır. İnternetin ve web hizmetlerinin telefonlara taşınması ile bu akıllı cihazlarda belirli görevleri yerine getiren mobil uygulamalar geliştirilmiş ve bu sayede kullanım daha fazla artmıştır.. Bugünlerde en yaygın olarak kullanılan işletim sistemlerinden birisi Android işletim sistemidir.

Bu çalışmada geliştirilen mobil uygulama, Android Studio, SDK, JDK araçları kullanılarak ilk prototipi JAVA (veri alış verişi JSON formatında) ve sonraki aşamada algoritma ile beraber son prototipi Javascript programlama dili ile geliştirilmiştir.

Çalışmada istasyonlardan alınan hava durum nitelikleri (Şekil 5.1) filtrelendikten - bir önişlemden geçirildikten - sonra eğitim veri seti olarak kullanılmıştır.



WeatherDatasLinearReggSpreadSub3wf.csv - Exce

Dosya Giriş Ekle Sayfa Düzeni Formüller Veri Gözden Geçir Görünüm Load Test Team Ne yapmak istediğinizi söyl

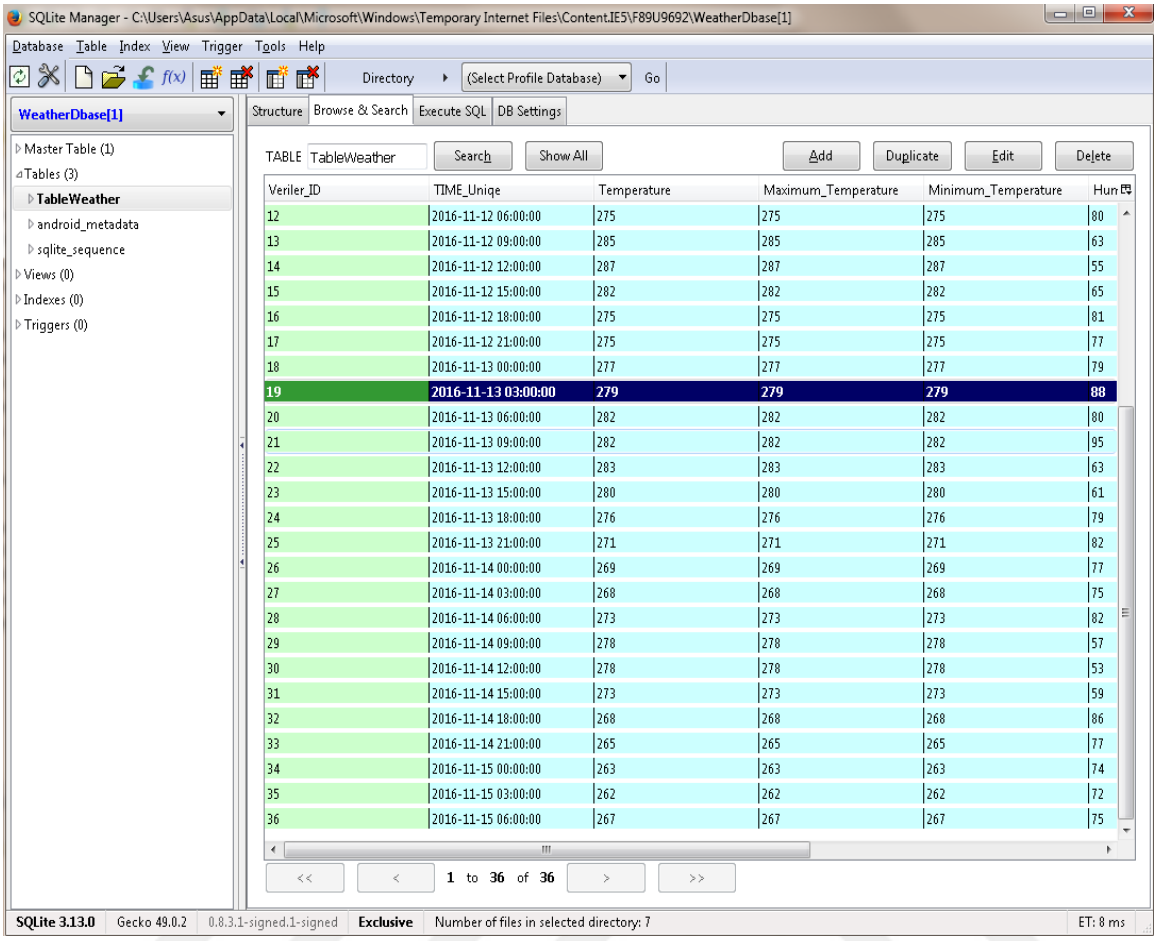
A1 Temperature

	A	B	C	D	E	F	G	H
145	-2.131	-4.054	-3.184	-0.08	86.677	6.398	216.703	Buzlanma_yok
146	-3.914	-4.573	-4.108	-2.23	95.176	1.398	269.797	Buzlanma_yok
147	0.977	-3.128	2.759	2.081	74.066	3.133	97.143	Buzlanma_yok
148	-3.403	-4.627	-3.317	-1.651	91.242	3.974	235.575	Buzlanma_yok
149	-2.235	-4.075	-2.693	-0.519	87.21	3.861	262.755	Buzlanma_yok
150	-3.667	-4.592	-4.053	-1.768	93.297	3.321	270.613	Buzlanma_yok
151	-2.286	-4.057	-3.179	-0.519	87.651	2.798	259.401	Buzlanma_yok
152	-1.946	-3.002	-2.008	0.03	92.482	2.663	274.396	Buzlanma_yok
153	-4.064	-4.763	-4.03	-2.116	94.882	3.143	283.048	Buzlanma_yok
154	-1.949	-3.929	-1.786	-0.08	86.328	1.768	262.524	Buzlanma_yok
155	-2.789	-3.441	-2.704	-0.631	95.267	3.291	242.622	Buzlanma_yok
156	-2.889	-3.392	-2.341	-0.941	96.33	2.65	246.052	Buzlanma_yok
157	-2.828	-4.067	-3.307	-1.086	91.177	1.7	273.253	Buzlanma_yok
158	-2.246	-4.198	-2.43	-0.313	86.482	2.392	235.333	Buzlanma_yok
159	-2.159	-3.339	-2.765	-0.08	91.628	1.551	272.254	Buzlanma_yok
160	-2.853	-3.635	-2.998	-0.519	94.345	1.996	233.931	Buzlanma_yok
161	-1.675	-4.308	0.031	-0.519	82.235	1.629	145.792	Buzlanma_var
162	-1.675	-4.321	0.054	-0.519	82.155	1.371	45.408	Buzlanma_var
163	4.338	-1.879	5.706	8.079	64.014	1.219	327.92	Buzlanma_var
164	4.439	-1.818	5.374	8.856	63.844	2.609	205.351	Buzlanma_var
165	-2.814	-3.518	-1.265	-0.095	94.903	1.948	232.506	Buzlanma_var
166	-2.775	-3.539	-3.082	-0.519	94.479	2.046	255.722	Buzlanma_var
167	-1.448	-3.21	0.073	-0.519	87.793	1.618	149.679	Buzlanma_var
168	-1.74	-4.331	-0.073	-0.631	82.484	1.635	145.966	Buzlanma_var
169	-2.236	-3.295	-2.3	-0.188	92.445	1.301	21.891	Buzlanma_var
170	0.103	-2.783	1.873	1.34	80.936	1.705	117.835	Buzlanma_var
171	-1.37	-3.257	0.269	-0.297	86.982	2.111	20.468	Buzlanma_var

Şekil 5.1. Geçmiş hava durumu verileri

Hava durumunu gözlemleyen meteoroloji sitesinden (AccuWeather.com), seçilen bölgenin (Şekil 5.2) günlük hava durumu tahmin bilgisi - önümüzdeki 12 saatin - alınmıştır ve bu hava durumu nitelikleri mobil uygulaması ve geliştirdiğimiz algoritma için test verisi olarak kullanılmıştır.

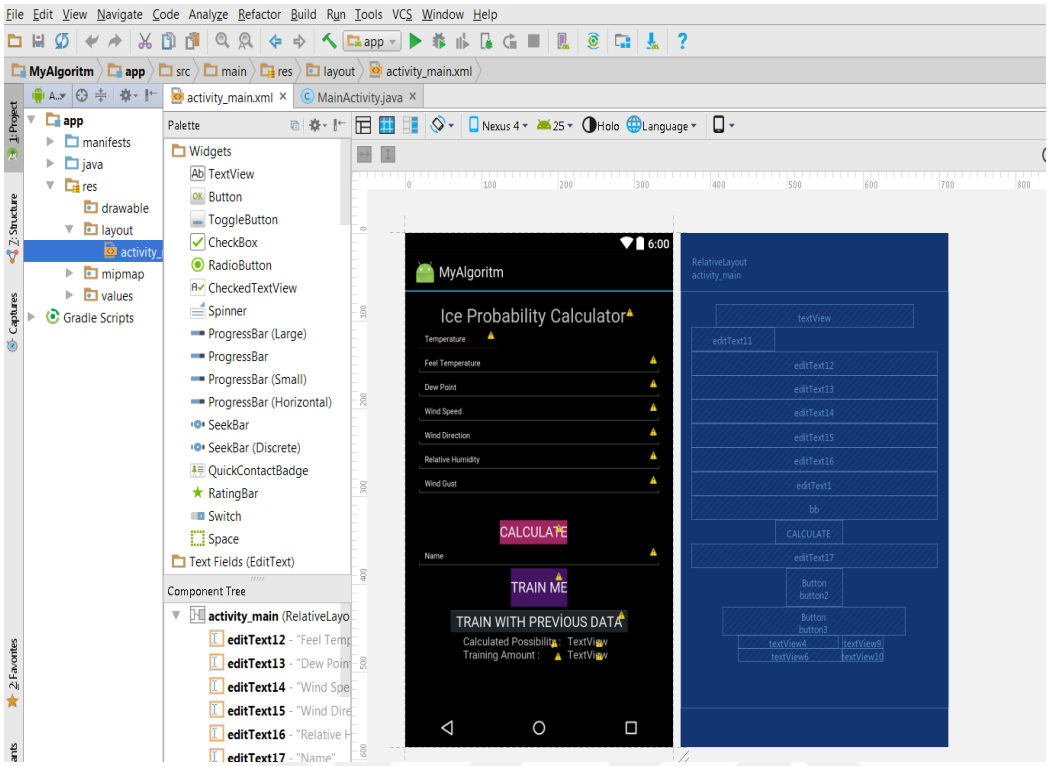
Web tabanlı erken uyarı sistemi mobil uygulamasında REST mimarisi kullanılmıştır. Veri alış verişinde JSON metodu tercih edilmiştir. Veri toplama kısmından sonra ikinci aşamada aynı işlemler - REST mimarisi ve JSON veri alış veriş yöntemi kullanılarak - geliştirilen algoritma ile beraber Javascript programlama dilinde yapılmıştır.



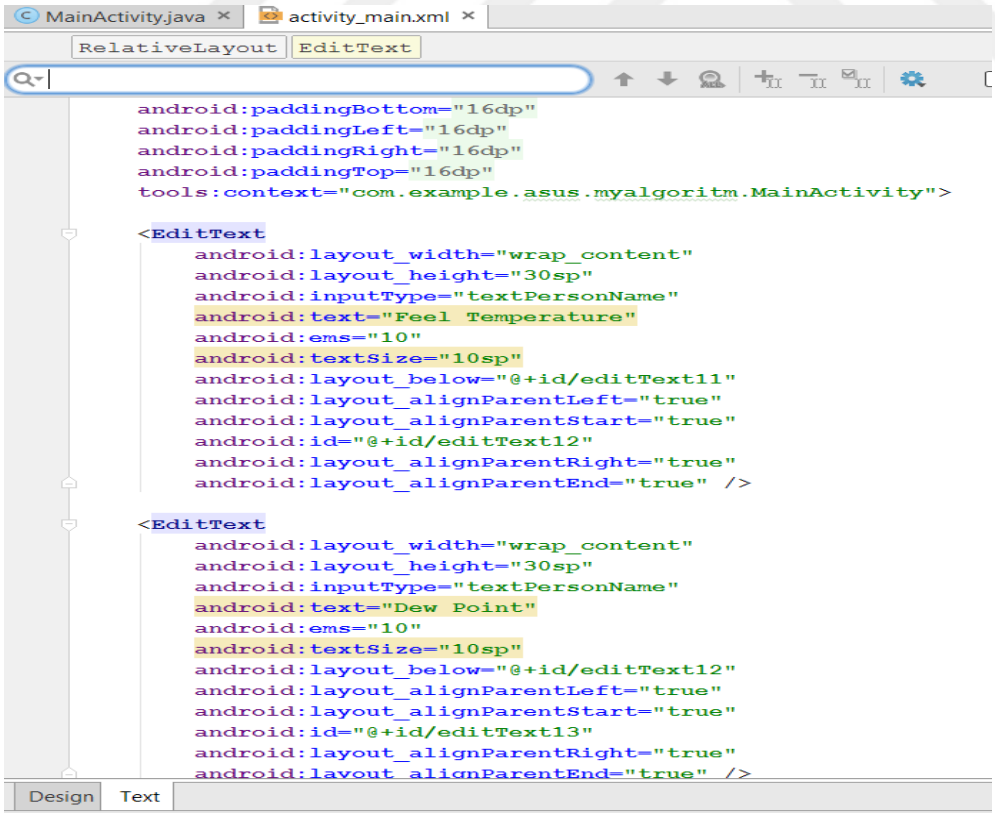
Şekil 5.2. Meteorolojiden alınan veriler

Erken uyarı sistemi mobil uygulamasının ara yüzü Android Studio tasarım aracı kullanılarak geliştirilmiştir. Tasarım XML tabanlı tanımlama dosyaları ve arayüz nesneleri kullanılarak tasarlanmıştır. Android Studio aracında tasarım yapmanın kolaylığı sürükle bırak yöntemi kullanılmasıdır.

Şekil 5.3 ve Şekil 5.4 ‘deki Android Studio Düzenleme Editörü ve Tasarım XML Dosyası ile geliştirilen erken uyarı sistemi mobil uygulamasının tasarlanan görsel yapısı gösterilmektedir.

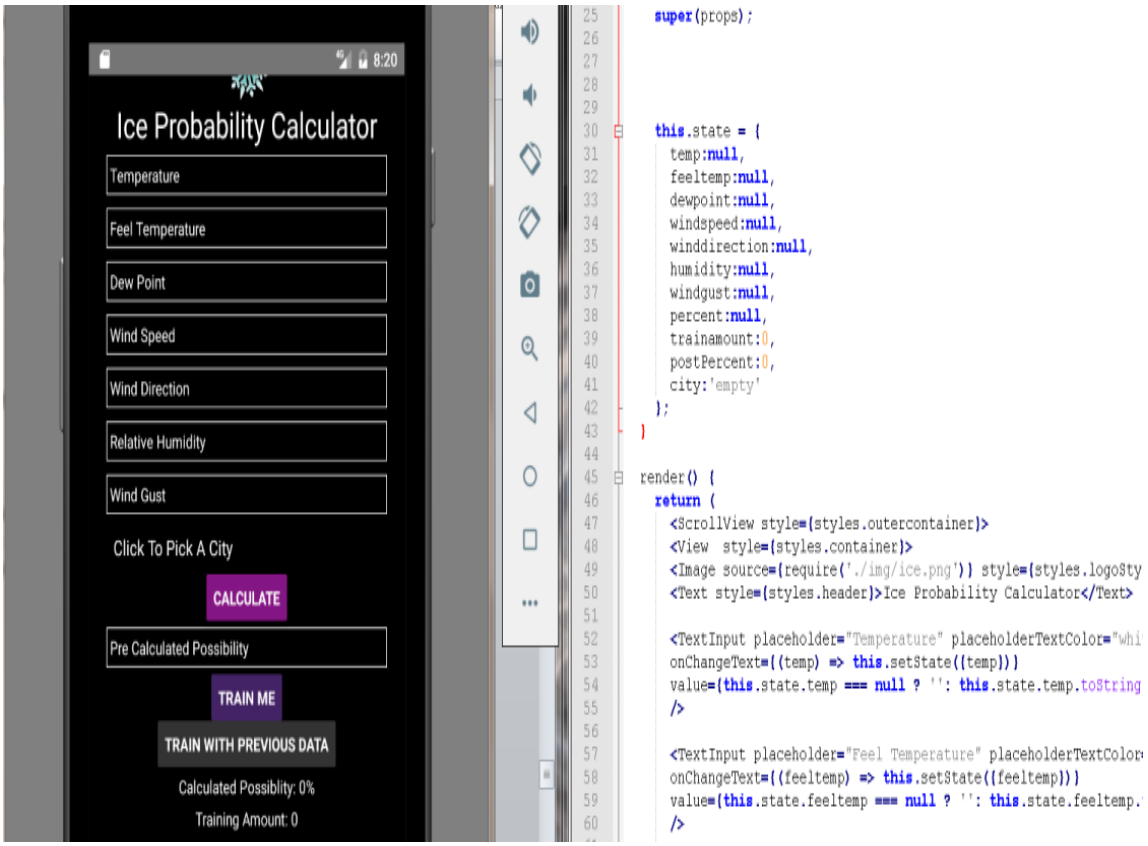


Şekil 5.3. Android Studio düzenleme editörü



Şekil 5.4. Android Studio tasarım XML dosyası

İkinci aşamada React Javascript Kütüphanesi (Şekil 5.5) kullanılarak dönüştürülmüştür. React Native mobil platformu Sanal DOM kullanarak hızını arttırsa da Android Studio editörleri ve tasarım araçları göz önüne alındığında daha avantajlıdır.



Şekil 5.5. React Native arayüz tasarımı

5.2. Veri Madenciliği Yöntem Uygulamaları

Bu çalışmada analiz edilen veri madenciliği yöntemlerini uygulamak için WEKA yazılım aracı kullanılmaktadır. Veri madenciliği aşamalarından sırasıyla, problem tanıma, geliştirme ve tahmin, sonuçların ortaya konulması ve karşılaştırma adımlarından yaralanılmıştır. Problem doğrultusunda yapılan buzlanma tahmininde, sensörlerden alınan geçmiş hava değişkenleri ve buzlanma tahminleri eğitim verisi olarak kullanılmaktadır.

Kulllanılan hava değişkenleri;

- Sıcaklık
- Çiğ noktası
- Hissedilen sıcaklık

- Rüzgar şiddeti
- Bağıl nem
- Rüzgar hızı
- Rüzgar yönü
- Buzlanma tahmini

Geçmişin açıklanmasında ve geleceğin tahmin edilmesinde veri madenciliğinden yararlanılmaktadır. Veri madenciliği modellerinden sınıflandırma, regresyon, kümeleme ve ilişkilendirme kurallarından, sınıflandırma ve regresyon modelleri bu çalışmada geleceğin tahmin edilmesi için kullanılmaktadır. Bu nedenle bu çalışmada veri madenciliği yöntemlerinden Destek Vektör Makinaları ve Çok Katmanlı Algılayıcı sinir ağları kullanılmıştır.

Çalışma kapsamında veri madenciliği yöntemlerinden senaryoya en uygun olanının seçilmesi hedeflenmektedir. Veri seti istenilen tahmin analizlerini yapmak için elverişli hale getirilmiştir. Aynı veri seti üzerinde farklı modeller denenerek performansı karşılaştırması yapılmıştır.

5.2.1. Destek Vektör Makinaları tahmin sonuçları

Destek vektör makinalarında doğrusal, Polinomiyal, Radyal tabanlı ve Sigmoid olmak üzere 4 Kernel fonksiyon vardır. Bu çalışmada filtrelenen eğitim veri seti ile birlikte her bir fonksiyon için doğruluk oranı tespit edilmiştir. Çizelge 5.1' de toplam doğru sınıflandırılan örnek yüzdesi verilmiştir.

Çizelge 5.1. DVM Kernel fonksiyonlarının toplam doğruluk yüzdeleri

Doğrusal Destek Vektör Makinaları	%86.3208
Polinomiyal Destek Vektör Makinaları	%58.0189
Radyal Tabanlı Destek Vektör Makinaları	% 84.434
Sigmoid Destek Vektör Makinaları	%75.00

Bu çalışma da amaç buzlanma tahmini yapmaktır. Bu sebeple "Buzlanma var" sınıfına ait örnekleri doğruluğunun tespiti, "Buzlanma yok" sınıfına ait örneklerin tespitinden daha fazla

önem arz etmektedir. Bu sebeple sadece buzlanma sınıfına ait sınıflandırma tahminlerinin doğruluk oranlarına bakıldığında toplam doğruluk oranına göre sonuç farklı olacaktır. “Buzlanma var” sınıfına ait doğruluk değerleri (Çizelge 5.2);

Çizelge 5.2 “Buzlanma Var” sınıfı doğru sınıflandırılan örnek yüzdesi

Doğrusal Destek Vektör Makinaları	%88.67924
Polinomiyal Destek Vektör Makinaları	%47.16981
Radyal Tabanlı Destek Vektör Makinaları	%37.73584
Sigmoid Destek Vektör Makinaları	%0.00

Çalışmada kapsamında, veri madenciliği yöntemlerinden destek vektör makineleri kernel fonksiyonlarının toplam doğru sınıflandırılan örnek yüzdesi ile “Buzlanma Var” sınıfı doğru sınıflandırılan örnek yüzdeleri incelenmiştir. Burada Sigmoid fonksiyonunun en düşük başarı oranına sahip olduğu ve Radyal tabanlı,polinomiyal, doğrusal fonksiyonlar arasında en iyi başarım performansının doğrusal fonksiyon olduğu gözlenmektedir. Sonuç olarak, tüm verilerin ve sadece “buzlanma var” sınıfı verilerinin doğruluk yüzdelerine bakıldığında en iyi başarımı Doğrusal DVM’ nin sağladığı görülmektedir.

5.2.2. Çok Katmanlı Algılayıcı sinir ağları tahmin sonuçları

Bu çalışmada kullanılan veri madenciliği yöntemlerinden ikincisi Çok Katmanlı Algılayıcı sinir ağı modelidir.

Çok Katmanlı Algılayıcı sinir ağı için parametreler;

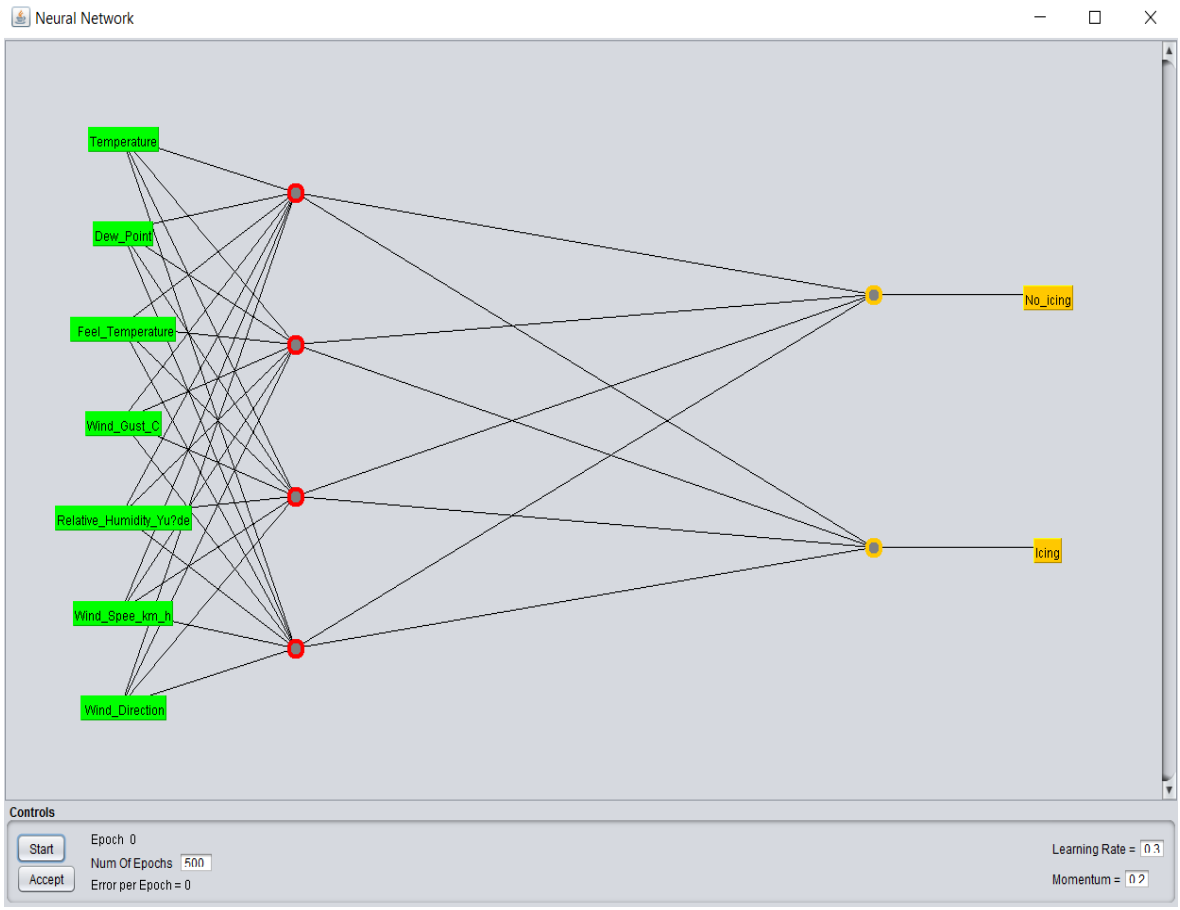
- Öğrenme Oranı = 0.3
- Momentum Oranı =0.2
- Eğitim Zamanı = 500 birim ‘dir.

Çok katmanlı Algılayıcı ağında gizli katman sayısı genellikle giriş katmanı ile çıkış katmanı düğüm sayısı arasında bir değer veya ortalamaları alınarak bulunmaktadır. Ara katman düğüm sayıları 1 ile 6 arasında bakıldığında, 4 düğüm sayısı en iyi doğru sınıflandırılan örnek yüzdesini vermektedir (Çizelge 5.3).

Çizelge 5.3. Düğüm sayısına göre doğru sınıflandırılan örnek yüzdesi

Ara Katman Düğüm Sayısı	Doğruluk yüzdesi
1	%83.9623
2	%83.9623
3	%85.8491
4	%87.2642
5	%86.3208
6	%86.3208

Ara katmanda daha fazla sayıda düğüm sayısı “Buzlanma var” sınıfı doğruluk oranında sapmalara neden olduğu için tercih edilmemiştir. Dahası, WEKA yazılım aracında çok katmanlı algılayıcı yönteminin parametrelerini seçerken gizi katman sayısı “a” olarak belirlenerek kendi atama yapması sağlanabilmektedir. (Şekil 5.6) Grafiksel kullanıcı arayüzü incelendiğinde ara katmanda 4 düğüm sayısı görülmektedir.



Şekil 5.6. Çok Katmanlı Algılayıcı sinir ağı grafiksel kullanıcı arayüz

Çok katmanlı algılayıcı sinir ağı modelinde en iyi toplam doğru sınıflandırılan örnek yüzdesi %87.2642' dir. Bununla beraber "Buzlanma Var" sınıfına ait 53 örnekten sadece 38 tanesi doğru sınıflandırılabilmiştir ve bu sınıf için doğru sınıflandırılan örnek yüzdesi %71.6981' dir.

5.2.3. Geliştirilen buzlanma tahmin algoritmasının sonuçları

Geliştirilen algoritmada tüm verilere bakıldığında %78 oranında başarım sağlanmaktadır. "Buzlanma Var" sınıfına ait 53 veriden 32 tanesi %50'nin üzerindedir. Bu oranlara bakıldığında "Buzlanma Var" sınıfı doğru sınıflandırılan örnek yüzdesi %60.37 olacaktır. Yapılan çalışmada doğru sınıflandırma oranı düşük olsa da, sistem daha fazla örnekle eğitildiğinde doğruluk oranı artmaktadır.

5.3. Buzlanma Tahmin Yöntem Uygulama sonuçlarının karşılaştırılması

Çalışma kapsamında geliştirilen algoritma ve veri madenciliği yöntemlerinden DVM ve YSA algoritmaları buzlanma tahmini için kullanılmıştır. Burada algoritmaların başarımı toplam buzlanma tahmini doğruluk oranına ve sadece "Buzlanma Var" sınıfına ait tahmin verilerinin doğruluk oranına bakılarak değerlendirilmiştir. Çizelge 5.4 ve Çizelge 5.5 de bu değerlendirme sonuçları verilmektedir.

Çizelge 5.4. Toplam doğru sınıflandırılan örnek yüzdesi

Geliştirilen Algoritma	%75.47
Doğrusal Destek Vektör Makinaları	%86.3208
Çok Katmanlı Algılayıcı sinir ağı	%87.2642

Çizelge 5.5. "Buzlanma Var" sınıfı doğru sınıflandırılan örnek yüzdesi

Geliştirilen Algoritma	%60.37
Doğrusal Destek Vektör Makinaları	%88.67924
Çok Katmanlı Algılayıcı sinir ağı	%71.6981

Sonuçlara bakıldığında toplam doğru sınıflandırılan örnek yüzdesinde en iyi başarımı ÇKA algoritmasıdır. “Buzlanma Var” sınıfına ait doğru sınıflandırılan örnek yüzdesinde en yüksek başarıya Doğrusal DVM algoritmaları sahiptir. Çalışma kapsamında geliştirilen algoritma doğru sınıflandırılan örnek sayısına bakıldığında ÇKA ve YSA veri madenciliği tahmin algoritmalarından daha düşük çıksa da, örneklerin tekrar sayısı arttırıldığı zaman başarı yüzdesindeki iyileşme artmaktadır. Örneklerin tekrar sayısı eğitimde arttıkça bu yöntemlerin başarı oranına yaklaşmaktadır.

6. SONUÇ VE ÖNERİLER

Bu çalışmada, ülkemizde yerel bölgelerde kullanılmak üzere sürücüleri buzlanma olup olmadığına dair uyarabilecek bir erken uyarı sistemi geliştirilmiştir. Araç kullanırken kolaylıkla erişilebilir olması amacıyla bu sistem mobil telefon üzerinde geliştirilmiştir. sistemin kullanım kolaylığı düşünülerek prototip olarak basit bir arayüz tasarlanmıştır.

Geliştirilen mobil uygulamada buzlanma tahmini yapılmaktadır. Bu tahminin yapılması için hava ve yer istasyonlarında yer alan sensörlerden alınan geçmiş hava verileri ve meteorolojiden alınan geleceğe dönük hava nitelikleri tahminleri eğitim ve test verisi olarak kullanılmıştır. Eğitim verisi olarak kullanılan geçmiş hava durumu niteliklerinde buzlanma tahmin bilgisi de yer almaktadır. Bu sayede hangi hava koşullarında buzlanma gerçekleşebileceğine dair sistem eğitilmiştir. Alınan geçmiş hava durumu nitelikleri eğitim veri kümesi olarak kullanılmadan önce sisteme uygun hale gelecek şekilde filtrelenmiştir. Bu sayede kullanılan algoritmaların doğruluk oranlarında hata olma olasılığı azaltılmıştır.

Veri madenciliği yöntemlerinden geleceğe dair analiz ve yorumlama hususunda sıklıkla tercih edilen çok katmanlı algılayıcı sinir ağı yöntemi ve destek vektör makinaları kullanılmıştır.

Yapay sinir ağları, KNN gibi birçok veri madenciliği tahmin algoritması farklı ölçütler ve ağırlıklandırma yöntemleri kullanmaktadır. Geliştirilen algoritmada sistem eğitilirken her verinin buzlanma tahminini ne yönde ve ne kadar etkilediğine bakılmıştır ve yönelim, ağırlık ölçütleri olarak sisteme atanmıştır. Bu tahmin algoritması eğer gelen veri kümesinde tahmin bulgusu var ise kendini eğitmektedir, yoksa yeni bir tahmin değeri oluşturmaktadır.

Erken uyarı sistemi için geliştirilen algoritma aynı veri seti kullanılarak çok katmanlı algılayıcı ve destek vektör makinaları veri madenciliği yöntemleri kullanılarak karşılaştırılmıştır. Destek vektör makinaları kendi içerisinde doğrusal, Polinomiyal, Radyal ve Sigmoid olarak 4 farklı tipi karşılaştırılmıştır. Bu karşılaştırma sonucunda buzlanma hava tahmini için en iyi yöntem, doğrusal destek vektör makinaları olarak gözlenmiştir.

Erken uyarı sistemi buzlanma tahmini için geliştirilen algoritma, Doğrusal destek vektör makinaları ve çok katmanlı algılayıcı sinir ağı modelleri karşılaştırıldığında toplam doğru sınıflandırılan örnek yüzdesine bakıldığında sırasıyla %75.47, %86.32 ve %87.2642'dır. Bulunulan sonuçlara bakıldığında en iyi sınıflandırmanın çok katmanlı algılayıcı yapay sinir ağı modeli olduğu görülmüştür. Bununla beraber sadece "buzlanma var" sınıfını göz önünde bulundurursak geliştirilen algoritma, Doğrusal destek vektör makinaları ve çok katmanlı algılayıcı sinir ağı modelleri karşılaştırıldığında yaklaşık sırasıyla %60.37, %88.68 ve %71.70'dır. DDVM algoritmasının performansının diğer yöntemlere göre daha iyi olduğu görülmüştür.

Erken uyarı sistemi için geliştirilen algoritma tahmin doğruluğu yaygın olarak kullanılan veri madenciliği yöntemleri tahmin algoritmaları ile karşılaştırıldığında daha az doğru sınıflandırılan örnek sayısına sahiptir, ancak sistem kendini eğittikçe tahmin oranında iyileşme ve doğru sınıflandırılan örnek sayısı artmaktadır.

KAYNAKLAR

1. İnternet: ORMAN VE SU İŞLERİ BAKANLIĞI Meteoroloji Genel Müdürlüğü. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.mgm.gov.tr%2F&date=2017-07-11, Son Erişim Tarihi: 2017-07-11.

2. Drezga, I. and Rahman, S. (1999). Short-term load forecasting with local ANN predictors. *IEEE Transactions on Power Systems*, *14*(3), 844-850.

3. Chen, S. M. and Hwang, J. R. (2000). Temperature prediction using fuzzy time series. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 30(2), 263-275.

4. Joshi, A., Kamble, B., Joshi, V., Kajale, K. and Dhange, N. (2015). Weather Forecasting and Climate Changing Using Data Mining Application. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(3), 19-21.

5. De Giorgi, M. G., Ficarella, A. and Tarantino, M. (2011). Assessment of the benefits of numerical weather predictions in wind power forecasting based on statistical methods. *Department of Engineering for Innovation*, 36(7), 3968-3978.

6. López, V. F., Medina, S. L. and de Paz, J. F. (2012). Taranis: Neural networks and intelligent agents in the early warning against floods. *Expert Systems with Applications*, 39(11), 10031-10037.

7. İnternet: Vodafone, Teknolojisiyle Doğanın Hizmetinde. . URL: http://www.webcitation.org/query?url=http%3A%2F%2Fmedyamerkezi.vodafone.com.tr%2Fbasin-bultenleri%2Fvodafone-teknolojisiyle-doganin-hizmetinde&date=2017-07-11, Son Erişim Tarihi: 2017-07-11.

8. Dregza, I. and Rahman, S. (1999). Short-term load forecasting with local ANN predictors. *IEEE Transactions on Power Systems*, 14(3), 844-850.

9. Taylor, J.W. and Buizza, R. (2002). Neural network load forecasting with weather ensemble predictions. *IEEE Transactions on Power Systems*, 17(3), 626-632.

10. Sapankevych, N., I. and Sankar, R. (2009). Time series prediction, using support vector machines: a survey. *IEEE Computational Intelligence Magazine*, 4(2), 24-38.

11. Lu, W., Wang, W., Leung, Y. T. A., Lo, S.-M., Yuen, R. K. K., Xu, Z. and Fan, H. (2002, May). *Air pollutant parameter forecasting using support vector machines*. Conference on Proceedings of the 2002 International Joint Neural Networks, 630-635.

12. Rao, T., Rajasekhar, N. and Rajinikanth, T. V. (2012, September). *An efficient approach for Weather forecasting using Support Vector Machines.* International Conference on Computer Technology and Science, 208-212.

13. Perrone, M. P. (1994). *General Averaging Results for Convex Optimization*. Proceeding of the 1993 Connectionist Models Summer School, 364-371.

14. Pal, N. R., Pal, S., Das, J. and Majumdar, K. (2003). SOFM-MLP: A hybrid neural network for atmospheric temperature prediction. *IEEE Transactions on Geoscience and Remote Sensing*, 41(12), 2783-2791.

15. Chen, N., Qian, Z., Nabney, l. T. and Meng, X. (2014). Wind power forecasts using Gaussian processes and numerical weather prediction. *IEEE Transactions on Power Systems*, 29(2), 656-665.

16. Costa, M. and Pasero, E. (2001, July). *Artificial Neural Systems for Verglass Forecast*. International Joint Conference on Neural Networks, 258-262.

17. Li, J. (2010, April). *A Combination of DE and SVM with Feature Selection for Road Icing Forecast.* 2nd International Asia Conference on Informatics in Control, Automation and Robotics, 509-512.

18. Mahura, A., Petersen, C. and Sass, B. H. (2008). *Road Icing Conditions in Denmark*. Ministry of climate and energy, Copenhagen, Scientific Report 08-03, 1-25.

19. Mandale, A. and Jadhawar, B. A. (2015). Weather forecast prediction: a Data Mining application. *International Journal of Engineering Research and General Science* 3(2), 1279-1284.

20. Lampton, A. and Valasek, J. (2012). Prediction of icing effects on the lateral/directional stability and control of light airplanes. *Aerospace Science and Technology*, 23(1), 305–311.

21. McCann, D. W. (2005). NNICE – A Neural Network Aircraft Icing Algorithm. *Environmental Modelling & Software*, 20(10), 1335–1342.

22. Wang, Z. and Li, C. (2009, Dec). *River Ice Forecasting Based on Genetic Neural Network*. 2009 International Conference on Information Engineering and Computer, 1-4.

23. Savadjiev, K. and Farzaneh, M. (2004). Modeling of icing and ice shedding on overhead power lines based on statistical analysis of meteorological data. *IEEE Transactions on Power Delivery*, 19(2), 715-721.

24. Luo, Y., Li, Y., Yao, Y. and Zhan, X. (2012, May). *Research on Power Transmission Line Ice Prediction System Based on BP Neural Network.* International Conference on Measurement, Information and Control, 817-820.

25. Chen, S. Q. and Guo, S. Z. (1994). Fuzzy Prediction. *Guiyang: Guizhou Science & Technology Press*, 332-342.

26. Liu, C., Liu, H. W., Wang, Y. S., Lu, J. Z., Xu, X. J. and Tan, Y. J. (2014, March). *Research of icing thickness on transmission lines based on fuzzy Markov chain prediction*. IEEE International Conference on Applied Superconductivity and Electromagnetic Devices, 327-330.

27. Huang, J., Yang, H., Hunan, Y. W. (2016, March). *Forecast of Line Ice-Coating Degree Using Circumfluence Index & Support Vector Machine Method.* International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, 327-330.

28. Li, Q., Li, P., Zhang, Q., Ren, W., Cao, M. and Gao, S. (2011, June). *Icing Load Prediction for Overhead Power Lines Based on SVM.* Proceedings of 2011 International Conference on Modelling, Identification and Control, 104-108.

29. Patel, B. G., Dabhi, V. K., Tyagi, U. and Shah, P. B. (2015, July). *A Survey on Location Based Application Development for Android Platform.* International Conference on Advances in Computer Engineering and Applications, 731-739.

30. Lee, M., Bak, C. and Lee, J. W. (2014). A prediction and auto-execution system of smartphone application services based on user context-awareness. *Journal of Systems Architecture*, 60(8), 702-710.

31. Mantas, V.M., Liu, Z., Pereira, A.J.S.C., (2015). A web service and android application for the distribution of rainfall estimates and earth observation data. Computers & Geosciences, 77, 66–76.

32. Internet: AFAD Deprem Mobil Uygulaması. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.deprem.gov.tr%2Ftr%2Fkategori%2Fmobil&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

33. Internet: WS600-UMB Smart Weather Sensor. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.lufft.com%2Fproducts%2Fcompact-weather-sensors-293%2Fws600-umb-smart-weather-sensor-1832%2F&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

34. Internet: Intelligent Active Road Sensor ARS31Pro-UMB. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.lufft.com%2Fproducts%2Froad-runway-sensors-292%2Fintelligent-active-road-sensor-ars31pro-umb-2305%2F&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

35. Internet: Android (işletim sistemi). URL: http://www.webcitation.org/query?url=https%3A%2F%2Ftr.0wikipedia.org%2Fwiki%2FAndroid+%28i%25C5%259Fletim_sistemi%29&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

36. Çınar, S. (2014). *Mobil Android Ortamında Parmak İzi Tanıma Ve Kimlik Doğrulama Sisteminin Geliştirilmesi*. Yüksek Lisans Tezi, Haliç Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

37. Internet: Android Kernel. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdenizkilinc.com%2F2013%2F07%2F02%2Fandroid+isletim-sistemi-mimarisi-kernel-libraries-runtime-framework-dvm&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

38. Internet: Hibben, M. W. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.technomicon.com%2FElectroPolitics%2FElectroPolitics-11-15-10.html&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

39. Internet: Android Studio'yu Tanıyalım. 2017-07-12. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fgelecegiyazanlar.turkcell.com.tr%2Fkonu%2Fandroid%2Fegitim%2Fandroid-201%2Fandroid-studioyu-taniyalim&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

40. Internet: View. http://www.webcitation.org/query?url=https%3A%2F%2Fdeveloper.android.com%2Freference%2Fandroid%2Fview%2FView.html&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

41. Internet: Getting Started. URL: http://www.webcitation.org/query?url=https%3A%2F%2Ffacebook.github.io%2Freactnative%2Fdocs%2Fgetting+started.html&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

42. Internet: Anatomy of an XML Web Service Lifetime. . 2017-07-12. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fmsdn.microsoft.com%2Fen-us%2Flibrary%2Fx05s00wz%28v%3Dvs.100%29.aspx&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

43. Internet: Android. 2017-07-12. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdeveloper.android.com%2Findex.html&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

44. İğci, E. (2014). *Android İşletim Sistemi İle Çalışan Mobil Cihazlarda Uygulama Geliştirilmesi*, Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir.

45. Internet: Layouts. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdeveloper.android.com%2Fguide%2Ftopics%2Fui%2Fdeclaring-layout.html&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

46. Internet: UI Overview. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdeveloper.android.com%2Fguide%2Ftopics%2Fui%2Foverview.html&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

47. Internet: WEKA. URL: http://www.webcitation.org/query?url=http%3A%2F%2F+http%3A%2F%2Fwww.cs.waikato.ac.nz%2Fml%2Fweka%2F&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

48. Cui, B. and He, S. (2016). *Anomaly Detection Model Based on Hadoop Platform and Weka Interface.* 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 84-89.

49. Lupu, E. and Emerich, S. (2010). Speaker identification approach based on time domain extracted features. *In Elmar, 2010 Proceedings*, 355-358.

50. Witten, I. H., Frank, E., Hall, M. A. and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. USA: Morgan Kaufmann.

51. Gu, C., Hornberger, G. M., Herman, J. S. and Mills, A. L. (2008). Effect of Freshets on the Flux of Groundwater Nitrate through Streambed Sediments. *Water Resources Research*, 44(5), 1-12.

52. Cios, K. J., Pedrycz, W., Swiniarski, R. W., & Kurgan, L. (2007). *Data mining: a knowledge discovery approach.* New York: Springer Science & Business Media.

53. Melek, C. (2012). *Metin madenciliği teknikleri ile şirketlerin vizyon ifadelerinin analizi.* Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi Sosyal Bilimleri Enstitüsü, İzmir.

54. Internet: Regresyon analizi. URL: http://www.webcitation.org/query?url=https%3A%2F%2Ftr.0wikipedia.org%2Fwiki%2FRegresyon_analizi&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

55. Franz, M. O. and Scholkopf, B. (2004). *Implicit Estimation of Wiener Series.* 2004 IEEE Workshop on Machine Learning for Signal Processing, 735 – 744.

56. Asad, M. (2015). *Optimized Stock Market Prediction Using Ensemble Learning*. 9th International Conference on Application of Information and Communication Technologies, 263-268

57. Awal, A., Rabbi, J., Hosssain, I. and Hashem, M. A. (2016). *Using Linear Regression to Forecast Future Trends in Crime of Bangladesh*. 5th International Conference on Informatics, Electronics and Vision, 333-338

58. Zhu, Y., Li, X. and Wu, G. (2016). *Face Expression Recognition Based On Equable Principal Component Analysis and Linear Regression Classification*. 3rd International Conference on Systems and Informatics,876-880

59. Internet: Regresyon analizi. . 2017-07-12. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fw3.balikesir.edu.tr%2F%7Ebsentuna%2Fwp-content%2Fuploads%2F2013%2F03%2FRegresyon-Analizi.pdf&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

60. Albayrak, A.S. and Yılmaz, S.K. (2009). Data Mining: Decision Tree Algorithms and An Application on ISE Data. *Süleyman Demirel University Journal of Faculty of Economics & Administrative Sciences,* 14(1), 31-52.

61. Sahami, M., Dumais, S., Heckerman, D. and Horvitz, E. (1998). *A Bayesian Approach to Filtering Junk Email*. Published In AAAI Workshop on Learning for Text Categorization, 1-8.

62. Vapnik, V. and Cortes, C. (1995). Support vector networks. *Machine Learning*, 20, 273-297.

63. Internet: Support vector machine. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fen.0wikipedia.org%2Fwiki%2FSupport_vector_machine&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

64. Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B.P. (2007). *Numerical Recipes The Art of Scientific Computing*. (3rd Edition). England: Cambridge University Press.

65. Cui, B. and He, S. (2016, Dec). *Anomaly Detection Model Based on Hadoop Platform and Weka Interface*. 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 84-89.

66. Dan, L., Lihua, L. and Zhaoxin, Z. (*2013). Research of Text Categorization on WEKA.* Third International Conference on Intelligent System Design and Engineering Applications (ISDEA), 1129-1131.

67. Dantas, L. and Valença, M. (2014, Dec). *Using Neural Networks In The Identification Of Signatures For Prediction Of Alzheimer's Disease*. IEEE 26th International Conference on Tools with Artificial Intelligence (ICTAI), 238-242.

68. Internet: Yapay Sinir Ağları (YSA) Nedir? URL: http://www.webcitation.org/query?url=http%3A%2F%2Fkod5.org%2Fyapay-sinir-aglari-ysa-nedir%2F&date=2017-07-12, Son Erişim Tarihi: 2017-07-12.

69. Maysanjaya, D., Nugroho, H. A and Setiawan, N. O. (2015). *A Comparison of Classification Methods on Diagnosis of Thyroid Diseases.* 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA), 89-93.

70. Zhang, G. P. (2007). Avoiding Pitfalls in Neural Network Research. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and reviews*, 37(1), 3-16.

71. Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control,Signals, and Systems*, (2), 303-314

72. Ruck, D. W., Rogers, S. K. and Kabrisky, M. (1990). Feature Selection Using a Multilayer Perceptron. *Journal of Neural Network Computing*, 2(2), 40-48.

73. Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. USA: Washington DC. Spartan Books.

74. Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, (1), 318-362.

75. Halder, C., Paul, J. and Roy, K. (2012). *Comparison of The Classifiers In Bangla Handwritten Numeral Recognition.* 2012 International Conference on Communication and Computing (ICRCC), 272-276.

76. Baron, A. R. (1989). *Statistical Properties of Artificial Neural Networks.* Proceedings of the 28th IEEE Conference on Decision and Control, 280-285

77. Heaton, J. (2008). *Introduction to Neural Networks for Java* (2nd Edition). USA: Heaton Research.

78. Gómez-Gil, P., López-Cruz, O. and Cruz-Martínez, A. B. (2012). Improving Neural-Based Classification of Databases With Overlapped Classes: The Case Of Star/Galaxy Segregation. *Third Latin American Symposium on Circuits and Systems*, 1-4.

79. Wibawa, M.S., Nugroho, H. A. and Setiawan, N. A. (2015). *Performance Evaluation Of Combined Feature Selection And Classification Methods In Diagnosing Parkinson Disease Based On Voice Feature*. International Conference on Science in Information Technology (ICSITech), 126-131.

80. Lippmann, R. P. (1987). An Introduction to Computing With Neural Nets. *ACM SIGARCH Computer Architecture News*, 16(1), 7-25.

81. Nabiyev, V.V. (2013). *Algoritmalar*. (Dördüncü Baskı). Türkiye: Seçkin Yayıncılık.

82. Sezgin, E., & Çelik, Y. (2013). *Veri madenciliğinde kayıp veriler için kullanılan yöntemlerin karşılaştırılması.* Akademik Bilişim Konferansı, Akdeniz Üniversitesi, 23-25.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : TIRAŞOĞLU, Hatice

Uyruğu : T.C.

Doğum tarihi ve yeri : 27.12.1990 Erzurum

Medeni hali : Bekâr

Telefon : 0 (507) 703 95 51

Faks : 0 (507) 703 95 51

e-mail : haticetrasoglu@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi / Teknoloji Fakültesi Bilgisayar Mühendisliği	Devam ediyor
Lisans	KTÜ / Bilgisayar Mühendisliği	2014
Lise	Trabzon Kanuni Anadolu Lisesi	2009

Yabancı Dil

İngilizce

Yayınlar

Tıraşoğlu, H., and Duran, F. (2017, May). A Multi-Purpose Mobile Application for Embedded Systems. *International Conference on Advanced Technology & Sciences, ICAT'17*

Burslar

TÜBİTAK 2210-D Sanayiye Yönelik Yurt İçi Yüksek Lisans Burs Programı (2015/2)

Hobiler

Resim, Yürüyüş, Seyahat

GAZİ GELECEKTİR..